

---

# Utilisation de Gnuplot pour tracer des courbes

Gnuplot est un outil permettant de dessiner facilement des courbes sous de nombreux systèmes. Ce logiciel sous licence GNU est libre d'utilisation.

Cette courte documentation est une introduction aux manipulations simples pour tracer des courbes avec Gnuplot. Pour de plus amples détails sur ces nombreuses capacités voir le site officiel <http://www.gnuplot.info> ou ce site en français très complet <http://logicielslibres.dane.ac-versailles.fr/spip.php?article70>

## Utilisation de Gnuplot

On peut considérer quatre façons d'utiliser Gnuplot

- En ligne de commande Linux, en lançant le logiciel `gnuplot`, on obtient une invite de commande `gnuplot >` qui attend vos commandes. Avec la commande `quit` pour ressortir.
- Avoir créé auparavant un fichier de commande Gnuplot que l'on peut appeler un **script**, par exemple sous le nom `MesCommandes.gnu`. Dans l'invitation de commande Gnuplot, on tape alors `load "MesCommandes.gnu"`.
- L'idée précédente peut aussi de lancer en ligne de commande en lançant `gnuplot "MesCommande.gnu -persist"` en ligne de commande Linux : toutes les commandes Gnuplot tapées dans le fichier script sont exécutées les unes après les autres. Le paramètre `persist` permet de conserver la fenêtre gnuplot ouverte.
- En langage C, on peut utiliser l'idée précédente directement dans un programme en lançant le script depuis le programme C. En effet, il existe la commande C `system` (dans `stdlib.h`) qui permet d'exécuter une commande Linux pendant votre programme C.  
C'est-à-dire ici `system("gnuplot \"MesCommande.gnu\" -persist");`

### *Sauvegarder le graphique*

Si vous avez une version de Gnuplot avec fenêtre graphique : Gnuplot affiche à l'écran dans une nouvelle fenêtre un dessin correspondant à vos commandes. Il est possible de sauvegarder alors le dessin sous un format par exemple **pdf** ou **svg** afin de l'insérer dans un document. Notez que si vous avez lancé Gnuplot en ligne de commande, vous pouvez compléter vos commandes les unes après les autres et voir évoluer votre dessin.

Si vous avez une version de Gnuplot en mode texte : Gnuplot affiche des courbes en mode texte... Mais vous pouvez définir au lancement :

```
set terminal png
set output "image.png"
```

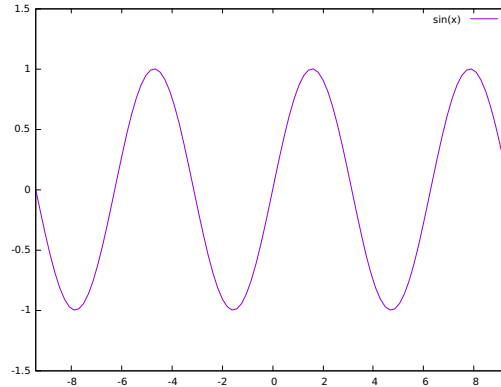
Les graphes sont alors écrit dans un fichier `image.png` sur disque au format `png`. Attention, chaque nouveau graphe produit écrasera l'ancien fichier (changez son nom dans `set output`).

## Tracer des courbes représentatives de fonctions connues analytiquement

Si l'on connaît l'expression analytique d'une fonction, par exemple  $f(x) = \sin(x)$  ou  $g(x) = \sqrt{3x^2 - 4}$ , on peut demander directement à gnuplot de décider comment représenter au mieux la courbe représentative de la fonction par la commande `plot sin(x)` ou `plot sqrt(3*x**2-4)`.

On peut vouloir préciser un intervalle de valeurs en  $x$  et  $y$ , par exemple :

```
plot [-3*pi:3*pi] [-1.5:1.5] sin(x)
```



## Tracer des courbes ou des histogrammes à partir de points

L'autre utilisation de la fonction `plot` est de tracer un nuage de points fournis par un fichier texte.

```
#an pingouins gnous
1970 1230 256
1975 987 320
1980 518 350
1985 780 302
1990 1024 150
1995 1250 281
2000 1580 520
```

Par exemple, si l'on désire représenter l'évolution de la population mondiale de gnous et de pingouins de 1970 à 2000, on crée le fichier `population.txt` ci-contre.

Les lignes avec un `#` sont des commentaires.

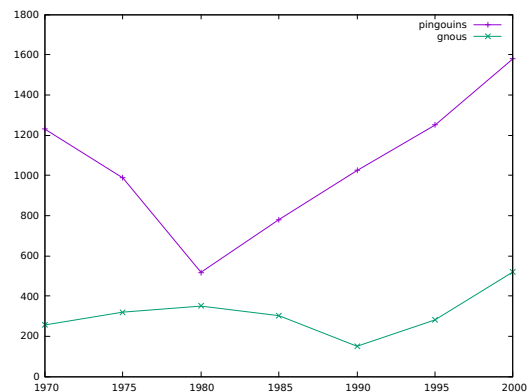
Le script suivant permet de tracer la courbe d'évolution de la population des pingouins et des gnous sur le même graphique

```
plot [*:~] [0:1800] "population.txt" using 1:2 title "pingouins" with linespoints 4
replot "population.txt" using 1:3 title "gnous" with linespoints 3
```

Cela donne ceci :

La première commande permet de dessiner une courbe en considérant les données des colonnes 1 et 2 par `using (1:2)` du fichier `population.txt`. Le titre de cette courbe sera "pingouins". cette courbe sera du type `linespoints` : c'est-à-dire des points reliés par des segments de droite et sera de couleur 4.

