

Introdução ao aplicativo Gnuplot

Universidade Federal do Pará
Instituto de Ciências Exatas e Naturais
Faculdade de Matemática
Patricia de Almeida Viana

Agosto 2011

Resumo

Neste trabalho foi feita uma investigação no aplicativo Gnuplot, um software livre de grande potencial que gera gráficos em duas e três dimensões e é usado em diversas áreas científicas. Pretende-se mostrar de forma clara como começar a trabalhar com este aplicativo no Linux. Como fontes de pesquisa foram consultadas o manual do Gnuplot e alguns tutoriais disponíveis na internet.

1 Introdução

O **Gnuplot** é um programa usado na criação de gráficos de funções e dados em duas ou três dimensões. Ele é bastante versátil e possui uma simplicidade capaz de fazer com que através da execução de apenas um comando de arquivo seja possível plotar diversos gráficos. Este aplicativo oferece ainda ferramentas de análise de dados, como ajuste de funções aos dados experimentais. Além de exibir gráficos na tela, possibilita gravá-los em diversos formatos, como postscript, fig e diretamente no formato tex. Ele é altamente portátil, possuindo versões para os sistemas operacionais linux, windows e unix. Iniciemos um estudo do aplicativo no Linux. Uma documentação detalhada pode ser encontrada através do help do programa (assim como suas "man pages"), <http://www.gnuplot.info/>, <http://www.duke.edu/~hpgavin/gnuplot.html> ou através de serviços de busca na internet.

2 Primeiro Contato

O primeiro passo é executar o programa **Gnuplot**, para isso basta digitar o comando **gnuplot** em alguma janela do terminal e pressionar a tecla [enter]. Feito isto, aparecerão algumas informações na tela, como a versão do programa que está sendo executado, os nomes dos criadores deste aplicativo bem como um site na internet onde podem ser encontradas maiores informações a respeito deste software.

```
Arquivo  Editar  Ver  Terminal  Ajuda
patricia@patricia-desktop:~$ gnuplot

G N U P L O T
Version 4.2 patchlevel 5
last modified Mar 2009
System: Linux 2.6.31-21-generic

Copyright (C) 1986 - 1993, 1998, 2004, 2007 - 2009
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 bug reports and suggestions to <http://sourceforge.net/pro
plot>

Terminal type set to 'wxt'
gnuplot> █
```

Figura 1:

A última linha deste terminal deve ser a linha de comando do gnuplot, a qual será utilizada como interface entre o usuário e o programa; esta linha de comando é da seguinte forma:

```
gnuplot >
```

A partir dessa linha de comando, digitaremos diversas instruções que o **Gnuplot** executará gerando o gráfico desejado.

E se algum momento desejarmos finalizar este programa e voltar para o terminal, basta digitar o comando *quit* (ou simplesmente a letra “q”) e, pressionar a tecla [enter].

3 Sintaxe

Existe um conjunto de regras que ditam a forma como se escrevem as expressões. Uma delas é a forma como se escrevem as potências $(x * *2) = x^2$.

Outra regra importante é que é obrigatório usar o asteristico (*) em todas as operações de multiplicação, isto quer dizer que não podemos escrever, por exemplo, plot 2x, temos que escrever plot 2 * x.

A divisão faz-se com uma barra (/), a soma e a multiplicação fazem-se com os sinais do costume (+ e *).

Além de gráficos o Gnuplot faz cálculos, e é isso que faz o comando **print**. Porém, uma

regra importante no Gnuplot, é que um número escrito sem casa decimal é interpretado como um número inteiro. Por exemplo, se pedirmos para o Gnuplot calcular a divisão $1/2$, a sua resposta será 0. Veja:

```
gnuplot> print 1/2
0
```

Isso ocorreu pois a divisão de um número inteiro por um número inteiro, é um inteiro. Se quisermos obter 0.5 devemos digitar $1.0/2$ ou $1/2.0$:

```
gnuplot> print 1/2.0
0.5
```

O que o comando `print` faz é mostrar valores calculados, ou seja, podemos usá-lo como uma 'calculadora', para efetuar as operações fundamentais, fatorial, valores de seno, cosseno, tangente, exponencial, o valor de uma função em um ponto, etc.

4 Comandos para Plotar Gráficos em 2D e 3D

Para plotarmos gráficos em duas ou três dimensões usamos respectivamente, os comandos `plot` e `splot`. Estes comandos sozinhos não fazem absolutamente nada; para que eles possam gerar qualquer tipo de gráfico devemos oferecer a eles algumas informações.

5 Comando *Plot*

O comando “`plot`” já mencionado anteriormente, é utilizado para elaborar gráficos em duas dimensões. Para isso, ele precisa receber informações como: a função ou nome do arquivo que contém o conjunto de dados a serem plotados, se é um histograma, se ele deve utilizar pontos, linhas ou ambos para a representação do gráfico, tipo de ponto a ser utilizado, cor destes pontos ou linhas, etc.

A entrada básica deste comando é a seguinte:

```
gnuplot > plot <limites> <função ou nome do arquivo> with <estilos>
```

Sendo:

< **limites** >: define que o gráfico será plotado entre dois valores de x e entre dois valores de y , ou seja, domínio e imagem, porém se esses limites ou um deles não for especificado o programa utilizará os que achar conveniente.

< **função ou nome do arquivo** >: define qual ou quais funções e/ou conjuntos de dados serão plotados. Para plotar mais de uma função ou conjunto de dados utilizando apenas um comando `plot`, basta separar cada função ou conjunto de dados por vírgulas, como no exemplo a seguir:

```
gnuplot > plot x**2, G(x), "dados.dat"
```

Neste exemplo será gerado um gráfico da função x^2 e neste mesmo gráfico será plotada a função $G(x)$, a qual deve ter sido definida anteriormente e, o gráfico do conjunto de dados contido no arquivo “*dados.dat*”.

< **estilos** >: define a aparência do gráfico, ou seja, se este gráfico será plotado como uma linha, como pontos, como linha e pontos além de definir qual será a cor utilizada. Por exemplo:

```
gnuplot > plot x**2 with points 2 2
```

Neste exemplo será gerado um gráfico da função x^2 , sendo que ela será representada por pontos (with points) na cor número 2 (with points 2) ou seja, cor ciano e, os pontos serão representados pelo sinal de adição (“+”) (with points 2 2).

Existem diversos tipos de pontos e cores, basta mudar o número de [-1,34] que a cor ou o tipo de ponto mudará.

6 Comando *Splot*

Este comando é extremamente semelhante ao **plot**, porém usa-se para plotar gráficos em três dimensões, sendo assim você deve especificar funções de três variáveis ou ter um arquivo de dados com três colunas no mínimo.

O comando **splot** usa as mesmas opções do comando **plot**, porém ele trabalha sempre ou com funções de duas variáveis, ou tabelas com no mínimo três colunas.

7 Comandos básicos

No Gnuplot podemos definir os tipos de pontos e linhas. Para visualizar as cores, bem como os formatos dos pontos, basta ativar na linha de comando, a palavra **test**, como mostrado abaixo:

```
gnuplot > test
```

A partir desse comando aparecerá a figura 2.

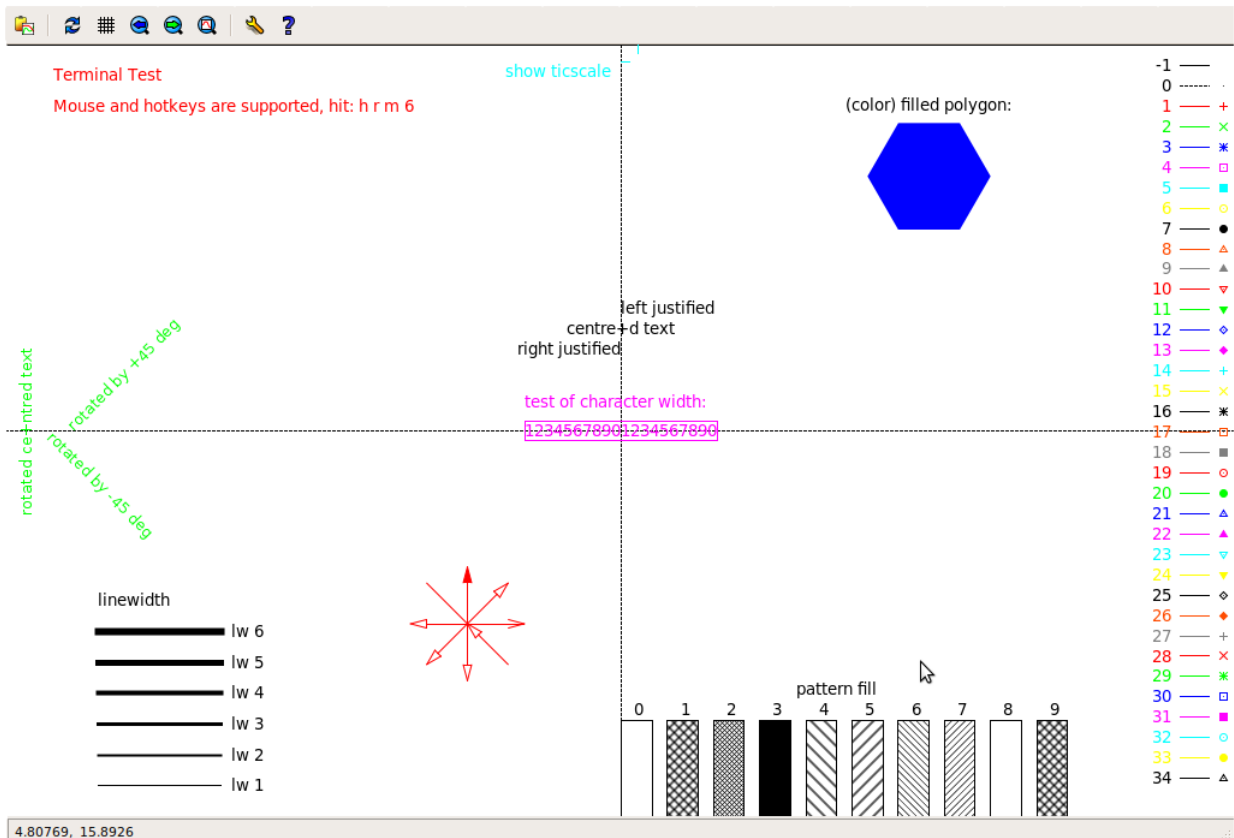


Figura 2: Tipos de pontos e cores disponíveis no gnuplot

Outra coisa muito importante que não podemos deixar de falar é sobre o comando **help**, basta digitá-lo na linha de comando do **Gnuplot**, que ele lhe dará uma ajuda de tudo o que ele pode fazer.

7.1 Outras definições do programa

No Gnuplot há uma série de comandos para complementar informações a respeito do gráfico de uma função, por exemplo para modificar as escalas de visualização, realizar desenhos de vários gráficos no mesmo sistema de eixos, colocar título em um gráfico, colocar algum comentário nos eixos coordenados, entre outros. Passamos então uma sintaxe destes comandos.

- Título:

```
gnuplot > set title "texto"
```

Esta linha de comando dá um título ao gráfico.

- Tipo e Espessura da linha:

Através da linha de comando, usando após **plot f(x)** os seguintes comandos: **linestyle n** altera o estilo da linha, **linetype n** altera o tipo de linha. e **linewidth n** altera a espessura da linha.

- Nomear eixos:

```
gnuplot > set xlabel "Eixo x"
```

Esta linha de comando nomeia o eixo x no gráfico da função.

```
gnuplot > set ylabel "Eixo y"
```

Esta linha de comando nomeia o eixo y no gráfico da função.

```
gnuplot > set zlabel "Eixo z"
```

Esta linha de comando nomeia o eixo z no gráfico da função. O eixo z só aparece quando estamos trabalhando no \mathbb{R}^3 , com o comando `splot`.

