

Gnuplot: Ajustes de curvas

Bruno Alexandre Rodrigues
Aluno do Curso de Matemática, UEL
E-mail: bruno_rodrigues@hotmail.com

Ulysses Sodré
Prof. Dep. Matemática, UEL
E-mail: ulysses@uel.br

Londrina-PR, 25 de Agosto de 2011

Conteúdo

1	Notas preliminares sobre o Gnuplot	2
2	A Matemática do Ajuste linear nos Mínimos Quadrados	2
3	Instalando o Gnuplot	3
4	Criando os primeiros gráficos no Gnuplot	3
5	Ajuste de curvas no Gnuplot	5
6	Preparando pastas e tabelas para os gráficos	6
7	Ajuste de curvas da forma $y = f(x)$ em 2D	7
7.1	Ajuste de uma reta	7
7.2	Ajuste de uma parábola	8
7.3	Ajuste de uma cúbica	9
7.4	Ajuste de curva trigonométrica	10
7.5	Ajuste de uma exponencial	11
8	Comandos para plotar superfícies $z = f(x, y)$ em 3D	12
9	Scripts no Gnuplot: Uma alternativa para plotar gráficos	13
10	Ajustes de superfícies no Gnuplot	15

Resumo

Este painel apresenta de uma forma bastante rápida os elementos gerais sobre o Gnuplot, que é uma programa para construir gráficos de funções 2D e 3D, que possui o comando `fit` que permite gerar parâmetros para construir funções de melhor ajuste no sentido dos mínimos quadrados utilizando o Método de Levenberg-Marquardt. São tratados diversos ajustes e o processo e construção de tabelas e de *scripts* que agilizam o processo gráfico.

1 Notas preliminares sobre o Gnuplot

O Gnuplot foi projetado por Thomas Williams e Colin Kelly, em 1986, para visualizar funções matemáticas e conjuntos de dados interativamente, em 2D e em 3D. Seu primeiro nome foi Newplot, mas, na época já existia outro programa com este nome, e os autores mudaram o nome para Gnuplot. Além de gerar gráficos de funções já definidas, o Gnuplot também realiza o ajuste de funções para conjuntos de dados fornecidos pelo usuário. O procedimento para realizar tais ajustes constitui o foco principal deste painel.

2 A Matemática do Ajuste linear nos Mínimos Quadrados

Seja um conjunto de pares ordenados (x_i, y_i) com $i = 1, 2, \dots, n$ de pontos no plano cartesiano. Para obter a reta $y = ax + b$ que melhor se ajusta a estes pontos, para cada $i = 1, 2, \dots, n$, definimos a distância entre a ordenada do ponto e a ordenada correspondente à reta $y = ax + b$, pela diferença

$$d_i = y_i - (ax_i + b)$$

Como cada $d_i = y_i - (ax_i + b)$ pode ser negativo, tomamos tais distâncias ao quadrado, que sempre serão não negativas, definimos a função $S = S(a, b)$ como a soma dos quadrados das distâncias, por

$$S(a, b) = \sum_{i=1}^n (y_i - ax_i - b)^2 \geq 0$$

Para determinar a reta que melhor se ajusta ao conjunto de pontos, deve-se determinar os valores de a e b para os quais $S = S(a, b)$ tem o mínimo valor. Esta função é polinomial, assim, calcularemos as derivadas parciais de $S = S(a, b)$ com relação às variáveis a e b e igualamos a zero, para obter os pontos críticos desta função.

$$\begin{aligned} \frac{\partial S}{\partial a} &= \sum_{i=1}^n 2(y_i - ax_i - b)(-x_i) = -2 \sum_{i=1}^n (y_i - ax_i + b)(x_i) \\ \frac{\partial S}{\partial b} &= \sum_{i=1}^n 2(y_i - ax_i - b)(-1) = -2 \sum_{i=1}^n (y_i - ax_i - b) \end{aligned}$$

Basta resolver o sistema com $\frac{\partial S}{\partial a} = 0$ e $\frac{\partial S}{\partial b} = 0$, para obter o ponto de mínimo da função $S = S(a, b) \geq 0$, isto é,

$$\begin{aligned} \sum_{i=1}^n (x_i y_i - ax_i^2 - bx_i) &= 0 \\ \sum_{i=1}^n (y_i - ax_i - b) &= 0 \end{aligned}$$

