

Introduction à gnuplot

Anne Cadiou

Laboratoire de Mécanique des Fluides et d'Acoustique

Informatique scientifique pour le calcul
Formations transverses des Écoles Doctorales
LyonCalcul, le jeudi 2 mai 2019



Gnuplot

Programme open-source en ligne de commandes pour tracer, visualiser et analyser des données.

- Développé en 1986 par Thomas Williams et Colin Kelley.
- Disponible sur tous les systèmes.
- Utilisé en interactif et en batch.
- Relativement rapide (différents modes de qualité graphique)

Pourquoi choisir gnuplot ?

- Stable, actif, maintenu
- Disponible sur toutes les plateformes
- Personnalisable
- 1D, 2D, 3D
- Nombreux formats graphiques
- Lit des fichiers en format texte et binaire (sous certaines conditions)
- Capable de traiter efficacement des millions de points
- Efficace et peu gourmand en ressources
- Facilité de manipulation (interactivité) et scripts (batch)
- S'interface avec L^AT_EX

Démarrer avec gnuplot

Dans un terminal, lancer la commande

```
$ gnuplot
```

```
G N U P L O T
```

```
Version 5.2 patchlevel 2 last modified 2017-11-01
```

```
Copyright (C) 1986-1993, 1998, 2004, 2007-2017
```

```
Thomas Williams, Colin Kelley and many others
```

```
gnuplot home: http://www.gnuplot.info
```

```
faq, bugs, etc: type "help FAQ"
```

```
immediate help: type "help" (plot window: hit 'h')
```

```
Terminal type is now 'qt'
```

```
gnuplot>
```

Environnement interactif (en ligne de commande)

Quitter par exit, quit ou CTRL+D :

```
gnuplot> exit
```

Aide

Pour obtenir de l'aide sur des commandes, faire

```
gnuplot> help commande
```

ou

```
gnuplot> ?commande
```

Documentation en ligne :

<http://www.gnuplot.info/>

Exemples d'utilisation :

<http://gnuplot.sourceforge.net/demo/>

Par exemple, au démarrage,

```
Terminal type is now 'qt'
```

signifie que la sortie graphique sera dans une fenêtre séparée. Un mode intéressant pour l'interactif est également `wxt` (multi-plateforme) :

```
gnuplot> set terminal wxt enhanced
Terminal type is now 'wxt'
Options are '0 enhanced'
```

Exemple avec terminal

```
gnuplot> help terminal
```

Gnuplot supports a large number of output formats. These are selected by choosing an appropriate terminal type, possibly with additional modifying options. See 'set terminal'.

This document may describe terminal types that are not available to you because they were not configured or installed on your system. To see a list of terminals available on a particular gnuplot installation, type 'set terminal' with no modifiers.

Terminals marked 'legacy' are not built by default in recent gnuplot versions and may not actually work.

Subtopics available for terminal:

```
cairolatex canvas cgm context
domterm dumb dxf eepic
emf emtex epscairo epslatex
fig gif hpgl jpeg
latex lua mf mp
pcl5 pdfcairo png pngcairo
pop postscript pslatex pstex
pstricks push qms qt
sixelgd svg tek40xx tek410x
texdraw tgif tikz tkcanvas
```

Press return for more:

```
tpic vttek wxt x11
xlib xterm
```

Tracé

gnuplot permet le tracé de fonctions - sans données, dont la syntaxe est similaire au Fortran.

```
gnuplot> f(x)=sin(x)
```

```
gnuplot> g(x,y)=exp(-x*x-y*y)
```

gnuplot permet le tracé de données issues de formats de type XY sur plusieurs colonnes.

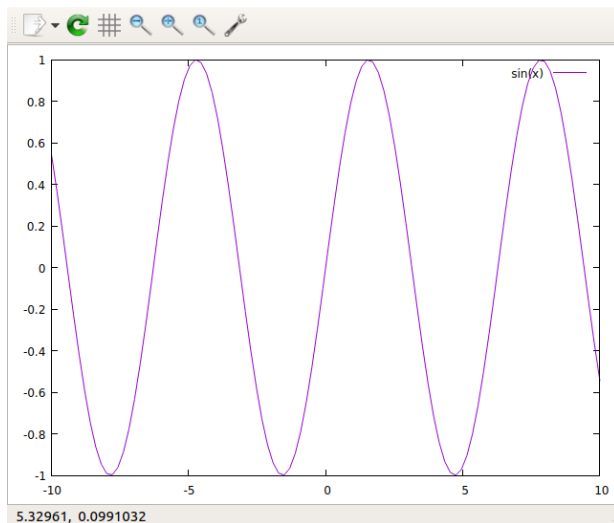
Les tracés 1D utilisent la commande plot

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

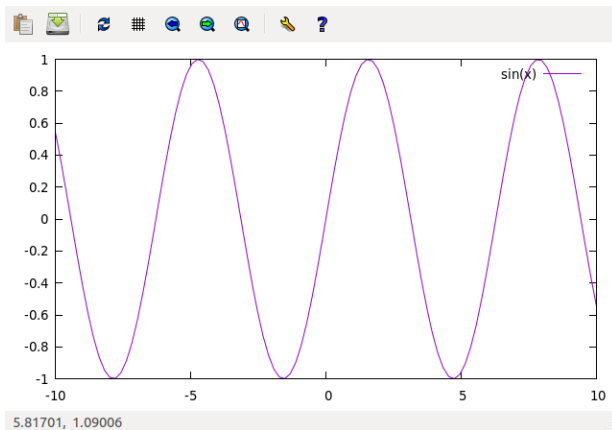
Les tracés 2D et 3D utilisent la commande splot

```
gnuplot> splot g(x,y)
```

Fenêtre qt



Fenêtre wxt



Sorties graphiques

On peut distinguer 3 catégories de sorties :

- les images : FIG, PNG, JPG
- les sorties pour publications : EPS, PDF
- les terminaux interactifs