- Definir parâmetros de escala:

```
gnuplot > set xrange [ninf : nsup]
```

Com esta linha de comando podemos determinar o intervalo de variação (domínio) da variável x no gráfico. O intervalo deve ser escrito entre colchetes e os números separados por dois pontos [:].

```
gnuplot > set yrange [ninf : nsup]
```

Esta linha de comando define o intervalo de variação possível para os valores da função (contradomínio) no caso de funções de uma variável. E no caso de funções de duas variáveis, definirá o domínio da variável y.

```
gnuplot > set zrange [ninf : nsup]
```

No caso de funções de duas variáveis, esta linha de comando define o contradomínio da função.

Observação: Uma vez modificada a faixa de variação de uma variável, esta modificação permanecerá válida para todos os gráficos feitos posteriormente.

- Definir valores dos eixos:

Para definir os valores que aparecerão nos eixos podemos usar o comando `set xtics`, `set ytics` e `set ztics`. A sintaxe do comando é:

```
gnuplot > set xtics xinicial,incremento,xfinal
```

onde xinicial e xfinal são opcionais.

Exemplo:

```
gnuplot > set xtics -2,1,3
```

No exemplo acima, aparecerão os valores -2, -1, 0, 1, 2, 3 no eixo x, caso estes valores estejam dentro do limite de variação para a variável x definido no comando `set xrange`. Os comandos `set ytics` e `set ztics` são análogos ao `set xtics`.

- Mostrar eixos x e y:

```
gnuplot > set zeroaxis
```

Com este comando é exibido o sistema de eixos e quando possível a sua origem, tornando o gráfico mais completo e fácil de entender. É recomendado colocar este comando logo no início para que apareçam os eixos em todos os demais gráficos.

- Mostrar grades:

```
gnuplot > set grid
```

Mostra as grades nos eixos x e y. Ou ainda, se quisermos desativar a opção grade basta utilizar o comando `unset grid`.

- Efeito de Sólido:

```
gnuplot > set hidden3d
```

O comando acima é utilizado no modo 3D para “esconder” o que fica “atrás” da superfície.

- Resolução do Gráfico:

Para traçar o gráfico de uma função, o Gnuplot escolhe alguns pontos e calcula o valor da função nestes pontos. A quantidade padrão de pontos escolhidos no eixo x para gráficos 2D é 100. No caso de gráficos tridimensionais a quantidade padrão de pontos também é de 100, sendo tomados 10 pontos no eixo x e no eixo y. Esta quantidade pode ser alterada através dos comandos `set samples n` para gráficos no plano e `set isosamples k,m` para gráficos no espaço, sendo que n é o número de pontos no eixo x e k, m correspondem ao número de pontos nos eixos x e y, respectivamente.

- Bordas do gráfico:

O comando `set border` controla a exibição das bordas do gráfico gerado com os comandos `plot` e `splot`. Para excluir todo o contorno da janela gráfica utilizamos o comando `unset border`. Para incluir uma das bordas utilizamos o comando `set border n`, sendo n o número associado a borda desejada. Para colocar duas ou mais bordas somamos os respectivos números e utilizamos `set border k`, onde k é a soma dos números associados às bordas.

Outra opção é o comando `reset`, que faz com que todas as modificações de formatação sejam canceladas de uma vez só.

- Diretório: Para navegarmos pelo diretório:

```
gnuplot > pwd
```

Isto dará a sua localização atual. Se caso quisermos modificar o diretório basta digitar o comando:

```
gnuplot > cd ‘‘novo diretório’’
```

- Salvando arquivos:

Para salvarmos um arquivo, devemos direcionar a saída padrão para um arquivo, como o exemplo abaixo:

O formato postscript “eps”, é o formato original do \LaTeX . Para exportar o gráfico do Gnuplot para o \LaTeX basta utilizar o comando:

```
gnuplot > set terminal postscript eps color
```

```
gnuplot > set output ‘‘arquivo.eps’’
```

```
gnuplot > replot
```

- Para usarmos títulos acentuados devemos usar o seguinte comando:

`set encoding iso_8859_1`

Exemplo: `set title "Taxa de transmiss\3431 o'`

Resultando o título “Taxa de transmissão”

7.2 Modificações na posição das legendas

Podem ser realizadas através dos comandos: `set key left bottom`, `set key right bottom`, `set key left top` e `set key right top`.

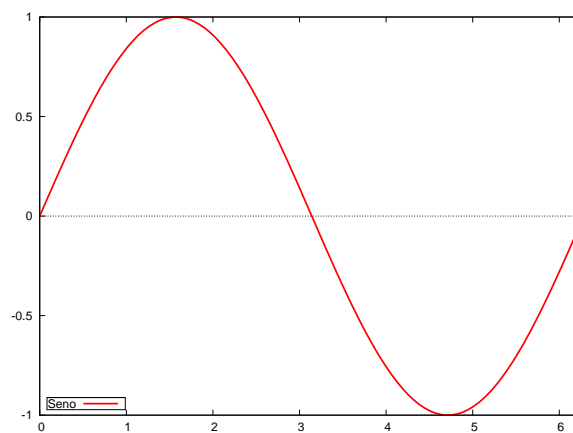


Figura 3: `set key left bottom` (Legenda no canto inferior esquerdo)

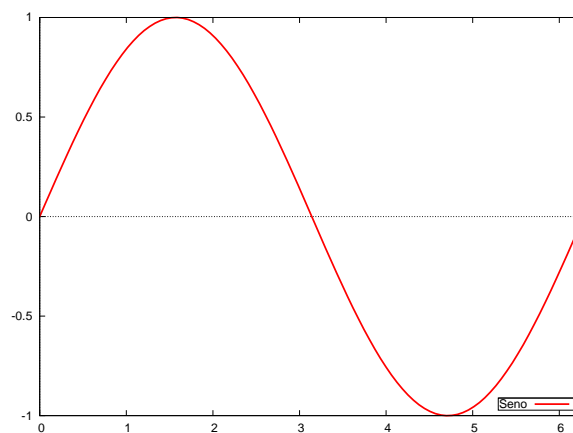


Figura 4: `set key right bottom` (canto inferior direito)

¹O código desses acentos pode ser obtido no site <http://www.ic.unicamp.br/~stolfi/EXPORT/www/ISO-8859-1-Encoding.html>

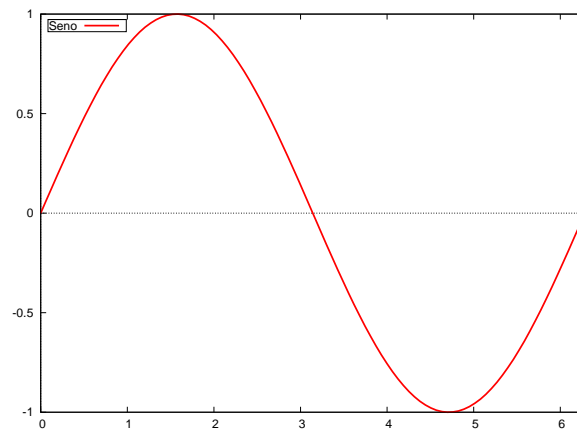


Figura 5: set key left top (canto superior esquerdo)

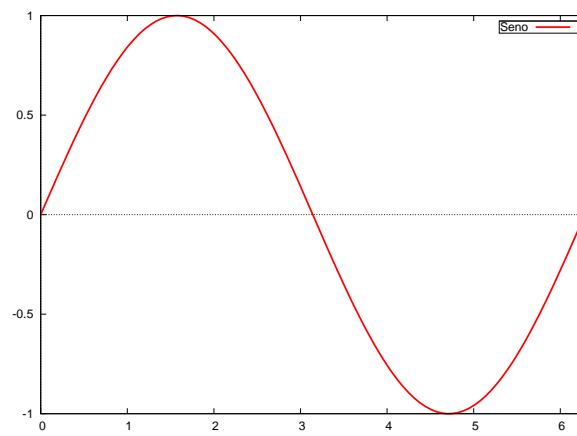


Figura 6: set key righ top (canto superior direito)

Para construir gráficos a partir de um conjunto de dados podemos utilizar plot ou splot, dependendo das dimensões desejadas.

```
gnuplot > plot "dados.txt"
```

Já os gráficos 3D (Ver figura 7) são construídos à partir do comando splot. Como, por exemplo:

$$\log \sqrt{x^2 + y^2}$$

```
gnuplot > splot log(sqrt(x**2+y**2))
```

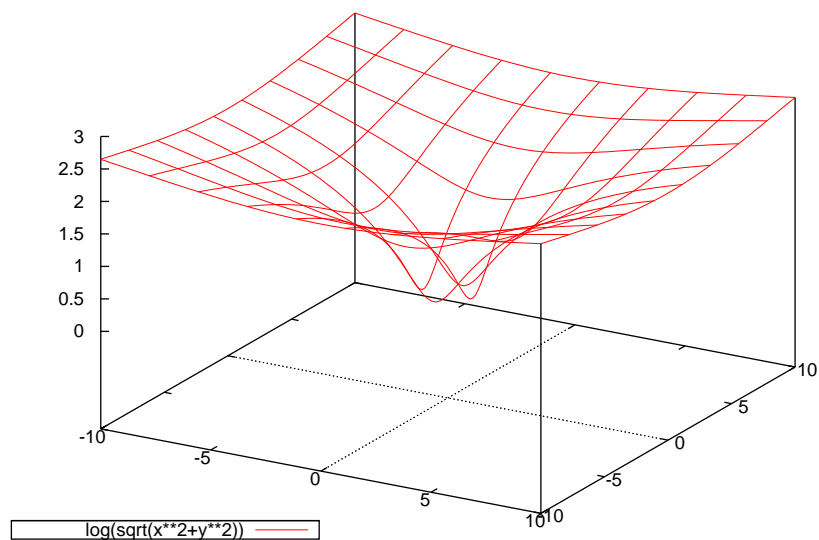


Figura 7: Gráfico 3d

Às vezes é interessante construir gráficos mais complexos, usando duas ou mais funções, podemos apenas separar as funções por vírgulas, pode-se usar ',' (vírgula e barra invertida) como terminador de linha e definir a função desejada na linha seguinte ou, outra possibilidade é usar a opção **rep** (de replot) a cada função adicional, onde são apresentadas as funções simultaneamente.

Figura 8 `gnuplot > plot sin(x),cos(x)`

ou apenas usar o comando **replot**

`gnuplot > plot sin(x)`

`gnuplot > replot cos(x)`

Podemos também usar `,` para continuar na outra linha

`gnuplot > plot sin(x), \`
`cos(x)`

Figura 9 $\left\{ \begin{array}{l} \text{gnuplot > set grid} \\ \text{gnuplot > set key box} \\ \text{gnuplot > plot sin(x), \} \\ \text{cos(x/3), \} \\ \text{\frac{x}{14} - 1} \end{array} \right.$

ou $\left\{ \begin{array}{l} \text{gnuplot > set grid} \\ \text{gnuplot > set key box} \\ \text{gnuplot > plot sin(x)} \\ \text{gnuplot > rep cos(x/3)} \\ \text{gnuplot > rep \frac{x}{14} - 1} \end{array} \right.$

O comando: **set key box** inseri uma caixa na legenda.

