

GLSE301 - TP séance 8 : GNUPlot

Objectifs

[Gnuplot](#) est un logiciel gratuit et portable qui permet de représenter des données et fonctions dans le plan et dans l'espace. Il est également capable de calculer des ajustements de fonctions à des données. Le programme peut être utilisé de manière interactive en ligne de commande mais peut également interpréter directement des scripts contenant une séquence de commandes.

Gnuplot peut produire ses résultats directement à l'écran ou dans des fichiers de divers formats : PNG, EPS, SVG, ... et peut même produire du code LaTeX directement. Il est notamment utilisé par [Octave](#), logiciel de calcul numérique libre similaire à Matlab ou Scilab. Il peut également être appelé depuis d'autres langages de programmation comme Perl, Python, Ruby, Smalltalk, ...

Consignes

Lire attentivement cet énoncé de TP en suivant les instructions. En cas de question, faire appel à l'enseignant qui est là pour vous répondre, et surtout ne pas rester pas bloqué sur une des tâches à réaliser. Essayer en tous cas de chercher sur internet la réponse à vos questions.

Introduction à GNUPlot

Gnuplot est un programme qui permet de dessiner des fonctions, des courbes, des histogrammes, des données ... très facilement. Vous pouvez utiliser le programme dans deux modes différents : le **mode interactif** vous permet d'encoder des commandes les unes à la suite des autres et le **mode script** vous permet d'écrire un script qui va pouvoir être exécuté d'un coup.

On va tout de suite s'intéresser à trois exemples simples en mode interactif qui vous montreront toute la puissance de Gnuplot. Pour travailler en mode interactif il faut simplement taper la commande `gnuplot` du terminal.

Une courbe dans le plan

Commençons avec une simple fonction dans le plan. On va dessiner la fonction sinus, et pour ça, il suffit d'utiliser la commande suivante :

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

La commande va dessiner la fonction sinus et va définir les limites des axes. L'axe des x s'étend de -3 à 3 et celui des y de -1,5 à 1,5. La courbe obtenue est la suivante :

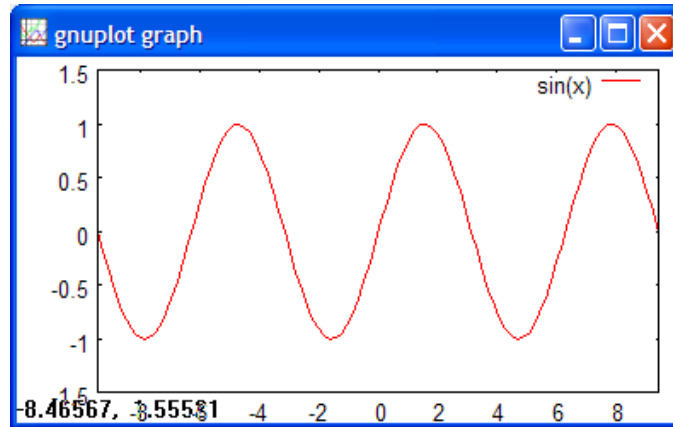


Illustration 1: Graphe de la fonction $\sin(x)$

Les deux chiffres en bas à gauche représentent la coordonnée actuelle visée par la souris. Elles se mettent à jour toutes seules lorsque vous déplacez votre souris sur le graphe.

Exercice 1

Tracer les courbes d'équations suivantes :

- ▲ $f(x) = 2x + 4$
- ▲ $g(x) = x^2 + 2x + 1$
- ▲ $h(x) = 2x \sin(\pi x/2)$

Il est possible de tracer plusieurs fonctions à la fois. Il suffit d'appeler la commande `plot` et donner la liste des fonctions. Chaque nom de fonction est séparé par une virgule.

Paramètres de tracé des courbes

Il est également possible de modifier certains paramètres de tracés (utiliser des points au lieu de ligne, changer la couleur, la légende de la courbe, ...).