Sauvegarder une image peut se faire

- via les boutons de dialogue des fenêtres interactives
- en ligne de commande, en changeant le type de terminal

Exemple

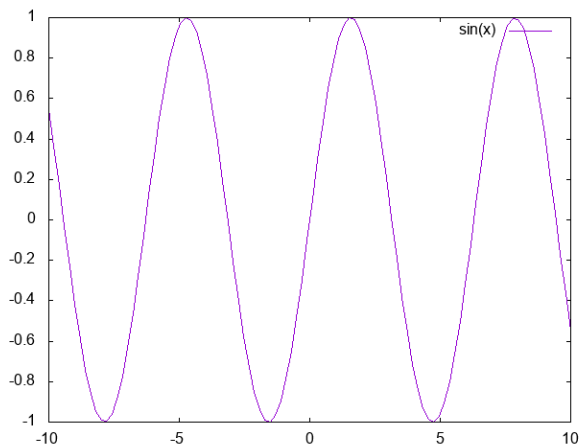
En ligne de commande, pour faire un aller-retour depuis le mode en cours :

```
gnuplot> set terminal push
gnuplot> set terminal png
gnuplot> set output "output.png"
gnuplot> replot
gnuplot> set output
gnuplot> set terminal pop
```

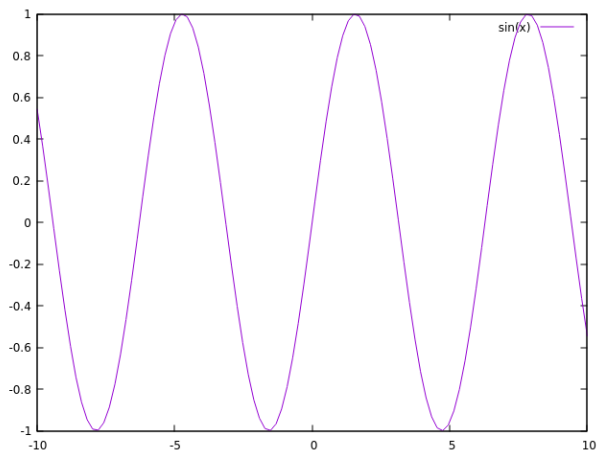
Attention :

- distinguer le canevas du rendu par le tracé
- la police de caractère est associée au terminal sélectionné

Sortie directe de la ligne de commande

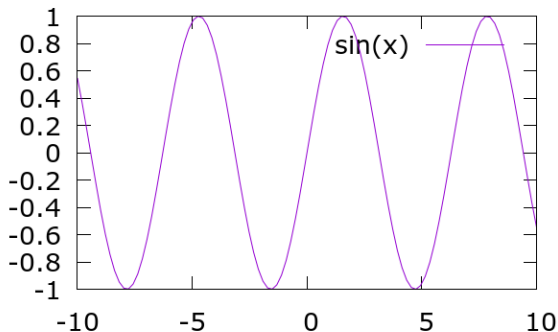


Sortie depuis la sauvegarde via l'interface qt



Modification de la police du terminal wxt

```
gnuplot> set terminal wxt font 'Verdana, 20'
```

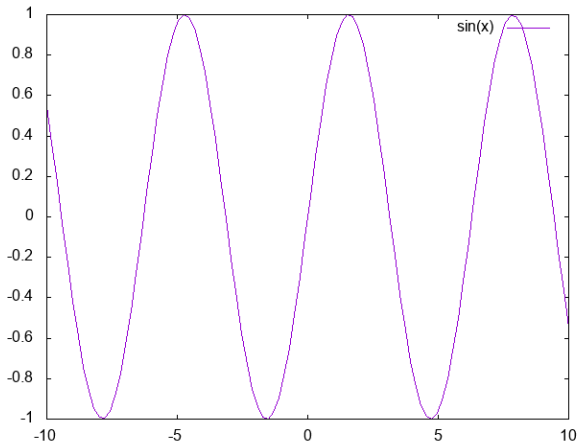


Sortie en png

```
gnuplot> set terminal png
```

Terminal type is now 'png'

Options are 'nocrop enhanced size 640,480 font "arial,12.0" '



Environnement interactif

Dans l'environnement interactif de gnuplot, les commandes peuvent être

- directement saisies
- lues depuis un fichier
- enregistrées vers un fichier (avec toutes les options et variables d'environnement)

Les modifications sur la sorties sont immédiatement effectives avec une commande de type `plot` ou `replot`.

Comme dans un terminal, l'historique des commandes est sauvegardé. Elles peuvent être accédées via les flèches $\uparrow$ et $\downarrow$ ou `CTRL+r`.

Commande history

Dans gnuplot

```
gnuplot> history quiet 5
```

affiche les 5 dernière commandes (sans la numérotation des commandes).
Sauvegarde de l'historique dans un fichier :

```
gnuplot> history "history.gnu"
```

Afficher les commandes de l'historique commençant par load

```
gnuplot> history ?load
```

Refaire la dernière commençant par load

```
gnuplot> history !load
```

save, load et call

- Lecture d'un script

```
gnuplot> load 'myscript.gnu'
```

- Lecture d'un script qui admet des arguments en entrée

```
gnuplot> call 'myscript.gnu' arg1 arg2
```

- Sauvegarde dans un fichier : syntaxe save [options] fichier

```
gnuplot> save 'session.gnu'
```

```
gnuplot> save set 'opts.gnu'
gnuplot> save functions 'functs.gnu'
gnuplot> save var 'vars.gnu'
gnuplot> save term 'term_options.gnu'
gnuplot> set title 'Fonction logarithme' tc rgb 'red'
gnuplot> save '|grep title >t.gnu'
```

Mise à jour de l'état de la session

Remettre toutes les options par défaut :

```
gnuplot> reset
```

Effacer la sortie :

```
gnuplot> clear
```

pause

La commande `pause` permet de mettre en pause l'exécution d'un script

Syntaxe : `pause [num] ['arg']`

Si le chiffre est positif, il spécifie un délai en secondes :

```
gnuplot> pause 5
```

attend 5 secondes.

```
gnuplot> pause -1 "Press Return to continue"
```

```
gnuplot> pause mouse "Cliquer pour continuer"
```

Commandes unix

Il est possible d'accéder à quelques commandes de base du terminal unix depuis l'environnement interactif de gnuplot :

```
gnuplot> !commande
```

Exemple :

```
gnuplot> system "ls -lrt"
```

Exemple :

```
gnuplot> !pwd
/home/acadiou/ASPICS/Cours_Gnuplot/exemples
```

La complétion automatique fonctionne (utile pour retrouver le chemin des données à lire).

