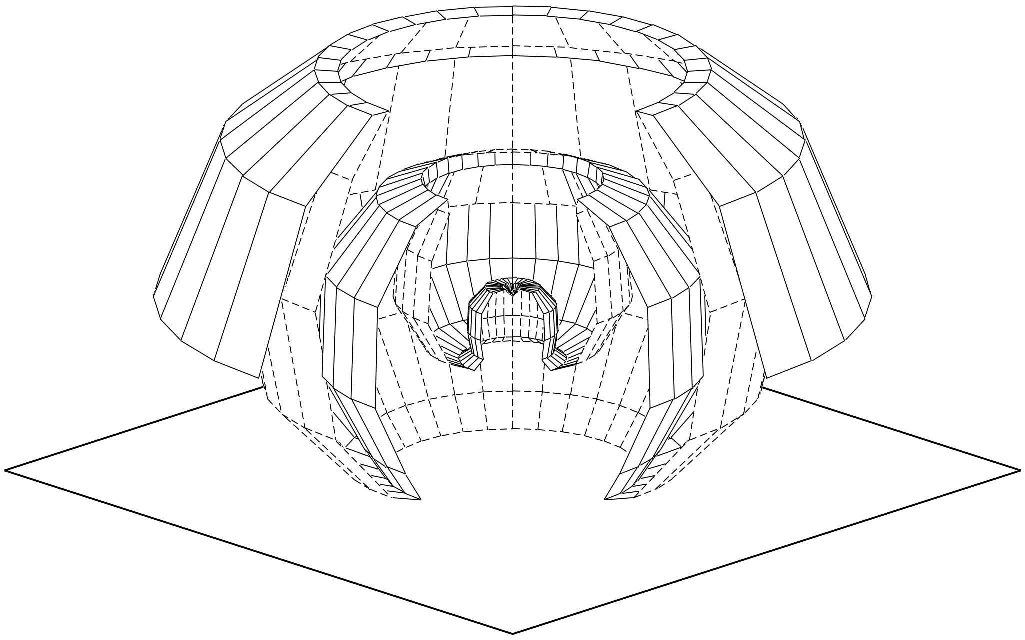
PREMIERS PAS AVEC GNUPLOT 3.5

.

Gérard SOOKAHET

version 1.0



PREMIERS PAS AVEC GNUPLOT 3.5

Gérard Sookahet
Juin 1997

Gnuplot est un traceur de fonction 2D/3D. Il est disponible pour de nombreuses plateformes UNIX, pour PC(Dos, Windows 3.1-95-NT et OS/2), Mac, Atari ST et Amiga. Son interface est quelque peu rudimentaire(commandes en lignes), mais simple à retenir et assez puissant pour permettre le dépouillement presque en temps réel de vos données. Les instructions se transmettent en ligne de commande ou avec un fichier script. Il permet de tracer soit des fonctions prédéfinies ou des fichiers de points.

---

1 Démarrage.

1.1 Démarrage.

Au prompt d'UNIX, il suffit de taper:

```
unix>gnuplot
```

Note: les symboles `unix>` et `gnuplot>` représentent, respectivement, le prompt d'UNIX et de Gnuplot.
On se retrouve alors en présence de l'écran d'acceuil de Gnuplot:

```
        G N U P L O T
        Unix version 3.5
        patchlevel 3.50.1.17, 27 Aug 93
        last modified Fri Aug 27 05:21:33 GMT 1993

        Copyright(C) 1986 - 1993 Thomas Williams, Colin Kelley

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

Terminal type set to 'x11'
gnuplot>
```

1.2 Sortie.

Au prompt de Gnuplot, il suffit de taper `quit`

1.3 Le terminal.

Avant de débuter, il est parfois raisonnable d'informer Gnuplot du type de terminal utilisé. En effet, cette information lui est primordiale pour choisir les ordres graphiques qui permettront d'afficher vos résultats à l'écran.
Si vous êtes sur un terminal Xwindow, vous devez voir apparaître, après avoir démarré, l'information suivante:

```
Terminal type set to 'x11'
gnuplot>
```

Si tel n'est pas le cas, alors il faut taper `set terminal x11`.
Gnuplot supporte d'autres types de terminaux. Pour en avoir la liste complète, il suffit de taper `set terminal` tout simplement.
Dans le cas où vous travaillez toujours sur le même type de terminal, il est souhaitable de ne pas avoir à retaper sans cesse `set terminal x11` par exemple. Pour cela, il suffit d'inclure dans votre fichier `.profile` la ligne suivante:

```
export GNUTERM=x11
```

2 Courbes 2D.

Les courbes 2D peuvent être tracées selon 3 modes différents:

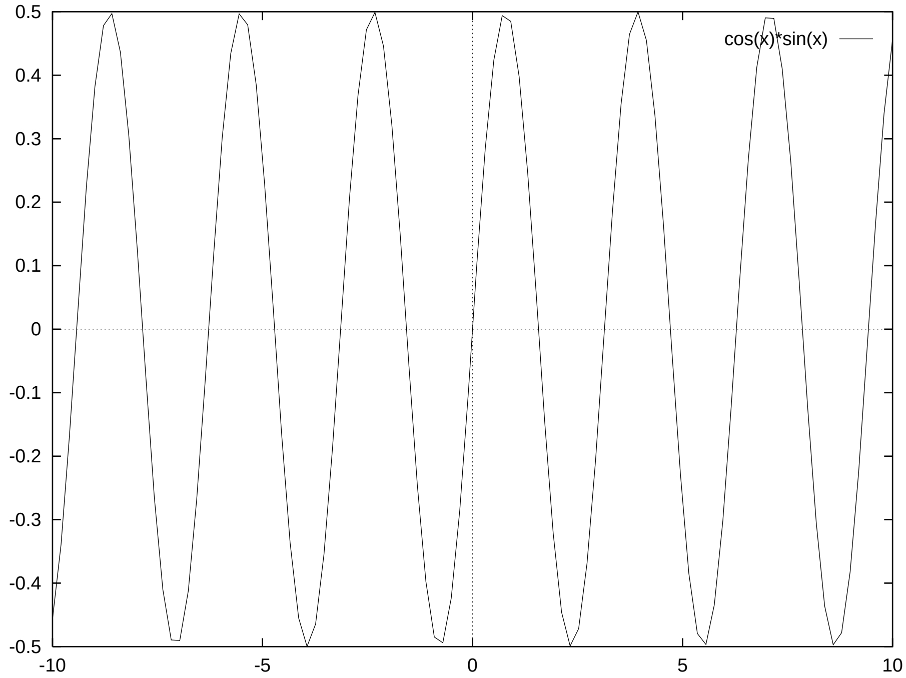
- Cartésien.
- Paramétrique.
- Polaire.

