Universidade Estadual Paulista
FCT – Campus de Presidente Prudente
Departamento de Cartografia

Tutorial: introdução ao uso do aplicativo *Gnuplot*

Autor: Prof. Mauricio Galo

Presidente Prudente
2021

SUMÁRIO

1. APRESENTAÇÃO / CARACTERÍSTICAS / APLICAÇÕES

O aplicativo *gnuplot* é destinado à visualização de gráficos e superfícies, úteis em aplicações científicas nas áreas de física, matemática, estatística, engenharias (cartográfica, mecânica, elétrica, ...), etc. Este aplicativo é de domínio público e tem versões para uma série de sistemas operacionais, entre os quais pode-se citar os seguintes: Windows, Unix, Linux, DOS, etc. Para a obtenção do aplicativo nas diversas plataformas sugere-se a página original:

- http://www.gnuplot.info/

Para alguns exemplos de aplicações são sugeridos os seguintes endereços eletrônicos:

- http://www.duke.edu/~hpgavin/gnuplot.html (manual/tutorial)
- http://seismic.yonsei.ac.kr/gnu_intro.html
- http://www.gnuplotting.org/
- http://www2.fct.unesp.br/docentes/carto/galo/web/gnuplot/fct.htm

Na sequência são apresentados alguns exemplos de gráficos gerados utilizando este aplicativo, para que o leitor tenha uma idéia do seu potencial de uso.

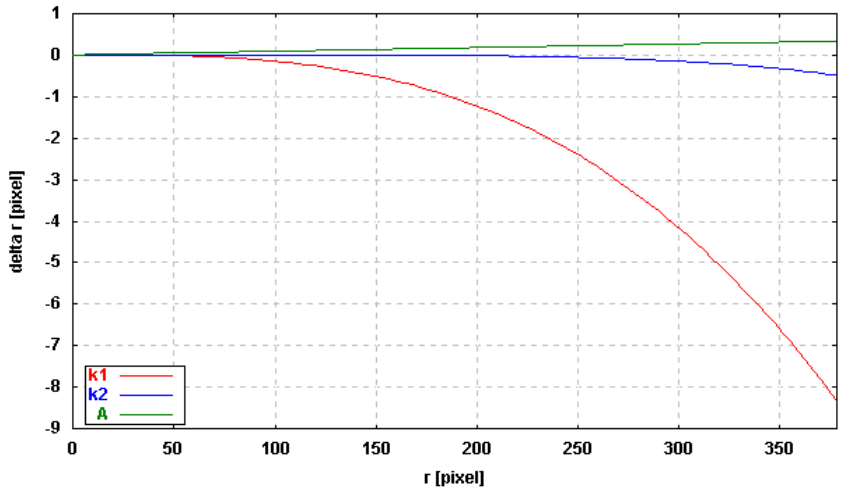


Figura 1.1 - Exemplo de um gráfico mostrando três curvas.

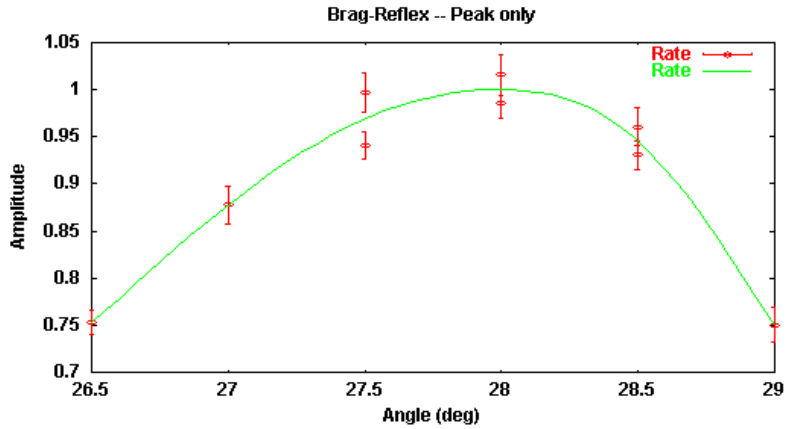


Figura 1.2 - Gráficos de funções com barra de erros (arquivo de demonstração que acompanha o aplicativo).

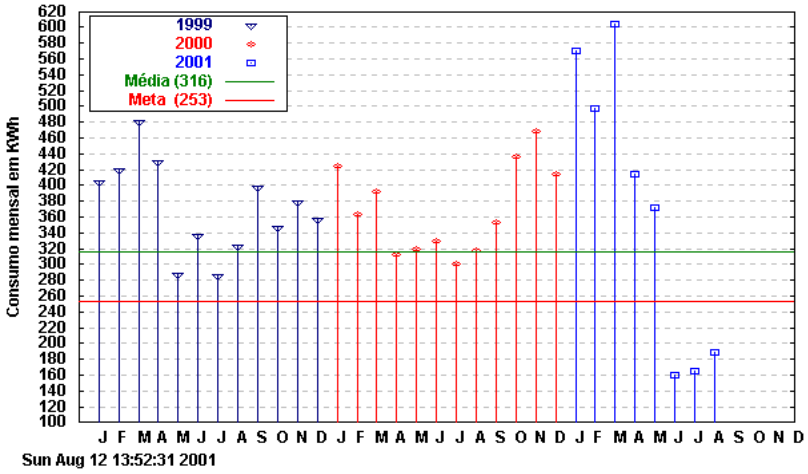


Figura 1.3 - Gráfico de consumo de energia com dados armazenados e lidos em arquivo.

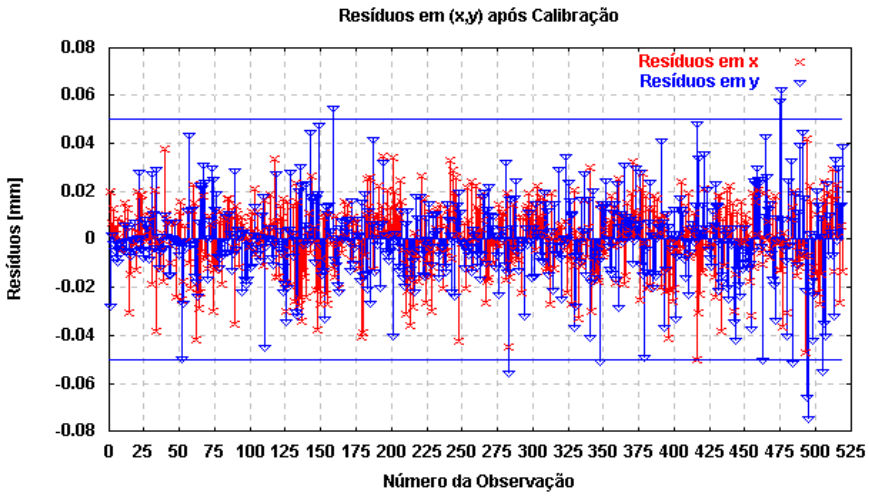


Figura 1.4 - Gráfico mostrando os resíduos lidos a partir de arquivos.

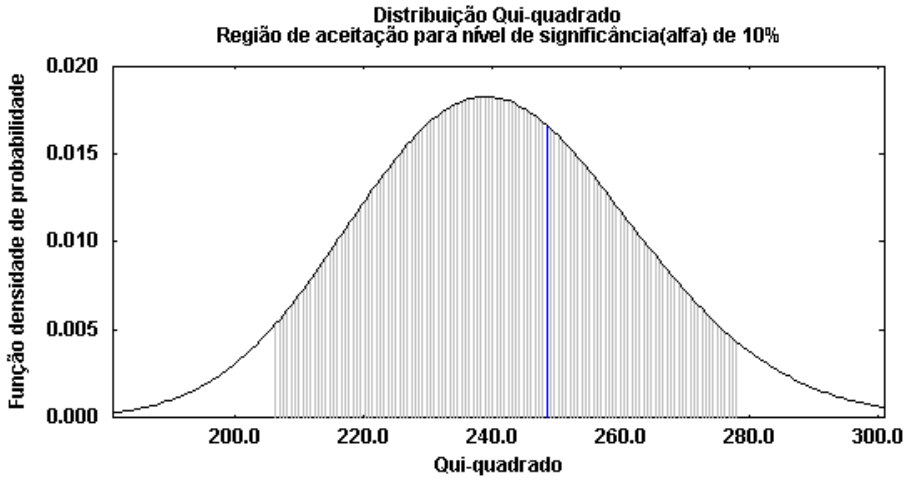


Figura 1.5 - Distribuição χ^2.