Les variables sont déclarées comme en bash

```
gnuplot> a=10
gnuplot> print a
10
```

Les commandes unix peuvent être utilisées pour manipuler les données et extraire les informations nécessaires au tracé (e.g. `awk`, `sed`, etc.)

Les variables lues dans un fichier sont indexées par `$#`

Lancement en batch

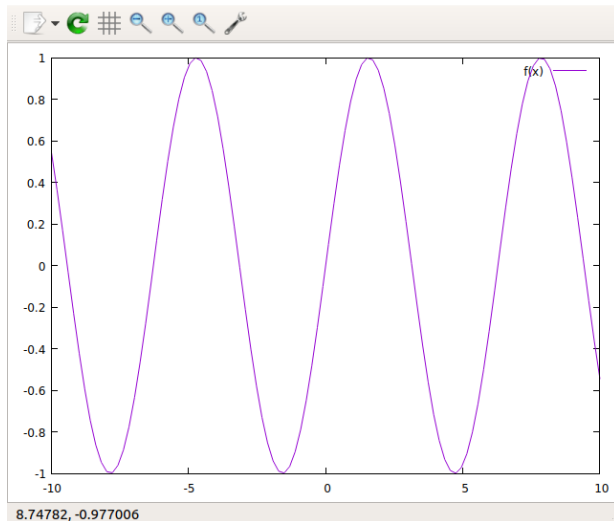
gnuplot peut être utilisé en batch :

```
$ gnuplot myscript.gnu
```

Penser à ne pas diriger la sortie vers du X11 si la machine n'en possède pas.

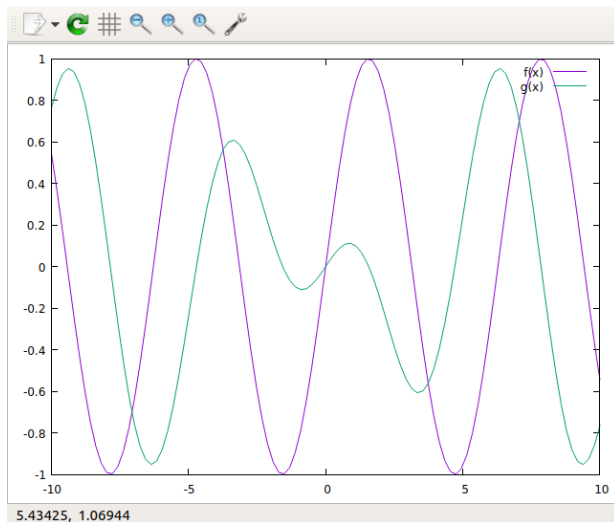
Exemple simple

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



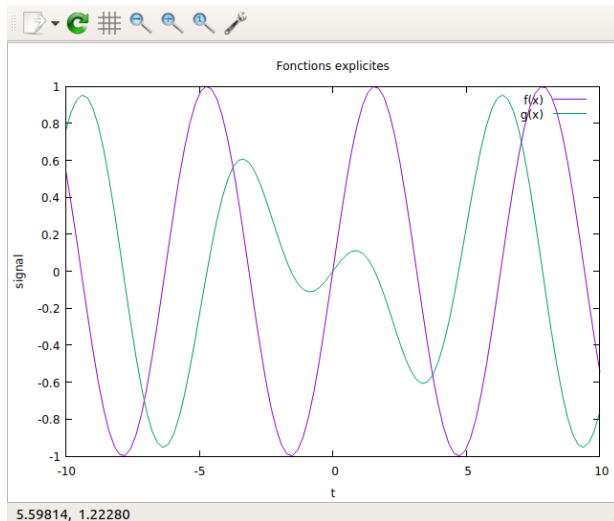

```

gnuplot> freq=5.
gnuplot> g(x)=cos(x)*sin(x/freq)
gnuplot> replot g(x)
    
```

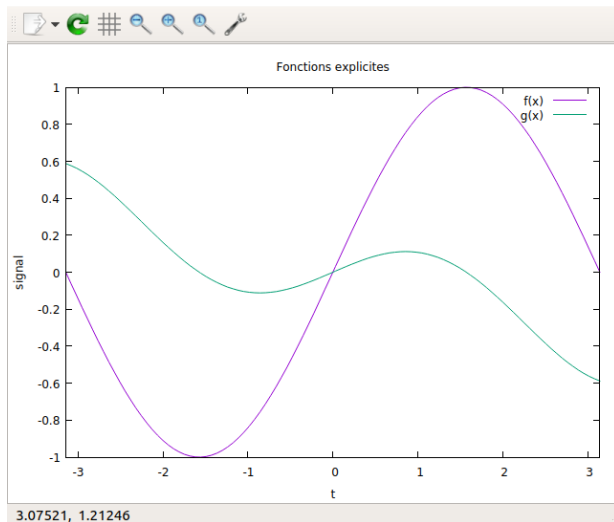


```

gnuplot> set title 'Fonctions explicites'
gnuplot> set xlabel 't'
gnuplot> set ylabel 'signal'
gnuplot> replot
    
```



```
gnuplot> set xrange [-pi:pi]
gnuplot> replot
```