Agora, isolando os termos que dependem de a e b :

$$\begin{aligned}\sum_{i=1}^n x_i y_i &= a \sum_{i=1}^n x_i^2 + b \sum_{i=1}^n x_i \\ \sum_{i=1}^n y_i &= a \sum_{i=1}^n x_i - b \sum_{i=1}^n 1\end{aligned}$$

Simplificamos este sistema, com $Sxy = \sum_{i=1}^n x_i y_i$, $Sx = \sum_{i=1}^n x_i$, $Sy = \sum_{i=1}^n y_i$ e $Sxx = \sum_{i=1}^n x_i^2$, para escrever o sistema na forma matricial:

$$\begin{pmatrix} Sxy \\ Sy \end{pmatrix} = \begin{pmatrix} Sxx & Sx \\ Sx & n \end{pmatrix} \cdot \begin{pmatrix} a \\ b \end{pmatrix}$$

Com uma calculadora ou computador, calculamos Sx , Sy , Sxy e Sxx , para obter inversa da matriz dos coeficientes, e, os coeficientes a e b são obtidos pelo produto de matrizes:

$$\begin{pmatrix} a \\ b \end{pmatrix} = \frac{1}{n \cdot Sxx - (Sx)^2} \begin{pmatrix} n & -Sx \\ -Sx & Sxx \end{pmatrix} \cdot \begin{pmatrix} Sxy \\ Sy \end{pmatrix}$$

3 Instalando o Gnuplot

1. Em algum pesquisador da Internet, como o Google, pesquise a palavra gnuplot. Se tiver sucesso, ela deverá ser a primeira que aparece.
2. Você deve entrar na página com o link para download e *baixar* o Gnuplot. Se você usa o Windows, deve ser um arquivo da forma `gp440win32.zip`.
3. Este arquivo deve ser descompactado na pasta `C:\gnuplot` do disco rígido.
4. Inicie o Gnuplot a partir do caminho completo `c:\gnuplot\bin\wgnuplot.exe`.
5. Ao abrir o programa, aparecem algumas informações e um prompt na forma:

```
Gnuplot: Linhas de comando
gnuplot>
```

4 Criando os primeiros gráficos no Gnuplot

1. O comando para plotar gráficos 2D é `plot` e o comando para gráficos 3D é o comando `splot`. A partir daqui, os códigos não mais terão o prompt `gnuplot>`.
2. No final de cada linha de comando, deve ser pressionada a tecla ENTER para obter o gráfico relativo ao código digitado.
3. Para plotar a função cosseno na variável x , digitamos:

```
Gnuplot: Linhas de comando
plot_t cos(x)
```

Após entrar com a linha de comando, será aberta uma janela com o gráfico da função cosseno. O pequeno sinal `_` que vemos entre as letras `t` e `c` é um espaço vazio.

4. O Gnuplot reconhece a constante $\pi=3.14159265$.

5. Para alterar o domínio da função $f = f(x)$ para $[-\pi, 2\pi]$, basta digitar

Gnuplot: Linhas de comando

```
set xrange [-pi:2*pi]
```

com o intervalo escrito entre colchetes e os extremos separados por dois pontos.

6. Para alterar o contradomínio da função $f = f(x)$ para $[-1, 2]$, basta digitar

Gnuplot: Linhas de comando

```
set yrange [-1:2]
```

7. Para colocar estas duas informações de uma única vez, basta digitar:

Gnuplot: Linhas de comando

```
set range [-pi:2*pi] [-1:2]
```

8. Para funções de duas variáveis, o gráfico de $z = f(x, y)$ pode ser plotado em $[p, q]$ com

Gnuplot: Linhas de comando

```
set zrange [p:q]
```

9. Para anexar os dois eixos do sistema no gráfico, basta digitar

Gnuplot: Linhas de comando

```
set zeroaxis
```

10. Para reunir todas tais orientações no gráfico da função $\cos()$, digitamos

Gnuplot: Linhas de comando

```
set zeroaxis # Este é um comentário
set range [-pi:2*pi] [-1:2]
plot cos(x)
```