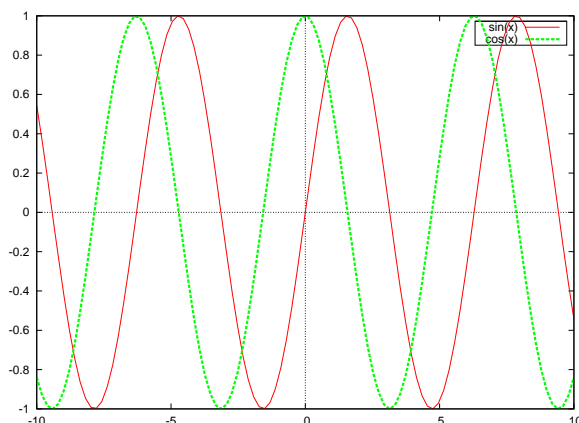


Figura 8:

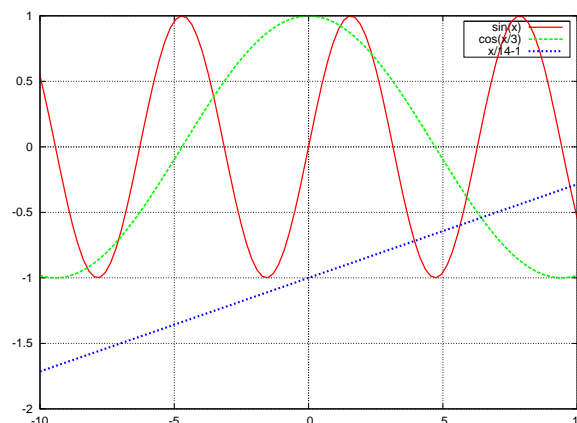


Figura 9:

8 Algumas definições a serem utilizadas com o argumento *with*

Você pode ver que tipos de gráficos o Gnuplot pode desenhar. Para especificar o estilo de gráfico temos os comandos: **lines**, **points**, **linespoints** e **impulses**.

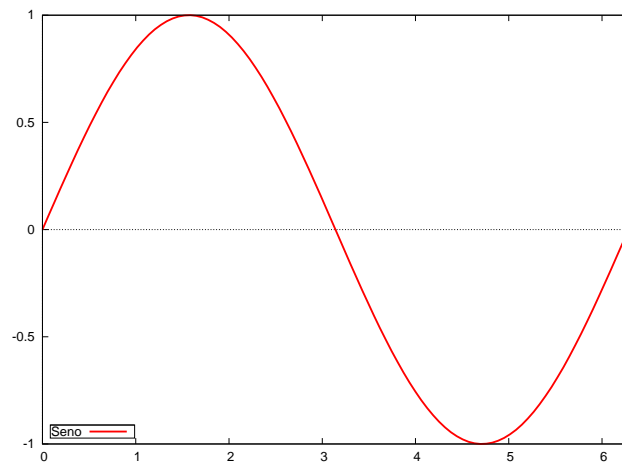


Figura 10: **lines**: Utiliza apenas uma linha que liga todos os pontos. Comando: **plot sin(x) with lines**.

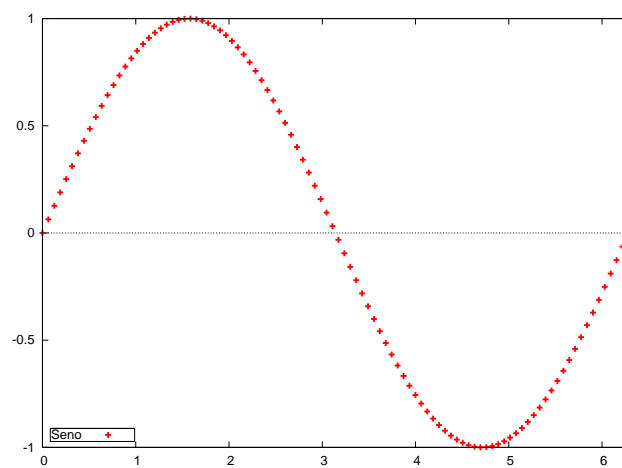


Figura 11: **points**: utiliza pontos com formatos diferentes como quadrados, sinais de mais, diamantes, asteriscos,... Comando: **plot sin(x) with points**.

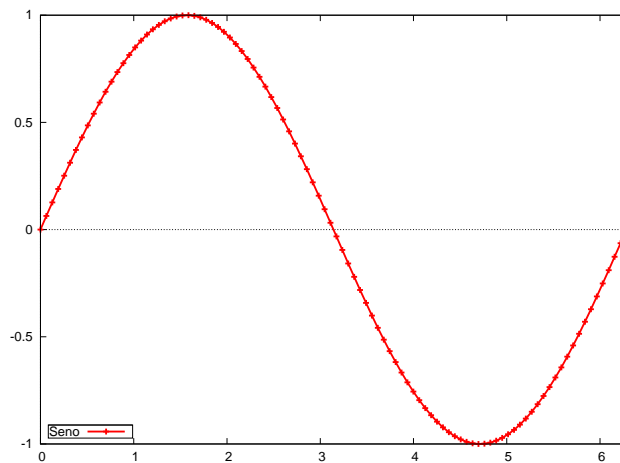


Figura 12: **linespoints**: é a combinação dos dois itens anteriores; utiliza uma linha que liga os pontos e além disso mostra os pontos como no item anterior. Comando: **plot sin(x) with linespoints**.

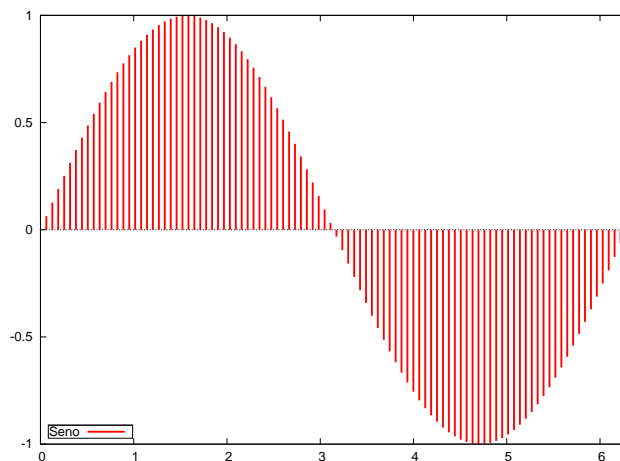


Figura 13: **impulses**: O gráfico aparece “rachurado”. Comando: **plot sin(x) with impulses**.

9 Redirecionamento de *Output*

Quando estamos trabalhando com o **Gnuplot**, a saída padrão do gráfico é para o **display**, ou seja, ele desenha o gráfico em uma nova janela e a exibe no monitor. Na maior parte dos casos, desejamos gerar gráficos em arquivos que possamos incluir em algum trabalho, como relatórios. Para isto o Gnuplot permite o redirecionamento de output para um arquivo, ou seja, permite gerar o gráfico em um arquivo cujo formato é reconhecido por outros programas, como o \LaTeX .

Para isto, devemos especificar o **tipo de terminal** que será utilizado, uma lista pode ser obtida a partir do comando dentro do Gnuplot.

`gnuplot > ?term` (Figura:14)

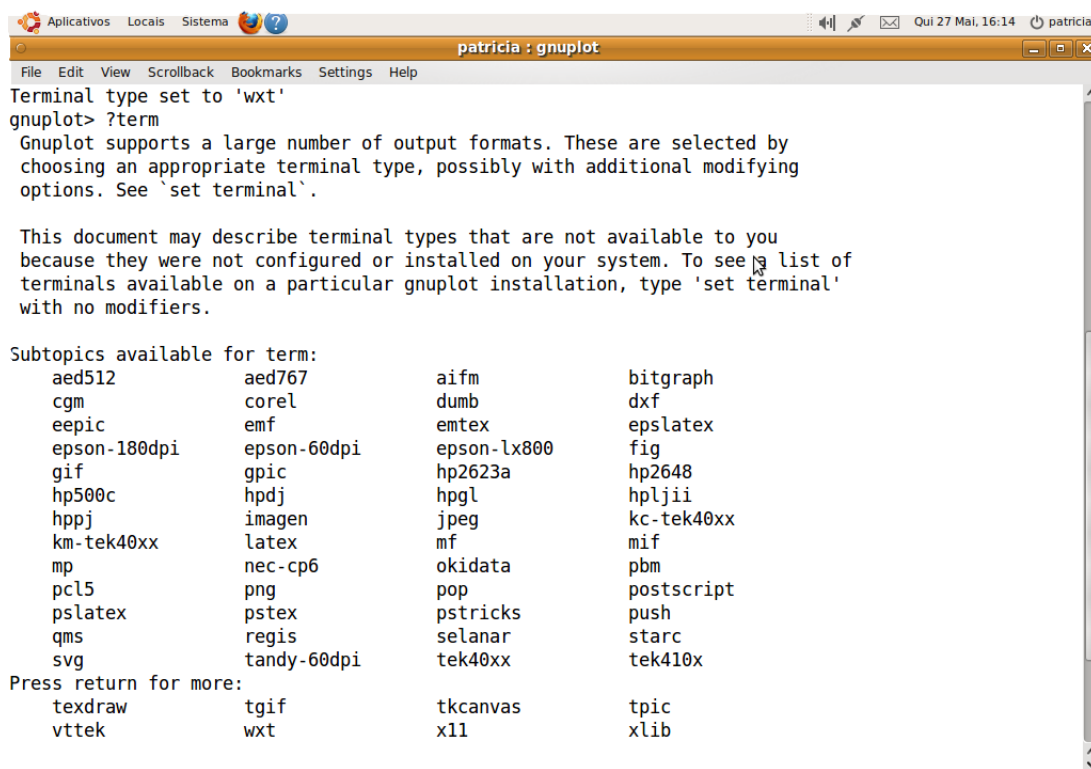


Figura 14:

10 Trabalhando a Matemática no Gnuplot

Vamos traçar o gráfico da função polinomial $9x^3 + \frac{x^2}{2} - \frac{7x}{8} + 4$. Observe que definimos o nome da função no caso, $f(x)$ e em seguida plotamos a mesma pelo seu nome.

```
gnuplot>f(x)= 9*x**3+0.5*x**2-7.0/8.0*x+4
gnuplot>plot f(x)
```

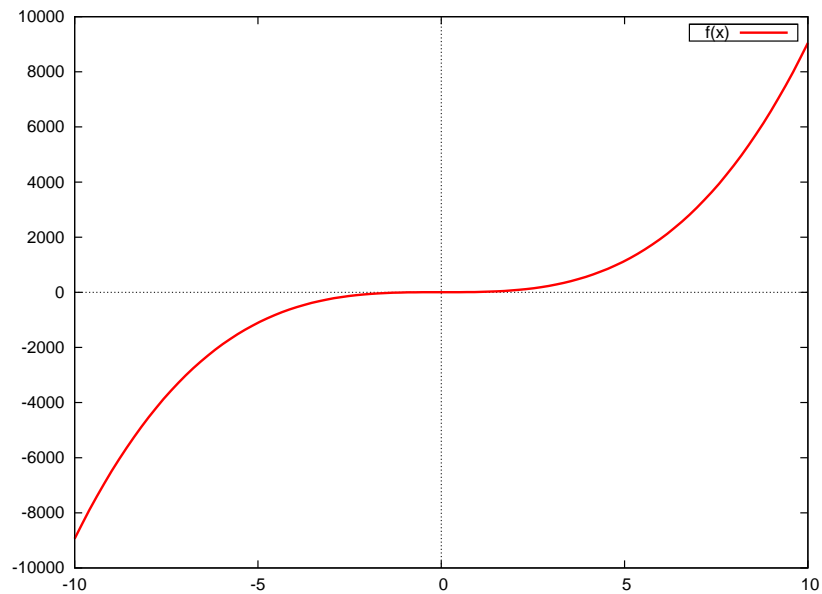


Figura 15: Função Polinomial

Agora vamos fixar o domínio $[-1 : 1] \times [-1 : 1]$ e plotar três superfícies sem grades ou marcas, com estilos diferentes. Com os comando **set xrange** e **set yrange**, podemos determinar o intervalo de variação (domínio) das variáveis x e y no gráfico. O intervalo deve ser escrito entre colchetes e os números separados por dois pontos $[:]$. Assim:

$$xy, x^2 + y^3 \text{ e } x^3 + y^2.$$

```
gnuplot>set xrange [-1:1]
gnuplot>set yrange [-1:1]
gnuplot>splot x*y with lines, x**2*y**3 with linespoints, x**3*y**2 with points
Obtemos a figura 16.
```