Vers une modification du style des tracés par défaut (explicitée dans un script)

gnuplot simple.gnu

Script simple.gnu

```
f(x)=sin(x)
freq=5.
g(x)=cos(x)*sin(x/freq)

set title 'Fonctions explicites'
set xlabel 't'
set ylabel 'signal'
set xrange [-pi:pi]

plot f(x) w l lw 2 lt 7 dt 2
replot g(x) w lp pi 5 lw 2 pt 6 ps 2 lc -1
```

w l with line

lw linewidth

lt line type

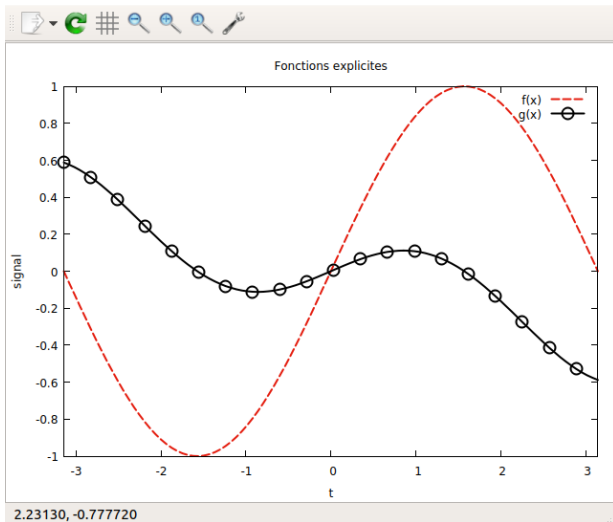
dt dash type

lp line and points

pi points interval

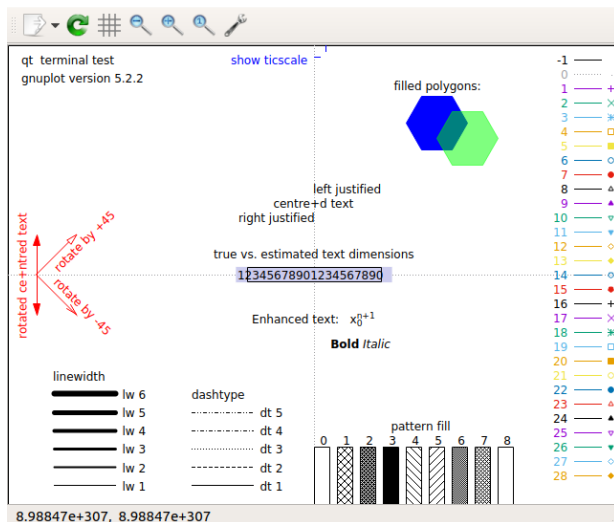
ps point style

lc line color

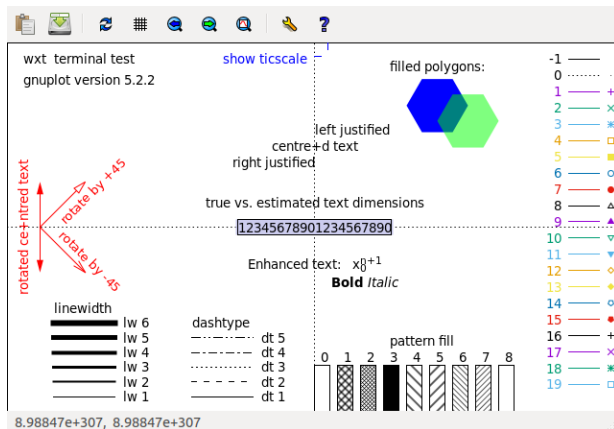


Aperçu des styles pour qt

```
gnuplot> test
```



Aperçu des styles pour wxt



Lecture de données

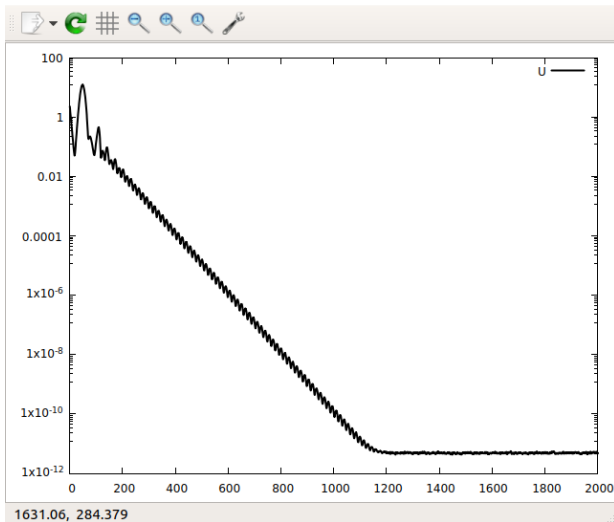
gnuplot permet de lire des données depuis un fichier ASCII ou binaire.
Les données doivent être rangées sous le format plotXY, avec plusieurs colonnes possibles. Par défaut, les lignes commençant par # sont des commentaires.

Exemple : fichier residus.res

```
# it resu resk rese
    1 2.25724651779934682 0.00689900468597994 0.00581914939155728
    2 1.80584087901295143 0.00551895908031626 0.00465617063673530
(...)
   2000 0.000000000000471255 0.000000000000082980 0.00000000001020731
```

Script de lecture et tracé (en log) :

```
set logscale y
plot "residus.res" u 1:2 w l lt -1 lw 2
```

Exemple à partir de données ASCII

```
#include <stdio.h>
#include <math.h>

int main(void)
{
    int i;
    double x,y,z1,z2,d;
    FILE *fptr;

    fptr = fopen("data.dat","w");

    if(fptr == NULL)
    {
        printf("Error!");
        exit(1);
    }

    d = 0.1;
    x = 0.0;
    for(i=0;i<=50;i++){
        y = 1.2*exp(-x);
        z1 = 1.1*(6 - 2*x)/(6 + 4*x + x*x);
        z2 = 0.8*(6 - 4*x + x*x)/(6 + 2*x);
        fprintf(fptr,"% 6.2f % 11.4e % 11.4e % 11.4e\n",
            x,y,z1,z2);
        x += d;
    }

    fclose(fptr);
    return 0;
}
```