2.1 Mode cartésien.

Par défaut, Gnuplot est en mode cartésien(variable `x`). Dans ce cas, essayons de tracer la fonction $f(x) = \cos x \sin x$:

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

Un fenêtre se crée où s'affiche notre premier graphe:



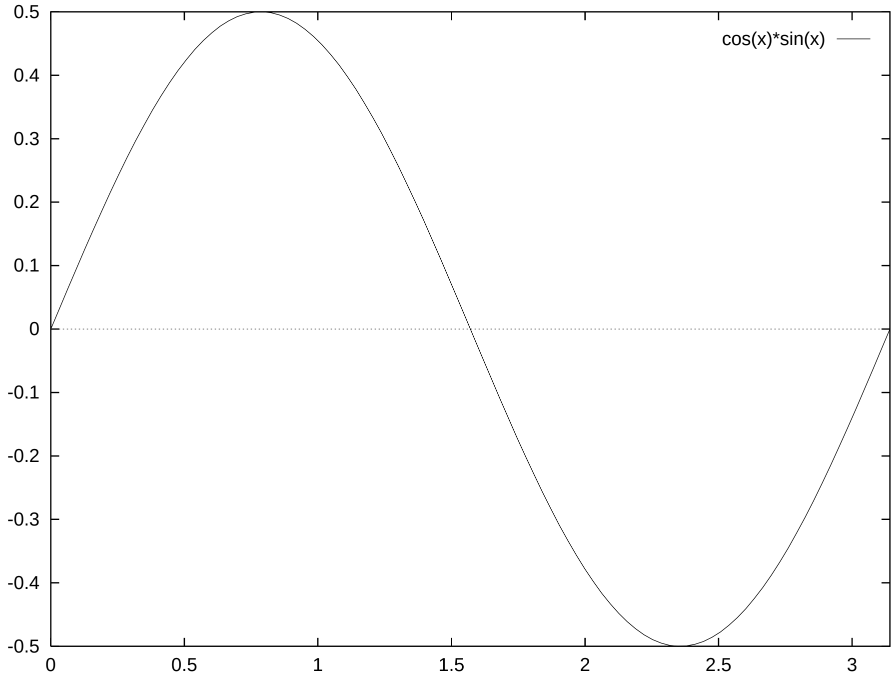
Cependant, comme nous n'avons pas spécifié d'intervalle pour `x`, c'est le programme qui se charge de choisir un intervalle par défaut. Si nous voulons spécifier un intervalle, cela se fait de deux manières:

```
gnuplot>set xrange[0:3.1416]
gnuplot>replot
```

La commande `replot` sert à retracer la fonction. Ainsi, nous avons tracé $f(x)$ sur un intervalle allant de 0 à π (approximativement!). Notons au passage, que le séparateur des deux bornes est `:` .
D'un autre côté, nous pouvons entrer directement:

```
gnuplot>plot [0:3.1416] cos(x)*sin(x)
```

Ce qui nous donne:



2.2 Mode paramétrique.

Il faut en premier lieu passer en mode paramétrique:

```
gnuplot>set parametric
```

Il s'affichera alors le message suivant:

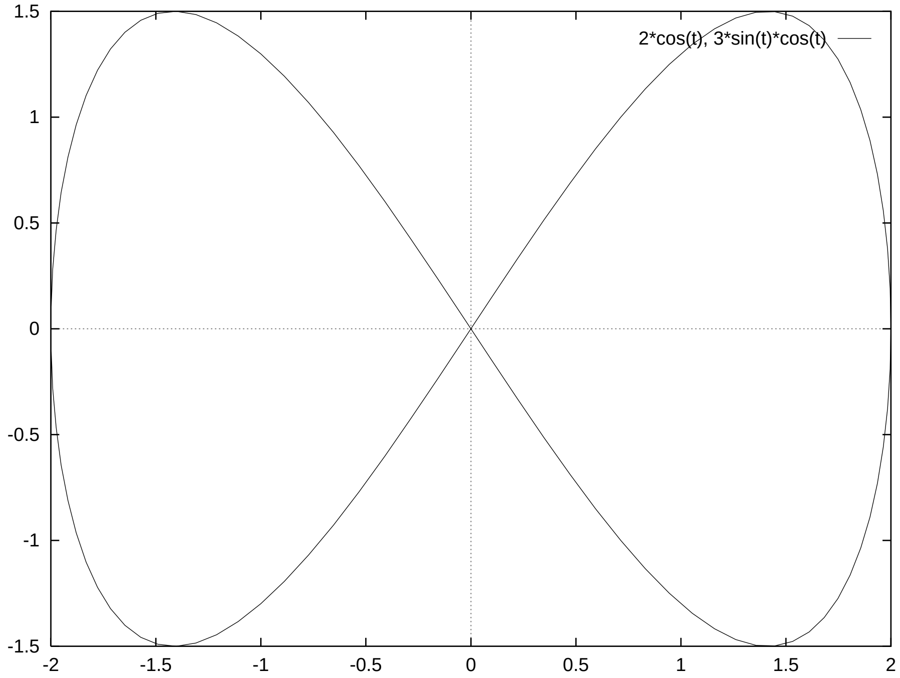
```
dummy variable is t for curves, u/v for surfaces
```

Vous l'avez compris, la variable d'usage est `t`.
Imaginons que l'on veuille représenter l'équation paramétrique suivante:

$$\begin{aligned} x(t) &= 2\cos t \\ y(t) &= 3\sin t\cos t \\ t &\in [0, 2\pi] \end{aligned}$$

Cela ressemble à l'∞!
La solution la plus compacte donne:

```
gnuplot>plot [0:2*pi] 2*cos(t),3*sin(t)*cos(t)
```



Vous avez peut-être noté que $\pi = 3.141592654....$ est prédéfini dans Gnuplot. Il ne faut pas oublier la virgule `,` qui sert de séparateur entre $x(t)$ et $y(t)$. Si jamais nous voulons revenir en mode cartésien, il faut faire:

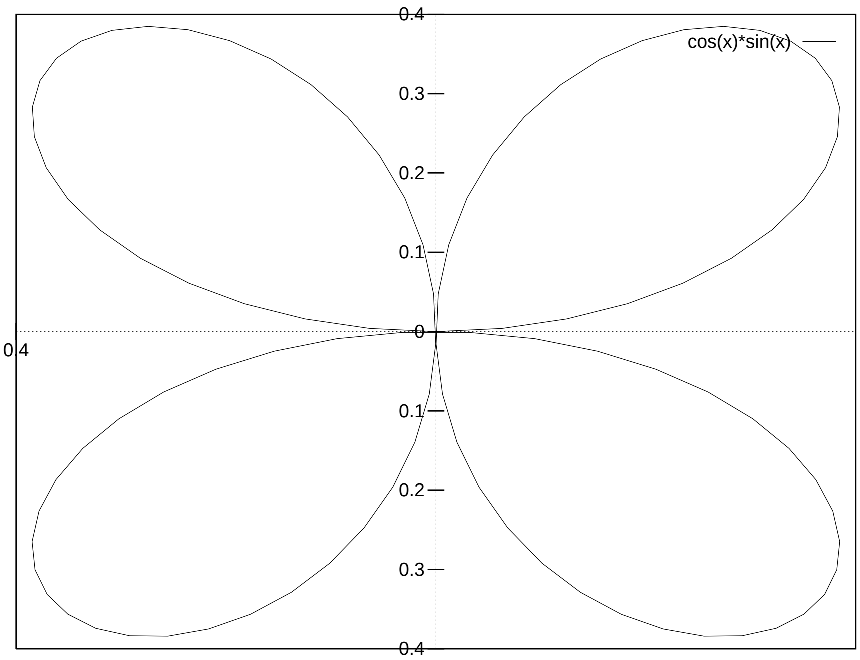
```
gnuplot>set noparametric
```

L'intervalle du paramètre `t` se contrôle par `set trange[ :  ]`.

2.3 Mode polaire.

Le passage en mode polaire se fait par `set polar`.
Pour revenir en mode cartésien: `set nopolar`.
La variable d'usage est `x` qui représente un angle en radian. Par défaut l'intervalle angulaire est $[0 : 2\pi]$ et se modifie avec `set xrange[ :  ]`.
Pour tracer la fonction $r(x) = \cos x \sin x$ sur $[0 : 2\pi]$, nous faisons:

```
gnuplot>plot [0:2*pi] cos(x)*sin(x)
```



2.4 Tracer un fichier de points 2D.

2.4.1 Structure du fichier de points.

Il s'agit d'un fichier qui comprend deux colonnes. Ces dernières correspondent aux coordonnées (x,y) de chaque point. Imaginons que nous ayons construit un fichier de points `parab.dat` représentant la fonction $f(x) = x^2$ pour x entier positif. Sous UNIX, si nous faisons `more parab.dat` nous verrons:

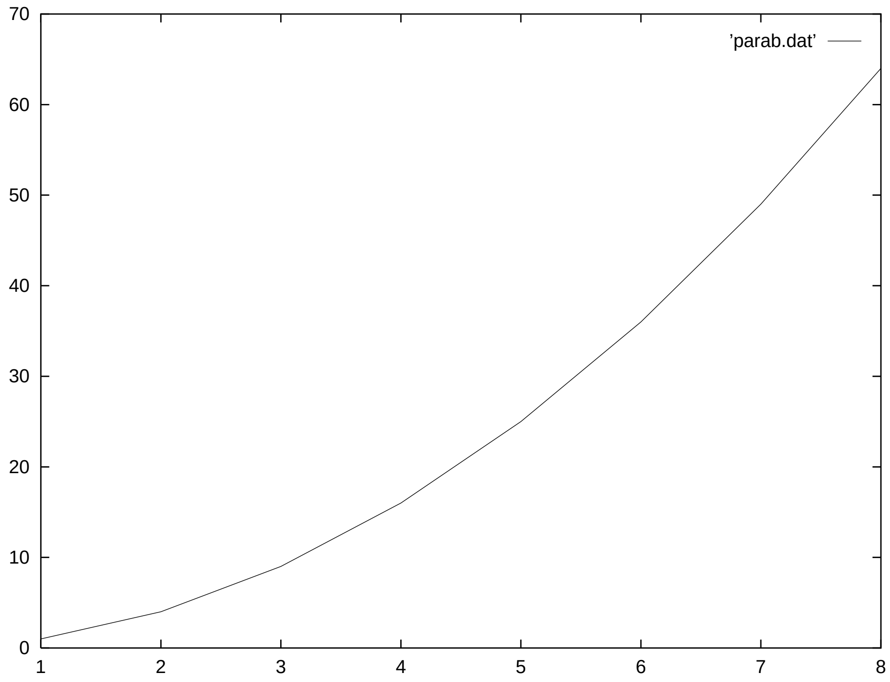
```
# x  y
  1  1
  2  4
  3  9
  4  16
  5  25
  etc ....
```

Chaque ligne contient les coordonnées (x,y) séparées par un(ou plusieurs) espace(s) ou une tabulation. La première ligne(facultative) commence par un dièse(♯), ce qui indique une ligne de commentaire qui sera ignorée par Gnuplot.

2.4.2 Tracé effectif du fichier.

Pour tracer le fichier `parab.dat`:

```
gnuplot>plot 'parab.dat' with lines
```

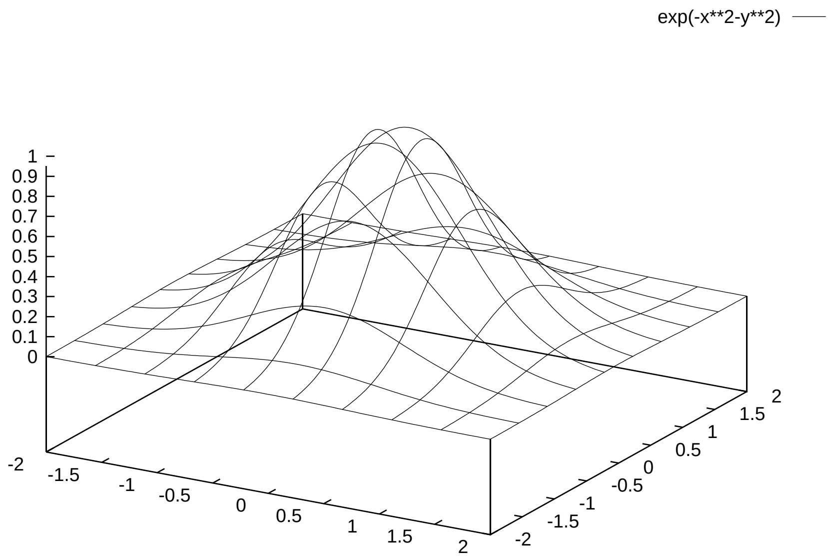


Ne pas oublier les apostrophes(') pour délimiter le nom du fichier à charger. L'option `with lines` permet de tracer un trait continu. Si l'on veut afficher des petits points: `with dots`. D'autres option sont disponibles(`points, linespoints, impulses, errorbars`.