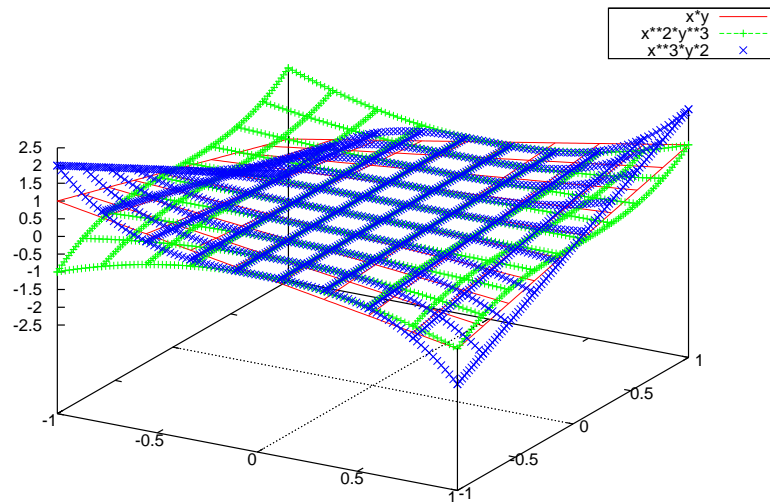


Figura 16:

Para traçar o gráfico de uma função, o Gnuplot escolhe alguns pontos e calcula o valor da função nestes pontos. A quantidade padrão de pontos escolhidos no eixo x para gráficos 2D é 100. No caso de gráficos tridimensionais a quantidade padrão de pontos também é de 100, sendo tomados 10 pontos no eixo x e no eixo y. Esta quantidade pode ser alterada através dos comandos **set samples n** para gráficos no plano e **set isosamples k,m** para gráficos no espaço, sendo que n é o número de pontos no eixo x e k, m correspondem ao número de pontos nos eixos x e y, respectivamente.

Uma verificação matemática, são para obter as curvas de níveis.

Para obter as curvas de nível de $z = xy$, utilizamos o **set contour** para curva de nível com altura dada, melhoramos a malha gráfica com **samples** e **isosamples**, para nomear os eixos x, y e z usamos o **xlabel**, **ylabel** e **zlabel**, respectivamente. Observe que aparecerão suas respectivas curvas de nível no plano.

```
gnuplot>set contour
gnuplot>set samples 20
gnuplot>set isosamples 21
gnuplot>set title "Curvas de nível"
gnuplot>splot x*y
```

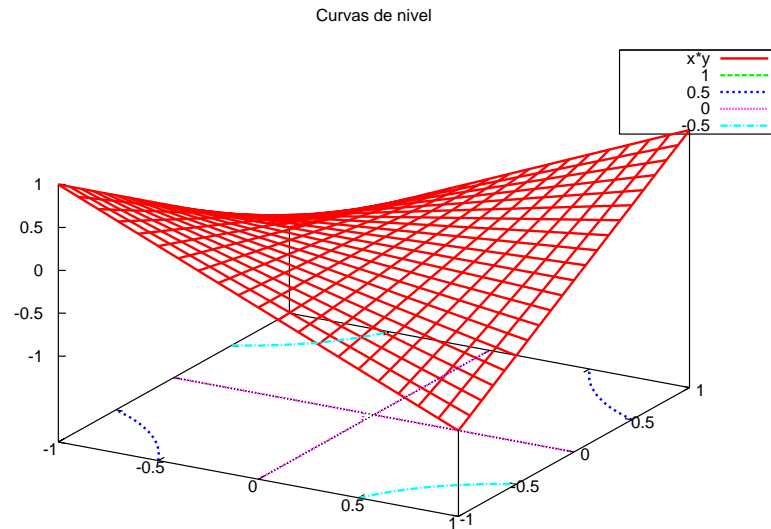



Figura 17: Curva de nível $z = xy$

Agora as curvas de nível podem aparecer na superfície e no plano, fazemos isso com os comandos **set contour surface** onde aparecerá a curva de nível na superfície e o **set contour both** que mostra a curva de nível no plano.

$$x^2 - y^2$$

```
gnuplot>set contour surface
gnuplot>set contour both
gnuplot>plot x**2-y**2
```

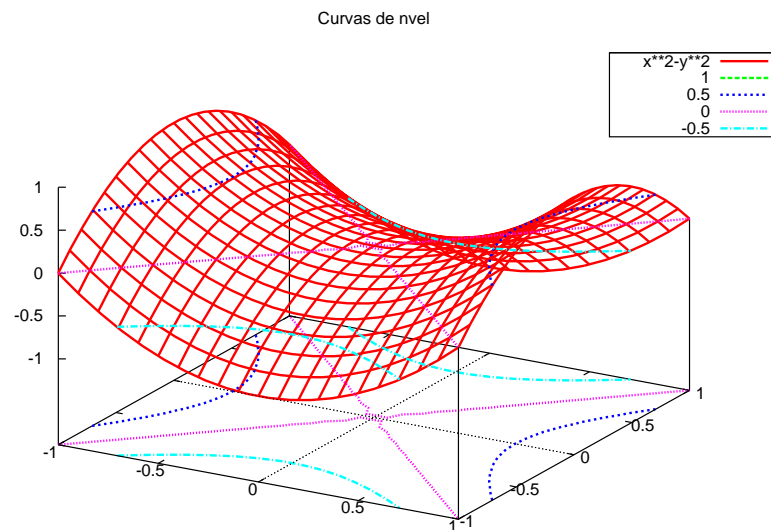


Figura 18:

Uma função definida por partes tem a forma:

$$f(x) = \begin{cases} \sin(x) & \text{se } x < 0, \\ 5 - x & \text{se } x > 0. \end{cases}$$

onde cada parte é associada a um intervalo. A notação 1/0 usada abaixo representa o infinito no Gnuplot.

```
set samples 300
f(x) = x < 0 ? sin(x) : x > 0 ? 5-x : 1/0
plot f(x)
```

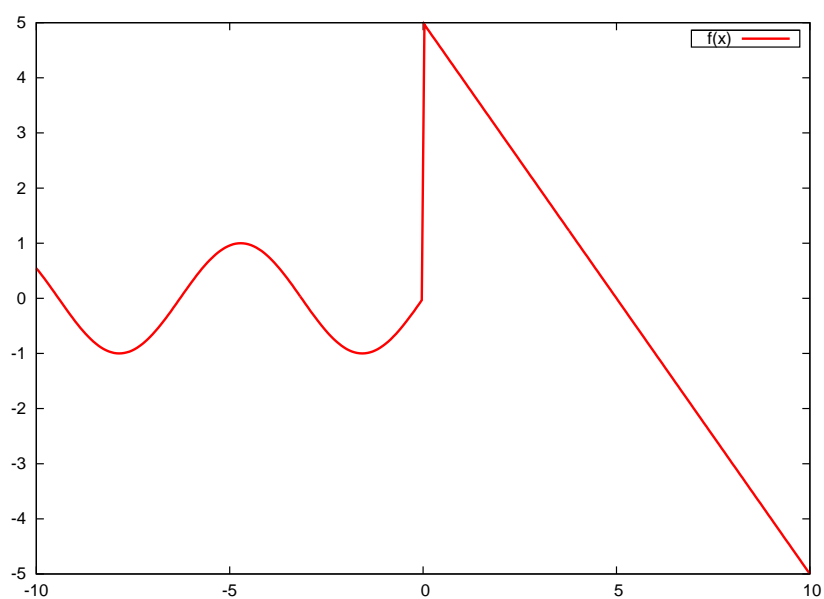


Figura 19: Função definida por partes.

Curvas parametrizadas no espaço tridimensional.

Para plotar curvas parametrizadas no espaço \mathbb{R}^3 , da forma $r(t) = (x(t), y(t), z(t))$, usaremos um truque que não está no Help do Gnuplot, mas que é similar a plotagem de uma superfície parametrizada. Desse modo, definiremos o padrão como:

```
gnuplot> set parametric
```

Para plotar a curva helicoidal $r(t) = (\cos(t), \sin(t), t)$ com t no domínio $[0, 4\pi]$, definiremos esse domínio com **set urange**, substituiremos o parâmetro t pelo parâmetro u , que é o padrão para plotar superfícies parametrizadas.

```
gnuplot> set urange [0:4*pi]
```

```
gnuplot> splot cos(u), sin(u), u
```

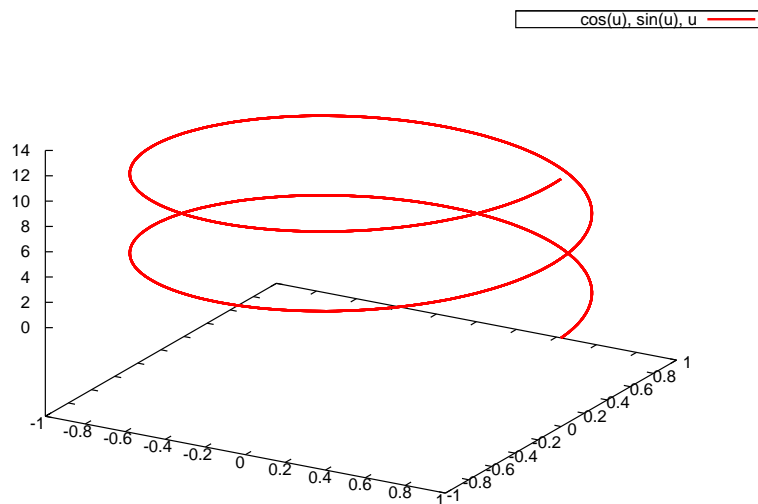


Figura 20:

A expressão paramétrica de um círculo é:

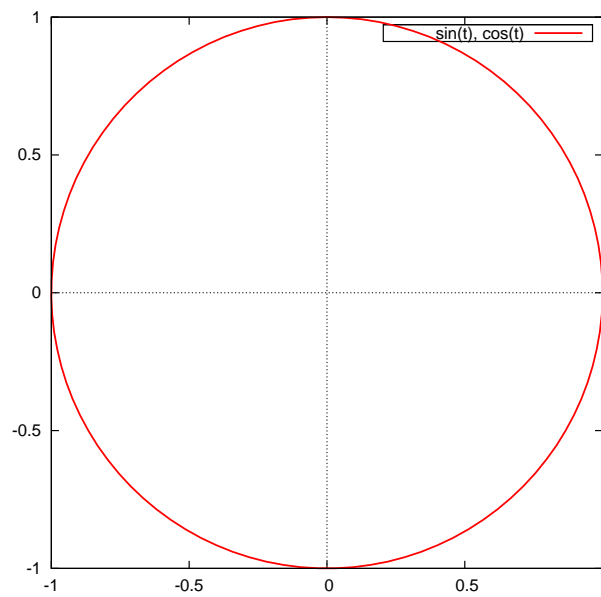
$$\begin{cases} x = \sin(t) \\ y = \cos(t), \end{cases} \text{ com } 0 < t < 2\pi.$$

```
gnuplot > set parametric
```

```
gnuplot > set xrange [-1:1]
```

```
gnuplot > set yrange [-1:1]
```

```
gnuplot > plot [0:2*pi] sin(t),cos(t) linewidth 4
```

