Introduction à Gnuplot

Un logiciel libre dédié aux graphiques scientifiques

Introduction

Gnuplot est un logiciel utile pour tracer graphiques de toutes les formes. S'il est utilisé correctement, il peut fournir des résultats très professionnels, souvent bien supérieurs à ceux que l'on peut obtenir avec Excel, par exemple. Exactement comme Maxima, il s'agit d'un logiciel qui existe depuis beaucoup de temps (1986, pour la précision) et il est complètement multi plateforme. Ceci est un avantage non négligeable... Dans mon travail, je me suis retrouvé à passer continûment entre Windows, Linux et MacOSX : depuis plusieurs années, j'essaye d'utiliser des logiciels qui ne me contraignent pas à utiliser un système d'exploitation en particulier. Une difficulté majeure de Gnuplot est le fait qu'il s'agit d'un logiciel interactif à ligne de commande qui peut être utilisé à partir du terminal. Les premières fois, il semble suivre une logique difficile à saisir, d'où la présence de cette petite page, afin de commencer à se faire une idée sur les possibilités de ce logiciel.
Cette page a été écrite sur la base de plusieurs interventions que j'ai faites sur le forum en italien de Macitynet, où je participe souvent avec le pseudonyme de DarwinNE :

http://www.macitynet.it/forum/showthread.php?t=59206

Comme j'ai fait pour Maxima, j'évite de décrire en détail chaque commande (pour cela, il suffit de taper `help`, suivi de la commande dont on cherche les détails). Je préfère fournir par contre des exemples concrets, qui peuvent être utiles pour bien comprendre la philosophie qui est utilisée par le logiciel.

Quelques idées pour commencer

La première chose dont il faut se rendre compte est que Gnuplot ne dessine pas directement sur l'écran ou sur une fenêtre, mais plutôt dialogue avec un *terminal* graphique. Ce dernier est un module logiciel qui a pour but de mettre à disposition un certain nombre de fonctions graphiques de base, qui sont utilisées ensuite par Gnuplot. En vertu de l'importance centrale du terminal, Gnuplot au démarrage nous fournit quelques informations sur sa version et spécifie le terminal qui est utilisé :

```
[davidebucci@davide-bucci-portable]$ gnuplot

        G N U P L O T
        Version 4.2 patchlevel 0
        last modified March 2007
        System: Darwin 8.10.1
```

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

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

        Send comments and help requests to
        Send bug reports and suggestions to


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

Pour terminer la session, il suffit de donner la commande `quit` :

```
gnuplot> quit
[davidebucci@davide-bucci-portable]$
```

La version de Gnuplot que je suis en train d'utiliser est la 4.2 (plutôt récente, du moins dans le moment où j'écris ce texte). Le terminal utilisé s'appelle Aquaterm et fonctionne très bien sous MacOSX et que l'on peut trouver sur VersionTracker et installer de façon indépendante. Si vous utilisez Linux ou Windows, vous trouverez probablement les terminaux x11 ou Windows à la place d'Aquaterm. MacOSX peut aussi utiliser le terminal x11, mais il n'y a aucune raison de faire cela, car Aquaterm fonctionne très bien et il est très léger.
Pour obtenir une liste de terminaux disponibles, par curiosité, il suffit de taper `set terminal` et ensuite retour. Il faut faire très attention au fait que Gnuplot distingue la casse des commandes qui sont insérées : `SeT TeRmInAl` ne va pas fonctionner, de la même façon que `SET TERMINAL`.

```
gnuplot> set terminal

Available terminal types:
             aqua  Interface to graphics terminal server for Mac OS X
             dumb  ascii art for anything that prints text
              emf  Enhanced Metafile format
            emtex  LaTeX picture environment with emTeX specials
         epslatex  LaTeX picture environment using graphicx package
              fig  FIG graphics language for XFIG graphics editor
              gif  GIF images using libgd and TrueType fonts
             jpeg  JPEG images using libgd and TrueType fonts
            latex  LaTeX picture environment
               mf  Metafont plotting standard
              png  PNG images using libgd and TrueType fonts
       postscript  PostScript graphics, including EPSF embedded files (*.eps)
              svg  W3C Scalable Vector Graphics driver
              x11  X11 Window System
gnuplot>
```

