

gnuplot
máquina de calcular gráfica

Praciano-Pereira, Tarcisio *

4 de maio de 2018
preprints da Sobral Matemática
no. 2018.03
Editor Tarcisio Praciano-Pereira
tarcisio@member.ams.org

Resumo

gnuplot, sim, escrito em minúsculas como é sugerido pelos autores, é um máquina de calcular gráfica com alguma capacidade de linguagem de script. Eu vou na primeira seção extender-me um pouco mais sobre o programa porque o resumo tem que ser resumido. Em suma você pode fazer o gráfico de qualquer função conhecida pelas linguagens de programação. Abra um terminal do gnuplot e digite

```
plot sin(x), 0
```

e você verá o gráfico do seno.

Em vez de escrever um grande artigo falando tudo sobre gnuplot, vou escrever uma serie de pequenos artigos tratando de tópicos relacionados com o programa e posteriormente farei uma recompilação de todo o trabalho.

palavras chave: gnuplot, pacote gráfico, linguagem de script.

gnuplot, yes, lowercase as suggested by the authors of the program, is a graphical calculator with a limited possibility as scripting language. In the first section I will go a bit deeper into describing the program to preserve the idea of the abstract not to be very long. But in summary, you may see the graph of any function known by any programming language, open the gnuplot terminal e code

```
plot sin(x), 0
```

and you will see the graph of sine.

Instead of writing a big paper about **gnuplot**, I shall write several small papers concentrated on topics of the program but later I shall recompile they all in a unique document. keywords: gnuplot, graphical calculator, scripting language.

*tarcisio@member.ams.org

1 gnuplot, uma calculadora gráfica

Este artigo ainda está em preparação, quando tiver atingido sua versão final esta observação irá desaparecer. A razão de que eu o publique ainda na versão preliminar se encontra na dedicatória que aparece mais abaixo, aproveitando a data, primeiro de maio.

gnuplot começou como uma brincadeira, e como toda brincadeira bem feita, deu certo. É um programa que tem dono, Copyright (C) 1986 - 1993, 1998, 2004, 2007 Thomas Williams, Colin Kelley e foi iniciado por Thomas Williams, Colin Kelley porque eles estavam precisando ver melhor alguma ondas quando estavam estudando um curso avançado de Mecânica e na brincadeira resolveram construir um programa que fizesse gráficos: gnuplot. [5]

Como acharam que estava interessante, divulgaram deixando que quem quisesse fizesse uso do programa, e realmente a coisa deu certo, muita gente usou e muito mais gente passou a cooperar transformando gnuplot no motor gráfico de vários programas importantes, como *octave*.

Abra um terminal do gnuplot, e você pode receber a mensagem

```
-bash: gnuplot: comando não encontrado
```

em que *bash*, que é a alma por trás do *Linux*, estaria lhe dizendo que o gnuplot ainda não estaria instalado. Se estiver usando *Debian/gnu/linux*, como eu, a coisa é simples de resolver

```
sudo apt-get install gnuplot
```

em alguns minutos o programa estará instalado. Digite então gnuplot e vai ser aberto um terminal rodando o programa. Digite a linha de comando que se encontra no resumo e você vai ver o gráfico do seno. Será um gráfico mal feito porque gnuplot otimiza a janela gráfica ocupando-a inteiramente com o gráfico mas você melhorar isto com digitando, antes do comando plot, os comandos

```
set xrange [-3.1415:3.1415]; set yrange [-4.:4];
```

que estão definindo as dimensões da janela gráfica. Digite agora o comando

```
replot
```

que refaz agora o gráfico de forma bem mais parecida como a gente espera que seja o seno. Neste ponto já tenho alguns comentários complementares:

1. Na linha em que foi redimensionada a janela gráfica há dois comandos. Você pode colocar quantos comandos quiser numa mesma linha, separando-os com *ponto-e-vírgula*. gnuplot não obriga que você termine os comandos com *ponto-e-vírgula*, se for um único comando na linha, o fim de linha é suficiente. Mas você pode, se quiser, terminar todos os comandos com *ponto-e-vírgula* e querendo colocar mais de um numa mesma linha, tem que separá-los com *ponto-e-vírgula*.
2. O comando **replot** é muito importante quando você tiver fazendo gráficos dentro dum laço, o **replot** atualiza a janela gráfica a cada passagem do laço. Eu vou voltar a apresentar este comando quando discutir laços com **replot**

Eu vou, de propósito, evitar de incluir gráficos neste artigo, porque todos os gráficos podem ser produzidos com comandos do gnuplot e eu vou deixar que você os execute numa janela gráfica ao lado do artigo que você está lendo. Você, certamente, já deve ter tentado, raspar e colar o comando da janela do navegador para o terminal do gnuplot: *funciona!*

Dedico este artigo ao
prisioneiro político
Luís Inácio Lula da
Silva

Espero que você já o tenha feito com os comandos que lhe apresentei, mas vá além, experimente com outras funções e vou lhe sugerir algo que torne a leitura deste artigo mais iterativa:

```
plot sin(x)*(x+3)*(x-5), 0
```

e o resultado pode ter sido desapontador, certo? Se você tiver visto apenas parte do gráfico a razão se encontra no comando

```
set xrange [-3.1415:3.1415]; set yrange [-4:4];
```

que continua sob efeito. Experimente

```
unset xrange; unset yrange;
```

e volte a pedir o gráfico da função que você tiver definido. Use o `replot`.