11. Valores e marcas que aparecem nas bordas dos gráficos podem ser definidos com os comandos `xtics`, `ytics` e `ztics` através de `set xtics a, i, b`, onde a é a primeira marca exibida, i é o incremento e b é a última marca. Podemos digitar:

Gnuplot: Linhas de comando

```
set xtics -pi, pi/2, pi
set ytics -0.5, 0.1, 0.5
replot # Replota a função com os novos parâmetros
```

12. Para formatar saídas de marcas com 3 dígitos no eixo x e 2 dígitos no eixo y , digitamos

Gnuplot: Linhas de comando

```
set format x "%.3f"
set format y "%.2f"
rep # Forma simplificada do comando replot
```

13. Para inserir uma grade no gráfico, pode-se digitar o comando

Gnuplot: Linhas de comando

```
set grid
```

14. Para desativar qualquer comando cuja linha iniciou com set, digite

Gnuplot: Linhas de comando

```
unset <comando a desativar>
```

15. É possível dar nomes aos eixos x, y e z, digitando:

Gnuplot: Linhas de comando

```
set xlabel "Eixo das abscissas"  
set ylabel "Eixo das ordenadas"  
set zlabel "Eixo vertical"
```

16. Funções também podem ser plotadas por pontos (points), impulsos (impulses), bolas (dots), linhas contínuas (lines), linhas e pontos (linespoints), etc. Para tal, pode-se utilizar qualquer um dos seguintes comandos:

Gnuplot: Linhas de comando

```
plot sin(x) with points  
plot sin(x) with impulses  
plot sin(x) with dots  
plot sin(x) with lines  
plot sin(x) with linespoints
```

17. A cor, espessura e estilo dos pontos e linhas podem ser alterados apenas com um número correspondente ao tipo desejado na frente do comando, como:

Gnuplot: Linhas de comando

```
plot sin(x) with points 5
```

18. Para acessar a lista com todos os estilos de linhas e pontos no gnuplot, digite:

Gnuplot: Linhas de comando

```
test
```

5 Ajuste de curvas no Gnuplot

1. O Gnuplot é um programa para plotar gráficos, cujo comando fit, gera parâmetros para uma função definida pelo usuário que *melhor* se ajusta aos dados fornecidos e os valores das funções avaliados nas referidas abscissas, no sentido dos *mínimos quadrados*. Este é o método *chi-quadrado*, sendo o modo mais simples de minimizar a *soma dos quadrados dos erros*, conhecida como o *Ajuste Linear pelos Mínimos Quadrados*.
2. Se $z=f(x)$ ou $z=f(x, y)$, z é a variável dependente e x e y são as variáveis independentes. Um parâmetro é um valor definido pelo usuário que será obtido no processo.

3. O comando `fit` utiliza o algoritmo de *Levenberg-Marquardt* para gerar parâmetros para a próxima iteração. O processo continua até ocorrer a convergência, sendo que as alterações devem ser menores que `FIT_LIMIT` ou que o ajuste seja obtido para um número máximo de iterações `FIT_MAXITER`.
4. Em geral, a função de ajuste utiliza um modelo para descrever ou prever o comportamento dos dados, e tais funções são escolhidas independentemente do modelo mas sempre levando em conta a experiência do *ajustador ou modelador* ou o modo de descrever a tendência dos dados com certa resolução e um número mínimo de funções e parâmetros.
5. Para obter uma curva suave em uma variável, o comando `fit` aceita a opção `smooth bezier` que pode ser usada em `plot` para gerar um melhor ajuste.
6. Antes de entrar na próxima seção, observamos que é muito importante ser organizado para obter bons resultados no processo de ajustar funções a dados experimentais. Esta foi a razão pela qual informamos como deve ser construída uma pasta para as tabelas e a forma como elas devem ser construídas.

6 Preparando pastas e tabelas para os gráficos

1. Antes de começar a trabalhar, deve ser preparada a limitação do erro permitido nos ajustes, a pasta com o material e as tabelas que serão usadas nos ajustes.
2. Inicie o Gnuplot, e digite na linha de comando do Gnuplot:

Gnuplot: Linhas de comando

```
FIT_LIMIT=1E-7
FIT_MAXITER=200
```

para que os erros máximos permitidos tenham 7 dígitos após o ponto decimal e o processo de convergência itere no máximo 200 vezes.