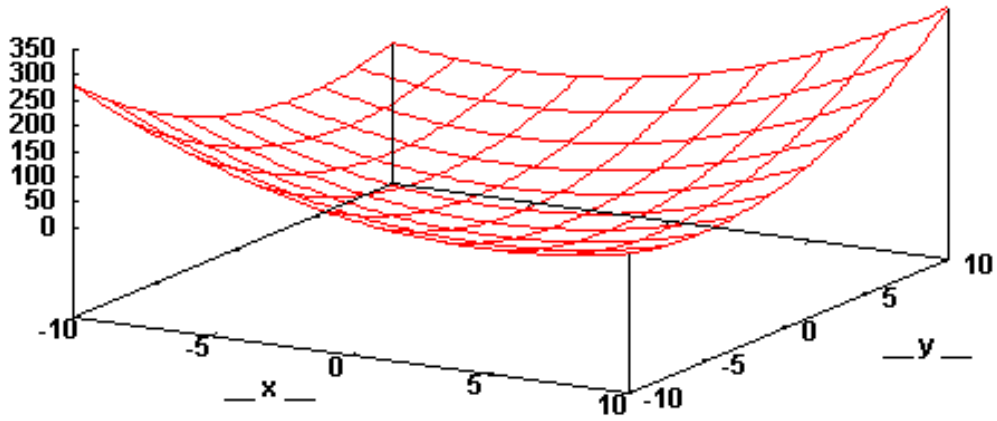


Figura 1.6 - Exemplo do gráfico de uma superfície.

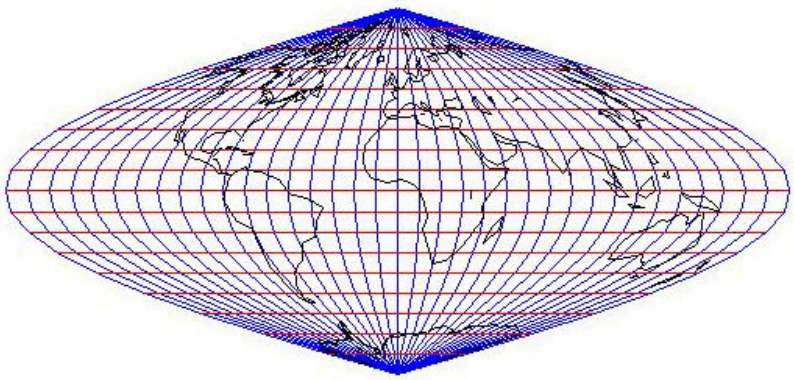


Figura 1.7 - Exemplo de uma projeção cartográfica feita usando o aplicativo *gnuplot*. (Projeção Sanson-Flamsteed)

Por meio dos sete exemplos anteriores tem-se uma idéia do potencial de aplicação deste programa. Neste material pretende-se apresentar uma introdução ao uso deste aplicativo, no qual serão mostrados alguns exemplos mais usuais.

Ambiente de trabalho

A Figura 1.8 mostra a tela principal do aplicativo *gnuplot* na qual são mostradas as principais funções. Esta tela é aberta no momento que o aplicativo é ativado, tanto usando o ícone:

localizado na área de trabalho, quanto o arquivo *wgnuplot.exe*.

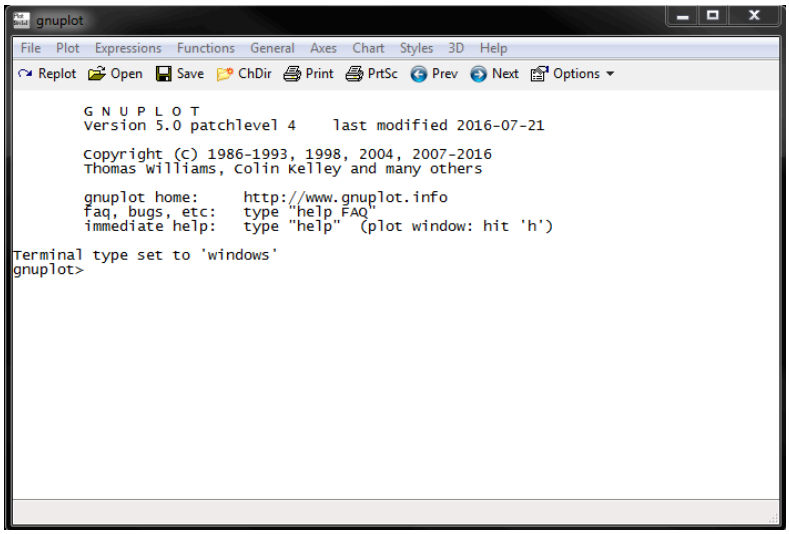


Figura 1.8 - Tela do aplicativo *gnuplot*.

A maneira mais simples de trabalhar com o aplicativo *gnuplot* é por meio da própria linha de comando, mostrada na janela da Figura 1.8, como ocorre em aplicativos como **MatLab**, **Octave**, **IDL**, etc. No entanto, esta não é a única possibilidade de trabalho. Uma alternativa é escrever os comandos em um arquivo *script* do tipo ASCII, e depois carregar esse arquivo usando a opção *load 'arquivo'*. Uma terceira possibilidade é através da criação de um arquivo em *bat*, no qual o aplicativo e o arquivo *script* são ativados simultaneamente, sem a necessidade de executar o programa *wgnuplot.exe*. Uma quarta possibilidade é o uso de bibliotecas em C, o que permite ativar o aplicativo diretamente a partir de um programa escrito em linguagem C. Mais informações sobre esta quarta opção podem ser obtidas no endereço http://ndevilla.free.fr/gnuplot/.

2. COMANDOS BÁSICOS PARA VISUALIZAÇÃO DE FUNÇÕES

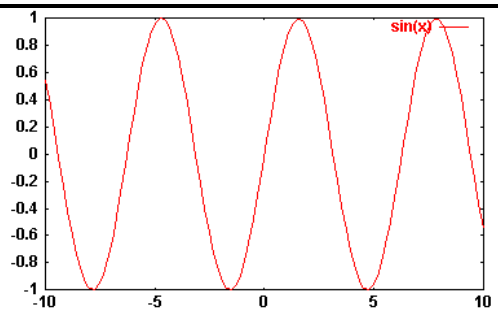
O comando utilizado para fazer a visualização de funções no plano cartesiano bidimensional se chama `plot`. Dentre as funções predefinidas disponíveis tem-se:

Função	Operação	Sintaxe
abs	Valor absoluto	abs(x)
sqrt	Raiz quadrada	sqrt(x)
exp	Exponencial	exp(x)
log	Logaritmo (base e)	log(x)
log10	Logaritmo (base 10)	log10(x)
sin	Seno de um ângulo	sin(x)
cos	Coseno de um ângulo	cos(x)
tan	Tangente de um ângulo	tan(x)
asin	Arco seno	asin(x)
acos	Arco coseno	acos(x)
atan	Arco tangente	atan(x)

Na Figura 2.1 são apresentados dois exemplos mostrando as funções seno e logaritmo. À esquerda é mostrado o comando utilizado e a direita o resultado.

```
gnuplot> plot sin(x)
```

Resultado >>



Observações: Pode-se notar que não foi definido o domínio da função e o intervalo padrão [-10:10] foi utilizado. Além disso pode-se observar que o domínio está em radianos.

```
gnuplot> plot log(x)
```

Resultado >>

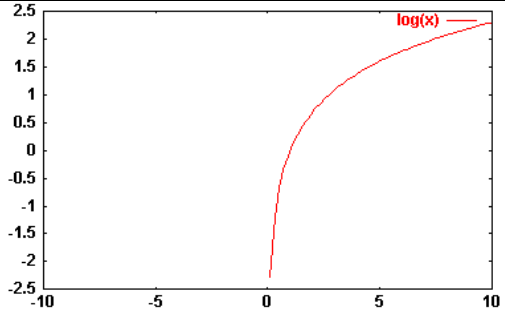


Figura 2.1 – Funções seno e logaritmo.

Ativação da grade (*grid*)

A ativação da grade (ou *grid*) pode ser feita usando o comando *set grid*, antes do comando *plot*, como mostra o exemplo da Figura 2.2. Para ver a sintaxe completa do comando *grid*, e de todos os demais, ative o comando de ajuda através de *help grid*.

```
gnuplot> set grid
gnuplot> plot log(x)
```

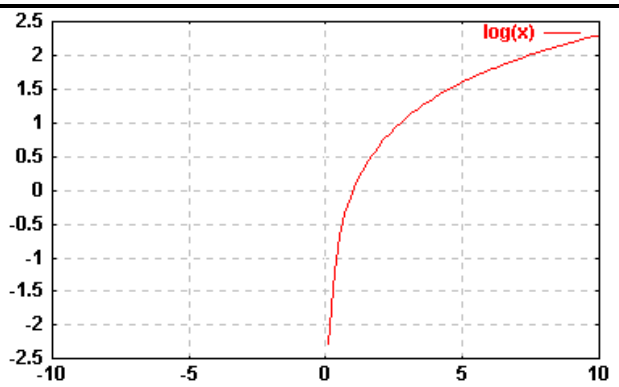


Figura 2.2 – Ativação do grid.