La commande `replot` permet de dessiner la deuxième courbe sur le même graphique.



Les courbes peuvent être de différents types, par exemple : **points** : points non reliés, **lines** : lignes sans représentation des points, **linespoints** : ligne et points, **impulses** : bâtons, **steps** : escalier, **boxes** : histogramme.

Les points par défaut de gnuplot sont des croix : pour obtenir une autre forme (correspondant à un nombre entier où 1 est la croix, 6 le cercle, 7 le disque,...) et une autre taille de points, on utilise les paramètres **pt** et **ps** : par exemple `with points pt 7 ps 1.2` indique que l'on veut des points en forme de disque de taille 1.2.

Gnuplot inclut également des routines d'interpolation et d'approximation de données. On utilise pour cela en fin de ligne `plot` (à la place de `using lines`), le mot-clef **smooth** suivi du type de routines : **acsplines** fonctions splines, **bezier** courbes de Bézier ou **csplines** splines cubiques.

### La commande set

La commande **set** permet d'initialiser des paramètres agissant sur l'allure du dessin. Elle contient un grand nombre de paramètres, utilisez la commande `help set` pour les connaître.

Par exemple

```
set xrange [-3*pi:+4*pi]
set yrange [-5:+10]
```

permet de fixer les intervalles d'affichages des données et les axes correspondants.

De même pour `set xlabel "Var T"`

permet de donner un nom à l'axe des abscisses.

Des réglages importants :

`set size square` : avoir une fenêtre carrée pour ne pas déformer l'affichage

`set xzeroaxis` et `set yzeroaxis` : avoir les axes des abscisses et des ordonnées sur le dessin

`set pointsize 0.1` : réduire l'épaisseur du trait d'une courbe à 0.1

`set key off` : supprime la légende

### Tracer des courbes données par une succession de points (lignes de niveaux)

On peut vouloir tracer une courbe représentative d'une fonction dont on n'a pas la connaissance analytique. On peut également vouloir dessiner une (ou plusieurs) ligne(s) de niveau d'une fonction de  $\mathbb{R}^2$  dans  $\mathbb{R}$ . Or dans ces deux cas, on ne peut pas utiliser l'expression analytique.

L'idée est de demander à un programme C de produire suffisamment de points de la courbe ou de la ligne de façon à donner l'illusion d'une continuité. Ces points sont stockés en deux colonnes (séparées par un espace) dans un fichier de données, par exemple "LigneNiveau.txt" On utilise alors tout simplement la commande `plot "LigneNiveau.txt" with points` uniquement avec des points.

### Surface représentative de fonctions à deux dimensions

Il est également possible de tracer des surfaces représentatives de fonctions de  $\mathbb{R}^2$  dans  $\mathbb{R}$ .

Voici un code exemple pour la fonction  $f(x, y) = \frac{\sin(\sqrt{x^2+y^2})}{\sqrt{x^2+y^2}}$

```
set hidden3d
set isosamples 30,30
splot sin(sqrt(x**2+y**2))/sqrt(x**2+y**2)
```

La commande `splot` dessine des surfaces avec des paramètres similaires à `plot`.

Les paramètres fixés ici avec `set` permettent d'afficher plus de lignes pour représenter la surface.

On peut vouloir aussi mettre de la couleur pour représenter la variations de  $f$  avec des couleurs froides (bleu) pour les petites valeurs et chaudes (rouge) pour les plus hautes.

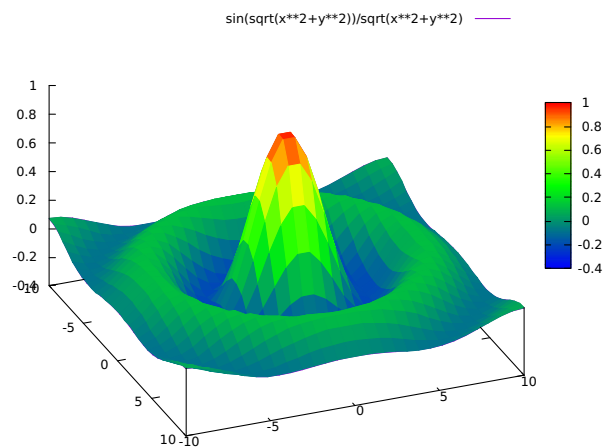
On utilise les lignes suivantes avant la commande `splot` :

```
set ticslevel 0
```

```
set pm3d
```

```
set palette defined (0 "blue", 0.5 "green", 0.8 "yellow", 1 "red")
```

On obtient alors :



Si vous avez Gnuplot installé avec le patch `gpmouse`, vous pouvez cliquer sur l'image pour voir la figure sous différents angles.