3. Existem outras variáveis que podem ser configuradas no Gnuplot.
4. Crie a pasta `SeuNome` dentro da pasta `c:\gnuplot`, para você guardar os seus arquivos e resultados.
5. No Gnuplot, pressione o menu `File`, opção `Change Directory`, digite o texto `c:\gnuplot\SeuNome` e pressione `OK`, ou use o navegador do Windows `...` para obter a pasta `SeuNome` e pressione `OK`.
6. Todo o material na forma de tabelas deve ser digitado em um editor de texto puro, como o Bloco de notas ou PFE, e salvo com a extensão `dat`, não esquecendo de alterar a opção `Salvar` como tipo: `Todos os arquivos`.

7 Ajuste de curvas da forma $y = f(x)$ em 2D

7.1 Ajuste de uma reta

1. No editor de textos, digite os dados abaixo e salve como `reta.dat`:

Gnuplot: Linhas de comando

```
#Tabela para o ajuste de uma reta
# X      Y
0.80  1.34
1.78  1.56
2.76  2.26
3.50  3.64
4.64  3.72
5.20  4.76
6.22  4.68
6.72  5.82
7.64  6.86
8.24  7.50
9.60  8.66
```

2. Para ajustar a reta $y = ax + b$ aos dados da tabela `reta.dat`, esta função deve ser inserida na linha de comando, com:

Gnuplot: Linhas de comando

```
f(x) = a*x+b
```

3. Para obter os parâmetros a e b do ajuste, use a coluna 1 da tabela `reta.dat` para a variável x e a coluna 2 para a variável y . Para ajustar `fit` a função $f=f(x)$ aos dados da tabela `reta.dat`, digite:

Gnuplot: Linhas de comando

```
fit f(x) 'reta.dat' using 1:2 via a,b
```

4. Para plotar apenas os dados da tabela, digite:

Gnuplot: Linhas de comando

```
plot 'reta.dat' using 1:2
```

5. Para plotar a reta de ajuste e os dados das colunas 1 e 2 do arquivo `reta.dat`:

Gnuplot: Linhas de comando

```
plot f(x), 'reta.dat' using 1:2
```

6. Reunindo todas as informações acima, você poderia digitar apenas:

Gnuplot: Linhas de comando

```
f(x) = a*x + b
fit f(x) 'reta.dat' using 1:2 via a,b
plot f(x), 'reta.dat' using 1:2 \
with points lw 2
```

A contrabarra é o comando para quebrar uma linha no Gnuplot.

7.2 Ajuste de uma parábola

1. Digite os dados abaixo e salve o arquivo com o nome `parab.dat`:

Gnuplot: Linhas de comando

```
#Tabela para o ajuste de uma parábola
#   X       Y
  1.46    2.30
  2.56    2.64
  3.18    4.18
  3.74    5.68
  4.64    6.40
  5.20    5.46
  5.74    7.80
  6.64    7.70
  7.68    8.44
  8.30    8.76
  9.26    8.68
  9.88    8.26
 10.34    7.36
 11.24    7.16
 12.18    7.54
 12.34    6.60
 13.32    6.67
 13.66    5.14
 14.66    4.50
```

2. Para ajustar a parábola $y = ax^2 + bx + c$ aos dados de `parab2.dat`, defina a função, com:

Gnuplot: Linhas de comando

```
f(x) = a*x**2 + b*x + c
```

3. Para obter a , b e c no ajuste, use a coluna 1 de `parab.dat` para a variável x e a coluna 2 para y . Para ajustar a função aos dados da tabela, digite:

Gnuplot: Linhas de comando

```
fit f(x) 'parab.dat' using 1:2 via a,b,c
```

4. Para plotar apenas os dados escolhidos, digitamos:

Gnuplot: Linhas de comando

```
plot 'parab.dat' using 1:2
```

5. A parábola e os dados das colunas 1 e 2 da tabela de `parab.dat` podem ser plotados com qualquer uma das linhas

Gnuplot: Linhas de comando

```
plot f(x), 'parab.dat' using 1:2
plot f(x), 'parab.dat' using 1:2 with points lw 2
```


6. Reunindo as informações acima, você poderia digitar apenas:

Gnuplot: Linhas de comando

```
f(x) = a*x**2 + b*x + c
fit f(x) 'parab.dat' using 1:2 via a,b,c
plot f(x) smooth bezier, 'parab.dat' \
using 1:2 with points lw 2
```