Neste ponto já temos alguns itens sobre o que conversar e umas outras tantas coisas para você experimentar. Experimentar é uma palavra chave em se tratando de programação, parte do conhecimento você obtém fazendo testes, experimentando. Fazendo pequenas modificações nos comandos e vendo que efeito se produzem.

1. O comando `set`, do inglês *ligue ou defina*, serve para dar valores à variáveis que controlam o comportamento da janela gráfica. Não se esqueça de que `gnuplot` é tocado à gráficos.
2. Aqui eu usei `set xrange [-3.1415:3.1415]` que define a variação no eixo *OX* e `set yrange [-4:4]` definindo a variação no eixo *OY* e também existe `set zrange [-31:14]` para estabelecer a variação no eixo *OZ* quando você trabalhar com funções de duas variáveis cujos gráficos vão usar uma janela gráfica 3D, experimente

```
splot x**2 + 3*x*y + y**3, 0
```

e o resultado deve ter sido desanimador! Certo? É que agora eu pedi, e você acompanhou, uma tarefa mais complicada ao `gnuplot`. Estou usando o comando `splot` e vou deixar para entrar nos seus detalhes num próximo texto, ele é o comando para fazer o gráfico de superfícies.

Mas o próximo item vai lhe dar uma saída imediata para esta questão caso você queira enfrentar a barra.

3. `gnuplot` traz uma ajuda inclusa no programa. Experimente digitar no terminal do `gnuplot`

```
help.
```

Desanimador? É texto em inglês! Infelizmente este é um problema que nós ainda temos que enfrentar enquanto ainda vivemos com uma ciência colonialista. Nossa língua é português mas somos obrigados a dominar inglês porque quase toda a literatura, que nós mesmos escrevemos, o fazemos em inglês. Mas você pode superar o problema se enfrentar o inglês e sair lendo, eliminando o receio. Você pode aprender inglês como aprendeu português, se lembra? sem nenhum auxílio, sem nenhuma didática, na marra. Foram falando com você e você foi adivinhando onde queriam chegar. Com o inglês pode ser mais fácil, existe um monte de auxílio on-line, mas comece usando o mesmo método do aprendizado do português: vá usando. Experimente, force a compreensão e aos poucos o vocabulário irá crescendo. Acredite, e enfrente!

Boa parte do que aparecer na primeira tela do `help` interessa pouco, é

Interessa pouco neste primeiro momento!

uma descrição genérica do `gnuplot` e você verá que vou tratar de muita coisa nos artigos que ainda virão. Pegue de leve, leia por cima, e sobre tudo obedeça a ordem que aparece no final da tela:

Press return for more:

dê `enter` para continuar, (tradução livre).

Na próxima tela aparece alguma coisa (duas coisas importantes), uma de que já falei acima. Não vou fazer uma tradução direta, mas vou dizer-lhe o conteúdo do primeiro parágrafo que termina com a palavra *Example* se referindo a linha seguinte.

- (a) os nomes das funções, quer dizer dos comandos do `gnuplot`, são sensíveis à caixa alta/caixa baixa. Quer dizer: `Plot`, não existe, e é diferente de `plot` o comando para fazer gráficos.
- (b) você pode colocar vários comandos numa mesma linha, já falei disto acima, desde que os separe com `ponto-e-vírgula`, como aparece no exemplo. Uma coisa interessante que você pode fazer é abrir dois terminais do `gnuplot`, num deles ler o `help` e no outro executar comandos. Esta é uma facilidade típica de quem usa terminais em `Linux`. Estou lhe dizendo para abrir dois exemplares do `gnuplot`, num deles ler o `help` e no outro executar o programa. É uma novidade, tente e vou mostrar-lhe como se faz usando `Linux` e possivelmente também funciona em outro “sistema”, mas não posso garantir porque uso exclusivamente `Linux`.

Abra um terminal e nele digite

```
gnuplot &
```

ao fazê-lo vai se abrir um novo terminal em que estará rodando `gnuplot` deixando o terminal anterior livre para novos comandos, é o significado do “&” digitado no fim do comando, é um comando de `Linux` que significa *deixar um processo rodando por trás da cena*. Retorne ao terminal anterior e acione a tecla “seta para cima” o que fará aparecer o comando anterior, dê `<enter>` e novo terminal rodando `gnuplot` estará no ar. Num deles acione `help` e o outro reserve para rodar os comandos do `gnuplot`. Você está com duas cópias do programa no ar e as duas cópias estarão rodando de forma independente. Isto é `Linux`!

