

# INTRODUCTION À GNUPLOT\*†

Licence de physique, Université Pierre et Marie Curie

## 1 Pour commencer

Le logiciel `gnuplot` sert à tracer des graphiques, de façon interactive, sur un écran ou sur une imprimante. Ce n'est probablement ni le plus convivial ni le plus puissant des logiciels de ce type, mais il a le mérite d'être très répandu sur les machines UNIX. Il est bon d'en connaître les fonctions essentielles.

Pour commencer, on tape la commande :

```
gnuplot
```

et on obtient quelque chose qui ressemble à :

```
G N U P L O T
Version 3.7 patchlevel 3
last modified Thu Dec 12 13 :00 :00 GMT 2002
System : Linux 2.4.20-20.9
```

```
Copyright(C) 1986 - 1993, 1998 - 2002
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/gnuplot-faq.html
```

```
Send comments and requests for help to
<info-gnuplot@dartmouth.edu>
Send bugs, suggestions and mods to
<bug-gnuplot@dartmouth.edu>
```

```
Terminal type set to 'x11'
```

```
gnuplot>
```

Derrière l'invite :

```
gnuplot>
```

il faut taper les commandes de dessin. Par exemple :

```
plot sin(x)
```

produira une sinusoïde.

Pour quitter `gnuplot`, taper

```
quit
```

## 2 Tracer les résultats d'un calcul

En général, ce que l'on veut tracer, ce sont les résultats d'un calcul placés dans un fichier. Par exemple, admettons que dans un programme FORTRAN, il y ait les lignes :

```
open(1,file='calcul.res')
write(1,*) x, y
close(1)
```

qui écrivent, dans le fichier `calcul.res`, les abscisses et

ordonnées d'un graphe, avec les  $x$  en colonne 1 et les  $y$  en colonne 2.

Pour tracer ces résultats dans `gnuplot` :

```
plot 'calcul.res'
```

Ceci trace  $y$  en fonction de  $x$  sur l'écran. ! Ne pas oublier les ' autour du nom de fichier. Si l'on veut que les différents points de la courbe obtenue soient reliés par des segments, on peut taper :

```
set data style lines
```

avant la commande `plot 'calcul.res'`, ou bien on peut taper directement :

```
plot 'calcul.res' w l
```

où `w l` sous-entend `with lines`.

On peut fixer les bornes du dessin avec :

```
set xrange [0.1 :0.2]
```

```
set yrange [1000 :1500]
```

qui limite  $x$  à l'intervalle  $[0.1, 0.2]$  et  $y$  à  $[1000, 1500]$ .

Il est possible de tracer plusieurs courbes sur le même dessin, par exemple, si l'on a créé deux fichiers `cal1.res` et `cal2.res`, alors :

```
plot 'cal1.res', 'cal2.res'
```

fera les deux courbes. De même, on peut comparer une courbe issue d'un fichier avec une fonction :

```
plot 'calcul.res', 5.05*exp(-(x/3.5)**2)
```

Si maintenant, l'écriture par le programme FORTRAN a été faite par la ligne :

```
write(1,*) x, y, z
```

la commande `plot 'calcul.res'` tracera, par défaut, la colonne 2 ( $y$ ) en fonction de la colonne 1 ( $x$ ). Pour tracer  $z$  (colonne 3) en fonction de  $x$  (colonne 1), il faut taper :

```
plot 'calcul.res' using 1 :3
```

ou  $y$  en fonction de  $z$  :

```
plot 'calcul.res' using 3 :2
```

La commande :

```
plot 'calcul.res', 'calcul.res' using 1 :3
```

tracera les courbes  $y = f(x)$  et  $z = g(x)$  sur le même graphe.

On peut faire un dessin à 3 dimensions par :

```
set parametric
```

```
splot 'calcul.res' using 1 :2 :3
```

## 3 Quelques commandes de finition

Les axes peuvent être étiquetés à l'aide de commandes

```
set xlabel 'X'
```

```
set ylabel 'axe des y'
```

\*Ph. Depondt.

†Améliorations, corrections et `couleurs` : D. Cabaret.