Par exemple, la commande suivante :

```
plot f(x) with points title "courbe 1";
```

permet de tracer la courbe f avec des points, en donnant comme légende à la courbe, le nom "courbe 1". L'option `linewidth 2.0` permet de doubler l'épaisseur de la ligne. `pointsize 2.0` permet de doubler la taille des points.

Exercice 2

Tracer les courbes précédentes en modifiant le type de tracé. Utiliser les types suivants et comparer les résultats : `lines`, `points`, `linespoints`, `impulses`, `dots`, `steps`, `fsteps`, `histeps`, ...

Essayer également de tracer les courbes en modifiant la légende de la courbe et l'épaisseur de tracé de la courbe.

Paramètres généraux de tracé

Les réglages précédents ne s'appliquent qu'à la courbe courante. Il est possible d'appliquer les options de tracé à toutes les courbes tracées par la suite. Ces réglages généraux s'effectuent via la commande `set`. Sa syntaxe est la suivante : `set <option> <valeurs>`.

Par exemple, la commande :

```
set xrange [-15:15];
```

permet de régler l'échelle des x de -15 à 15. Vous pouvez remplacer une borne de cet intervalle par *, ce qui laisse le logiciel gnuplot choisir la valeur.

Pour réinitialiser la valeur de tous ces réglages (ie. revenir aux réglages par défaut du logiciel), vous pouvez utiliser la commande `reset` ;

Exercice 3

Tracer la courbe g (de l'exercice 1) en réglant l'axe des x dans l'intervalle $[-20; 20]$ et l'axe des y de 0 à une valeur fixée par gnuplot.

Exercice 4

Grâce à la commande `help set`, rechercher les options de `set` permettant de modifier le titre du graphique et le label des axes x et y. Utiliser ces options pour donner un titre au graphique et des labels aux axes.

Une courbe dans l'espace

Voyons maintenant un exemple de fonction dans l'espace, en trois dimensions. Dessinons le graphe de la fonction $z = (x^2 + y^2 + 1)^{-1}$. Pour ce faire, on va devoir utiliser quatre commandes :

```
gnuplot> set samples 20
gnuplot> set isosamples 20
gnuplot> set hidden3d
gnuplot> splot [-3:3] [-2:2] 1 / (x*x + y*y + 1)
```

Les trois premières commandes permettent de paramétrer le mode de dessin et la quatrième commande fait le dessin de la fonction. Pour avoir plus d'information vous avez l'aide en ligne:

```
gnuplot> help samples
```

L'axe des x s'étend de -3 à 3 et celui des y de -2 à 2. Les limites de l'axe des z sont automatiquement choisies.

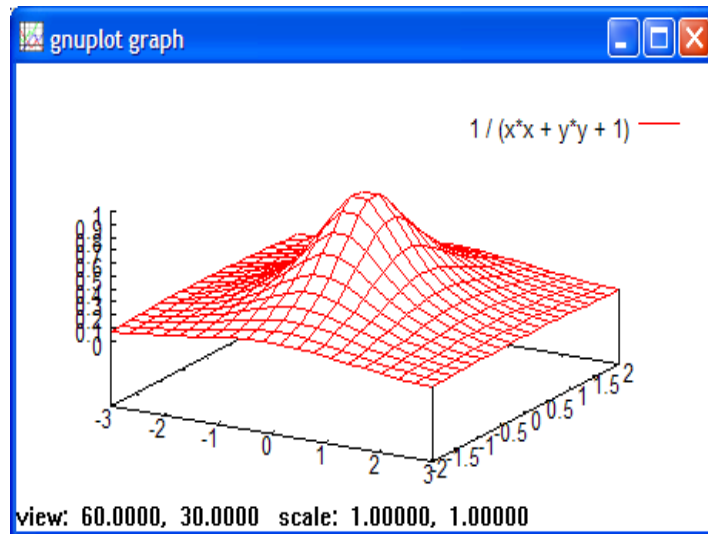


Illustration 2: Graphe de la fonction $z = (x^2 + y^2 + 1)^{-1}$

On peut donc déjà voir que pour dessiner en deux dimensions, on va utiliser la commande **plot**. Par contre, on utilise la commande **splot** pour dessiner en trois dimensions. De plus, on va pouvoir paramétrer et personnaliser les graphes obtenus en utilisant la commande **set**.