3 Surfaces 3D.

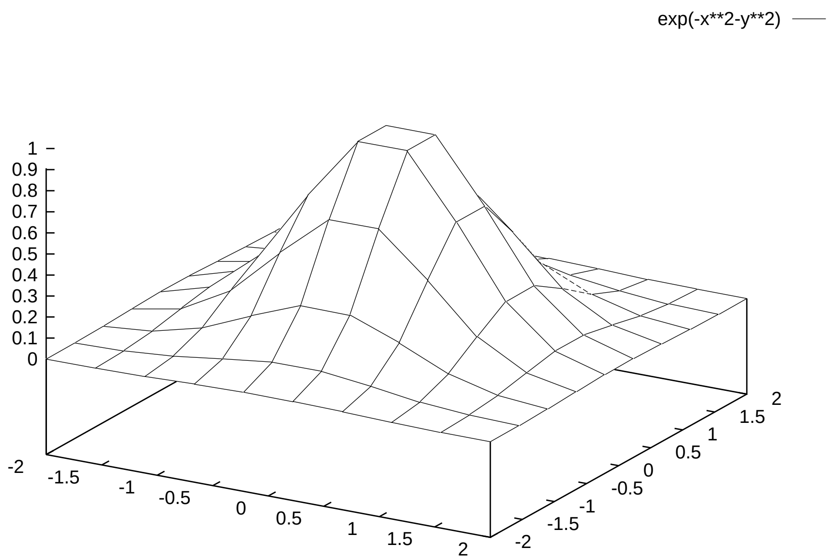
Comme il ne s'agit que d'une brève introduction, nous ne parlerons ici que des surfaces en coordonnées cartésiennes.
La commande utilisée est `splot` et les variables d'usages sont `x` et `y`. Si nous voulons tracer une gaussienne sur le compact $[-2, 2] \times [-2, 2]$, cela se fera comme suit:

```
gnuplot>splot [-2:2] [-2:2] exp(-x**2-y**2)
```



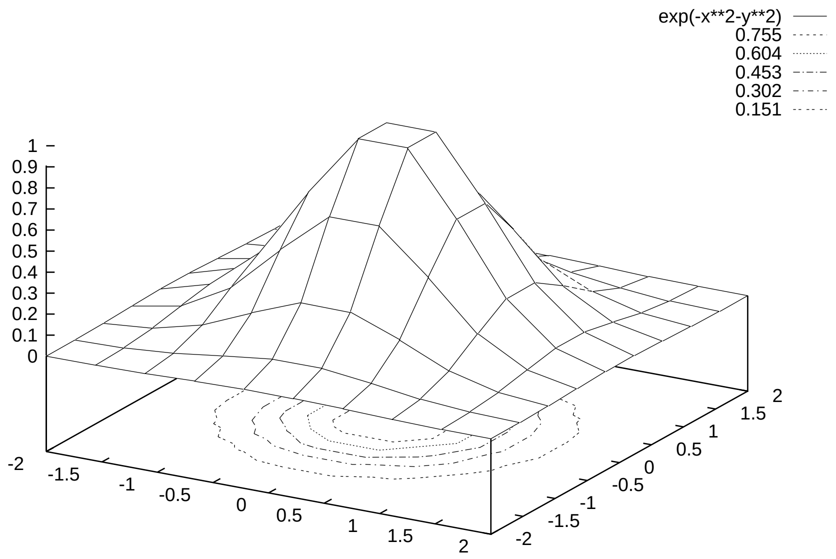
Le premier intervalle est pour `x` et le second pour `y`. Le symbole `**` est la fonction puissance(comme en Fortran).
Pour gérer les faces cachées, il faut écrire:

```
gnuplot>set hidden3d
gnuplot>replot
```



Pour tracer les lignes de niveaux qui correspondent à notre gaussienne:

```
gnuplot>set contour
gnuplot>replot
```



4 Fichiers de commandes

De nombreuses opérations sont parfois nécessaires avant de tracer une courbe. Dans cette optique, toutes ces opérations peuvent être regroupées dans un fichier de commandes afin de gagner du temps. Par exemple, on crée sous UNIX à l'aide d'un éditeur(vi, asedit, xedit ou emacs) le fichier `gauss.gnu`:

```
set title "Gaussienne Bidimensionnelle"
set xrange[-2:2]
set yrange[-2:2]
set hidden3d
set contour
splot exp(-x*x-y*y)
```

Ensuite dans Gnuplot, il suffit de faire:

```
gnuplot>load 'gauss.gnu'
```

Notons au passage que `set title` permet de donner un titre à notre graphe. Nous pouvons aussi faire de même avec un fichier de points:

```
set title "Demi-Parabole"
set xlabel "Axe des abscises"
set ylabel "f(x)"
set xrange [0:10]
plot 'parab.dat'
```

Comble du raffinement, la commande `set xlabel` permet de donner un titre à l'axe des x(idem pour y).
Si l'on décide d'afficher plusieurs graphes à la suite dans un même fichier de commande ,il est intéressant de temporiser entre chaque affichage. Cela se fait grâce à la fonction `pause`:

```
pause -1 "Appuyez sur RETURN pour continuer"
```

5 Fichiers de sortie.

L'une des grandes leçon de l'informatique, c'est de songer à sauvegarder son travail de temps en temps(au moins pour en garder une trace!). Cette sauvergarde salutaire peut être faite via une impression graphique ou dans un fichier de commandes.

5.1 Impression d'un graphe.

Afin d'imprimer un graphe il y a plusieurs possibilités. Retenons en au moins deux très simples:

- Dans l'environement Xwindow les graphiques s'affichent dans une fenêtre à part.Par conséquent, nous pouvons invoquer l'option *imprimer la fenêtre active* dans un des menus disponible.
- Une autre solution consiste à sauvegarder notre graphe au format *postscript*^{tm} et ensuite de lancer une impression à partir d'UNIX. Voyons comment procéder:

```
gnuplot>set terminal postscript
gnuplot>set output 'graphe.ps'
gnuplot>plot [0:pi] cos(x)
gnuplot>quit
```

Comme c'est souvent le cas, il suffira de lancer une impression sous UNIX avec la commande `lpr -Pxxx`(le xxx représente le petit nom de votre imprimante: qms, lw, tek, etc ....). Par défaut les fichiers *postscript*^{tm} sont configurés en mode `landscape`(paysage), et sont monochromes. Si l'on veut un mode portrait et en couleur:

```
gnuplot>set terminal postscript portrait color
```

5.2 Sauvegarde dans un fichier.

L'opération se fait par la commande `save`:

```
gnuplot>save 'mywork.gnu'
```

Gnuplot sauvera dans le fichier `mywork.gnu` toutes les commandes que vous avez utilisé.

