



Ceci est un extrait électronique d'une publication de  
Diamond Editions :

<http://www.ed-diamond.com>

Ce fichier ne peut être distribué que sur le CDROM offert  
accompagnant le numéro 100 de **GNU/Linux Magazine France**.

La reproduction totale ou partielle des articles publiés dans Linux  
Magazine France et présents sur ce CDROM est interdite sans accord  
écrit de la société Diamond Editions.

Retrouvez sur le site tous les anciens numéros en vente par  
correspondance ainsi que les tarifs d'abonnement.

Pour vous tenir au courant de l'actualité du magazine, visitez :

<http://www.gnulinuxmag.com>

Ainsi que :

<http://www.linux-pratique.com>

et

<http://www.miscmag.com>



## → Des courbes harmonieuses avec Gnuplot 4.0

Cyril Buttay, Florent Morel,

**EN DEUX MOTS** Gnuplot fait partie de ces logiciels libres dont l'origine remonte à la nuit des temps (1986 !). Pourtant, malgré son âge avancé, il fait l'objet d'une activité «brûlante» : sa version 4.0 est sortie en avril 2004 (la précédente remontait à un peu plus de 10 ans...). La version 4.1 est actuellement en développement, et sortira peut-être pour les 20 ans de cet outil de visualisation indémodable !

Gnuplot permet de tracer des données scientifiques en deux ou trois dimensions, sous forme de courbes, surfaces, etc. Cet article est basé sur l'utilisation de Gnuplot par les auteurs dans le cadre de publications scientifiques.



Comme tous les logiciels de cette époque, Gnuplot possède une interface un peu rebutante, mais il mérite qu'on s'intéresse à lui (ce ne sont pas les inconditionnels de LaTeX, Vi ou Emacs qui me contrediront). C'est un outil en ligne de commande [1], qui fonctionne comme LaTeX : on lui passe une série de directives, on compile, et on récupère le résultat sous forme de fichier graphique. Fonctionnant sur les mêmes principes, Gnuplot et LaTeX sont donc tout naturellement destinés à travailler ensemble (les graphiques produits par Gnuplot peuvent bien entendu être inclus dans un autre outil de mise en page). Cependant, si LaTeX génère par défaut des documents irréprochables (dans leur forme, hein... pour le fond, c'est votre boulot), Gnuplot nécessite un peu plus de travail de finition. Dans cet article, nous allons donc voir comment le dompter pour en tirer la substantifique moelle. Nous allons nous consacrer à la mise en forme d'une figure 2-D qui pourrait être destinée, par exemple, à une revue scientifique. Sachez cependant que Gnuplot est capable de générer bien d'autres formes de tracés, auxquels les concepts étudiés ici s'appliquent tout autant.

### Premier exemple

L'équivalent du `Hello world` pour un logiciel de visualisation, c'est de tracer la fonction

$\sin(x)$  ! Pour cela, il vous suffit, en ligne de commande, de taper la commande `gnuplot` pour lancer l'interpréteur :

```
$ gnuplot

G N U P L O T
Version 4.0 patchlevel 0
last modified Thu Apr 15 14:44:22 CEST 2004

[..]

Send bugs, suggestions and mods to
<gnuplot-bugs@lists.sourceforge.net>

Terminal type set to 'x11'
```

Notez la dernière ligne du message d'accueil (`Terminal type set to 'x11'`), qui signifie que les tracés seront dirigés vers votre serveur graphique ; c'est-à-dire qu'ils seront directement affichés au lieu d'être enregistrés dans des fichiers. Entrez alors la commande suivante : `plot sin(x)`, ce qui devrait produire le tracé de la figure 1. Vous pouvez remarquer que, par convention, la variable  $x$  représente l'abscisse. Si vous préférez écrire  $\sin(t)$ , il vous faut d'abord déclarer  $t$  comme variable libre par `set dummy t`.

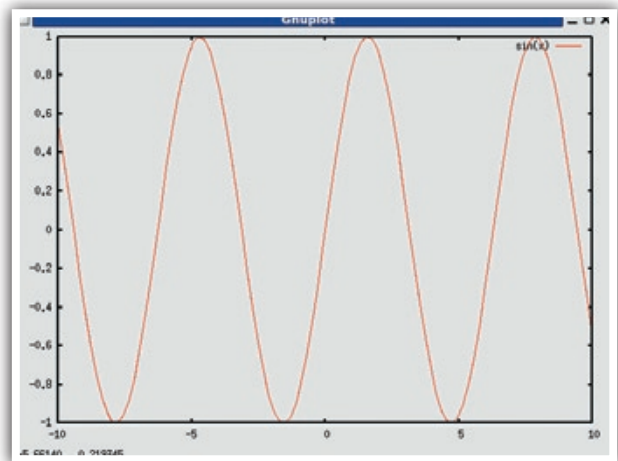


Fig 1 : Tracé de  $\sin(x)$  sur le terminal X11.