Para desativar a opção *grid* pode-se utilizar o comando *set nogrid*.

Como sugestão de exercícios, faça a visualização dos gráficos das seguintes funções algébricas.

$$y = 5x^2 - 2x - 50 \qquad y = e^{\left(-\frac{x}{5}\right)} \qquad y = \text{sen}(x)$$

$$y = \text{sen}\left(\frac{x}{2}\right)\cos(x) \qquad y = \left|\text{sen}\left(\frac{x}{2}\right)\cos(x)\right| \qquad y = \sqrt{\left|\text{sen}\left(\frac{x}{2}\right)\cos(x)\right|}$$

$$y = \log\left(\sqrt{\left|\text{sen}\left(\frac{x}{2}\right)\cos(x)\right|}\right) \qquad y = 2\cos x + \text{sen}(2x) + \frac{\text{sen}(4x)}{2} \qquad y = \log\left(\arctan\left(x\,\frac{pi}{4}\right)\right)$$

Em caso de dúvida quando à sintaxe de alguma função pré-definida, ative a opção *functions* da barra de tarefas (Figura 2.3).

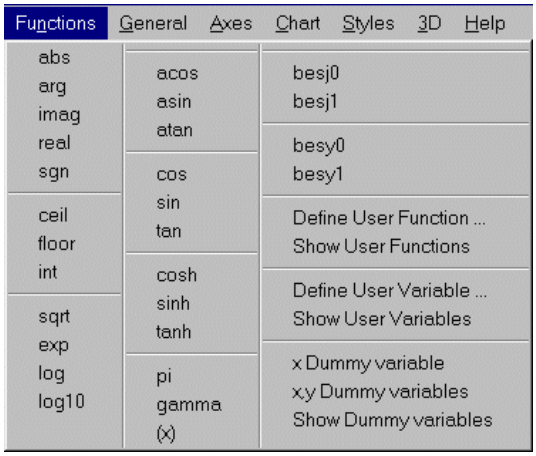


Figura 2.3 – Funções pré-definidas no aplicativo *gnuplot*.

Modificação do domínio de funções

Nas funções mostradas na Figura 2.1 e 2.2 pode-se notar que os domínios são iguais. Na verdade como o domínio não foi definido o valor utilizado é o *default*. A modificação do domínio das funções pode ser feita utilizando o comando *set xrange*, como mostra o exemplo abaixo.

```
gnuplot> set xrange [0:3]
gnuplot> plot log(x)

ou

gnuplot> plot [0:3] log(x)
```

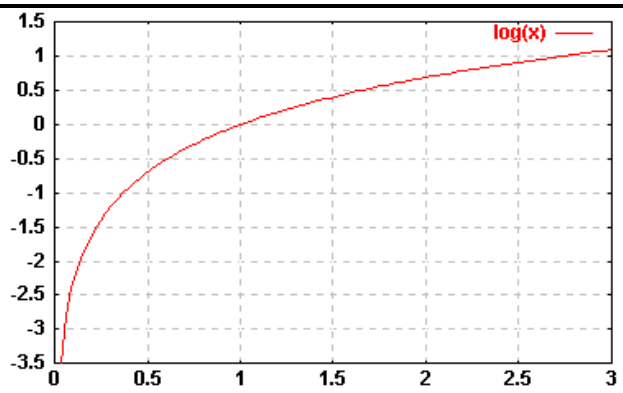


Figura 2.4 – Modificação do domínio.

De modo análogo pode-se definir o intervalo para as coordenadas y e z (caso 3D). Nestes casos os comandos são *set yrange* e *set zrange*, respectivamente.

Visualização de múltiplas funções

Nos gráficos anteriores fez-se a visualização de uma função por vez. Pode-se também fazer a visualização de mais de uma função ao mesmo tempo. Para isto pode-se usar ',\' como terminador de linha e definir a função desejada na linha seguinte. Outra possibilidade é usar a

opção *rep* (de *replot*) a cada função adicional, como mostrado no exemplo da Figura 2.5, onde são apresentadas três funções simultaneamente.

```
gnuplot> set grid
gnuplot> plot sin(x),\
gnuplot> cos(x/3),\
gnuplot> x/14-1

ou

gnuplot> set grid
gnuplot> plot sin(x)
gnuplot> rep cos(x/3)
gnuplot> rep x/14-1
```

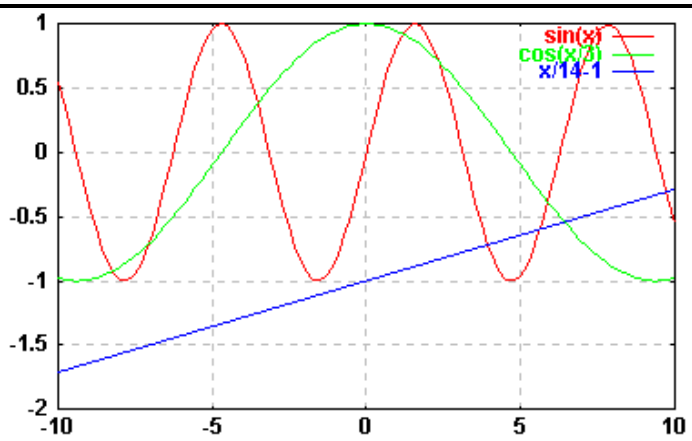


Figura 2.5 – Visualização de múltiplas funções.

Para mais detalhes sobre o comando *replot* ative *help replot*.

3. Modificação de Atributos

Nesta seção são apresentados comandos que possibilitam mudar alguns atributos dos gráficos construídos com o *gnuplot*.

Cor, tipos de pontos e linhas

O *gnuplot* permite que os gráficos sejam desenhados em diversas cores. No caso da representação de curvas usando pontos podem-se utilizar diferentes formatos (cruz, círculo, triângulo, etc). Para visualizar as cores, bem como o formato das entidades pontuais basta ativar, na linha de comando, a palavra *test*, como mostrado abaixo:

```
gnuplot> test
```

Na Figura 3.1 são apresentadas as cores disponíveis, bem como os tipos de representação para pontos.

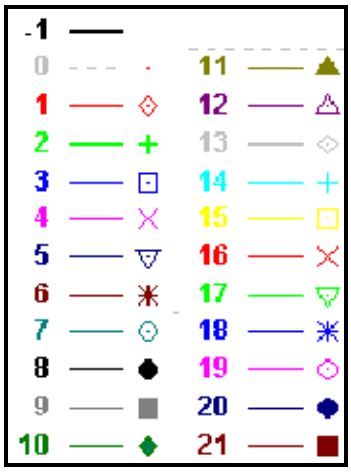


Figura 3.1 – Tipos de pontos e cores disponíveis no *gnuplot*.

É importante ressaltar que diferenças nas cores podem ocorrer dependendo do *hardware* utilizado.

Em todos os gráficos apresentados nas seções anteriores as funções foram representadas por linhas contínuas. Como exemplos de outros modos de representação tem-se: pontos, impulsos, linhas, etc. No caso de utilizar estes elementos deve-se usar a opção *with* seguida do tipo desejado. Como exemplo, pode-se citar:

```
gnuplot > plot sin(x/2) with points
gnuplot > plot sin(x/2) with lines
gnuplot > plot sin(x/2) with linespoints
gnuplot > plot sin(x/2) with dots
gnuplot > plot sin(x/2) with impulses
```

A definição da cor pode ser feita incluindo as letras "lc" (de *line color*) seguidas do número correspondente à cor, de acordo com a tabela de cores mostrada na Figura 3.1. Supondo que se deseja associar os elementos *points*, *lines*, *linespoints*, *dots*, e *impulses*, respectivamente às cores azul (3), azul marinho (5), vermelho (1), preto (8) e cinza escuro (9), deve-se escrever:

```
gnuplot > plot sin(x/2) with points lc 3
gnuplot > plot sin(x/2) with lines lc 5
gnuplot > plot sin(x/2) with linespoints lc 1
gnuplot > plot sin(x/2) with dots lc 8
gnuplot > plot sin(x/2) with impulses lc 9
```

Para o caso de entidades pontuais pode-se definir o tipo do ponto, utilizando as letras "`pt`" (de *point type*) seguidas o número correspondente, como mostrado na Figura 3.1. Assim, para visualizar a função seno(x/2), no intervalo [0:pi/2], na cor verde (10) e com asteriscos (6), juntamente com a função coseno(4x), na cor azul (3) e com impulsos pode-se escrever:

```
gnuplot > plot [0:pi/2] sin(x/2) with points lc 10 pt 6
gnuplot > rep cos(4*x) with impulses lc 3

Observação: No exemplo da primeira linha o número 10
representa a cor e o número 6 representa o tipo do ponto.

Deste modo tem-se:

   Primeiro número (precedido de lc)    >>>    Cor
   Segundo número (precedido de pt)     >>>    Tipo do ponto
```

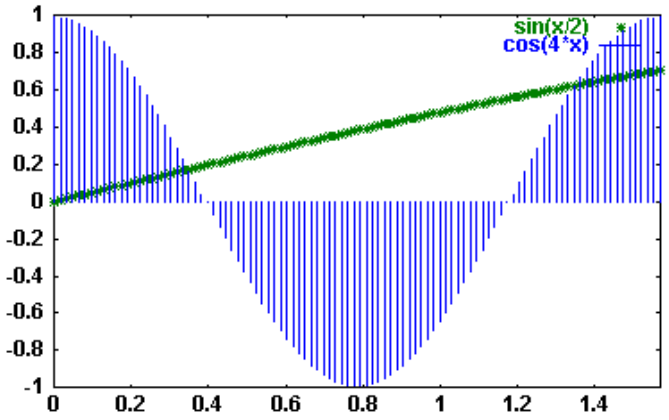


Figura 3.2 – Representação de múltiplas funções, com mudança de cor e uso de pontos e impulsos.