5.3 Sauvegarde dans d'autres formats.

Il se peut que l'on souhaite conserver une image de son graphe plutôt que des commandes car certains fichiers de points sont parfois volumineux. Eventuellement, il est possible de conserver le fichier *postscript*^{tm} à condition de le compresser car il peut être lui aussi volumineux. L'idéal c'est de sauvegarder son graphe au format `pbm` par un `set terminal pbm`. Ensuite sous UNIX, il suffit d'utiliser le fameux programme de Jef Poskanzer(PBMPlus) afin de convertir notre fichier dans un format plus adequat et qui occupe moins de place(GIF,JPG,PNG).
D'autres format sont aussi disponibles suivant l'utilisation que l'on veut en faire par la suite. Parmi ceux-la:

- aifm (Adobe Illustrator).
- corel (Corel Draw).
- dxf (Autocad).
- hpgl (traceur HP).
- latex (LaTeX).

6 Aide interactive.

Gnuplot dispose d'une aide interactive et hiérarchique. Pour y avoir accès il suffit de taper `help` sans paniquer. L'écran suivant apparaît:

```
Help topics available:

autoscale        binary-data     bugs           cd
clear            comments        copyright      environment
exit             expressions     help           introduction
line-editing     load            pause          plot
print            pwd             quit           replot
reread           save            set            shell
show             splot           startup        substitution
userdefined


Help topic:
```

Il faut choisir une de ces rubriques afin d'accéder à une sous-rubrique pour avoir plus de précision. Par contre, si on a une idée précise, on peut invoquer une aide directe(`help grid` par exemple).

POUR ALLER PLUS LOIN AVEC GNUPLOT 3.5

Vous trouverez dans ce qui suit des notions un peu plus avancées mais très utiles dans la pratique afin de faire un bon usage de Gnuplot et d'exploiter au mieux vos données.

- Courbes 2D.
- Surfaces 3D.
- Fonctions, variables et constantes.
- Animations.
- Des astuces pour accroître votre productivité.

---

7 Courbes 2D.

7.1 Affichage d'un fichier multicolonne.

Soit un fichier(`crois.dat`) de la forme suivante:

```
# Croissances
#  t    x     y     z     w
   1    2     1     1     2.71
   2    4     4     8     7.38
   3    6     9     27    20.08
   4    8     16    64    54.59
   5    10    25    125  148.41
   etc....
```

Est-ce quelqu'un a reconnut la fonction de la dernière colonne(approximativement!)? Si nous faisons `load 'crois.dat'`, par défaut, Gnuplot affichera la représentation graphique des deux premières colonnes. Si nous souhaitons afficher la première et la troisième colonne, il nous faudra utliser la commande `using:` comme suit:

```
gnuplot>plot 'crois.dat' using 1:3 with lines
```

7.2 Affichage toutes les n lignes.

L'utilisation de `awk` permet d'afficher un fichier de points toutes les n lignes. Par exemple pour afficher le fichier `parab.dat` toutes les 4 lignes:

```
gnuplot>plot "< awk '{if(NR%4==0)print}' parab.dat"
```

7.3 Echelle logarithmique.

L'échelle est disponible pour les trois axes(x,y,z). Pour fixer l'échelle logarithmique sur les axes x et y:

```
gnuplot>set logscale xy
```

Pour désactiver la commande on utilise `set nologscale xy`.

7.4 Grille.

Pour afficher une grille sur votre graphique:

```
gnuplot>set grid
```

Pour désactiver la commande: `set nogrid`.

8 Afficher plusieurs graphes simultanément.

L'affichage simultané se fait en séparant chaque formule ou nom de fichier par une virgule. Exemple:

```
gnuplot>plot [0:3] sin(x),x with dots,x-x**3/6 with points
```

9 Surfaces 3D.

9.1 Fichiers de points 3D.

La structure d'un fichier de points 3D est la suivante:

```
# Fichier 3D
# x   y   z
  0   0   1
  0   1   2
  0   2   3
  0   3   4

  1   0   4
  1   1   3
  1   2   3
  1   3   1

  2   0   6
  2   1   6
  2   2   6
  2   3   6

  3   0   6
  3   1   1
  3   2   6
```

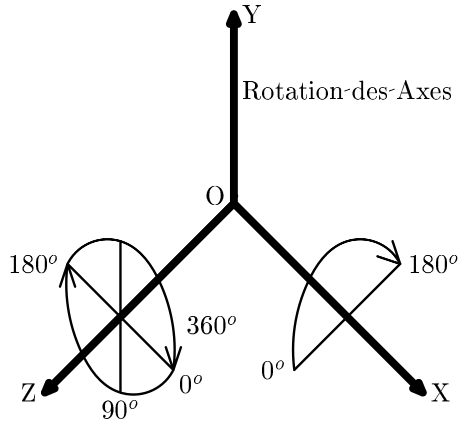
```
3   3   1

etc....
```

Chaque ligne correspond aux coordonnées d'un point dans l'espace. Chaque bloc représente un ensemble de points sur une même ligne de maillage. Ces blocs sont séparés par une ligne vide. Dans cet exemple, nous avons choisit de garder les isolignes selon x constantes, mais nous tout aussi bien utiliser les y.

9.2 Angles de vue.

La commande `set view` permet de modifier l'angle de vue selon lequel on perçoit un graphe sur notre écran bidimensionnel. La syntaxe générale est:

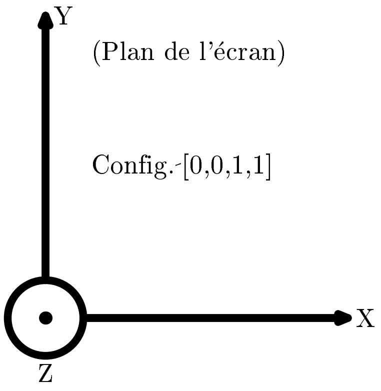
```
set view rot_x,rot_z,scale,scale_z
```



- `rot_x` est l'angle de rotation(en degré) autour de l'axe Ox du trièdre (xyz). Il varie de 0^o à 180^o.
- `rot_z` est l'angle de rotation(en degré) autour de l'axe Oz du trièdre (xyz). Il varie de 0^o à 360^o.
- `scale` est l'échelle globale de la figure.
- `scale_z` est l'échelle de l'axe Oz.