Exercice 5

Tracer les courbes d'équations suivantes :

- ⤴ $f(x, y) = 2x + 3y + 2$
- ⤴ $g(x, y) = x^2 + y^2 - 2x + y + 1$
- ⤴ $h(x, y) = \sin(x) + 2\cos(x + y)$

Exercice 6

Utiliser la commande **help** pour trouver ces différents styles. Essayer en quelques-uns. Faire de même pour renommer les courbes.

Quelques options de tracé

Il existe également des options globales pour tracer les courbes 3D, activables et réglables via la commande `set` (idem aux courbes 2D). Parmi ces options, il en existe une pour afficher ou non les parties masquées d'une vue 3d. Cette option est l'option **hidden3D** activable via la commande :

```
set hidden3d;
```

et se désactive avec la commande :

```
unset hidden3d;
```

Exercice 7

Comparer le tracé de la courbe h en activant et en désactivant l'affichage des parties masquées. Faire de même pour le tracé des trois courbes en même temps.

Exercice 8

Il est possible de modifier la finesse avec laquelle les courbes sont tracées. Pour cela, il faut modifier la fréquence d'échantillonnage des courbes. La commande **set isosamples** permet de régler la fréquence d'échantillonnage. Elle est suivie du nombre d'échantillons à prendre sur l'axe de x et le nombre d'échantillons sur l'axe des y . Exemple :

```
set isosamples 10,10;
```

Tracer les courbes associées à f , g et h pour différentes valeurs d'échantillonnage :

- ⤴ 10,10
- ⤴ 20,10
- ⤴ 10,20
- ⤴ 20,20

Exercice 9

Activer l'option `set contour`. Tracer la courbe h .

Que fait l'option `set contour` ?

Idem en activant l'option `set grid`.

Couleur

Il est également possible de colorer les surfaces tracées. Pour cela, il faut utiliser la commande :

```
set pm3d;
```

Exercice 10

Tracer la courbe h en couleur.

Exercice 11

Les couleurs peuvent être modifiées via la commande `set palette defined` (<liste des couleurs>). Exemple avec un dégradé de couleurs prédéfinies par gnuplot :

```
set palette defined (0 "red", 1 "blue", 2 "green", 3 "dark-yellow");
```

Essayer de tracer la courbe `h` avec différents dégradés de couleurs.

Dessin de données

Pour terminer cette introduction, voyons un dernier exemple dans lequel on va dessiner des données brutes. Supposons que l'on veuille tracer un graphe reprenant les données suivantes : (1, 5), (2, 12), (3, 17), (4, 13) et (5, 2). Il va falloir utiliser la commande suivante :

```
gnuplot> plot [0:6] [0:20] "-" with linespoint
```

Le terminal passe ensuite dans un mode spécial dans lequel vous devez saisir les données. Une fois toutes les données saisies, entrez n'importe quel mot commençant par la lettre `e` :

```
input data ('e' ends) > 1 5
input data ('e' ends) > 2 12
input data ('e' ends) > 3 17
input data ('e' ends) > 4 13
input data ('e' ends) > 5 2
input data ('e' ends) > e
```

Gnuplot va maintenant dessiner ces points et les relier par des lignes puisqu'on a utilisé l'option `with linespoint`.