Données :

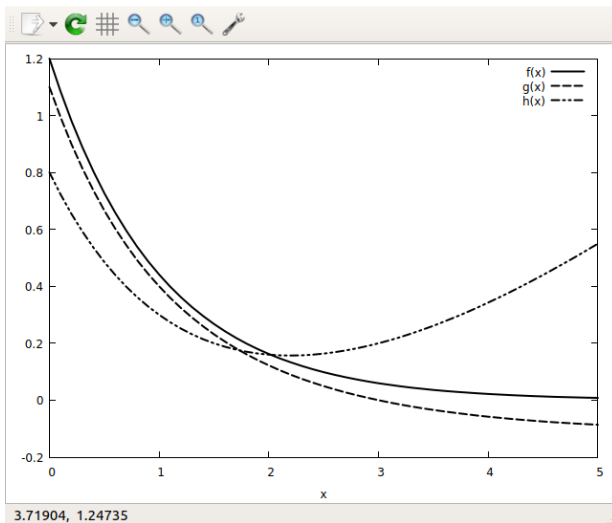
```
-rw-r--r-- 1 2,2K data.dat
```

```
0.00 1.2000e+00 1.1000e+00 8.0000e-01
0.10 1.0858e+00 9.9532e-01 7.2387e-01
0.20 9.8248e-01 9.0058e-01 6.5500e-01
0.30 8.8898e-01 8.1481e-01 5.9273e-01
0.40 8.0438e-01 7.3711e-01 5.3647e-01
0.50 7.2784e-01 6.6667e-01 4.8571e-01
(...)
4.40 1.4733e-02 -7.1695e-02 4.1946e-01
4.50 1.3331e-02 -7.4576e-02 4.4000e-01
4.60 1.2062e-02 -7.7261e-02 4.6105e-01
4.70 1.0914e-02 -7.9761e-02 4.8260e-01
4.80 9.8757e-03 -8.2090e-02 5.0462e-01
4.90 8.9359e-03 -8.4257e-02 5.2709e-01
5.00 8.0855e-03 -8.6275e-02 5.5000e-01
```

Script :

```
set xlabel 'x'

plot 'data.dat' u 1:2 w l lt -1 lw 2 title 'f(x)'
replot 'data.dat' u 1:3 w l lt -1 lw 2 dt 2 title 'g(x)'
replot 'data.dat' u 1:4 w l lt -1 lw 2 dt 5 title 'h(x)'
```



Exemple à partir de données binaires

```

include <stdio.h>
#include <math.h>

struct vect{
    double x,y,z1,z2;
};

int main(void)
{
    int i;
    double d;
    struct vect vec;
    FILE *fptr;

    fptr = fopen("data.bin","w");

    if(fptr == NULL){
        printf("Error!");
        exit(1);
    }

    d = 0.1;
    vec.x = 0.0;
    for(i=0;i<=50;i++){
        vec.y = 1.2*exp(-vec.x);
        vec.z1 = 1.1*(6. - 2.*vec.x)/(6. + 4.*vec.x + vec.x*vec.x);
        vec.z2 = 0.8*(6. - 4.*vec.x + vec.x*vec.x)/(6. + 2.*vec.x);
        fwrite(&vec,sizeof(struct vect),1,fptr);
        vec.x += d;
    }

    fclose(fptr);
    return 0;
}

```

Données :

```
-rw-r--r-- 1 1,6K data.bin
```

Script :

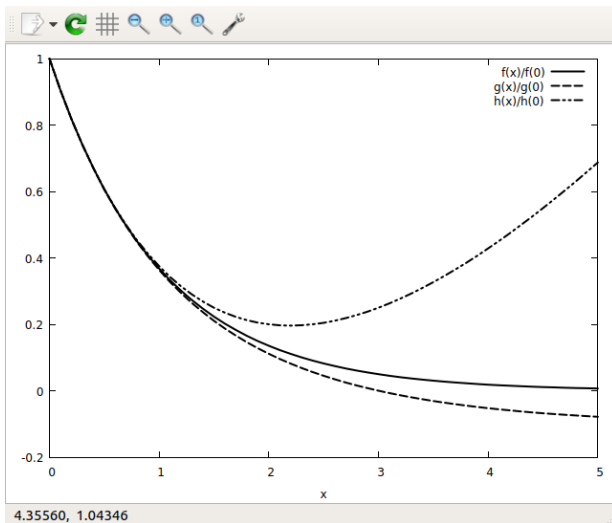
```
set xlabel 'x'
fmt="%double %double %double %double"

plot 'data.bin' binary format=fmt u 1:2 w l lt -1 lw 2 title 'f(x)'
replot 'data.bin' binary format=fmt u 1:3 w l lt -1 lw 2 dt 2 title 'g(x)'
replot 'data.bin' binary format=fmt u 1:4 w l lt -1 lw 2 dt 5 title 'h(x)'
```