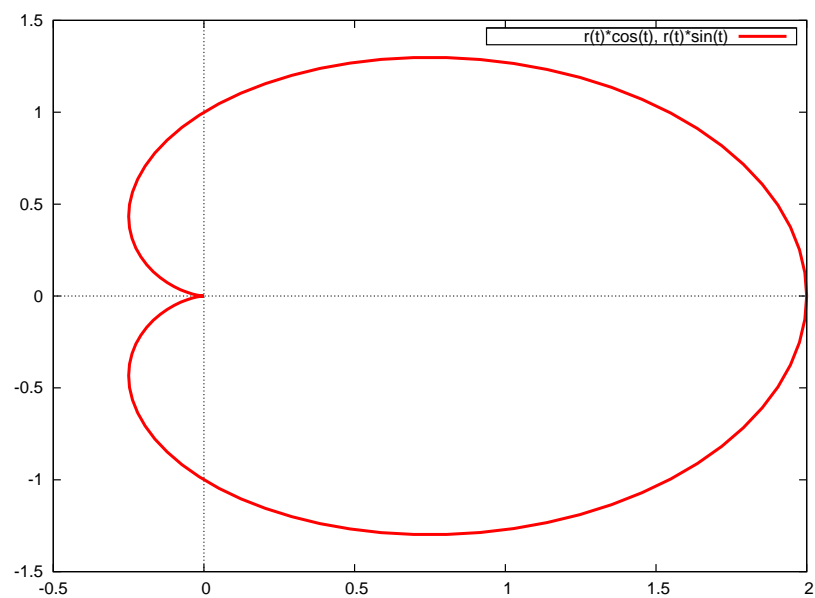


Exemplo 10.1 Agora observemos $r(t) = 1 + \cos(t)$, a chamada cardióide.

```
gnuplot > set parametric
```

```
gnuplot > r(t) = 1+cos(t)
```

```
gnuplot > plot [0:2*pi] r(t)*cos(t),r(t)*sin(t)
```



Derivadas

Dada uma função f derivável no ponto $a \in \text{Dom}(f)$, sabemos que a reta tangente ao gráfico de f no ponto $(a, f(a))$ tem a seguinte equação:

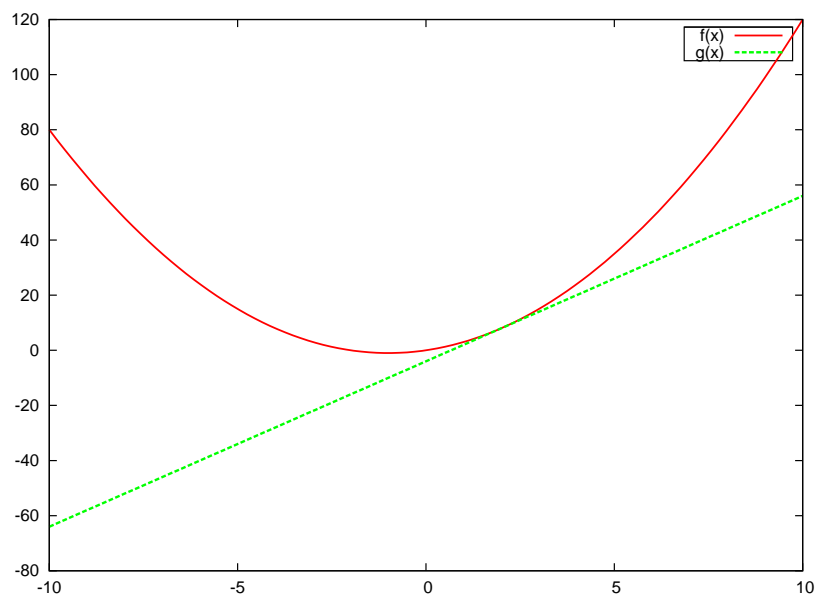
$$y - y_0 = m(x - a)$$

onde $y_0 = f(a)$ e $m = f'(a)$. Assim, podemos escrever a equação da reta tangente ao gráfico de f no ponto $(a, f(a))$ da seguinte forma:

$$g(x) = f(a) + f'(a) \cdot (x - a)$$

Usando o Gnuplot para visualizar o gráfico de uma função e a reta tangente num ponto $(a, f(a))$ pertencente ao gráfico. Vamos tomar a função $f(x) = x^2 + 2x$. Então sua derivada é $f'(x) = 2x + 2$. Para visualizar a reta tangente ao gráfico de f no ponto $(2, f(2))$, passemos ao Gnuplot os seguintes comandos:

```
gnuplot > reset
gnuplot > f(x)=x**2+2*x
gnuplot > df(x)=2*x+2
gnuplot > g(x)=f(2)+df(2)*(x-2)
gnuplot > plot f(x), g(x)
```



Superfícies

A visualização de superfícies é análoga à visualização de curvas planas. Neste caso o comando básico é `splot`.

Na figura 21 é mostrada uma superfície dada pela função $f(x, y) = x + y$, que tem como gráfico um plano. Basta considerar $f(x, y) = z$, temos:

$$x + y - z = 0$$

Para constatar com o Gnuplot, façamos:

```
gnuplot > f(x,y)=x+y
```

```
gnuplot > splot f(x,y)
```

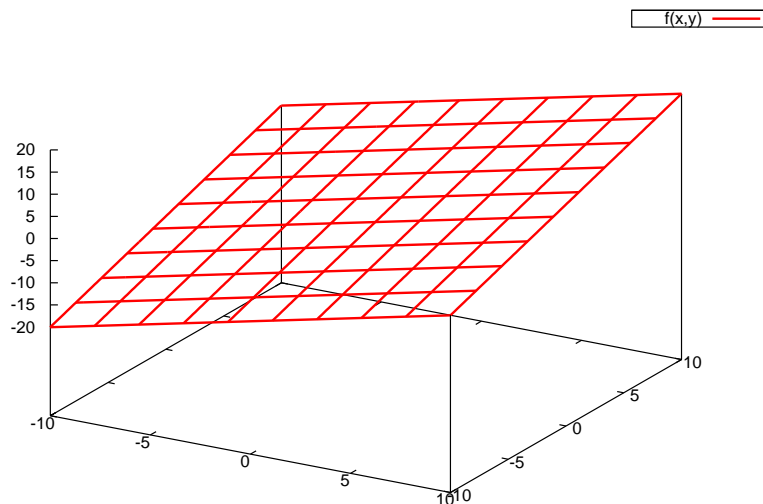


Figura 21:

Agora podemos observar outras superfícies com comandos diferentes.

Exemplo 10.2 (Parabolóide) O gráfico da função $f(x, y) = x^2 + y^2$ o que chamamos de parabolóide. Com o Gnuplot:

O comando `set isosample 20,20` é para melhorar a malha, deixá-la mais fechada. Já o comando `set hidden3d` é para diferenciar as faces da superfície.

```
gnuplot > set isosample 20,20
```

```
gnuplot > set hidden3d
```

```
gnuplot > f(x,y)=x**2+y**2
```

```
gnuplot > splot f(x,y),0
```

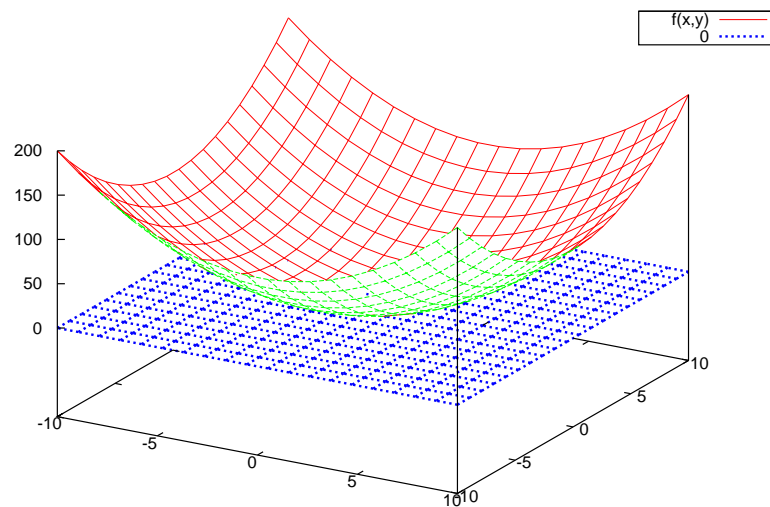


Figura 22:

Se considerarmos a função $f(x,y) = x^2 + (y-5)^2$ teremos uma pequena modificação no ponto mais baixo do parabolóide. Antes, o “bico” do parabolóide era o ponto $(0,0,0)$ no plano XY , agora será o ponto $(0,5,0)$.

`gnuplot > f(x,y),5`

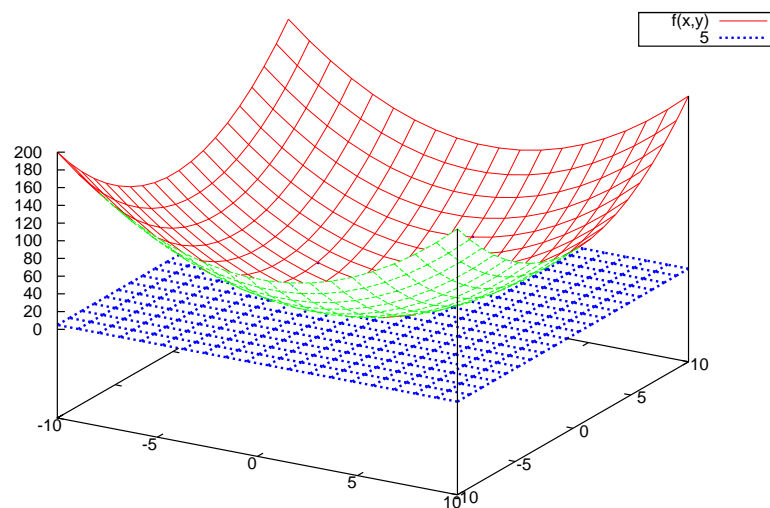


Figura 23:

Agora, considerando a função $f(x,y) = x^2 + y^2 + 50$, o bico do parabolóide será o ponto $(0,0,50)$. Basta digitar o comando

`gnuplot > splot f(x,y),50`

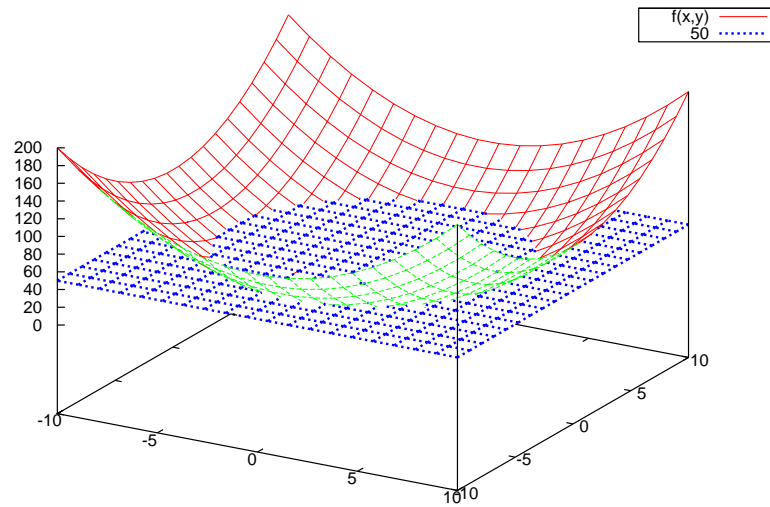


Figura 24:

Exemplo 10.3 (Sela) Considere a função $f(x,y) = x^2 - y^2$ e passemos ao Gnuplot os seguintes comandos:

```
gnuplot > set isosample 20,20
gnuplot > set hidden3d
gnuplot > f(x,y)=x**2-y**2
gnuplot > splot f(x,y)
```