### Utilisation avancée

Maintenant que les présentations sont faites, nous allons pouvoir aller plus loin. Dans cette partie, nous allons chercher à tracer l'évolution de la population mondiale, d'après les données disponibles sur le site du bureau américain du recensement (<http://www.census.gov/ipc/www/worldpop.html>). Le fichier récupéré sur ce site est enregistré sous `population.dat` et possède la forme suivante :

1950	2556517137	1.47	37798160
1951	2594315297	1.61	42072962
1952	2636388259	1.71	45350197
1953	2681738456	1.77	47979452
1954	2729717908	1.87	51465740

1955	2781183648	1.89	52974870
1956	2834158518	...	

La première colonne correspond à l'année du recensement, la seconde à la population mondiale correspondante, la troisième est le taux d'accroissement annuel, et la quatrième la variation nette de population. Nous ne nous occuperons ici que des deux premières colonnes. Toutes les personnes qui trouvent qu'il est idiot d'afficher un résultat numérique avec autant de chiffres significatifs sont invitées à en faire part au bureau américain du recensement.

## Les terminaux

Nous avons vu que par défaut, après le lancement de Gnuplot, les tracés sont dirigés vers le serveur graphique. Sous Linux, il s'agit en général du terminal X11, mais Gnuplot en connaît bien d'autres. Pour avoir un aperçu des terminaux existants, tapez `set terminal` à l'invite de Gnuplot. Dans la longue liste qui se déroule alors, on retrouve d'autres serveurs d'affichage, des imprimantes, mais aussi des formats de fichiers (\*.png, \*.pbm, etc.). Ce sont ces derniers qui vont nous intéresser, puisqu'un tracé dirigé vers ces terminaux sera en fait simplement enregistré dans un fichier du format correspondant.

Reste donc à choisir le terminal adapté à notre usage. Pour l'utilisation avec LaTeX, il vaut mieux se tourner vers le format Postscript, qui permet l'enregistrement sous forme vectorielle et qui est bien accepté par LaTeX. Il faut de plus savoir que tous les terminaux n'offrent pas les mêmes possibilités sous Gnuplot (certains n'autorisent pas les lettres accentuées, d'autres la couleur, etc.). De ce point de vue, le terminal `postscript` est le plus avancé. Il est même parfois plus simple de générer une sortie Postscript et de la convertir après coup (avec ImageMagick par exemple). Notons tout de même que certains terminaux sont conçus spécifiquement pour LaTeX (`psLaTeX`, `epsLaTeX`, `emtex`), et permettent de profiter de la qualité typographique offerte par ce dernier [2].

## Tracé de données

Dans l'exemple de la figure 1, nous avons tracé une fonction. Mais Gnuplot peut également (et avec la même commande) tracer des données. Dès que le volume de commande devient important, il est préférable de passer par un fichier texte, plutôt que de taper les commandes en mode interactif. Voici le contenu de `population.plt` (vous pouvez prendre l'extension de votre choix, Gnuplot s'en moque) :

```
1 # fichier Gnuplot pour tracer l'évolution de la population mondiale
2 # entre 1950 et 2050
3
4 set term post eps enhanced
5 set output "population.eps"
6 plot 'population.dat' using 1:2
```

Pour exécuter ce fichier, vous pouvez taper `gnuplot population.plt` en ligne de commande ou `load 'population.plt'` si vous avez déjà lancé Gnuplot. Il y a une différence fondamentale entre les deux méthodes : dans la première, Gnuplot se lance, exécute votre commande, puis se termine. À chaque exécution de votre script, Gnuplot repart à zéro et se comporte donc de la même manière. Dans la seconde, il reste ouvert après

l'exécution, et garde en mémoire les actions effectuées. Le résultat des commandes que vous pourrez entrer par la suite dépend donc du script que vous avez exécuté auparavant. Pour vous en convaincre, faites la manipulation suivante (en ligne de commande) :

```
$ gnuplot
..
gnuplot>load 'population.plt'
gnuplot>plot sin(x)
```

Et comparez avec le résultat du premier exemple, en début d'article : le terminal vers lequel est dirigé le résultat de `plot sin(x)` est le dernier utilisé (X11 pour le premier exemple, `postscript` ici). Pour obtenir un comportement répétable, il est donc conseillé d'exécuter vos scripts par `gnuplot nom_du_script`.