Pour mettre un titre :  
`set title 'distance focale de la lentille'`

Pour changer les dimensions du graphe, on utilise la commande `set size`. Par exemple, la commande `set size 0.721,1` produira un graphe de forme carrée. Les valeurs par défaut sont 1,1.

Par défaut, la légende associée à la commande `plot 'calcul.res'` sera 'calcul.res'. Pour changer la légende par défaut :  
`plot 'calcul.res' title 'nouvelle légende'`

Pour supprimer la légende :  
`set nokey`  
Pour la rétablir :  
`set key`

Pour tracer deux courbes de la même couleur (utile pour le TD1) :  
`plot 'trajet.res' lt 1, 'dioptre.res' lt 1`  
1 désigne le rouge, 2, le vert, 3, le bleu...  
De la même façon, l'option `lw` permet de contrôler l'épaisseur du trait (1 est la valeur par défaut).

## 4 Pour imprimer

Pour imprimer sur l'imprimante laser (qu'on appellera 'nom\_imprimante'), il faut, après les commandes de dessin, taper les quatre commandes suivantes :

```
set term postscript
set output "| lpr -Pnom_imprimante"
replot
quit
```

L'imprimante reproduira le dessin obtenu sur l'écran. L'instruction `replot` sert à refaire le dessin avec les nouvelles options. La commande UNIX :

```
lpq -Pnom_imprimante
```

donne la file d'attente sur l'imprimante. Cela permet donc de vérifier d'une part, que son propre dessin y figure bien, et d'autre part, de voir si d'autres impressions précèdent la sienne.

**!** **NE PAS** faire de copie d'écran sur l'imprimante : on n'obtient ainsi **QUE** la résolution de l'écran à la place de celle de l'imprimante qui est bien meilleure. De plus, celle-ci met beaucoup plus de temps à traiter les copies d'écran.

## 5 Quelques éléments de confort...

Les commandes de `gnuplot` peuvent être abrégées. Par exemple  
`se da sty li`  
remplace  
`set data style lines`  
et

```
p 'calcul.res' u 1 :3
remplace
plot 'calcul.res' using 1 :3
```

Il est possible de faire des fichiers de commande. Ainsi, mettons que l'on doive imprimer divers graphes sur l'imprimante laser : la séquence de commandes – un peu lassante à taper de façon répétée – est toujours la même. On peut mettre cette séquence dans un fichier baptisé, par exemple, `lp.gnu`. Ainsi la commande :

```
load 'lp.gnu'
```

enverra l'impression, à chaque fois que l'on en aura besoin.

De façon plus générale, quand on doit répéter souvent la même séquence, il est bon de l'inclure dans un fichier et de l'appeler par `load`. Un autre exemple de ce qui pourrait être mis dans un fichier serait :

```
set title 'T=300K'
set xlabel 't';set ylabel 'P'
plot 'calcul.res'
pause -1 'Appuyez sur la touche Return'
set xlabel 'P';set ylabel 'dP/dt'
plot 'calcul.res' using 3 :2
```

La commande `pause` écrit le texte et attend que l'on appuie sur `return` pour continuer. Les point-virgules permettent de mettre plusieurs instructions sur la même ligne.

Quand on travaille toujours avec les mêmes options, il peut être utile de le faire savoir automatiquement à `gnuplot`, par exemple, si l'on aime particulièrement le style `lines`, on peut aussi le faire savoir. Il suffit de créer dans son répertoire principal (`$HOME`) un fichier obligatoirement baptisé `.gnuplot` (ne pas oublier le point) dans lequel on place les commandes correspondantes, qui s'exécuteront automatiquement à chaque appel de `gnuplot`.

## 6 Aide en ligne

`gnuplot` offre de nombreuses autres ressources qu'on peut découvrir petit à petit. En effet, il fournit une aide en ligne quand on tape un point d'interrogation. Si le point d'interrogation est tout seul, on obtient une présentation générale, et une liste de commandes. Si l'on fait suivre le point d'interrogation par un nom de commande, on obtient des explications sur cette commande. Par exemple :

```
?set
```

donne des explications sur `set`. Des explications plus détaillées sur toutes les commandes décrites ici peuvent, en particulier, être obtenues de cette façon.