7.3 Ajuste de uma cúbica

1. Usando o editor de textos, salve os dados abaixo com o nome poli.dat:

Gnuplot: Linhas de comando

```
#Tabela para ajustes polinomiais
#   X       Y
1.38 2.98
2.30 4.60
2.46 5.06
5.44 6.01
6.64 7.21
8.34 7.97
9.46 8.16
11.36 8.10
12.38 8.24
13.44 8.62
14.24 8.76
14.76 9.40
16.52 10.55
17.44 11.08
```

2. Para ajustar a cúbica $y = ax^3 + bx^2 + cx + d$ aos dados da tabela em poli.dat, insira esta função na linha de comando do Gnuplot, com:

Gnuplot: Linhas de comando

```
f(x) = a*x**3 + b*x**2 + c*x + d
```

3. Para obter os parâmetros a , b , c e d , utilize a coluna 1 para a variável x e a coluna 2 para a variável y . Ajuste a função aos dados, com:

Gnuplot: Linhas de comando

```
fit f(x) 'poli.dat' using 1:2 via a,b,c,d
```

4. Para plotar somente os dados escolhidos na tabela, utilize:

Gnuplot: Linhas de comando

```
plot 'poli.dat' using 1:2
```

5. Para plotar a cúbica ajustada e os dados da tabela, pode-se utilizar:

Gnuplot: Linhas de comando

```
plot f(x), 'poli.dat' using 1:2
```

6. Reunindo as informações acima, você poderia digitar apenas:

Gnuplot: Linhas de comando

```
f(x) = a*x**3 + b*x**2 + c*x + d
fit f(x) 'poli.dat' using 1:2 via a,b,c,d
plot f(x) smooth bezier lw 2, 'poli.dat' \
using 1:2 with points 3
```

7. Quando maior o grau do polinômio utilizado no ajuste, parece melhor o resultado. Em outras palavras, para ajustar funções polinomiais a conjuntos de pontos, quanto maior for o grau escolhido, deveremos obter pontos mais próximos da curva.
8. Como exercício, ajuste os dados da tabela poli.dat a um polinômio completo de sexto grau com os parâmetros a , b , c , d , e , g e h .

7.4 Ajuste de curva trigonométrica

1. Ajuste a curva $y = a \sin(bx) + c \cos(dx) + e$ aos dados da tabela trig.dat:

Gnuplot: Linhas de comando

```
#Tabela para o ajuste de uma função trigonométrica
# X Y
1.44 5.64
2.46 6.08
2.64 6.18
4.36 7.46
5.58 7.27
6.03 6.59
7.12 5.87
8.20 5.53
9.14 6.55
10.84 7.49
12.44 6.91
13.98 5.83
15.76 6.04
16.50 6.74
18.00 7.50
18.40 7.46
```

inserindo a função com:

Gnuplot: Linhas de comando

```
f(x) = a*sin(b*x)+c*cos(d*x)+e
```

2. Obtém-se a , b , c , d e e para o melhor ajuste usando a coluna 1 para x e a coluna 2 para y . Ajusta-se a função aos dados, digitando:

Gnuplot: Linhas de comando

```
fit f(x) 'trig.dat' using 1:2 via a,b,c,d,e
```

3. Plota-se a curva ajustada e os dados da tabela com:

Gnuplot: Linhas de comando

```
plot f(x), 'trig.dat' using 1:2
```

4. Resumo das informações acima:

Gnuplot: Linhas de comando

```
f(x) = a*sin(b*x)+c*cos(d*x)+e
fit f(x) 'trig.dat' using 1:2 via a,b,c,d,e
plot f(x), 'trig.dat' using 1:2
```

7.5 Ajuste de uma exponencial

1. Salva a tabela abaixo com o nome expo.dat.

Gnuplot: Linhas de comando

```
#Tabela para o ajuste exponencial
# X Y
0.44 0.46
1.40 0.70
2.70 1.48
3.78 1.74
4.50 2.72
5.50 3.46
5.74 5.24
6.72 4.26
7.10 5.70
7.50 6.54
7.86 8.06
8.64 7.70
8.84 8.78
9.68 9.64
10.44 9.26
10.76 11.18
```