- (c) O exemplo a que se refere `help` é bem interessante e vou traduzi-lo com a sugestão de que você o execute transformando-o a seu gosto e eu mesmo já o traduzi modificado. Compare como está na página do `gnuplot` e como traduzi aqui. Raspe e cole no outro terminal e verifique o resultado.

```
set title "meu primeiro gráfico"; plot sin(x);  
pause -2 "Enter para continuar"
```

- (d) No exemplo eu usei o comando `pause` que está em inglês mas é exatamente a mesma palavra portuguesa... inglês é considerada uma língua latina porque uma grande percentual das palavras inglesas são nossas, eles foram colonizados pelos latinos! `title` é uma corruptela de `título` pelo que se pode ver que inglês é português mal escrito!

O comando `pause` recebe dois argumentos, o primeiro é `-2` e o segundo é uma frase e o seu efeito é parar o motor esperando que você bata no `enter`. É muito útil quando você for apresentações com vários gráficos aparecendo cada na sua vez com este comando controlando quando a sessão deve prosseguir. Eu vou voltar mais a frente quando falar de laços que é quando este comando será muito importante.

Na próxima seção vou sugerir alguns exercícios usando o material que foi exposto para que você faça algumas experiências que irão levá-la a adquirir um uso prático do `gnuplot` para resolver algumas questões de Matemática elementar mas logo irão servir para exemplos mais avançados.

2 Exemplos simples com `gnuplot`

Como falei na primeira seção, o objetivo é que você tenha dois terminais abertos um que esteja rodando `gnuplot` e no outro em que você esteja consultando a ajuda (`help`) do `gnuplot`, afinal não tem sentido aprender a usar um programa sem rodar o programa.

Uma das formas de usar `help` consiste em pedir um `help` específico, por exemplo

```
help function
```

em que estou pedindo ajuda para saber como definir uma função em `gnuplot`. A primeira tela do `help` vai mostrar-lhe como definir a função `approx(ang)`. O autor do `help` selecionou como variável “`ang`” e poderia ser qualquer cadeia de caracteres que seja válida como nome para um função. Consulte a este respeito um livro qualquer de linguagem de programação que as condições são as mesmas em qualquer delas, mas resumidamente é um aglomerado de letras, sem espaços. Em Matemática se usam comumente: x, y, z e no exemplo está definida a função `approx(ang)` em que a variável é `ang`. O primeiro exemplo poderia ser

```
f(x) = (x+3)*(x-5);
```

1. Raspe e cole as duas linhas depois de “Example:”, de `enter` para ver o resultado. “*Example*” é como se diz *exemplo* em inglês, e eu já lhe avisei que inglês é português mal falado!
2. Não parece ter acontecido nada, certo? Use o ratinho, vulgarmente chamado de `mouse`, com botão da direita pressionado para raspar em volta do que parece ser a linha do eixo OX no gráfico. Selecione um pequeno cubo em volta desta linha. Você vai ver porque a função definida pelo `help` se chama `approx`. Seu gráfico é uma curva tangente ao gráfico do `seno` na origem.
3. Foi bom este exemplo, para que você acostume com a ideia de que o `help` nem sempre é uma boa ajuda. Seja crítica e procure entender *o que está por baixo*.
4. Execute o exemplo do `help` substituindo `approx(ang)`, `f(x)`, e o título por “minha primeira funcao”. Não se esqueça de definir a função $f(x)$ no terminal do `gnuplot`.

5. Defina o título como um comando separado, o primeiro, e repita o exemplo: a sintaxe é `set title "minha primeira funcao"`.

6. A função potência em gnuplot é `x**n` que significa em Matemática x^n . Você pode alterar a definição:

```
pow(x,n)=x**n;
```

defina uma função do terceiro grau e faça o seu gráfico junto com uma função do primeiro grau. Não se esqueça de usar `set xrange` que já apareceu na primeira seção. Se preferir usar a minha notação para a função potência, defina primeiro `pow()`;

7. Defina, comandos separados

(a) `f1(x)`, `f2(x)`, `f3(x)`, `f4(x)` como `pow(x,3)`, `pow(x,2)`, `x`, `4`, respectivamente.

(b) `g(x) = pow(x,3) + pow(x,2) + x + 4`

(c) `set title "gnuplot sabe somar funções "`

e execute `plot g(x), f1(x)+f2(x)+f3(x)+f4(x), 0`

No próximo artigo eu vou mostrar-lhe como se pode memorizar o trabalho feito sem precisar, em cada ocasião, começar do começo.

Índice Remissivo

gnuplot, 1

bash, 1

octave, 1

por trás da cena
 processo, 3

Referências

- [1] Tom M. Apostol. *Calculus vol II*. Blaisdell Publishing Company, 1962.
- [2] David I. Bell Landon Curt Noll and other. Calc - arbitrary precision calculator. Technical report, <http://www.isthe.com/chongo/>, 2011.
- [3] Harold J. Larson. *Introduction to Probability Theory and Statistical Inference*. John Wiley & Sons, Inc - Wiley International Edition, 1969.
- [4] Tarcisio Praciano-Pereira. *Cálculo Numérico Computacional*. Sobral Matemática, 2007.
- [5] Thomas Williams, Colin Kelley, and many others. gnuplot, software to make graphics. Technical report, <http://www.gnuplot.info>, 2010.