La configuration par défaut est visible au démarrage de Gnuplot:

```
gnuplot>show view
        view is 60 rot_x, 30 rot_z, 1 scale, 1 scale_z
```

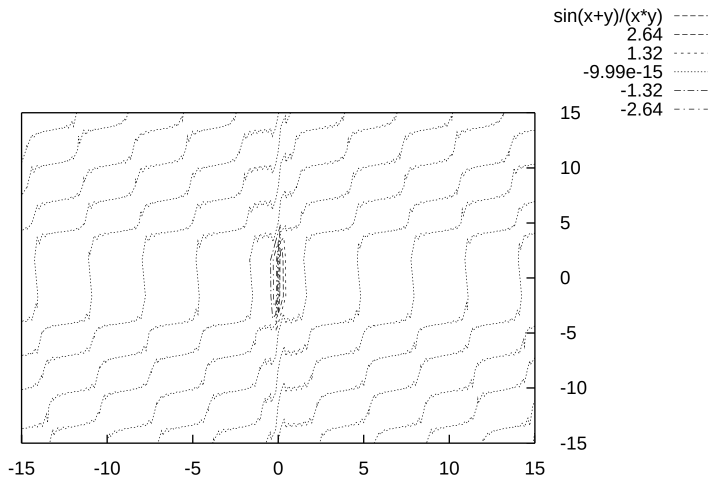


La configuration $(0, 0, 1, 1)$ correspond à un trièdre tel que le plan (xOy) est contenu sur la surface de votre écran sachant que l'axe (Ox) est horizontal et pointe vers votre droite. L'axe (Oy) est verticale et pointe vers le haut. L'axe (Oz) est ortogonal à la surface de l'écran et pointe vers votre visage.

9.3 Courbes de niveaux.

Il peut être intéressant de disposer des courbes de niveau d'une surface sans avoir à visualiser la surface en question:

```
gnuplot>set contour
gnuplot>set view 0,0,1,1
gnuplot>set nosurface
gnuplot>splot [-15:15] [-15:15] sin(x+y)/(x*y)
```

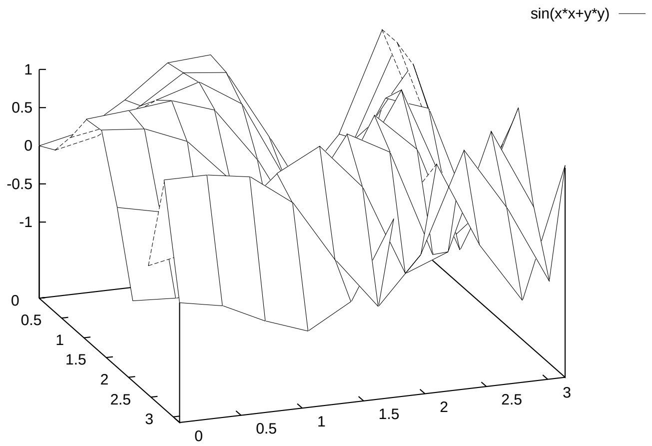


9.4 Finesse du maillage.

Certaines surfaces varient beaucoup. Il faut donc en affiner le tracé afin de pouvoir visualiser quelquechose de convenable. La commande utilisée est `set isosample`. Elle comprend deux paramètres qui correspondent à la finesse sur l'axe des x et des y. Les valeurs par défaut sont visibles grâce à `show isosample`. Par exemple:

```
gnuplot>set view 60,70,1,1
gnuplot>set hidden3d
gnuplot>set isosample 10,10
gnuplot>splot [0:pi] [0:pi] sin(x*x+y*y)
```

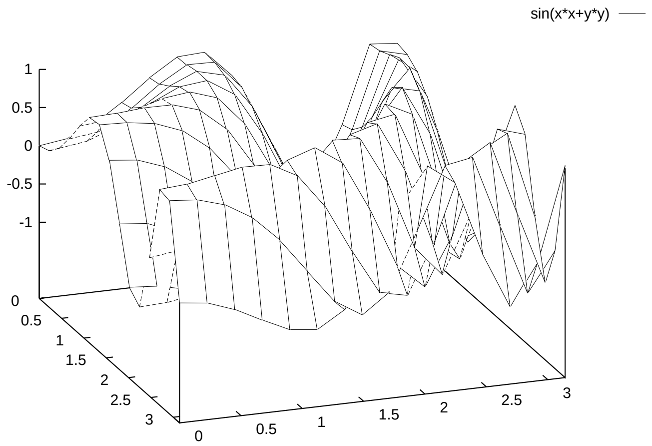
Cela donne:



Avec:

```
gnuplot>set isosample 15,15
gnuplot>replot
```

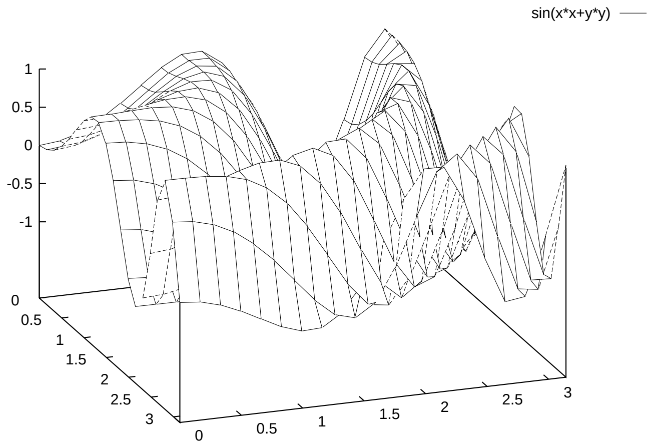
Nous obtenons:



Avec:

```
gnuplot>set isosample 20,20
gnuplot>replot
```

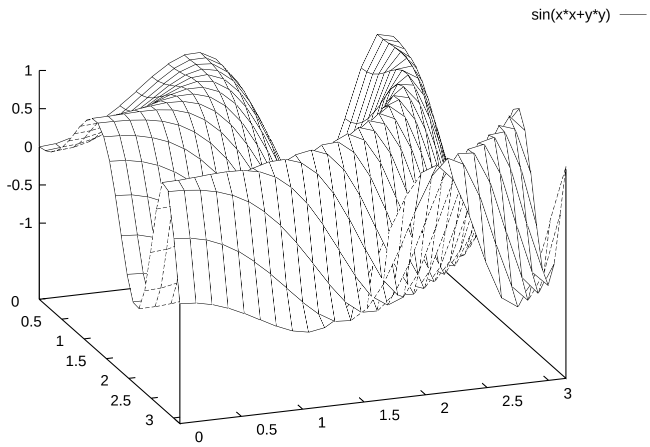
Nous obtenons un tracer encore plus fin:



Et pour finir, avec:

```
gnuplot>set isosample 25,25
gnuplot>replot
```

Nous obtenons:



C'est beau, mais toutefois il faut savoir que plus on augmente les valeurs de `isosample`, plus le temps de calcul est long.