Para verificar outros tipos de estilos de dados ativar a opção *`Style`* e depois *`Data Style`*, a partir da barra de tarefas (Figura 3.3).

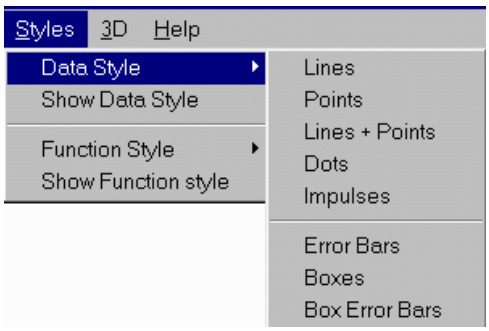
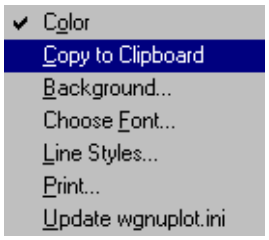


Figura 3.3 – Opções de estilos disponíveis.

Mudança dos atributos como cor de fundo, fonte, etc.

Uma vez mostrados os gráficos é possível modificar o tamanho da janela de visualização, como se faz em qualquer janela do Windows. Modificado o tamanho, ou outro atributo qualquer da janela, pode-se salvar esta configuração ao clicar com o botão direito sobre o gráfico, escolhendo a opção *Update ... wgnuplot.ini*. Assim, as próximas janelas abertas terão a aparência da última configuração salva.

Além de modificar o tamanho da janela, ao clicar como o botão direito do mouse sobre o gráfico tem-se as opções mostradas na Figura 3.4



⇒ Ativar ou desativar a cor.
⇒ Copiar o conteúdo da janela para a área de transferência.
⇒ Modificar a cor do fundo.
⇒ Escolher a fonte usada na janela gráfica.
⇒ Modificar o tipo de linha bem como as cores.
⇒ Impressão.
⇒ Salvar as configurações atuais.

Figura 3.4 – Mudanças de atributo da tela gráfica apresentada pelo aplicativo. As opções mostradas podem mudar de acordo com a versão instalada.

4. Definição de Funções pelo Usuário

Nas seções anteriores mostrou-se como se constrói gráficos usando funções predefinidas. A partir destas funções predefinidas pode-se definir uma infinidade de outras funções. Na sequência são apresentadas algumas maneiras diferentes de mostrar a função

$$y = \text{sen}\left(\frac{x}{2}\right),$$

no intervalo [-2π:2π].

```
(Opção 1)
gnuplot > set xrange [-2*pi:2*pi]
gnuplot > plot sin(x/2)

(Opção 2)
gnuplot > set xrange [-2*pi: 2*pi]
gnuplot > f(x)=sin(x/2)
gnuplot > plot f(x)

(Opção 3)
gnuplot > set xrange [-2*pi:2*pi]
gnuplot > f(x,b)=sin(x/b)
gnuplot > plot f(x,2)

(Opção 4)
gnuplot > set xrange [-2*pi:2*pi]
gnuplot > f(x)=sin(x*a)
gnuplot > plot f(x), a=0.5

(Opção 5)
gnuplot > set xrange [-2*pi:2*pi]
gnuplot > f(x,a)=sin(a*x)
gnuplot > plot f(x,0.5)
```

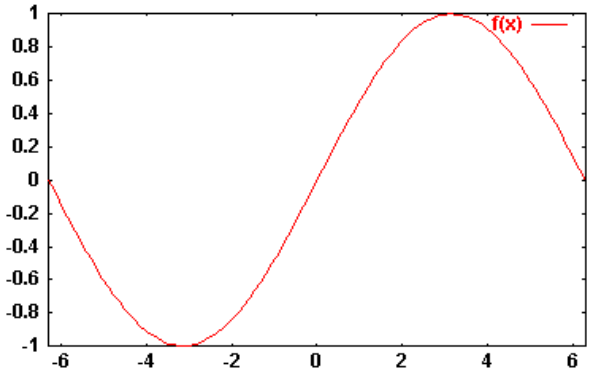


Figura 4.1 – Exemplo de visualização de uma única função, usando diferentes opções.

Como pode ser observado, são várias as opções que podem ser utilizadas para representar uma única função. As opções 3, 4 e 5, em especial, são interessantes para escrever funções mais genéricas, nas quais algumas constantes podem ser modificadas no momento em que for utilizado o comando *plot*.

Considerando as seguintes funções:

$$\begin{aligned} y_1 &= f_1(x) = 180 \\ y_2 &= f_2(x) = 13x - 200 \\ y_3 &= f_3(x) = 4 - 2x^2 - 3x \\ y_4 &= f_4(x) = 0.5x^3 - 34x + 2x^2 - 22 \end{aligned} \quad ,$$

e a tarefa de visualizá-las simultaneamente no domínio [-10:10], uma primeira opção seria definir as quatro funções separadamente. Neste caso pode-se escrever o seguinte *script*:

```
gnuplot > set xrange [-10:10]
gnuplot > f1(x)=180
gnuplot > f2(x)=13*x-200
gnuplot > f3(x)=4-2*x*x-3*x
gnuplot > f4(x)=0.5*x*x*x-34*x+2*x*x-22
gnuplot > plot f1(x)
gnuplot > rep f2(x)
gnuplot > rep f3(x)
gnuplot > rep f4(x)
```

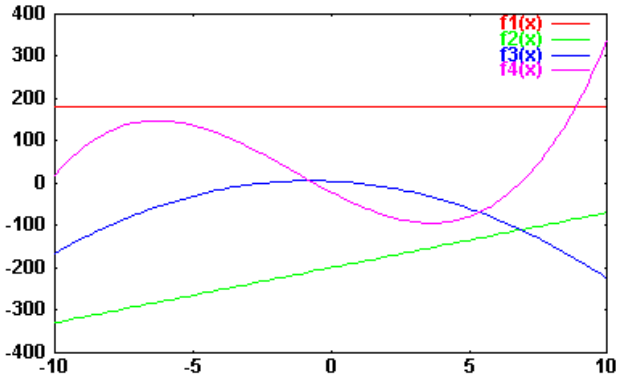


Figura 4.2 – Visualização simultânea de quatro polinômios.

Uma opção mais geral para visualizar os polinômios mostrados na Figura 4.2 seria escrever um único polinômio de grau 3, genérico, da seguinte forma:

$$y = f(x) = a + bx + cx^2 + dx^3,$$

onde a, b c e d são constantes. Deste modo apenas uma função precisa ser definida e o seguinte *script* pode ser utilizado:

```
gnuplot > reset
gnuplot > set xrange [-10:10]
gnuplot > f(x,a,b,c,d)=a+b*x+c*x**2+d*x**3
gnuplot > plot f(x,180,0,0,0)
gnuplot > rep  f(x,-200,13,0,0)
gnuplot > rep  f(x,4,-3,-2,0)
gnuplot > rep  f(x,-22,-34,2,0.5)
```

Figura 4.3 – Visualização simultânea de quatro polinômios, usando uma única função.

Como pode-se observar no exemplo anterior, apenas uma função é definida, sendo os coeficientes modificados de acordo com a função desejada, no momento em que é utilizado o comando *plot* (ou *rep*, no caso de múltiplas funções).