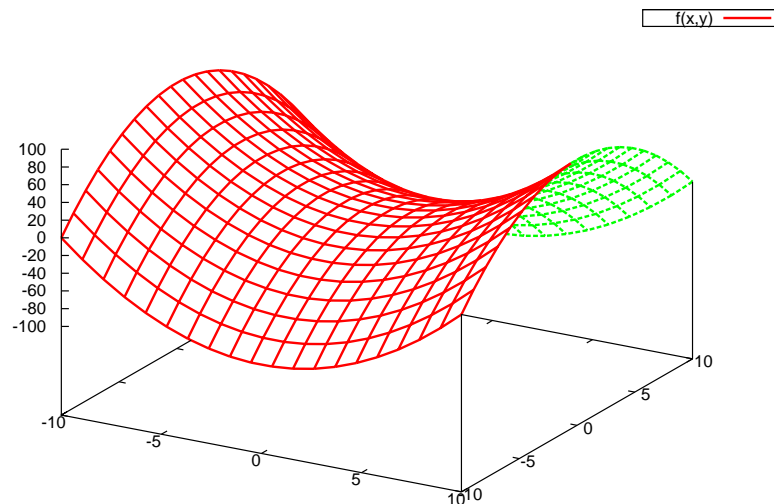


Figura 25:

Observe que cortes paralelos ao plano XZ nos dão parábolas com concavidade para cima, pois isso equivale a y constante em $f(x, y)$, ou seja,

$$f(x, y) = f(x, k) = x^2 + k$$

Cortes paralelos ao plano YZ nos dão parábolas com concavidade para baixo, uma vez que nesse caso, x é constante e temos:

$$f(x, y) = f(k, y) = k - y^2$$

Para ver isso, passe ao Gnuplot os comandos:

```
gnuplot > splot f(x,2)
```

```
gnuplot > splot f(3,y)
```

Exemplo 10.4 (Chapéu de Cowboy) É a aparência do gráfico da função:

$$f(x, y) = \frac{\sin(\sqrt{x^2 + y^2})}{\sqrt{x^2 + y^2}}$$

Que para o Gnuplot passaremos:

```
gnuplot > set isosample 40,20
```

```
gnuplot > set hidden3d
```

```
gnuplot > set zrange[-0.5:1.5]
```

```
gnuplot > f(x,y)=(sin(sqrt(x**2+y**2)))/(sqrt(x**2+y**2))
```

```
gnuplot > splot f(x,y)
```

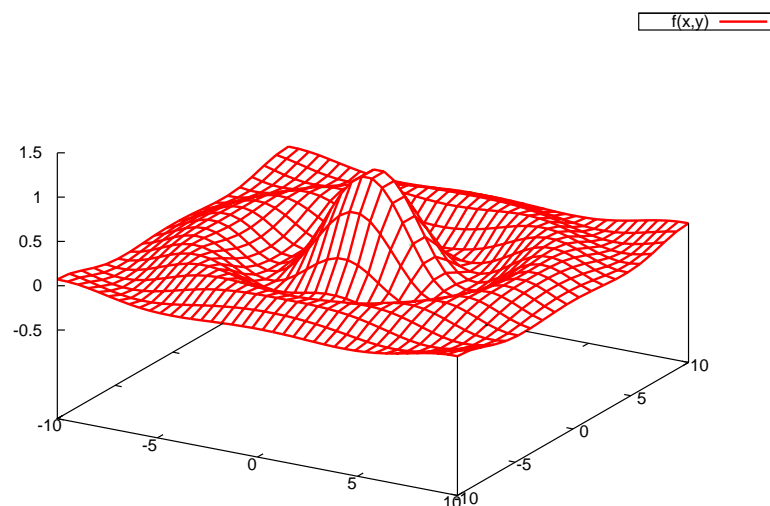


Figura 26:

Exemplo 10.5 (Caixa de ovos) O gráfico da função $f(x, y) = \sin x + \cos x$ lembra bastante esse objeto:

```
gnuplot > set isosample 60,20  
gnuplot > set hidden3d  
gnuplot > set xrange[-2:2]  
gnuplot > f(x,y)=sin(x)+cos(x)  
gnuplot > splot f(x,y)
```

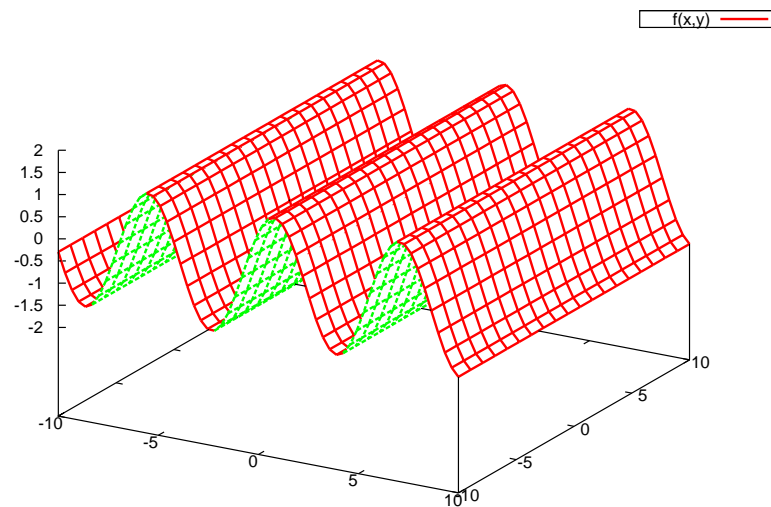


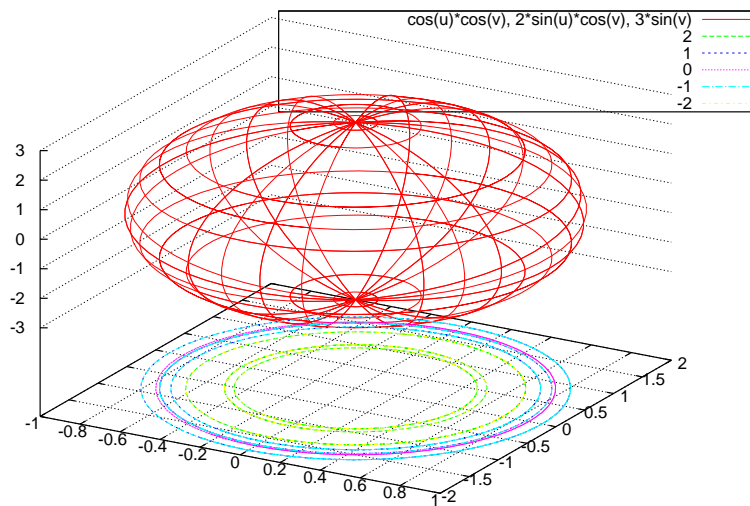
Figura 27:

11 Superfícies Parametrizadas

Esfera Parametrizada

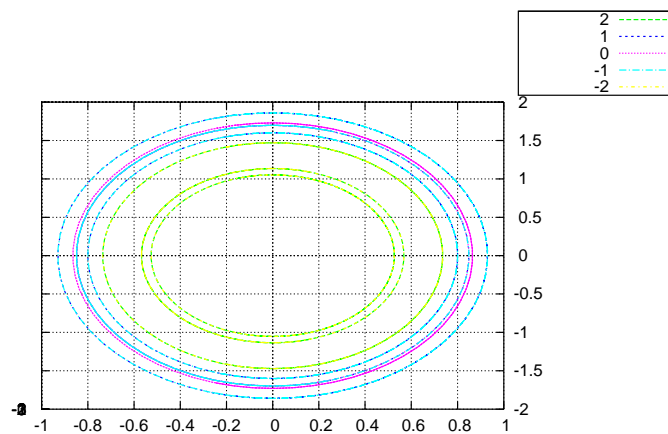
$$(\cos(u)\cos(v), 2\sin(u)\cos(v), 3\sin(v))$$

```
gnuplot>set parametric
gnuplot> set zeroaxis
gnuplot > set key box
gnuplot > splot cos(u)*cos(v),2*sin(u)*cos(v),3*sin(v)
gnuplot > set contour base
gnuplot > set grid xtics ytics ztics
```



Se quisermos verificar somente as curvas de nível da superfície, adicionamos os seguintes comandos:

```
gnuplot > unset surface Este comando esconde a superfície
gnuplot > set view 0,0 Este comando mostra como queremos visualizar o gráfico.
```

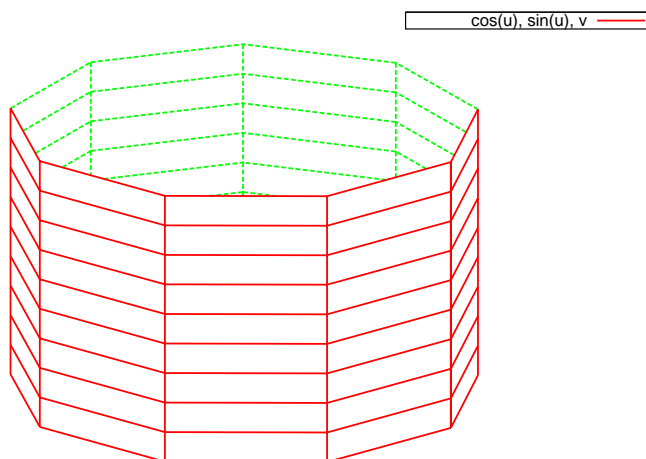


Cilindro Circular Reto

Uma observação diferente nesta figura é em relação ao comando `unset tics` e `unset border`, com estes comando escondemos os eixos e observamos somente a figura.

$$(\cos(u), \sin(u), v)$$

```
gnuplot>set parametric
gnuplot>set urange [0:2*pi]
gnuplot>set hidden3d
gnuplot>unset border
gnuplot>unset tics
gnuplot>set vrange [-4:4]
gnuplot>splot cos(u), sin(u), v
```

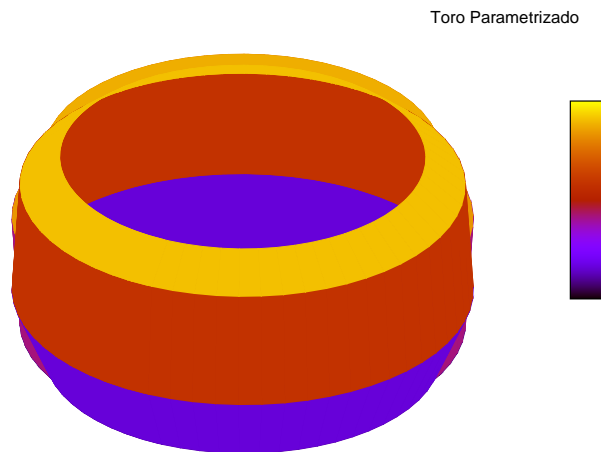


Toro Parametrizado

Uma outra forma de visualização de gráficos é o efeito usado com o `with`, usando o comando `with pm3d`, é algo como uma distribuição de densidade da cor, o que significa que cada pedaço de superfície colorida é mostrada na parte inferior do gráfico; e ainda se colocarmos logo após do `plot` ou `splot` junto com a função que queremos plotar o comando `title 'Nome'` isto modificará o que aparece na legenda, ou seja, é uma edição do que queremos nomear na legenda, observemos o seguinte efeito:

$$((1 - 0.2\cos(v))\cos(u), (1 - 0.2\cos(v))\sin(u), 0.2\sin(v))$$

```
gnuplot>set parametric
gnuplot>splot (1-0.2*cos(v))*cos(u), (1-0.2*cos(v))*sin(u), 0.2*sin(v) with
pm3d title 'Toro Parametrizado'
```



Prisma Hexagonal

$$(\cos(v)^3 \cos(u)^3, \sin(v)^3 \cos(u)^3, \sin(u)^3)$$

```
gnuplot>set parametric
gnuplot>unset border
gnuplot>unset tics
gnuplot>set isosamples 100,20
gnuplot>splot cos(v)**3*cos(u)**3, sin(v)**3*cos(u)**3, sin(u)**3
```