2. Para ajustar a exponencial $y = ax^b$ à tabela expo.dat, insira a função na linha de comando, com:

Gnuplot: Linhas de comando

```
f(x) = a*x**b
```

3. Para obter a e b , use a coluna 1 para a variável x e a coluna 2 para a variável y , com:

Gnuplot: Linhas de comando

```
fit f(x) 'expo.dat' using 1:2 via a,b
```

4. Plote a exponencial ajustada e os dados da tabela expo.dat, com:

Gnuplot: Linhas de comando

```
plot f(x), 'expo.dat' using 1:2
```

5. Resumindo:

Gnuplot: Linhas de comando

```
f(x) = a*x**b
fit f(x) 'expo.dat' using 1:2 via a,b
plot f(x), 'expo.dat' using 1:2
```

8 Comandos para plotar superfícies $z = f(x, y)$ em 3D

1. Plotaremos o parabolóide $z = 100 - x^2 - y^2$ com alguns comandos complementares para uma melhor visualização 3D

2. Para plotar o parabolóide acima, digite:

Gnuplot: Linhas de comando

```
splot 100-x**2-y**2
```

3. Para dar nomes aos eixos coordenados, digite

Gnuplot: Linhas de comando

```
set xlabel "Eixo OX"
set ylabel "Eixo OY"
set zlabel "Eixo OZ"
```

4. Escolha as marcas nos eixos x, y e z do seguinte modo:

Gnuplot: Linhas de comando

```
set xtics 004
set ytics 004
set ztics 050
```

5. Para dar um efeito de sólido ao gráfico, escondendo as partes que ficam *por trás*, use o comando:

Gnuplot: Linhas de comando

```
set hidden3d
```

6. Com o comando abaixo, pode-se exibir as curvas de nível da superfície plotada

Gnuplot: Linhas de comando

```
set contour base
```

7. Para indicar o intervalo onde ficam as curvas de nível e o incremento de uma curva para outra, com as curvas **marcadas a cada 25 unidades, de 0 a 100**, digite:

Gnuplot: Linhas de comando

```
set cntrparam levels incremental 0,25,100
```

8. Podemos alterar a resolução gráfica 2D e 3D. Em geral, para funções 2D, são tomados 100 pontos no eixo OX. Para funções 3D, o Gnuplot toma 10 pontos no eixo OX e 10 pontos no eixo OY com uma grade de 100 pares ordenados no plano e então calcula a função nestes 100 pontos, traçando o o gráfico depois.

9. Para melhorar os gráficos 2D, pode-se alterar a quantidade de pontos com:

Gnuplot: Linhas de comando

```
set samples 200
```

10. Para melhorar os gráficos 3D, alteramos a quantidade de pontos com a número de pontos no eixo 0X e b número de pontos no eixo 0Y, digitando:

Gnuplot: Linhas de comando

```
set isosamples 200, 300
```

9 Scripts no Gnuplot: Uma alternativa para plotar gráficos

1. O Gnuplot pode gerar gráficos através do carregando de um arquivo com extensão gnu que contém todas as informações para o gráfico. Este arquivo é denominado script e pode ser criado em um editor de texto puro.
2. Script para plotar $f(x) = \sin(x)/x$ e a sua reta tangente no ponto $x = 7$. Abra o editor de textos e digite os comandos:

Gnuplot: Linhas de comando

```
# Script - Plotar f(x)=sin(x)/x e sua reta tangente em x=7.
reset
cd 'C:\gnuplot\SeuNome'
set xrange [5:9]
set yrange [-0.2:0.4]
set xtics 5,1,9
set ytics -0.2,0.2,0.4
set xlabel 'Eixo-x'
set ylabel 'Eixo-y'
set grid
f(x)=sin(x)/x
df(x)=cos(x)/x-sin(x)/x**2
g(x)=f(7)+df(7)*(x-7)
plot f(x), g(x)
# Fim do script (Este é um comentário)
```

3. Salve este arquivo na pasta SeuNome com o nome script01.gnu, não se esquecendo de alterar a opção Salvar como tipo: Todos os arquivos, para que o arquivo seja salvo com a extensão gnu ao invés de txt.
4. Em seguida, abra o Gnuplot, e digite na linha de comando

Gnuplot: Linhas de comando

```
cd 'c:\gnuplot\SeuNome'
load 'script01.gnu'
```

5. Do mesmo modo, como foi criado o script acima, podemos criar scripts para ajustar todas as curvas já apresentadas nas seções anteriores.
6. A vantagem do script é que se ocorrer um erro na digitação, basta corrigir o script e executar de novo o script.