Normalisation

```
set xlabel 'x'
f0='awk 'NR == 1 {print $2}' data.dat'
g0='awk 'NR == 1 {print $3}' data.dat'
h0='awk 'NR == 1 {print $4}' data.dat'

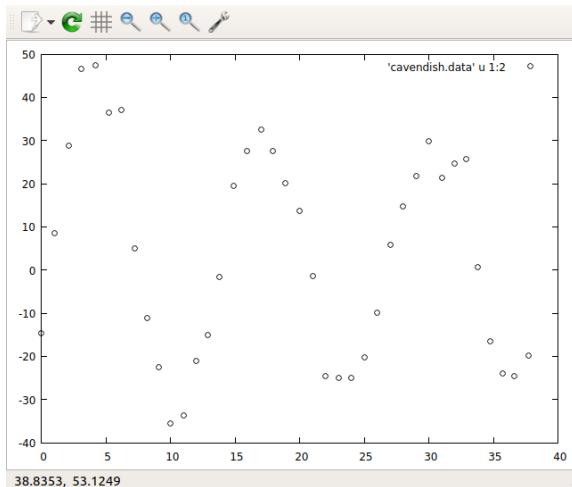
plot 'data.dat' u 1:($2/f0) w l lt -1 lw 2 title 'f(x)/f(0)'
replot 'data.dat' u 1:($3/g0) w l lt -1 lw 2 dt 2 title 'g(x)/g(0)'
replot 'data.dat' u 1:($4/h0) w l lt -1 lw 2 dt 5 title 'h(x)/h(0)'
```



Fitting

(<https://www.cs.hmc.edu/~vrable/gnuplot/cavendish.data>)

```
plot 'cavendish.dat' u 1:2 w p pt 6 lc -1
```



Fonction :

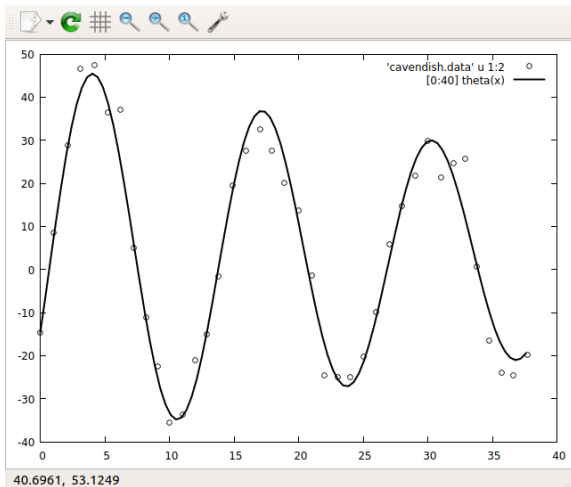
$$\theta(t) = \theta_0 + a \exp(-t/\tau) \sin(2\pi t/T + \phi)$$

Script :

```
theta(t) = theta0 + a * exp(-t / tau) * sin(2 * pi * t / T + phi)
fit theta(x) "cavendish.data" using 1:2 via a, tau, phi, T, theta0
replot [0:40] theta(x) w l lw 2 lt -1
```

Résultat de gnuplot :

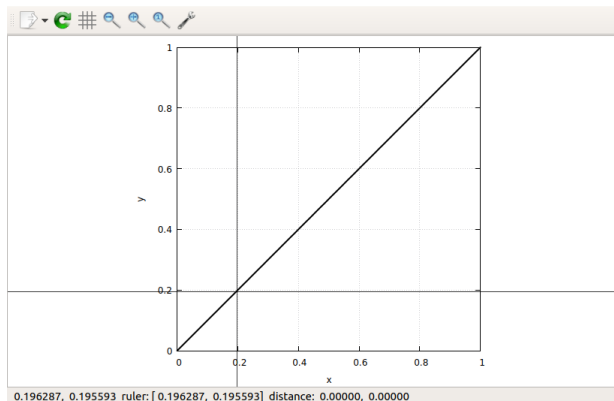
```
Final set of parameters Asymptotic Standard Error
=====
a = 45.5239 +/- 2.767 (6.078%)
tau = 57.8326 +/- 10.71 (18.51%)
phi = -0.401155 +/- 0.05306 (13.23%)
T = 13.109 +/- 0.08364 (0.638%)
theta0 = 3.02268 +/- 0.8144 (26.94%)
```



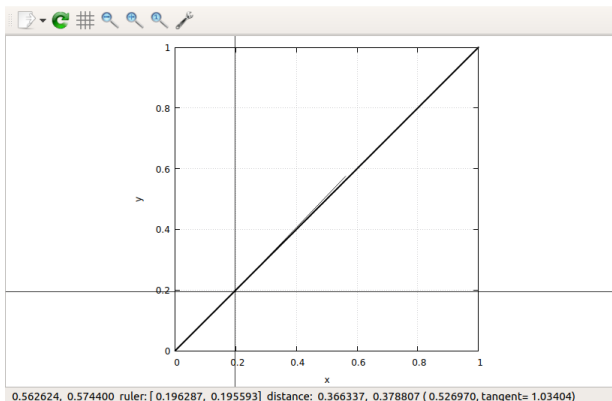
Estimation de pente

En coordonnées cartésiennes, en interactif, taper 'r' pour faire apparaître un point de référence (mode ruler on de la souris)

```
set xrange [0:1]
set yrange [0:1]
plot x w l lt -1 lw 2
```



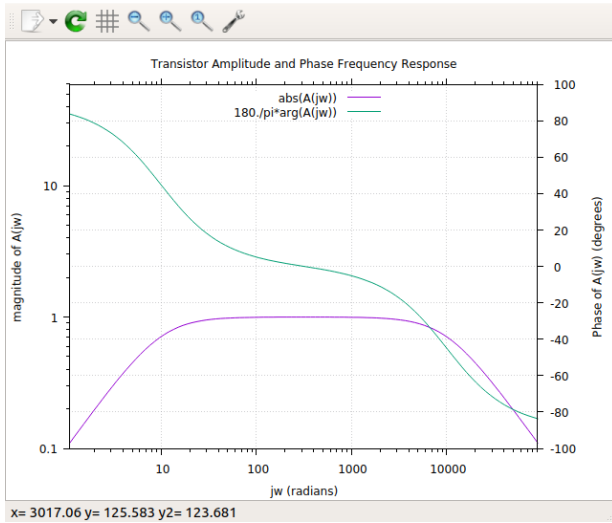
En pressant la touche '5' s'affiche la distance entre la référence et la position de la souris, en coordonnées polaire (angle) ou tangente (Presser '5' une deuxième fois).