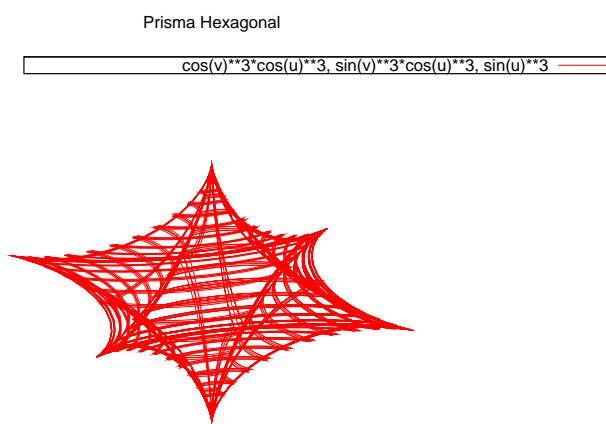


Figura 28:

Hélice Parametrizada

$$((1 - 0.1\cos(v))\cos(u), (1 - 0.1\cos(v))\sin(u), 0.1(\sin(v) + u/1.7 - 10))$$

```
gnuplot>set parametric
gnuplot>set isosamples 100,20
gnuplot>set urange [0:10*pi]
gnuplot>set vrange [0:2*pi]
gnuplot>unset border
gnuplot>unset tics
gnuplot>set hidden3d
gnuplot>
splot (1-0.1*cos(v))*cos(u),(1-0.1*cos(v))*sin(u),0.1*(sin(v)+u/1.7-10)
```

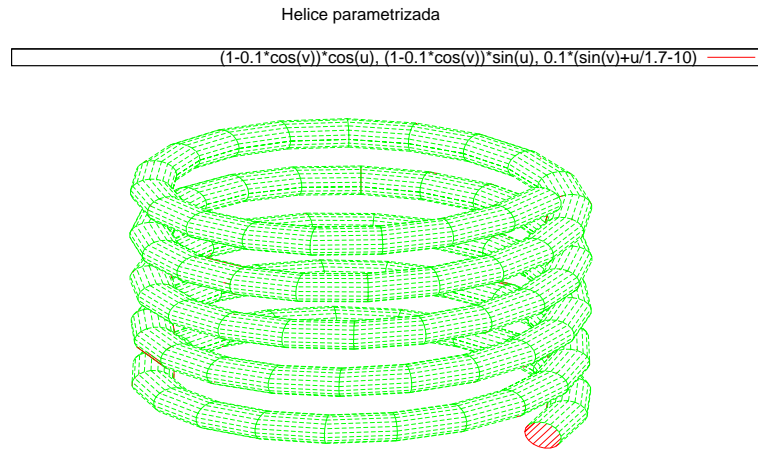


Figura 29:

Concha Parametrizada

$$\left(\cos(u) \cdot u \cdot \frac{1 + \cos(v)}{2}, \sin(v) \frac{u}{2}, \sin(u) \cdot u \cdot \left(\frac{1 + \cos(v)}{2} \right) \right)$$

```
gnuplot>set isosamples 40,20
gnuplot>set urange [0:2*pi]
gnuplot>set vrange [0:2*pi]
gnuplot>set zrange [-3:1.5]
gnuplot>splot cos(u)*u*(1+cos(v)/2), sin(v)*u/2, sin(u)*u*(1+cos(v)/2)
gnuplot> set autoscale z
```

Concha Parametrizada

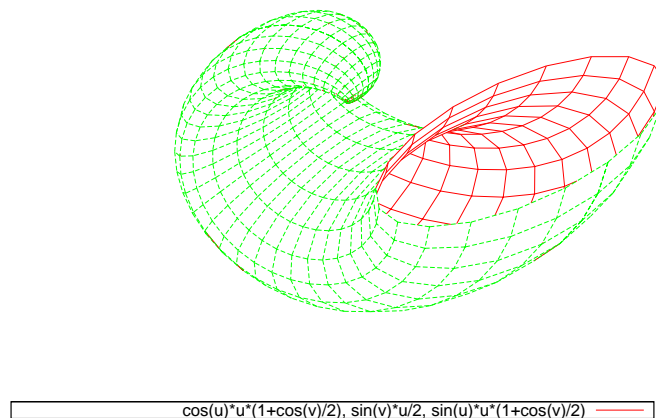


Figura 30:

Dois Toros Parametrizados Entrelaçados

$$(\cos(u) + .5\cos(u)\cos(v), \sin(u) + .5\sin(u)\cos(v), .5\sin(v), 1 + \cos(u)$$

$$+.5\cos(u)\cos(v), .5\sin(v), \sin(u) + .5\sin(u)\cos(v))$$

```
gnuplot>set urange [-pi:pi]
gnuplot>set vrange [-pi:pi]
gnuplot>set isosamples 50,20
gnuplot>splot cos(u)+.5*cos(u)*cos(v), sin(u)+.5*sin(u)*cos(v), .5*sin(v)
with lines, 1+cos(u)+.5*cos(u)*cos(v), .5*sin(v), sin(u)+.5*sin(u)*cos(v) with
lines
```

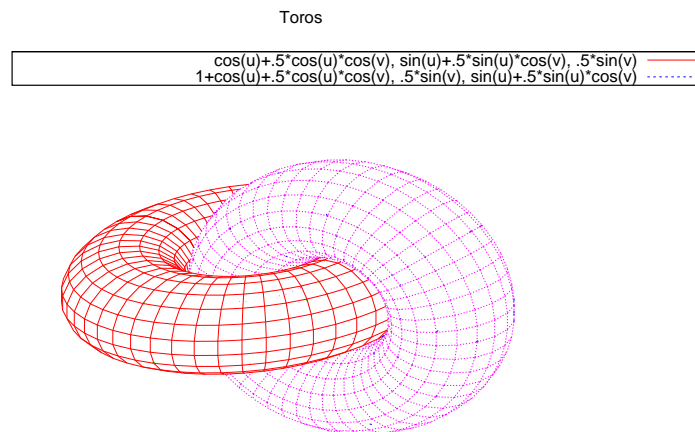


Figura 31:

Interseção de Dois Cilindros

$$(u, \sqrt{4-u^2}, v, u, -\sqrt{4-u^2}, v, \sqrt{4-v^2}, u, v, -\sqrt{4-v^2}, u, v)$$

Comandos:

```
set parametric
```

```
set hidden3d
```

```
set isosamples 30,30
```

```
set urange [-2:2]
```

```
set vrange [-2:2]
```

```
set zrange [-2:2]
```

```
unset tics
```

```
splot u, sqrt(4-u**2), v, u, -sqrt(4-u**2), v, sqrt(4-v**2), u, v, -sqrt(4-v**2), u, v
```

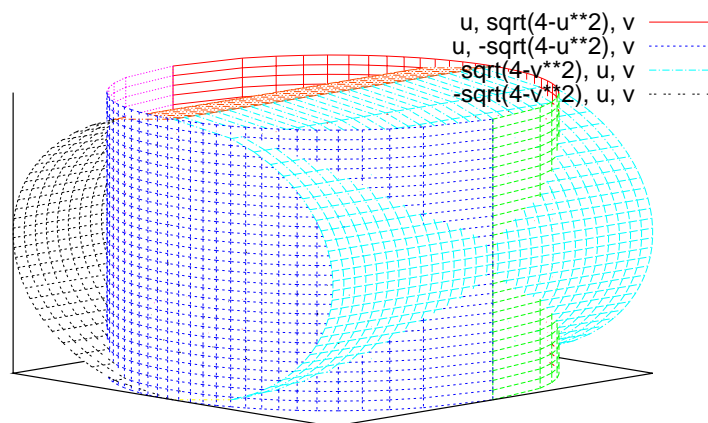


Figura 32:

12 Editando Arquivos com Comandos do Gnuplot

Observamos que até agora, todas as instruções foram feitas diretamente na linha de comando. Para que não seja necessário digitarmos todos os comandos a cada novo gráfico a ser feito, é possível criarmos um arquivo em um editor de texto, contendo todas as instruções para a construção do gráfico.

Depois de criado, o arquivo é lido e interpretado pelo Gnuplot através do comando `load`, cuja sintaxe é dada a seguir:

```
gnuplot> load "nome do arquivo".
```

13 Criar arquivo no bloco de notas

Após digitarmos todos os comandos necessários para traçar o gráfico desejado, salvamos o arquivo numa pasta de trabalho. Todos os comandos utilizados, podem ser colocados em um mesmo arquivo (exe1.plt) da seguinte maneira:

```
#  
# arquivo exe1.plt  
# comandos para realizar o gráfico da função  $\cos(2x)$   
#  
set title "teste1-cosseno"  
set xrange [0:2*pi]  
set yrange [-1.2:1.2]  
set xlabel "eixo x"  
set ylabel "eixo y"  
plot cos(2*x)
```

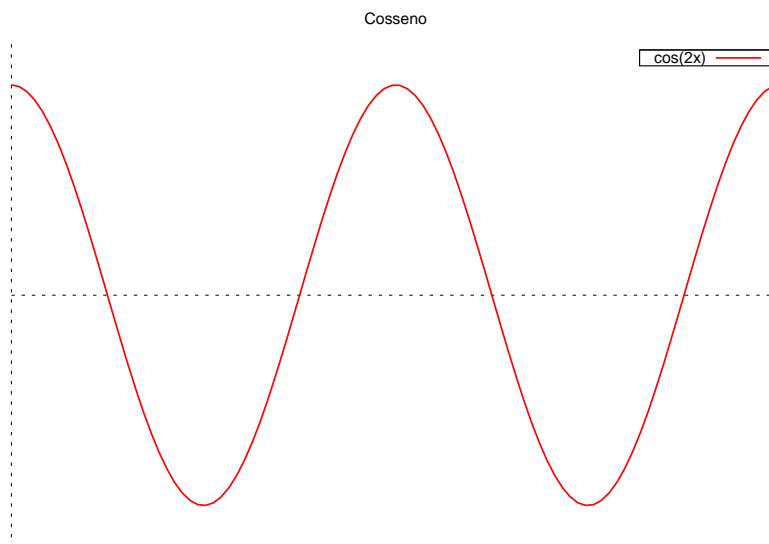


Figura 33: Figura feita com Gnuplot a partir da leitura de dados.

Observação: Se utilizarmos o sinal $\#$ (sustenido) no início de uma linha do arquivo, o Gnuplot não irá ler esta linha, então usamos $\#$ para fazer comentários.

14 Visualizando o Modo Multiplot

O modo multiplot permite a visualização de vários gráficos. Observe este gráfico e seu script:

```
set multiplot  
set size 1,0.5;  
set origin 0.0,0.5; plot sin(x);  
set origin 0.0,0.0; plot cos(x)  
unset multiplot
```

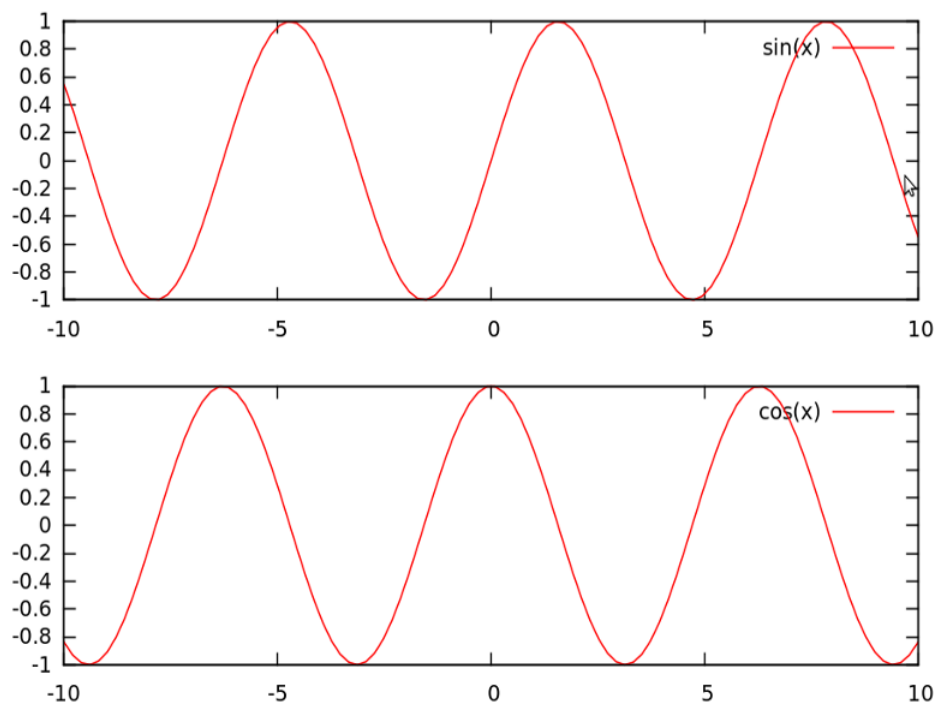


Figura 34:

Veja a função de cada comando: Com o comando **set multiplot** passamos para o modo 'multiplot', **set origin** fixa a origem do gráfico, **set size**, fixa o tamanho do gráfico e **unset multiplot**, sai do modo 'multiplot'.