5. LEITURA E VISUALIZAÇÃO DE DADOS A PARTIR DE ARQUIVOS / TEXTO / LEGENDA

Até este ponto, as funções foram definidas de modo algébrico e não foi feita a visualização de dados lidos em arquivo. Como exemplo, considerar que se dispõe de um arquivo ASCII (com nome *desniv.txt*), no qual se tem armazenado as altitudes (em metros) de 10 pontos de uma linha de nivelamento. Além das altitudes têm-se, para cada um dos pontos, informações de temperatura e pressão, úteis na realização de correções nas altitudes. A Figura 5.1 mostra o arquivo montado com estas informações.

```
# Arquivo desniv.txt
# Coluna 1 > Número do ponto
# Coluna 2 > Altitude (m)
# Coluna 3 > Temperatura (graus celsios)
# Coluna 4 > Pressão mmHg
10      420.100     22.5     745
20      422.430     23.7     732
30      428.701     22.8     720
40      426.482     23.3     729
50      421.320     23.6     732
60      419.240     22.4     749
70      415.400     22.1     735
80      417.804     22.0     729
90      422.500     23.4     740
100     427.306     23.0     737
```

Figura 5.1 – Exemplo de um arquivo de dados.

Considerando que este arquivo esteja no diretório em que se está trabalhando, para visualizar o gráfico "Número do ponto x Altitude", basta especificar o arquivo, conforme ilustrado na Figura 5.2.

```
gnuplot > reset
gnuplot > plot "desniv.txt"
```

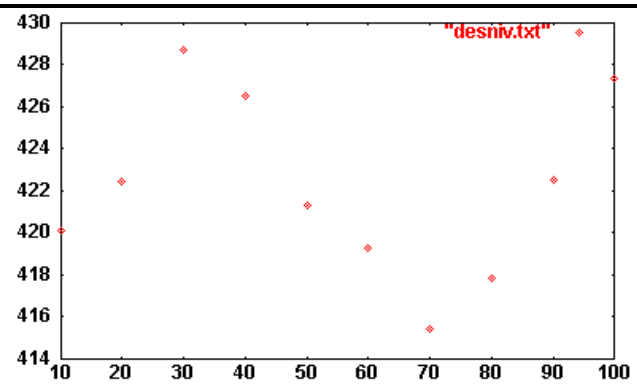


Figura 5.2 – Visualização de dados lidos em arquivo.

Pode-se notar que, automaticamente, as colunas 1 e 2 são mostradas. É possível também explicitar as colunas que se quer utilizar, usando a opção *using*. Os exemplos a seguir mostram duas maneiras de reproduzir o gráfico anterior:

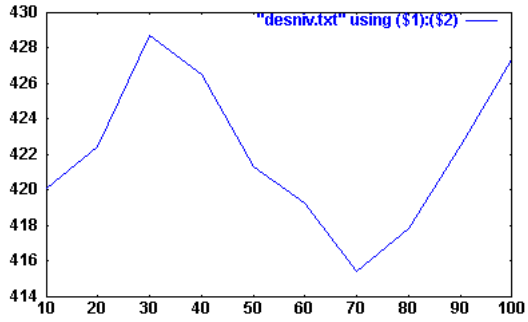
```
gnuplot > reset
gnuplot > plot "desniv.txt" using 1:2
```

ou

```
gnuplot > reset
gnuplot > plot "desniv.txt" using ($1):($2)
```

Caso o usuário queira representar os dados usando impulsos, linhas, etc, ou ainda mudar a cor, pode-se usar a opção *with* (ver seção 3) na mesma linha, como mostram os exemplos da Figura 5.3.

```
gnuplot > reset
gnuplot > plot "desniv.txt" using ($1):($2) with lines lc 3
```




```
gnuplot > reset
gnuplot > plot "desniv.txt" using ($1):($2) with impulses lc 3
gnuplot > rep  "desniv.txt" using ($1):($2) with lines lc 8
```

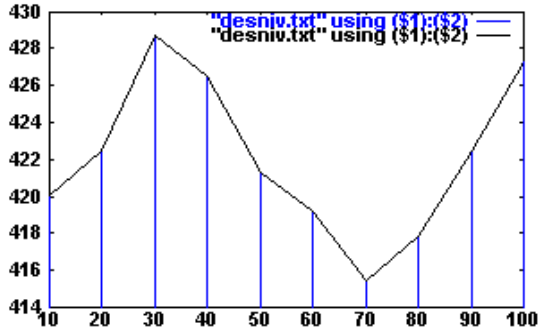


Figura 5.3 – Visualização de dados lidos em arquivo, com mudança de atributos.

A princípio, o uso de ***plot "..." using 1:2*** e ***plot "..." using ($1):($2)*** não faz diferença. Realmente, isso ocorre se a intenção for apenas visualizar a coluna 1 "contra" a coluna 2. No entanto, o uso de $, antes do número da coluna ($2 por exemplo), permite que se interprete $2 como uma variável. Um exemplo do uso de "$coluna" seria na visualização do

desnível de cada um dos pontos do arquivo anterior, em relação ao ponto 10, que possui altitude igual a 420,100 m. Deste modo, para calcular o desnível em relação a este ponto basta usar

```
gnuplot > reset
gnuplot > plot "desniv.txt" using ($1): ($2-420.100) with lines lc 3
```

Usando as informações e a sintaxe vista nos exemplos anteriores, tente representar na abscissa e ordenada dos gráficos os seguintes elementos:

Abscissa	**Ordenada**
Número do ponto	Temperatura (graus celsios)
Número do ponto	Pressão atmosférica
Número do ponto	Diferença de pressão em relação à média*
Numero do ponto	Temperatura (graus celsios) e Diferença em relação à temperatura do primeiro ponto.
Numero do ponto	Temperatura em ºF e Diferença em relação à temperatura do primeiro ponto (ºF).

* O valor médio deve ser calculado fora do aplicativo

Como se pôde perceber, é fácil modificar as variáveis que se quer representar na abscissa e ordenada. Além disso, pode-se fazer operações usando colunas, como mostram os exemplos:

```
using ($1):($2 +$3)
using ($1):($2 +3*($3))
using ($2):($5)/100
```

Para mais detalhes sobre as opções do comando *using* utilize a ajuda (*help using*).

Inserção de título e texto nos eixos x e y

Para a inserção de título e rótulo na abscissa e ordenada, os seguintes comandos podem ser utilizados:

```
set title "texto que corresponde ao título"
set xlabel "texto corresponde à abscissa"
set ylabel "texto corresponde à ordenada"
```

O exemplo da Figura 5.4 mostra o uso destes três comandos.

```
gnuplot > reset
gnuplot > set grid
gnuplot > set xrange [0:25]
gnuplot > set title "Função Parabólica \n Teste 1"
gnuplot > set xlabel "X - Tempo (s)"
gnuplot > set ylabel "Y - Aceleração (m/s2)"
gnuplot > f(x)=0.1*x**2-5*x+20
gnuplot > plot f(x) with lines lc 8
```

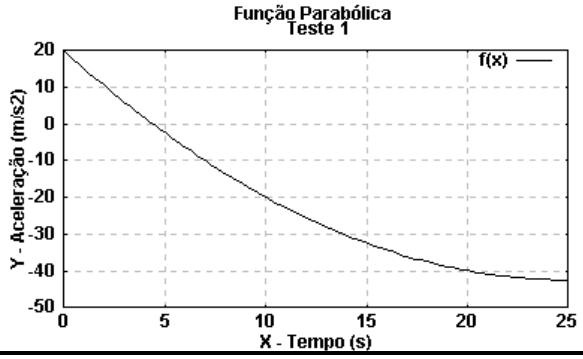


Figura 5.4 – Inserção de título e rótulos na abscissa e ordenada.

Modificação do espaçamento da grade

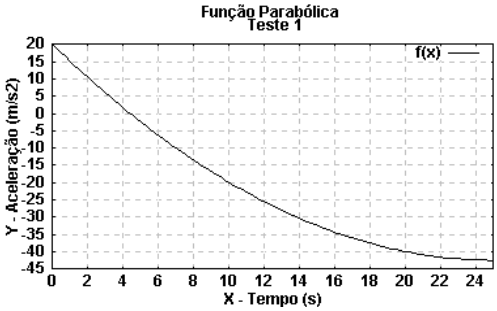
No exemplo anterior mostrou-se como se faz a inserção do título e dos rótulos nos eixos x e y. Pode-se notar que a separação do *grid* (grade) não foi definida. Caso seja necessário fazê-lo, pode-se usar as opções *set xtics* e *set ytics*. Considerando que os incrementos desejáveis em x e y sejam respectivamente ix e iy, a seguinte sintaxe pode ser utilizada:

```
set xtics ix
set ytics iy
```

Além desta possibilidade pode-se, ainda, dividir os intervalos com *tics* menores, usando as opções *set mxtics* e *set mytics*.