Échelle double

http://gnuplot.sourceforge.net/demo_cvs/multiaxis.html

```
A(jw) = ({0,1}*jw/({0,1}*jw+p1)) * (1/(1+{0,1}*jw/p2))
p1 = 10
p2 = 10000
set dummy jw
set grid x y2
set key center top title " "
set logscale xy
set log x2
unset log y2
set title "Transistor Amplitude and Phase Frequency Response"
set xlabel "jw (radians)"
set xrange [1.1 : 90000.0]
set x2range [1.1 : 90000.0]
set ylabel "magnitude of A(jw)"
set y2label "Phase of A(jw) (degrees)"
set ytics nomirror
set y2tics
set tics out
set autoscale y
set autoscale y2
plot abs(A(jw)) axes x1y1, 180./pi*arg(A(jw)) axes x2y2
```



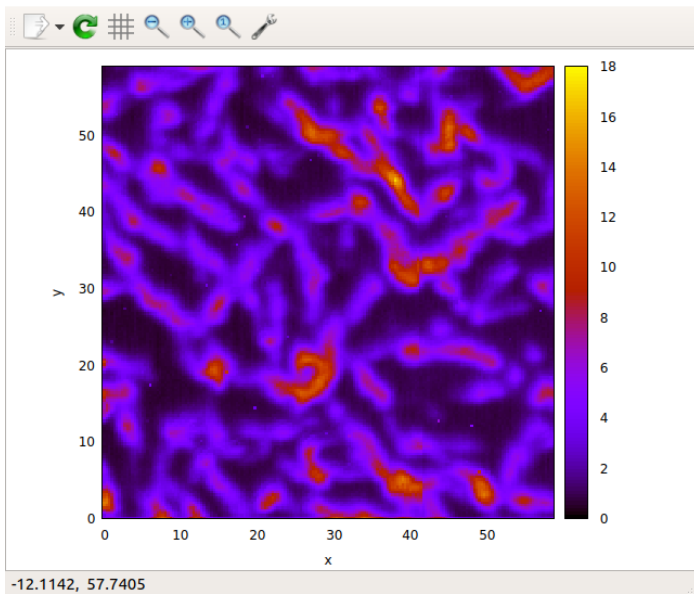
2D

(http://www.gnuplotting.org/data/matlab_colormap.txt)

```
#!/usr/bin/gnuplot
reset
set size ratio -1
unset key

set xrange [0:59]
set yrange [0:59]
set xlabel 'x'
set ylabel 'y'
set cbtics scale 0

plot 'matlab_colormap.txt' u ($1/3.0):($2/3.0):($3/1000.0) matrix with image
```

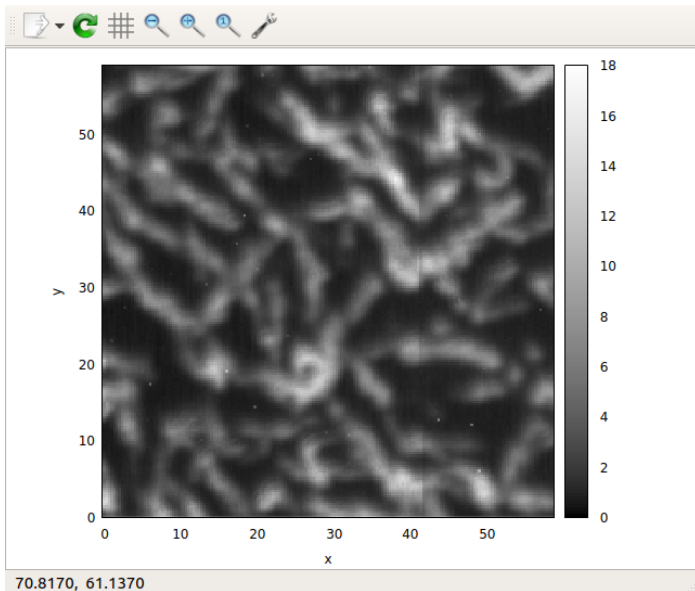



Palette

```
#!/usr/bin/gnuplot
reset
set size ratio -1
unset key

set xrange [0:59]
set yrange [0:59]
set xlabel 'x'
set ylabel 'y'
set cbtics scale 0
set palette gray

plot 'matlab_colormap.txt' u ($1/3.0):($2/3.0):($3/1000.0) matrix with image
```



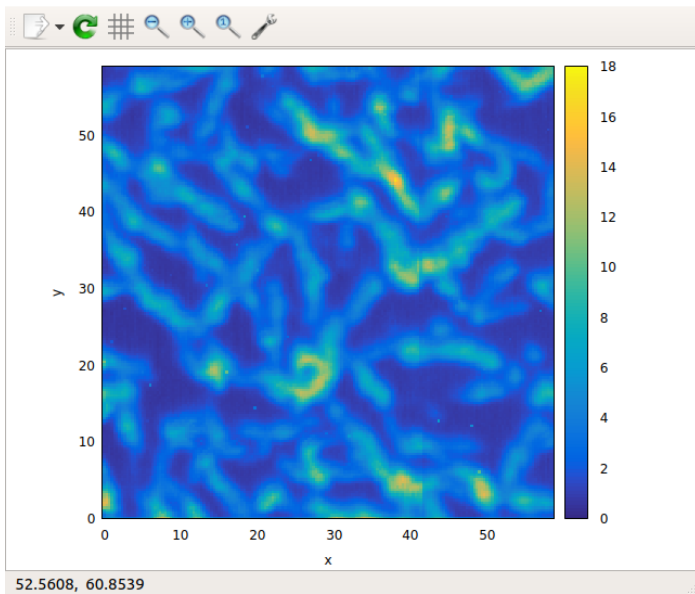
Palette comme matlab

(<https://github.com/Gnuplotting/gnuplot-palettes>)

```
#!/usr/bin/gnuplot
reset
set size ratio -1
unset key

set xrange [0:59]
set yrange [0:59]
set xlabel 'x'
set ylabel 'y'
set cbtics scale 0
load 'parula.pal'

plot 'matlab_colormap.txt' u ($1/3.0):($2/3.0):($3/1000.0) matrix with image
```