15 Aplicações Científicas

O aplicativo *gnuplot* é destinado à visualização de gráficos e superfícies, úteis em aplicações científicas nas áreas de física, estatística, engenharias (cartográfica, mecânica, elétrica, ...). Vejamos a seguir algumas dessas aplicações.

Aplicação na Estatística

O gráfico estatístico é uma forma de apresentação dos dados estatísticos, cujo objetivo é o de produzir no investigador ou no público em geral, uma impressão mais rápida e viva do fenômeno em estudo. Os estatísticos usam o *Gnuplot* com o objetivo de obter diversificações gráficas na forma de linhas, barras...

Vejamos uma aplicação clássica na estatística, o gráfico de barras:

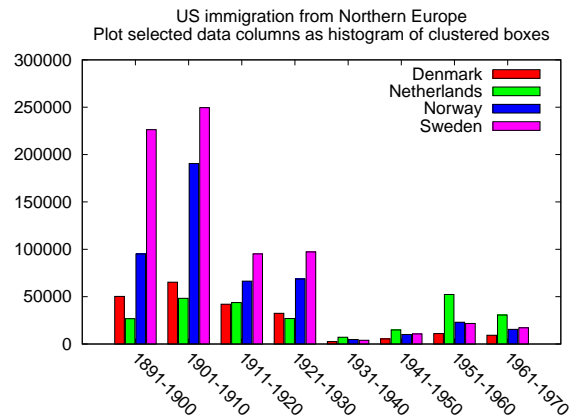


Figura 35:

Comandos:

Para exibir este gráfico, criamos um arquivo de dados 'immigration.dat', o qual terá todas as informações das investigações feita.

# IMMIGRATION BY REGION AND SELECTED COUNTRY OF LAST RESIDENCE													
#	Regiao	Austria	Hungary	Belgica	Czechoslovakia	Dinamarca	Franca	Alemanha	Espanha	Grecia	Irlanda	Italia	
	Netherlands	Norway	Suica	Polonha	Portugal	Romenia	Uniao	Sivitica		Suecia	United_Kingdom		
	Iugoslavia	Other_Europe	TOTAL										
1891-1900	234081	181288	18167	-	50231	30770	505152	15979	388416	651893	26758	95015	226266
96720	27508	12750	505290	8731	31179	271538	-	282	3378014				
1901-1910	668209	808511	41635	-	65285	73379	341498	167519	339065	2045877	48262	190505	
249534	-	69149	53008	1597306	27935	34922	525950	-	39945	7387494			
1911-1920	453649	442693	33746	3426	41983	61897	143945	184201	146181	1109524	43718	66395	95074
4813	89732	13311	921201	68611	23091	341408	1888	31400	4321887				
1921-1930	32868	30680	15846	102194	32430	49610	412202	51084	211234	455315	26948	68531	97249
227734	29994	67646	61742	28958	29676	339570	49064	42619	2463194				
1931-1940	3563	7861	4817	14393	2559	12623	144058	9119	10973	68028	7150	4740	3960
17026	3329	3871	1370	3258	5512	31572	5835	11949	377566				
1941-1950	24860	3469	12189	8347	5393	38809	226578	8973	19789	57661	14860	10100	10665
7571	7423	1076	571	2898	10547	139306	1576	8486	621147				
1951-1960	67106	36637	18575	918	10984	51121	477765	47608	43362	185491	52277	22935	21697
9985	19588	1039	671	7894	17675	202824	8225	16350	1325727				
1961-1970	20621	5401	9192	3273	9201	45237	190796	85969	32966	214111	30606	15484	17116
53539	76065	3531	2465	44659	18453	213822	20381	11604	1124492				

Em seguida passamos os seguintes comandos ao Gnuplot:

set title "US immigration from Northern Europe \nPlot selected data columns as histogram of clustered boxes"

set auto x

set yrange [0:300000]

set style data histogram

set style histogram cluster gap 1

set style fill solid border -1

set boxwidth 0.9

set xtic rotate by -45

set bmargin 10

plot 'immigration.dat' using 6:xtic(1) ti col, " u 12 ti col, " u 13 ti col, " u 14 ti col

Aplicação na Engenharia Cartografica

Uma interessante aplicação do Gnuplot é a visualização de Projeções Cartográficas, podendo ser útil no aprendizado de Cartografia Matemática, uma vez que o aluno pode escrever as equações das projeções, modificar o paralelo padrão, modificar o ponto de tangência, modificar a escala, calcular o fator de escala, visualizar a projeção, dentre outras possibilidades.

A seguir observemos algumas figuras cartográficas. Estas figuras podem ser encontradas na página oficial do Gnuplot (www.gnuplot.info/).

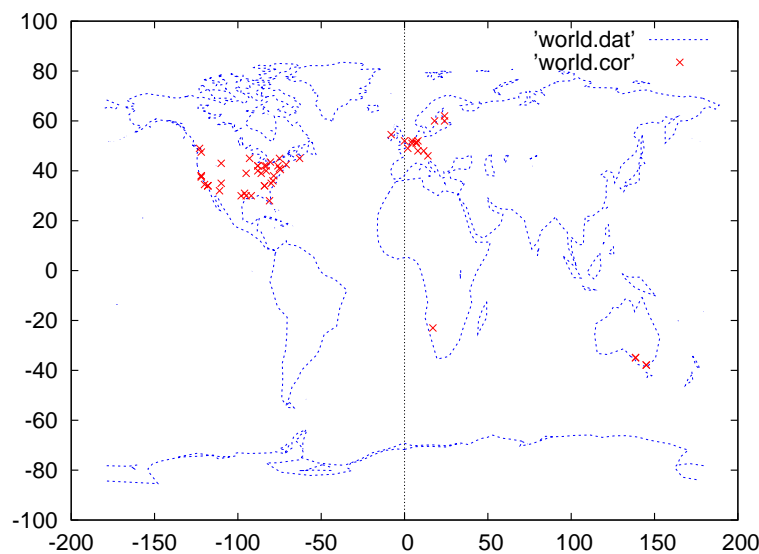


Figura 36:

Versao 3D usando o sistema de coordenadas esfericas

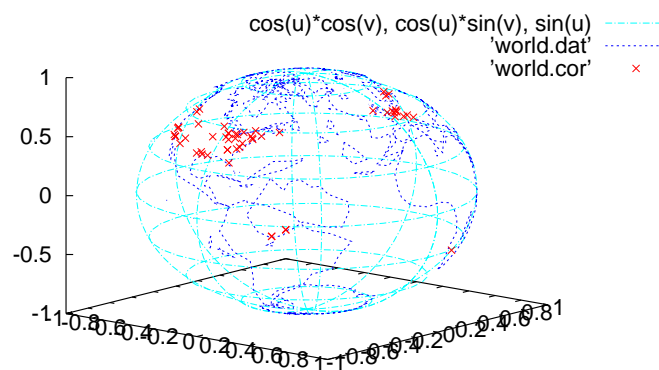


Figura 37:

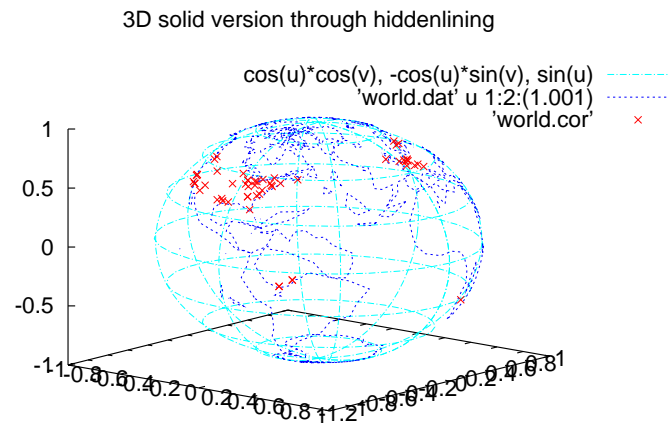


Figura 38:

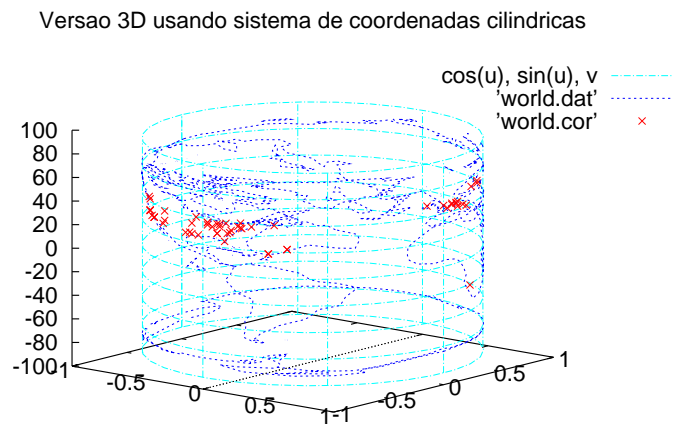


Figura 39:

Para produzirmos as figuras anteriores, teremos que construir três arquivos: *world.dat*, *world.cor* e *world.dem*, estes arquivos podem ser baixados no site do Gnuplot. A partir destes arquivos basta digitar na linha de comando do Gnuplot **load 'world.dem'**, arquivo de dados já criado, que visualizaremos tais figuras.

Considerações Finais

O Gnuplot foi desenvolvido inicialmente por Thomas Williams e Colin Kelly em 1986 com o objetivo de criar um programa que lhes permitisse visualizar graficamente propriedades das equações matemáticas relacionadas ao fenômeno do eletromagnetismo. Primeiramente foi denominado de NEWPLOT, mas ao descobrirem que já existia outro programa com esse mesmo nome, mudaram para Gnuplot. O programa destina-se a criação de gráficos a partir de funções e conjuntos de dados em duas ou três dimensões. O Gnuplot é bastante utilizado devido a sua simplicidade, versatilidade e automatização das tarefas, o que possibilita traçar diversos gráficos a partir da execução de apenas um arquivo de comandos. Segundo Galo (2003) o aplicativo Gnuplot é destinado à visualização de gráficos e superfícies, úteis em aplicações científicas nas áreas de física, matemática, estatística e engenharias. Este aplicativo é de domínio público e tem versões para uma série de sistemas operacionais, tais como: Windows, Unix, Linux.

Referências

- [1] FONTENELE, F. C. F. GNUPLOT: comandos básicos e aplicações em sala de aula. Monografia do Curso de Matemática, 2007.*
- [2] GALO, Mauricio. Instruções iniciais para o uso do Gnuplot. Disponível no site: www.prudente.unesp.br/dcartog/galo/gnuplot*
- [3] Página central do Gnuplot: www.gnuplot.info.*
- [4] Gnuplot um programa para fazer gráficos e alguns cálculos: Disponível no seguinte site: <http://t16web.lanl.gov/Kawano/gnuplot/index-e.html>.*