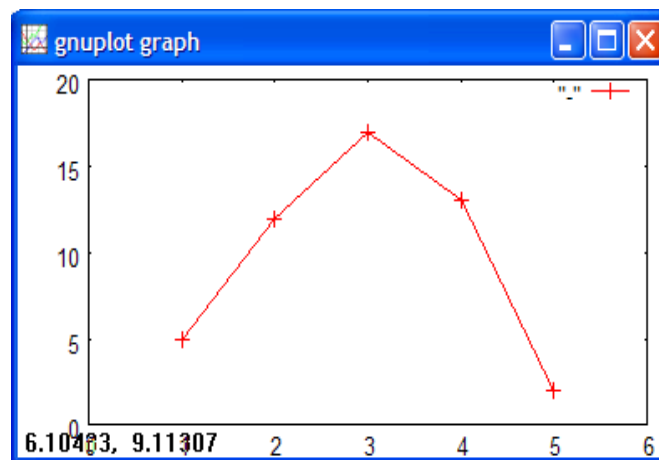


Illustration 3: Graphe de données brutes

On verra bien entendu comment charger des données brutes directement depuis un fichier plutôt que de chaque fois devoir les saisir manuellement.

Commençons simplement par voir comment dessiner simplement des données. Les données sont simplement stockées dans un fichier texte qui contient une donnée par ligne. On peut ajouter des commentaires dans ce fichier grâce au caractère # . Le listing suivant vous montre un exemple de fichier de données.

```
# Données
12
17
22
38
21
19
49
89
44
57
```

Rien de plus simple que de dessiner ces données dans un graphique. Comme on dessine en 2D, on va donc utiliser la commande plot et il suffit de lui donner en paramètre le nom du fichier contenant les données entre guillemets simples comme le montre le listing suivant:

```
# Paramètres
set key outside bmargin left box
# Dessin de la courbe
plot 'samples.dat'
```

Un point va être ajouté pour chacune des données présentes dans le fichier. Les coordonnées en x seront simplement 0, 1, 2, ... et les coordonnées en y seront les données présentes dans le fichier. La figure suivante vous montre le résultat pour les données de l'exemple précédent.

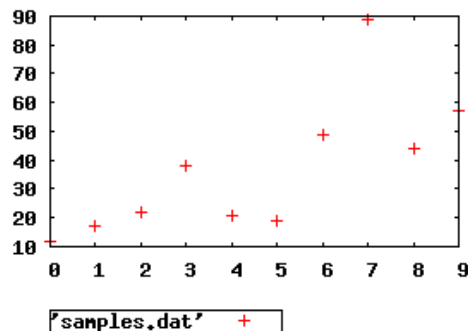


Illustration 4: simple dessin des données

Reprendre l'exemple et vérifier le résultat avec l'option **with lines** pour la commande plot.

Histogramme

Pour tracer des histogrammes, il faut dans un premier temps activer le mode de tracer des histogrammes via la commande :

```
set style data histograms
```

puis régler le type d'histogramme que l'on souhaite tracer via la commande

```
set style histogram <type d' histogramme>.
```

Exercice 12

Dessiner le fichier donné (de l'exemple précédent) en utilisant l'option histogramme

Exercice 13

Telecharger le fichier [data.dat](#) puis essayer les commandes suivantes

```
set style data histograms;  
set style histogram clustered;  
plot './data.dat' using 1, '' using 2;
```

Plusieurs courbes

On peut bien entendu dessiner plusieurs courbes sur la même figure, il suffit de les séparer par une virgule. L'exemple suivant dessine deux courbes de données ainsi que le graphe de la fonction x^2 .

```
# Paramètres  
set key outside bmargin left box  
set yrange [0:100]  
  
# Dessin de la courbe  
plot 'samples.dat' with lines title 'Échantillon 1', 'other_samples.dat'  
title 'Échantillon 2' with linespoint, x**2
```