Os exemplos da Figura 5.5 ilustram o uso de alguns destes comandos. No segundo gráfico, as setas (**>>>**) são sobrepostas ao desenho apenas para mostrar a posição dos *tics* menores.

```
gnuplot > reset
gnuplot > set grid
gnuplot > set xtics 2
gnuplot > set ytics 5
gnuplot > set xrange [0:25]
gnuplot > set title "função parabólica \n teste 1"
gnuplot > set xlabel "x - tempo (s)"
gnuplot > set ylabel "y - aceleração (m/s2)"
gnuplot > f(x)=0.1*x**2-5*x+20
gnuplot > plot f(x) with lines lc 8
```




```
...
gnuplot > set grid
gnuplot > set xtics 2
gnuplot > set mxtics 2
gnuplot > set ytics 5
gnuplot > set mytics 2
gnuplot > set xrange [0:25] ...
```

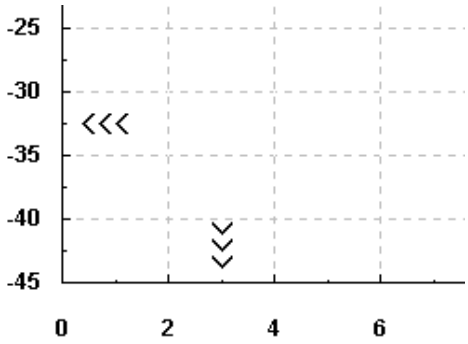


Figura 5.5 – Uso dos comandos *xtics* e *mxtics*.

Modificação da legenda

Em todos os exemplos vistos anteriormente, não se fez a modificação do conteúdo e nem da posição da legenda. Para modificar o texto da legenda pode-se usar a opção *t* seguido do texto a ser escrito, na mesma linha em que se usa o comando *plot* (ou *rep*) como mostra o exemplo da Figura 5.6. Para não ser incluído nenhum texto na legenda basta usar t "".

```
gnuplot > reset
gnuplot > set grid
gnuplot > set xtics 2
gnuplot > set mxtics 2
gnuplot > set ytics 40
gnuplot > set mytics 2
gnuplot > set xrange [0:25]
gnuplot > set title "Função Parabólica \n Teste 1"
gnuplot > set xlabel "X - Tempo (s)"
gnuplot > set ylabel "Y- Aceleração (m/s2)"
gnuplot > f(x,a,b,c)=a+b*x+c*x**2
gnuplot > plot f(x,120,-5,0.1) t"Função 1" with points lc 3 pt 5
gnuplot > rep  f(x,80,+10,-0.15) t"Função 2" with lines lc 8
```

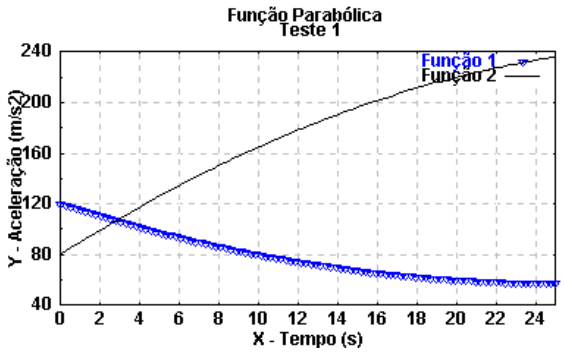


Figura 5.6 – Modificação do texto da legenda usando t "...".

Nos exemplos anteriores, a posição da legenda foi sempre no canto superior direito. Para modificar a localização da legenda pode-se utilizar o comando *set key*. Na sequência, são apresentados quatro exemplos da utilização deste comando, cada um considerando uma posição:

```
set key left bottom          (Canto inferior esquerdo)
set key right bottom         (Canto inferior direito)
set key left top             (Canto superior esquerdo)
set key right top            (Canto superior direito)
```

Para outras opções de configuração da legenda consulte *help key*.

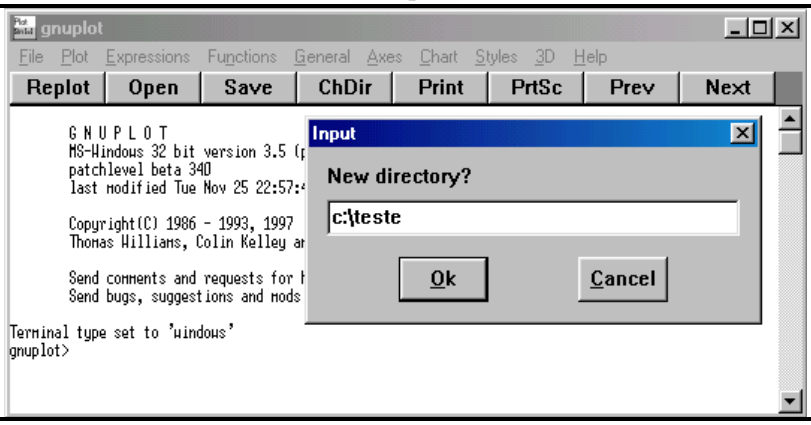
Criação de *scripts* em arquivo

À medida que novas funções e opções de processamento vão sendo incorporadas no *script*, seu tamanho aumenta e uma alternativa mais prática de trabalho é escrever o *script* em arquivo, para depois "carregar" no aplicativo.

Considerando que o *script* apresentado na página anterior foi salvo no diretório *c:\teste* com o nome *curva.gnu*[1], o roteiro indicado na Figura 5.7 pode ser utilizado para carregar este arquivo:

Ativar o aplicativo *gnuplot*.

Mudar o diretório (usando a opção *ChDir* localizada na parte superior da janela) e digitar o nome do diretório onde está localizado o arquivo.



Após definir o diretório é mostrado o direcionamento dado, como pode ser visto ao lado.

```
Terminal type set to 'windows'
gnuplot> cd 'c:\teste'
gnuplot>
```

Para carregar o *script*, armazenado no arquivo *curva.gnu* deve-se utilizar a opção *load*, como mostrado ao lado. Pode-se também usar haspas duplas (“) ao invés de haspas simples. Outra possibilidade é simplesmente arrastar o arquivo criado para a área de trabalho.

```
Terminal type set to 'windows'
gnuplot> cd 'c:\teste'
gnuplot> load 'curva.gnu'
```

Figura 5.7 – Como carregar um arquivo *script*.

6. OPERADOR TERNÁRIO

Um operador disponível no aplicativo *gnuplot*, importante em várias situações, é o operador ternário. Normalmente este operador é utilizado quando se deseja, por exemplo, trabalhar com duas ou mais funções, dependendo de alguma condição pré-determinada.

A sintaxe deste operador é a seguinte:

<Expressão E> **?** *< Opção A >* **:** *< Opção B >*

[1] Não existe uma extensão obrigatória, sendo utilizado neste tutorial a extensão “*.gnu*” apenas por conveniência. Acrescente uma última linha a este arquivo contendo o seguinte comando: '*pause -1 "Continua?"*'. Deste modo, o programa mostra o resultado e espera que o usuário feche a janela gráfica.

Ao ser avaliada a "Expressão E", se ela for verdadeira a opção A é considerada e caso contrário, a opção B passa a ser válida.

Exemplo de aplicação

Deseja-se visualizar um gráfico, cujo domínio é [0:10], composto por duas funções, de acordo com as condições mostradas abaixo:

$$H(x)=\begin{cases} f(x)=2x-60 & \\ g(x)=-x^2-2x-2 & \end{cases} \quad \begin{matrix} se \\ caso \end{matrix} \quad \begin{matrix} 4\le x\le 7 \\ contrário \end{matrix}$$

Na Figura 6.1 está ilustrado um exemplo do uso o operador ternário, para a função anterior.

```
reset
set xtics 1
set grid
set xrange [0:10]

f(x)=2*x-60
g(x)=-x**2+2*x-2
f(x)=(4<=x && x<=7) ? f(x):g(x)

plot f(x) t"Função Composta"
Pause -1 "Fechar ?"
```

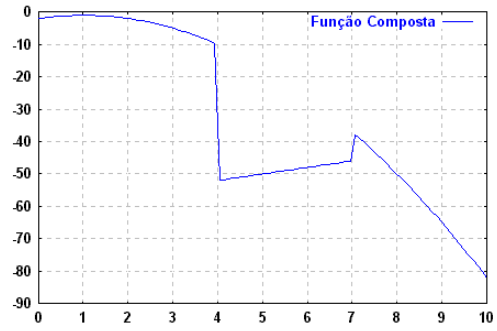


Figura 6.1 - Exemplo de uso do operador ternário.