7. Script para ajustar uma reta a um conjunto de dados.

Gnuplot: Linhas de comando

```
#Script para o ajuste de uma reta
reset
cd 'c:\gnuplot\SeuNome'
set terminal png
set output 'reta.png'
f(x) = a*x+b
fit f(x) 'reta.dat' using 1:2 via a,b
plot f(x), 'reta.dat' using 1:2 \
with points lw 2
#Fim do script
```

8. No Gnuplot, o comando `set terminal png` informa que a saída será um arquivo gráfico de extensão `png` e o comando `set output 'reta.png'` informa que este arquivo terá o nome `reta.png`.

9. Para ver o resultado do ajuste na própria janela do Gnuplot, use o comando

Gnuplot: Linhas de comando

```
set terminal win
```

omitindo o comando `set output 'reta.png'`

10. O comando `reset` serve para restaurar as configurações originais do Gnuplot, caso o Gnuplot já esteja sendo usado quando foi carregado o arquivo `gnu`.

11. Script para ajustar uma parábola a um conjunto de dados.

Gnuplot: Linhas de comando

```
#Script para o ajuste de uma parábola
reset
cd 'c:\gnuplot\SeuNome'
set terminal png
set output 'parab.png'
f(x) = a*x**2+b*x+c
fit f(x) 'parab.dat' using 1:2 via a,b,c
plot f(x) smooth bezier, 'parab.dat' \
using 1:2 with points lw 2
```

12. Um script para ajustar uma curva cúbica a um conjunto de dados.

Gnuplot: Linhas de comando

```
#Script para o ajuste de uma curva cúbica
reset
cd 'c:\gnuplot\SeuNome'
set terminal png font "arial,11"
set output 'cub.png'
f(x) = a*x**3+b*x**2+c*x+d
fit f(x) 'cub.dat' using 1:2 via a,b,c,d
plot f(x) smooth bezier lw 2, \
'cub.dat' using 1:2 with points lw 2
```

13. Script para fazer o ajuste exponencial:

Gnuplot: Linhas de comando

```
cd 'c:\gnuplot\SeuNome'; FIT_LIMIT=1E-10
set terminal png font "arial,11"
set output 'expo.png'; f(x) = a*x**b
set grid; set size 1,1
set xtics 1; set ytics 1
fit f(x) 'expo.dat' using 1:2 via a,b
plot f(x) smooth bezier lw 2, 'expo.dat' \
using 1:2 with points lw 2
```

Este script possui vários comandos na mesma linha, separados por ; (ponto e vírgula), produzindo o mesmo resultado.

Exercícios

1. Criar scripts para gerar ajustes trigonométrico e polinomial, bem como criar scripts com todas as informações necessárias para a plotar superfícies em 3D.
2. Obtenha um ajuste linear com a tabela `freio.dat`, realizando:
 - (a) Ajustar a tabela `freio.dat` a uma parábola $v = ax^2 + bx + c$.
 - (b) Comparar graficamente a função obtida com os pontos da tabela.
 - (c) Com o gráfico, obter os valores de x para os quais $v = 72$ e $v = 85$.
 - (d) Ajustar tabela `freio.dat` à reta $w = dx + e$.
 - (e) Use este ajuste linear para obter os valores x tal que $w = 72$ e $w = 85$.
 - (f) Comparar os ajustes linear e quadrático, chegando a alguma conclusão.
3. Agora, obtenha ajustes lineares com as tabelas `peso.dat`, `inde.dat`.

10 Ajustes de superfícies no Gnuplot

Como está escrito na página do Gnuplot, este programa não é *forte* para gerar ajustes de superfícies a conjuntos de dados 3D ou 4D ou ainda em espaços com mais dimensão. Deixamos este assunto para uma outra ocasião, mas observamos que ajustar uma superfície em 3D é um processo similar ao desenvolvido neste painel.