9.5 Surfaces paramétriques.

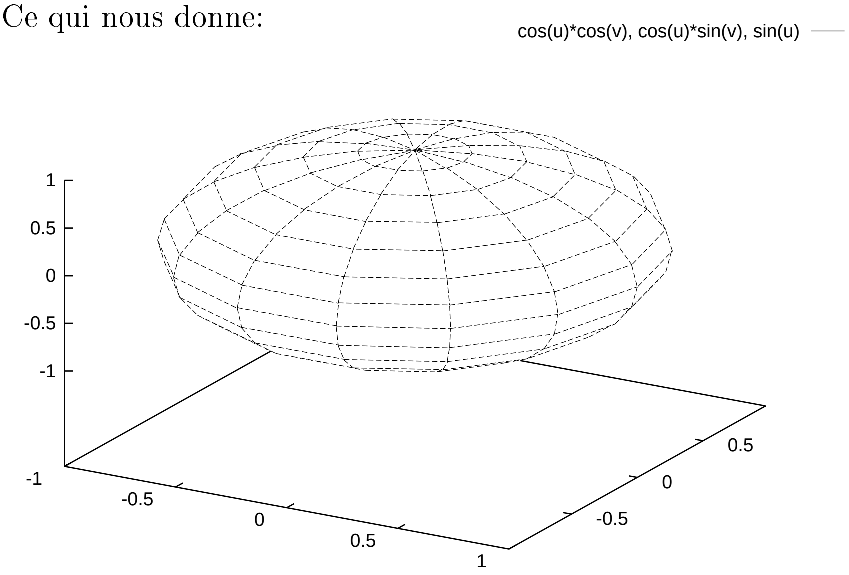
Une surface paramétrique est décrite par trois fonctions du couplet (u, v):

$$\begin{aligned} x &= f(u, v) \\ y &= g(u, v) \\ z &= h(u, v) \end{aligned}$$

Le passage en mode paramétrique se fait par `set parametric`. Les intervalles se contrôlent par `set urange[ : ]` et `set vrange[ : ]`. Les fonctions $x(), y()$ et $z()$ sont séparées par des virgules.
Voici l'exemple très simple de la sphère:

```
gnuplot>set parametric
gnuplot>set hidden3d
gnuplot>isosample 15,15
gnuplot>set urange[-pi/2:pi/2]
gnuplot>set vrange[0:2*pi]
gnuplot>splot cos(u)*cos(v),cos(u)*sin(v),sin(u)
```

Ce qui nous donne:



Vous avez sans nul doute remarqué que la sphère était un peu plate. Pour remédier à ce petit problème, il suffit de modifier le dernier paramètre dans `set view`: celui du facteur d'échelle selon Oz.
Ainsi:

```
gnuplot>set view ,,,1.5
```

Notez au passage qu'il n'est pas nécessaire de faire figurer les paramètres inchangés.

10 Fonctions, variables et constantes.

10.1 Un exemple concret.

Gnuplot accepte la déclaration des variables et des fonctions un peu comme dans un logiciel de calcul formel. Les fonctions doivent dépendre des variables usuelles servant pour les représentations graphiques (x, y, u, v, t).
Pour en comprendre l'utilisation, le mieux est d'exploiter un exemple. Dans le cadre de l'Hydrodynamique et de la Théorie des Transformations Conformes, on étudie la transformation de Joukovski:

$$w(z) = V_i(z + \frac{R^2}{z})$$

(z appartenant au champ complexe)
Celle-ci peut décrire l'écoulement plan stationnaire d'un fluide autour d'une circonférence de rayon R. On y associe une fonction de courant $\Psi(r, \theta)$ en coordonnées polaires telle que:

$$\Psi(r, \theta) = V_i \sin\theta(r - \frac{R^2}{r})$$

En représentant les courbes de niveaux associées à la fonction de courant, nous pouvons avoir une idée qualitative de la représentation des lignes de courant de l'écoulement.
Cette représentation se fera en coordonnées cartésiennes. D'où:

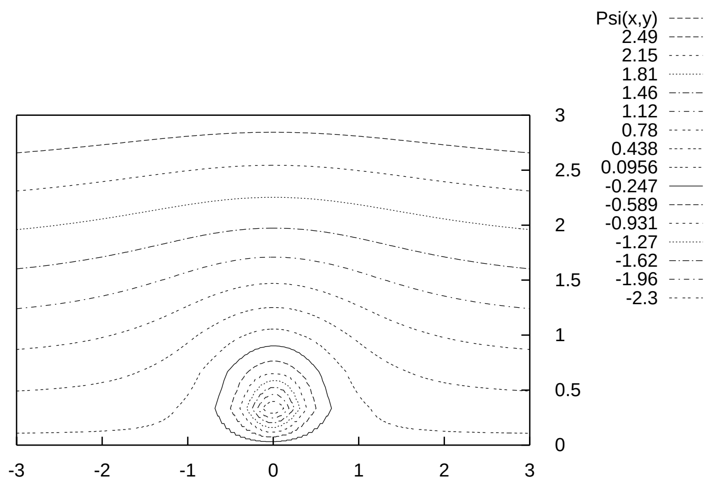
$$\begin{aligned} r^2 &= x^2 + y^2 \\ \sin\theta &= \frac{y}{\sqrt{x^2 + y^2}} \end{aligned}$$

Voici le code source:

```
# ============================================
#  Lignes de courant autour d'une ciconference
# ============================================
#
# Angle de vue tel que le plan (xOy) soit confondu avec celui de l'ecran
set view 0,0,1,1
set nosurface
set contour
#
# Augmentation de la densite des courbes de niveau
set cntrparam levels 15
```

```
#
# Rayon de la circonference
rc=1.0
#
# Constante associee a la vitesse
vi=1.0
r(x,y) = sqrt(x*x+y*y)
sint(x,y) = y/r(x,y)
#
# Fonction de courant
Psi(x,y) = vi*sint(x,y)*(r(x,y)-rc/r(x,y))
set xrange[-3:3]
set yrange[0:3]
splot Psi(x,y)
```

Voici le résultat obtenu:



Note:
Nous avons choisit de représenter uniquement le demi-plan suprérieur. L'autre demi-plan se déduit par symétrie par rapport à l'axe Ox. Il faut aussi faire abstraction de la zone intérieure du cercle de centre $(0,0)$ et de rayon 1. En effet, cette dernière n'a pas de sens "physique".