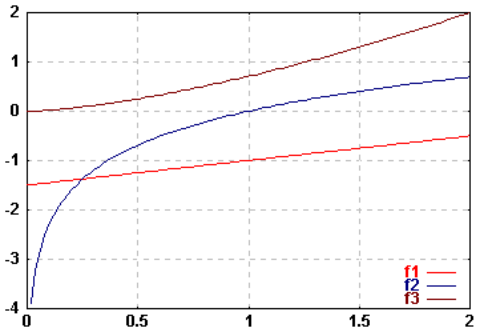
O próximo exemplo do uso do operador ternário mostra uma situação na qual se tem uma função F(x), composta por três outras funções:

$$F(x)=\begin{cases} f_1(x)=\dfrac{x}{2}-0.5 & \\ f_2(x)=\log(x) & \\ f_3(x)=\sqrt{\dfrac{x^3}{2}} & \end{cases} \quad \text{se} \quad \begin{matrix} x<0{,}5 \\ 0{,}5\le x\le 1{,}0 \\ 1{,}0<x\le 2{,}0 \end{matrix}$$

Neste caso deve-se usar mais de uma condição. A função F(X) será igual à função $f_1(x)$, ou seja, $F(x)=f_1(x)$, se $x<0{,}5$. Assim, para $x\ge 0{,}5$ as outras condições devem ser testadas. Deste

modo a função $f_1(x)$ deverá ser ignorada para o caso em que $x \geq 0{,}5$. Isto pode ser feito usando o "0/0" (ou 1/0), como mostra o exemplo da Figura 6.2.

```
reset
set key bottom right
set grid
set xrange [0:2]
set yrange [-4:2]
f1(x)=0.5*x-1.5
f2(x)=log(x)
f3(x)=sqrt(x*x*x/2)
plot f1(x) t"f1" 1
rep  f2(x) t"f2" 5
rep  f3(x) t"f3" 6
```




```
reset
set key bottom right
set grid
set xrange [0:2]
set yrange [-4:2]
f1(x)=0.5*x-1.5
f2(x)=log(x)
f3(x)=sqrt(x*x*x/2)
g(x)=(     x<0.5       ) ? f1(x) : 0/0
h(x)=( x>=0.5 && x<=1 ) ? f2(x) : 0/0
j(x)=(      x>1        ) ? f3(x) : 0/0
plot g(x) t"f1" 1
rep  h(x) t"f2" 5
rep  j(x) t"f3" 6
```

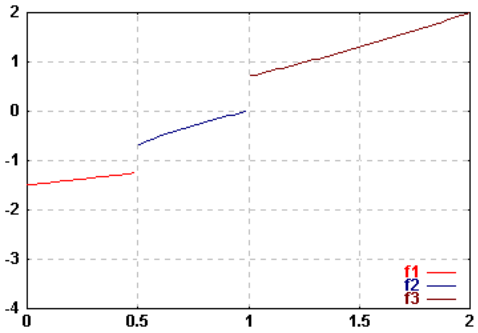


Figura 6.2 - Exemplo de uso do operador ternário, no qual são utilizadas três funções. Na parte superior são mostradas as três funções em todo o domínio.

Observe que ao usar uma indeterminação do tipo **0/0** ou **1/0** no operador ternário, o aplicativo `Gnuplot` simplesmente ignora o comando, deixando de traçar a função.

7. SUPERFÍCIES

A visualização de superfícies é análoga à visualização de curvas planas. Neste caso o comando básico é *`splot`*. Na Figura 7.1 é mostrada uma superfície dada pela função $f(x,y) = \log\left(\sqrt{x^2+y^2}\right)$.

```
reset
set grid
set format z "%4.2f"
f(x,y)=log(sqrt( x*x + y*y))
splot f(x,y)
```

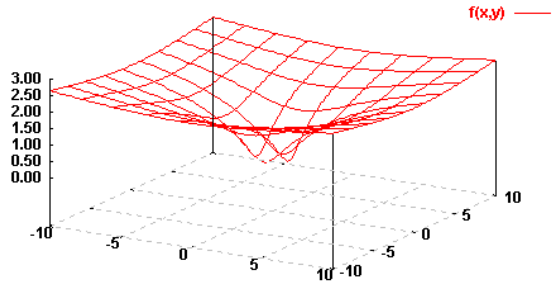
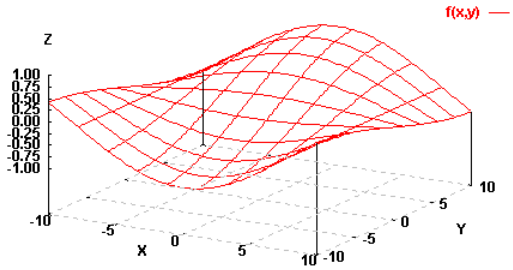


Figura 7.1 - Exemplo da visualização de uma superfície na forma f(x,y).

No próximo, ilustrado na Figura 7.2, exemplo pode-se ver outra superfície, onde são utilizados outros comandos, como por exemplo:

set xlabel	Usado para escrever no gráfico o rótulo no eixo x (análogo para y e z)
set ztics	Usado para modificar o espaçamento das coordenadas em z (análogo para x y y)
set format	Usado para escrever valores numéricos com formato predefinido.
set title	Usado para mostrar o título
set hidden3D	Usado no modo 3D para "esconder" o que fica "atrás" da superfície

```
reset
set grid
set xlabel "X"
set ylabel "Y"
set zlabel "Z"
set ztics 0.25
set format z "%4.2f"
f(x,y)=sin(y/7)*cos(x/5)
splot f(x,y)
```




```
reset
set grid
set xlabel "X"
set ylabel "Y"
set zlabel "Z"
set title "Exemplo de Superfície"
set ztics 0.50
set format z "%4.2f"
f(x,y)=sin(y/7)*cos(x/5)
set hidden3d
splot f(x,y) t""
```

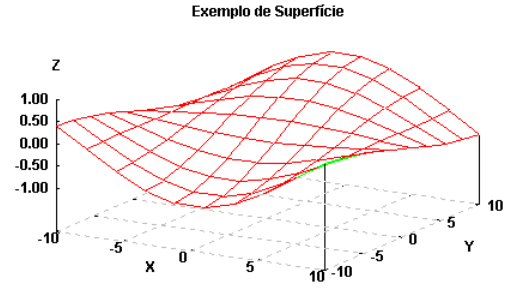


Figura 7.2 - Exemplos para a visualização de uma superfície na forma f(x,y).

A mesma superfície é mostrada na Figura 7.3, sendo incluídas algumas curvas de nível, através do comando *set contour*.

```
reset
set grid
set xlabel "X"
set ylabel "Y"
set zlabel "Z"
set title "Exemplo de Superfície"
set ztics 0.50
set format z "%4.2f"
f(x,y)=sin(y/7)*cos(x/5)
set cntrparam levels incremental -1,0.25,1
set contour base
set hidden3d
splot f(x,y) t""
```

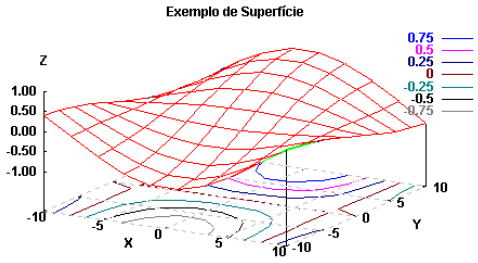


Figura 7.3 - Superfície com as curvas de nível.

A opção *set cntrparam* permite definir, por exemplo, qual o desnível entre as curvas sucessivas, dentre outros elementos, como se pode ver ao ativar *help cntrparam*.

As superfícies mostradas anteriormente são definidas por meio de funções algébricas. Outra possibilidade é através de um conjunto de coordenadas 3D, armazenadas em um arquivo ASCII, e que determinam uma superfície. Na Figura 7.4 é mostrado um conjunto de pontos, armazenados no arquivo *pontos3d.dat*. Este arquivo é composto por três colunas, contendo respectivamente as coordenadas X, Y, e Z.

```
# Conjunto de pontos 3d
# Coluna 1 x
# Coluna 2 y
# Coluna 3 z
#
    10    10   100
    10    30   150
    10    50   150
    10    70   130
    10    90   170
    20    10   100
    20    30   150
    20    50   120
    20    70   093
    20    90   135
    30    10   099
    30    30   189
    30    50   138
...
```

```
...
    30    70   152
    30    90   174
    50    10   200
    50    30   153
    50    50   128
    50    70   203
    50    90   142
    70    10   173
    70    30   157
    70    50   142
    70    70   201
    70    90   210
    90    10   173
    90    30   216
    90    50   142
    90    70   152
    90    90   132
```

Figura 7.4 - Arquivo de dados (*pontos3d.dat*) composto por um conjunto de pontos 3D. O caractere "#" no início da linha indica que a linha é um comentário.