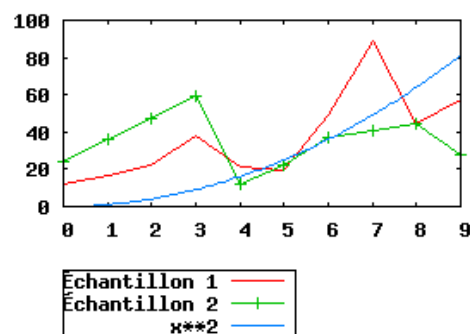


Illustration 5: Dessiner plusieurs courbes

La première courbe dessine les données du fichier samples.dat avec des lignes et le titre de la légende doit être Échantillon 1. La seconde courbe dessine les données du fichier

other_samples.dat avec des lignes et des points et la légende doit être Échantillon 2. Enfin, la troisième courbe est le graphe de la fonction x^2 .

A vous de créer et saisir des données dans le fichier samples.dat et other_samples.dat

Vous remarquerez qu'il n'y a pas d'ordre entre les paramètres de la commande plot, on peut mettre d'abord le titre puis le type de courbe ou alors l'inverse, ça ne pose pas de problèmes.

Mode script

Dans le mode script, on commence par écrire toutes les commandes dans un fichier texte, avec une commande par ligne. Ce fichier est un script Gnuplot. Le listing suivant vous montre le script qui a été utilisé pour générer la courbe 3D présentée dans l'exemple précédent.

```
# Paramétrage
set samples 20
set isosamples 20
set hidden3d

# Dessin de la courbe
splot [-3:3] [-2:2] 1 / (x*x + y*y + 1)
```

Exécution du script

Une fois un script écrit et enregistré dans un fichier, il faut l'exécuter. On peut le faire de deux manières différentes. La première consiste à lancer Gnuplot en mode interactif et d'utiliser la commande load pour charger et exécuter un script. Vous devez tout d'abord vous placer dans le même répertoire que le script en utilisant la commande cd, et ensuite, il vous reste à lancer l'exécution du script :

```
gnuplot> load '3dsample.plt'
```

La seconde manière de faire consiste à lancer le programme Gnuplot en lui donnant comme argument le chemin vers le fichier de script. Pour ce faire, vous devez lancer le programme depuis une console (invite de commandes sous Windows).

```
> gnuplot 3dsample.plt
```

Dans ce cas, Gnuplot se lance, exécute complètement le script puis est automatiquement quitté, ce qui n'était pas le cas lorsqu'on utilise la commande load en mode interactif.

Il n'y a aucune règle particulière pour choisir le nom du fichier de script. Nous utiliserons, par convention, l'extension .plt pour indiquer qu'un fichier est un script Gnuplot.

Terminaux

Les terminaux déterminent comment Gnuplot va générer ses résultats. Si l'on tape la commande **show term**, il nous dira que ce paramètre vaut 'x11' ou bien 'wxt'.

Cela signifie que la sortie standard de Gnuplot est le terminal X. C'est la raison pour laquelle les courbes s'affichent dans des fenêtres. Pour pouvoir sauvegarder les courbes dans des fichiers, il faut donc changer la sortie standard. Les possibilités offertes dépendent de votre serveur X. Pour les connaître, il suffit de taper la commande **set terminal**.

Lorsqu'on souhaite créer un fichier de sortie, il faut utiliser la commande **set output** pour spécifier le nom du fichier que Gnuplot doit créer. Le listing suivant montre le script qui permet de générer la courbe 3D de la figure 2 dans un fichier PNG.

```
# Sortie
set terminal png size 400,250
set output '3dsample.png'

# Paramétrage
set samples 20
set isosamples 20
set hidden3d

# Dessin de la courbe
splot [-3:3] [-2:2] 1 / (x*x + y*y + 1)
```

La première commande change le terminal courant pour le terminal png et définit la taille du terminal qui est de 400 pixels de large sur 250 pixels de haut. La seconde commande définit le fichier de sortie qui est 3dsample.png. Exécuter ce script et vérifier le résultat.