10.2 Fonctions définies par morçeaux.

Un astuce intéressante permet à Gnuplot de tracer des fonctions définies par morçeaux. Supposons que l'on veuille représenter la fonction:

$$\begin{array}{ll} \sin x & \text{si } x < 0 \\ x & \text{sinon} \end{array}$$

Il suffira d'écrire:

```
gnuplot>f(x)=(x<0)?sin(x):x
gnuplot>plot [-10:10] f(x)
```

Cette syntaxe peut paraître quelque peu ésotérique, mais les initiés au langage C auront tout de suite reconnu un opérateur ternaire:
`<condition>?<expression1>:<expression2>`
Autrement dit, si la condition est vrai, on évalue l'`expression1` sinon on évalue l'`expression2`.
Mieux encore, cette fois-ci on voudrait tracer la fonction f qui vaut:

$$\begin{array}{ll} -(x+\frac{\pi}{2}) & \text{si } x < -\frac{\pi}{2} \\ \cos x & \text{si } -\frac{\pi}{2} \leq x \leq \frac{\pi}{2} \\ x-\frac{\pi}{2} & \text{si } x < \frac{\pi}{2} \end{array}$$

Pour ce faire, nous écrivons:

```
gnuplot>pi_2=pi/2.0
gnuplot>g(x)=(x<=-pi_2)?-(x+pi_2):cos(x)
gnuplot>f(x)=(x<=pi_2)?g(x):x-pi_2
gnuplot>plot [-10:10] f(x)
```

Si vous avez compris le principe des opérateurs ternaires, il ne tient qu'à vous de définir une fonction avec autant de morçeaux désirés.
Note:
A la lumière de ces explications, les fonctions Min et Max se définissent aisément:

```
min(x,y)=(x<y)?x:y
max(x,y)=(x>y)?x:y
```

11 Animation.

Gnuplot parvient à réaliser des animations sommaires grâce à la fonction `reread`. En fait il s'agit d'une astuce qui consiste à rafraîchir la fenêtre graphique en écrasant le graphe précédent.

11.1 Un exemple simple d'animation.

Celui-la va consister à déplacer une gaussienne bidimensionnelle selon la droite $y = x$ dans le plan xOy. A l'aide d'un éditeur, on crée le fichier `anim1.gnu`:

```
# anim1.gnu
# Deplacement d'une surface gaussienne selon y=x
#
a=a+0.1
splot exp(-(x-a)**2-(y-a)**2)
reread
```

A chaque fois que Gnuplot arrive sur l'instruction `reread`, il se repositionne sur la première ligne du fichier `anim1.gnu` et relit ce dernier. L'incrémentation du paramètre `a` va donc modifier le centrage de la gaussienne. Pour réaliser l'animation, il suffit d'entrer au prompt de Gnuplot les lignes qui suivent:

```
gnuplot>a=0.1
gnuplot>set xrange[0:15]
gnuplot>set yrange[0:15]
gnuplot>load 'anim1.gnu'
```

Néanmoins, un petit inconvénient demeure. Effectivement, l'animation a tendance à continuer indéfiniment!. Pour palier à cela, un magistral CONTROL-C fera l'affaire faute de mieux.

12 Des astuces pour accroître votre productivité.

- Si vous en avez assez de saisir des instructions en entier, Gnuplot est très conciliant et accepte des instructions tronquées dans la mesure où il n'y a pas d'ambiguïté.
  Par exemple au lieu de taper:

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

  Vous pouvez écrire:

  ```
  gnuplot>p sin(x) w d
  ```

  De même, au lieu de:

```
gnuplot>set terminal postscript portrait
```

Nous avons:

```
gnuplot>set ter post p
```

Il y a d'autres raccourcis. A vous de les découvrir.

- Comme c'est souvent le cas dans de nombreux logiciels(`vi` par exemple), vous pouvez faire usage d'une commande shell sans sortir de Gnuplot en utilisant un point d'exclamation. Ainsi pour imprimer un fichier `sinus.ps`, il suffit de faire:

```
gnuplot>! lpr -Pxxx sinus.ps
```

- Gnuplot dispose d'un rappel historique des commandes. La touche fléchée $\uparrow$ accède aux commandes précédentes. La touche $\downarrow$ accède aux commandes suivantes. Il est à noter que cela ne fonctionne pas toujours. En effet, ces touches dépendent fortement de la configuration du terminal.

- La fonction `thru` permet d'agir sur les données d'un fichier avant l'affichage. Plus précisément, dans un fichier de deux colonnes, il est possible de calculer une fonction de la seconde colonne. Reprennons notre fichier de points `parab.dat` qui contient des entiers et leurs carrés respectifs. Nous écrivons:

```
gnuplot>plot 'parab.dat' thru sqrt(x)
```

Si tout se passe bien, nous devrions voir une droite. Ainsi, nous pouvons utiliser n'importe quelle fonction prédéfinie dans Gnuplot ou en définir une:

```
gnuplot>f(x)=x**2
gnuplot>plot 'parab.dat' thru f(x)
```

Notons au passage que la variable `x` porte sur la seconde colonne du fichier. Pour lever l'ambiguïté, il ne tient qu'à vous d'utiliser `y`.

13 Des liens extérieurs.

Voici des liens intéressants disponibles sur internet:

- La Page Web de Gnuplot. `http://www.cs.darmouth.edu/gnuplot_info.html`
- Le Site Officiel de Gnuplot. `ftp://ftp.darmouth.edu/pub/gnuplot`
- Le Newsgroup de Gnuplot. `news://comp.graphics.apps.gnuplot`

14 Auteurs.

Les auteurs et principaux contributeurs de Gnuplot sont:

- Thomas Williams, Colin Kelley, Russel Lang, Dave Kotz, Gershon Elber, John Campbell, Alexander Woo et bien d'autres....
- Peut-être vous dans un futur proche.
- Documentation rédigée par Gérard Sookahet.
  `http://mucha.obspm.fr/~grouss/GE/index.html`