A Figura 7.5, a seguir, apresenta um *scrip* que permite gerar a superfície definida pelos pontos presentes no arquivo *pontos3d.dat*.

```
# Exemplo de Visualização de Pontos no espaço 3D
#
# Mauricio Galo / UNESP / Dep. de Cartografia
# Aplicativo: gnuplot

#
# Comandos preliminares: grid, tics, rótulos e título
reset
set grid
set xtics 20
set ytics 20
set ztics 50
set title  "Exemplo de uma superfície gerada com Gnuplot\n\
(Dados lidos em arquivo)"
set xlabel "X"
set ylabel "Y"
set zlabel "Z(m)"

#
# Comandos específicos para visualização 3D
set hidden3d
set view 40,30,1,1
set data style points
set dgrid3d 30,30,2
splot 'pontos3d.dat' using ($1):($2):($3) t"" with lines lc 5
pause -1 "Fecha?"

#
# Curvas de nível
set contour base
set cntrparam levels incremental 50,15,300
set dgrid3d 30,30,2
splot 'pontos3d.dat' using ($1):($2):($3) t"" with lines lc 5
pause -1 "Fecha?"
```

Figura 7.5 - Exemplo de *script* que faz a visualização de um arquivo de pontos e gera a superfície.

A Figura 7.6 mostra as superfícies geradas ao ser executado o *scrip* apresentado na Figura 7.5, que utiliza como dados de entrada os pontos da Figura 7.4.

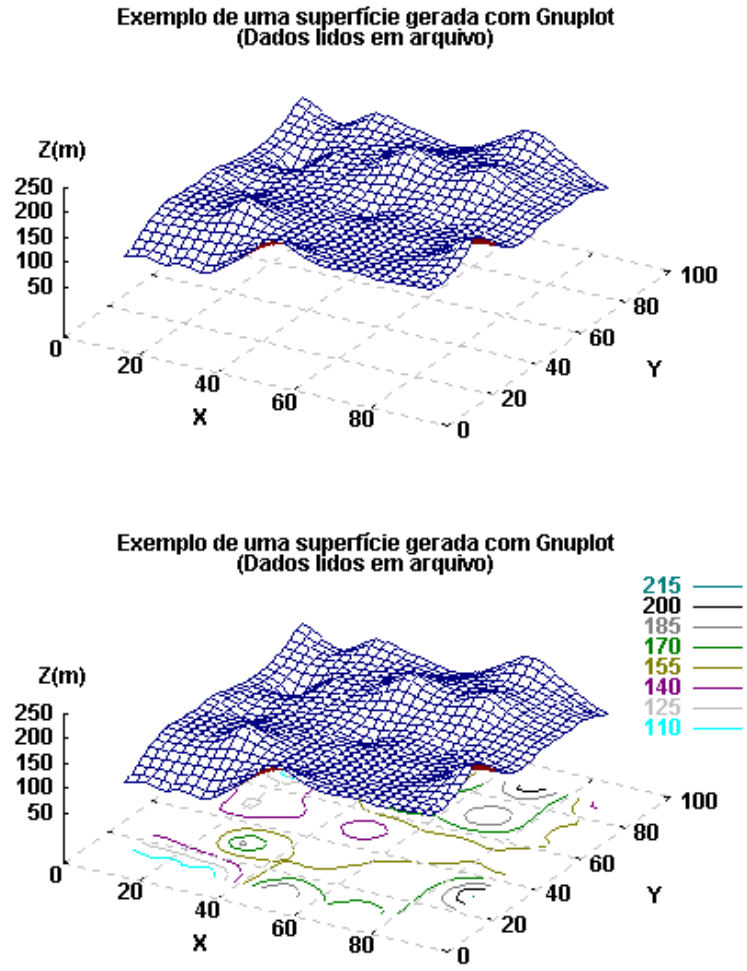


Figura 7.6 - Superfícies geradas a partir de uma nuvem de pontos 3D.

8. TÓPICOS ADICIONAIS

Representação de ângulos em graus ou radianos

Por *default*, o modo de representação dos ângulos é em radianos e para modificar esse modo de representação pode-se usar o comando *set angles*. As opções são as seguintes:

```
set angles degrees
set angles radians
```

Comando de pausa

Este comando é normalmente usado em um arquivo *script* no qual é apresentada uma sequência de gráficos, devendo-se colocar uma pausa antes do segundo gráfico, do terceiro, e assim sucessivamente. Como exemplo de uso pode-se considerada a sintaxe:

```
pause -1 " Continua ? "
```

Troca de variáveis

Geralmente, ao escrever funções, as variáveis utilizadas são x, y e z. No entanto, os nomes podem ser modificados usando o comando *set dummy*. Como exemplos de uso tem-se:

```
set dummy lat
set dummy h
set dummy lat, lon
set dummy h,s
```

Na sequência é apresentado um gráfico (Figura 8.1), no qual são usados os comandos *dummy*, *angles* e *pause*. Considerando que se tem um ponto numa altitude geométrica **h** sobre um elipsóide de semi-eixo maior **a** (a=6378,160 km) e excentricidade **e** (e= 0.08182), deseja-se obter o gráfico da função que fornece o raio de um paralelo em função da latitude. A equação que permite o cálculo do raio do paralelo para uma latitude φ, é dada por:

$$r_\varphi = (N + h)\cos\varphi$$

onde $N = a\left(1 - e^2\,\mathrm{sen}^2\,\varphi\right)^{-1/2}$.

Considerando que h seja 450 m e que latitude (que varia no intervalo [-π/2, π/2]) será considerada no intervalo 0 a π/2, o *script* que cria os gráficos desejados, pode ser escrito da seguinte maneira:

```
#
# Cálculo do Raio do Paralelo
#
# M. Galo, UNESP, Dep. de Cartografia
#

reset
set grid
set time
set angles radians
set dummy lat
set xlabel "Latitude em radianos"
set ylabel "Raio do paralelo em km"
a=6378.160
e=0.08182
set xrange [0:pi/2]
N(lat)=a*( 1 - e*e*sin(lat)*sin(lat) )**(-0.5)
raio(lat,alt)=( N(lat) + alt )*cos(lat)
plot raio(lat,0.450) t"Raio para h=450m"
pause -1 "Fecha?"

reset
set grid
set time
set angles degrees
set dummy lat
set xlabel "Latitude em graus"
set ylabel "Raio do paralelo em km"
a=6378.160
e=0.08182
set xrange [0:90]
N(lat)=a*( 1 - e*e*sin(lat)*sin(lat) )**(-0.5)
raio(lat,alt)=( N(lat) + alt )*cos(lat)
plot raio(lat,0.450) t"Raio para h=450m"
pause -1 "Fecha?"
```

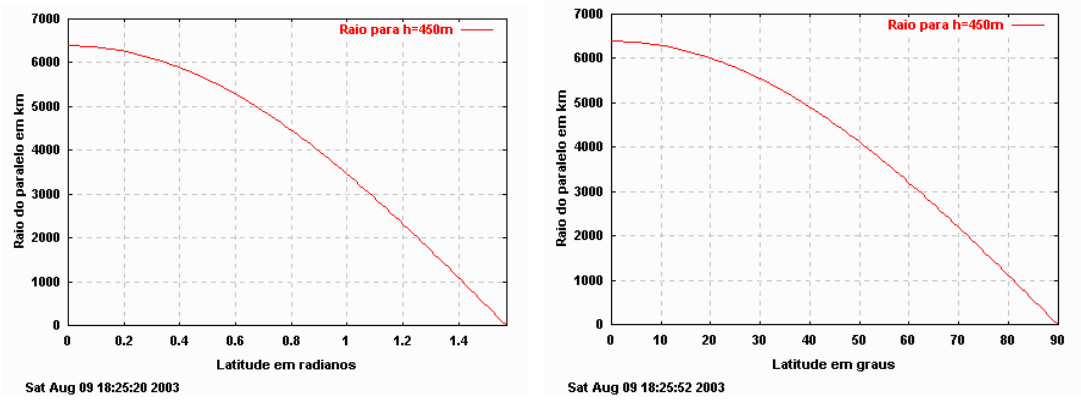


Figura 8.1 – Exemplo do uso dos comandos *dummy*, *angle* e *pause*.

AGRADECIMENTO

O autor agradece à Prof. Maria de Lourdes Bueno Trindade Galo pelas sugestões apresentadas.

BIBLIOGRAFIA

COLLEGE OF NATURAL SCIENCES COMPUTING LABORATORIES. *Introduction to GnuPlot*. University of Northern Iowa, Cedar Falls, IA. Disponível em <http://seismic.yonsei.ac.kr/gnu_intro.html>. Acesso em: Março 2021.

GAVIN, H. P. *GNUPLOT 4.2 - A Brief Manual and Tutorial*. Disponível em: <http://people.duke.edu/~hpgavin/gnuplot.html>. 2008. Acesso em: Março 2021.

Gnuplot Homepage. Disponível em: <http://www.gnuplot.info/>. Acesso em: Março 2021.

Gnuplotting – Create Scientific Plots Using Gnuplot. Disponível em: < http://www.gnuplotting.org/ >. Acesso em: Março 2021.