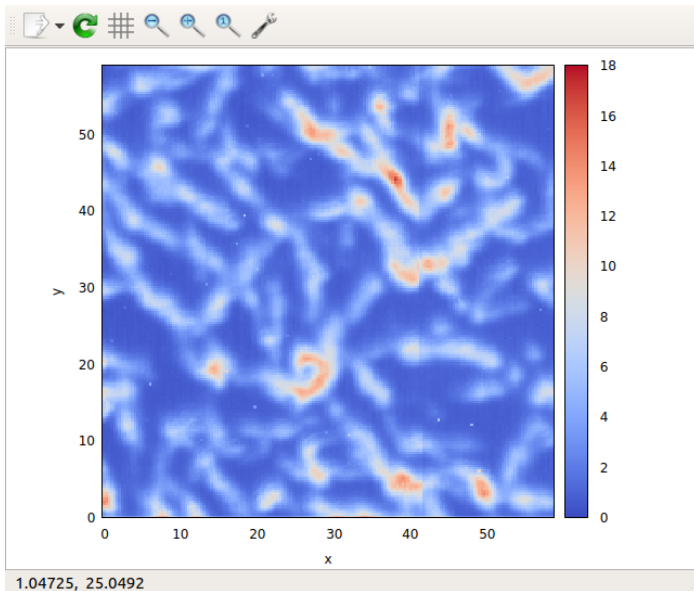
Palette comme paraview

(<https://github.com/Gnuplotting/gnuplot-palettes>)

```
#!/usr/bin/gnuplot
reset
set size ratio -1
unset key

set xrange [0:59]
set yrange [0:59]
set xlabel 'x'
set ylabel 'y'
set cbtics scale 0
load 'moreland.pal'

plot 'matlab_colormap.txt' u ($1/3.0):($2/3.0):($3/1000.0) matrix with image
```



3D

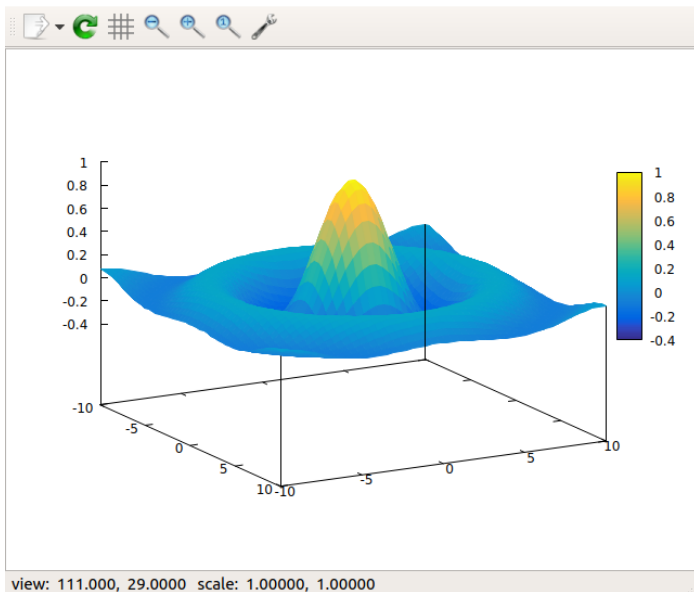
```
#!/usr/bin/gnuplot
reset
unset surface
set size ratio -1
unset key

set cbtics scale 0

set view 111, 29, 1, 1
set isosamples 50, 50

set pm3d implicit at s
load 'parula.pal'

splot sin(sqrt(x**2+y**2))/sqrt(x**2+y**2)
```

Sortie pour L^AT_EX

Il est possible d'écrire des EPS et PDF, à inclure dans le L^AT_EX via `includegraphics`.

```
gnuplot> set terminal postscript eps enhanced
gnuplot> set output 'output.eps'
gnuplot> replot
```

```
gnuplot> set terminal pdfcairo enhanced
gnuplot> set output 'output.pdf'
gnuplot> replot
```

De gnuplot vers L^AT_EX

Le terminal epslatex permet de sortir directement un fichier L^AT_EX, à intégrer dans L^AT_EX par `\input{output.tex}`

Script gnuplot :

```
set terminal latex
set output 'figLatex.tex'
set xlabel '$\theta$'
set ylabel '$F(\theta)$'
set label '$\left\{\begin{array}{l}\mu=0\\\sigma^2=1\end{array}\right\}.$' \
at -1.5,0.8
set label '$P(X\leq\theta)=\frac{1}{\sqrt{2\pi}} \cdot \int_{-\infty}^{\theta} e^{-t^2/2}dt$' \
at 0.2,0.3
set title 'Loi normale $\mathcal{N}(0,1)$'
set xrange [-2:2]
set zeroaxis
plot norm(x) notitle
```

Fichier L^AT_EX :

```

\documentclass{report}
\usepackage[utf8]{inputenc}
\begin{document}
\begin{figure}
\input{figLatex.tex}
\caption{
  Exemple de figure g'en'er'ee sous \texttt{gnuplot}
}
\end{figure}
\end{document}

```

Résultat

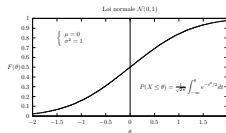


Figure 1: Exemple de figure générée sous `gnuplot`

Références

- <http://www.gnuplot.info/files/tutorial.pdf>
- http://gnuplot.sourceforge.net/demo_cvs/
- <http://lowrank.net/gnuplot/index-e.html>
- <http://www.bersch.net/gnuplot-doc/gnuplot.html>
- <https://www.cs.hmc.edu/~vrable/gnuplot/using-gnuplot.html>