Dans l'exemple que l'on vient de fournir, en réalité la liste était beaucoup plus longue, mais je l'ai raccourcie un peu, en me limitant aux terminaux les plus intéressants. Nous allons voir quelques exemples dans la suite de notre discussion. On peut aussi se trouver dans la situation où, dans des installations incomplètes par exemple, Gnuplot n'ait pas un terminal défini par défaut. Dans ce cas, au démarrage, on aura quelque chose du style :

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

Pour choisir le terminal en ce cas, il suffit d'utiliser la commande `set terminal`, suivie par le terminal que l'on souhaite utiliser :

```
gnuplot> set terminal aqua
Terminal type set to 'aqua'
Options are '0 title "Figure 0" size 846 594 font "Times-Roman,14" noenhanced solid'
gnuplot>
```

Comme il peut être ennuyant de spécifier le terminal à chaque démarrage de Gnuplot, on peut définir une variable de système, `GNUTERM`, avec le terminal à utiliser. Voici un extrait du fichier .bash_profile, qui décrit la configuration à utiliser pour le terminal Gnuplot.

```
export GNUTERM=aqua
```

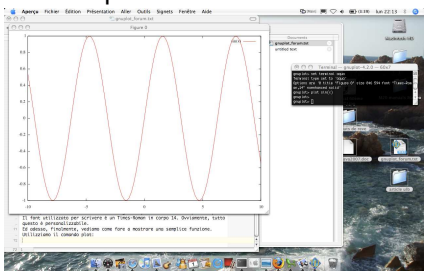
Le logiciel ira donc utiliser le terminal "aqua" ; au moment de dessiner un graphique, une fenêtre de 846 pixel fois 594 pixel sera ouverte à l'écran. La police est Times-Roman en corps 14. Tout ceci est bien sûr modifiable.

Comment tracer une fonction

Et maintenant, on va voir comment il faut faire pour tracer une fonction simple, par exemple y=sin(x). Nous allons utiliser la commande `plot`:

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

Si tout s'est bien passé, nous devrions nous trouver devant la fenêtre qui est montrée ici :



Néanmoins, le graphique que l'on vient d'obtenir n'est pas très satisfaisant pour plusieurs raisons. D'abord, il pourrait être pratique que la légende n'apparaisse pas, car il y a seulement une courbe qui est montrée. Nous allons aussi donner un titre au dessin et des noms aux axes :

```
gnuplot> set title "Grafico della funzione sin(x)"
gnuplot> set xlabel "angolo x in radianti"
gnuplot> set ylabel "sin(x)"
gnuplot> unset key
gnuplot> replot
gnuplot>
```

Avec la commande `set` nous pouvons modifier la configuration de Gnuplot pour la session courante. La première, deuxième et troisième ligne du code que l'on vient d'écrire nous spécifient le titre du graphique et les noms des axes. La quatrième ligne (`unset key`) spécifie de ne pas tracer la légende, qui dans la terminologie de Gnuplot est appelée "key". La commande `replot` ré-dessine le graphique avec la nouvelle configuration.

Il y a un détail important qu'un oeil averti n'aura pas manqué de remarquer. Dans un graphique qui doit apparaître dans un rapport en français, il faudrait utiliser la virgule comme séparateur décimal. Cela peut être obtenu à l'aide de la commande `set decimalsign`, suivie du symbole à utiliser, dans notre cas donc la virgule, mis entre guillemets, car il s'agit d'une chaîne de caractères.

```
gnuplot> set decimalsign ','
gnuplot> replot
gnuplot>
```

Il serait sympa aussi de visualiser la courbe entre -2*pi et 2*pi, en utilisant comme graduation de l'axe des abscisses des sous-multiples de pi. On peut obtenir tout ceci avec un code à

vrai dire un peu tordu :

```
gnuplot> set xrange [-2*pi: 2*pi]
gnuplot> set xtics ("-2*pi" -2*pi, "-3*pi/2" -(3*pi
-pi/2, "0" 0)
gnuplot> set xtics add ("2*pi" 2*pi, "3*pi/2" (3*pi
gnuplot> replot
gnuplot>
```

La commande `set xrange`, suivie d'un intervalle de valeurs comprises entre crochets spécifie au logiciel de représenter seulement la portion des abscisses comprise entre les deux valeurs spécifiées. La commande `set xtics` introduit des gradations dans des endroits très précis de l'axe x et les représente dans le graphique avec les étiquettes que l'on voit dans la figure. On peut remarquer en passant comme `set tics add` nous permet d'ajouter de nouvelles étiquettes, sans retirer celles ajoutées précédemment.
La fonction sin(x) que nous avons utilisée est représentée par le logiciel en faisant un échantillonnage à intervalles régulières. Par défaut, le logiciel divise en 100 points la région montrée dans le graphique. Voici comment on peut les mettre plus en évidence :

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

Le nombre de points qui sont calculés peut être modifié à l'aide de la commande `set samples` :

```
gnuplot> set samples 1000
gnuplot> plot sin(x) with points
gnuplot> set samples 20
gnuplot> replot
gnuplot>
```

Maintenant, nous revenons à la visualisation avec une ligne continue, mais nous allons changer le style du trait. Le résultat dépend de la configuration du terminal que l'on va utiliser. Dans mon cas, avec Aquaterm, j'obtiens une ligne verte plutôt marquée pour l'exemple suivant :

```
gnuplot> plot sin(x) with lines lt  2 lw 3
```

Le modificateur `lt` est une abréviation qui indique "line type" et indique le style (souvent la couleur) de la ligne. Dans certains terminaux, surtout si l'on ne peut pas disposer de la couleur, ce qui change est un pointillage de différents types. Cette technique est fondamentale à l'impression, lorsque l'on veut produire une dissertation qui sort bien à l'impression en noir et blanc. Le modificateur `lw` indique plutôt "line width", c'est-à-dire la largeur de la ligne. Dans des

commandes complexes comme `plot`, il vaut mieux de respecter attentivement l'ordre des paramètres, afin d'éviter des ambiguïtés :

```
gnuplot> plot sin(x) w l lt  2
```

Pour terminer cette discussion, nous allons tracer le graphique de la tangente, du sinus et du cosinus aux alentours de l'origine. Comme Gnuplot tente d'ajuster automatiquement les axes, il risquerait de donner une indication faussée, car la tangente présente une asymptote verticale que le logiciel ne sait pas traiter. Il faut donc spécifier un intervalle de visualisation sur l'ordonnée :

```
gnuplot> set yrange [-5:5]
gnuplot> set samples 1000
gnuplot> set xrange [-2*pi: 2*p
gnuplot> set xtics ("-2*pi" -2*
-pi/2, "0" 0)
gnuplot> set xtics add ("2*pi"
gnuplot> plot tan(x) with lines
title 'coseno'
gnuplot>
```

Graphiques à partir d'un fichier

Nous allons continuer cette visite guidée de Gnuplot en parlant un petit peu de graphiques tracés à partir de données expérimentales. Gnuplot est bien adapté à accomplir cette tâche et il est capable d'effectuer toute une série de traitements qui peuvent être très utiles dans beaucoup de situations.
Dans beaucoup de situations pratiques, on doit traiter des fichiers qui sont composés par beaucoup de données organisées en colonnes

(par exemple) issues d'un instrume de mesure automati ou quasi-automati Très souvent, il n'est pas pratique de traiter ce type de données à l'aide d'une feuille de calcul, car par exemple il y a trop de points à traiter. Nous allons voir comme Gnuplot est parfaitem à son aise dans ces situations Nous allons donc préparer le fichier comme il suit et nous allons l'appeler `data_gnu`