Décortiquons maintenant le contenu du script `population.plt` :

- ▶ Lignes 1 et 2, des commentaires débutant par le classique caractère # ;
- ▶ Ligne 4, on sélectionne le terminal `postscript`, dans sa version `eps enhanced`. Les fichiers `eps` sont identiques aux fichiers `ps` (Postscript), mis à part qu'il ne peuvent contenir qu'une seule page, et qu'ils sont destinés à être incorporés dans un document. L'ajout de la directive `enhanced` signifie que l'on fait appel aux fonctions de typographie « avancée » du terminal, comme les exposants et les indices. Remarquez aussi que lorsqu'il n'y a pas d'ambiguïté, il est possible de raccourcir les instructions de Gnuplot (`term` au lieu de `terminal`, `post` pour `postscript`...). Attention cependant à ne pas perdre en lisibilité ce que vous gagnez en rapidité de frappe ;
- ▶ Ligne 5, on définit le nom du fichier de sortie (ici `population.eps`) ;
- ▶ Enfin, ligne 6, on demande le tracé du fichier de données `population.dat`, en utilisant la colonne 1 du fichier pour les abscisses et la colonne 2 pour les ordonnées.

À la fin de l'exécution de ce script, vous devriez obtenir dans votre répertoire courant le fichier `population.eps`, dont le contenu correspond à la figure 2.

Dans ce script très court, nous avons quasiment fait le tour de toutes les commandes de Gnuplot que nous utiliserons. D'une manière générale, un script consiste à définir les valeurs de propriétés à l'aide de la commande `set`, puis à effectuer le tracé par `plot`.

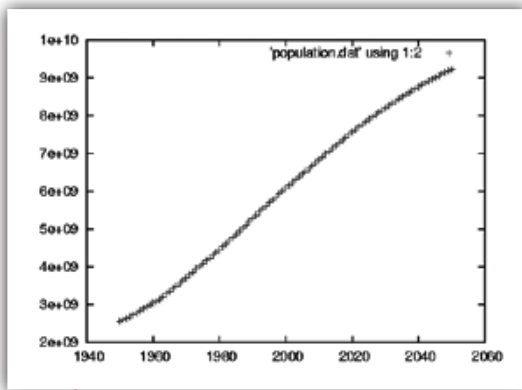


Fig. 2 : Le résultat obtenu « brut de décoffrage ».

La complexité et la puissance de Gnuplot résident dans la multitude des propriétés modifiables (pour en obtenir la liste, il vous suffit de taper `help set`). C'est à ces propriétés que nous allons maintenant nous intéresser en détail.

### Commandes de mise en forme

Le tracé de la figure 2 est clair, mais il est trop brut pour être acceptable. Commençons par mettre en forme les axes.

```
1 # fichier Gnuplot pour tracer l'évolution de la population mondiale
2 # entre 1950 et 2050
3
4 set term post eps enhanced
5 set output "population.eps"
6 set encoding iso_8859_15
7 set xlabel "Année"
8 set ylabel "Population mondiale (en milliards)"
9 set format y "%.0s"
10 set mxtics
11 set grid ytics xtics mxtics
12 plot [1950:2050][0:]'population.dat' using 1:2
```

Par rapport au script précédent, nous avons ajouté ici les lignes 6 à 11. Voyons leur effet :

- ▶ Par la ligne 6, on spécifie l'encodage de texte utilisé pour le script. Cette fonction est très importante dès que l'on utilise des caractères autres qu'anglais (comme les caractères accentués, ç, œ, etc.). L'`iso_8859_15` est le plus utilisé pour un Linux francisé ;
- ▶ Ligne 7 et 8, les mots-clés `xlabel` et `ylabel` correspondent aux titres des axes ;
- ▶ Ligne 9, on définit le format d'affichage de l'échelle des ordonnées. Pour obtenir de plus amples renseignements sur ces formats tapez `help format specifiers` dans Gnuplot. Le format défini par `%.0s` correspond à la partie entière de la mantisse de `y` en notation scientifique

(c'est-à-dire avec des puissances multiples de 3). Bref, regardez plutôt la différence dans l'affichage de l'axe des ordonnées entre la figure 2 et la figure 3 !

- ▶ La ligne 10 définit les graduations de l'axe des abscisses. En langage Gnuplot, les « tics », ce sont les graduations des axes. Pour chaque axe, il y a les graduations majeures (`tics`), en face desquelles sont affichées des valeurs numériques, et les graduations mineures (`mtics`) qui sont muettes. Ces dernières sont désactivées par défaut, mais nous allons les afficher (grâce à la ligne 10) pour remplir un peu l'espace entre deux graduations majeures de l'axe des abscisses.
- ▶ La commande `set grid` permet d'afficher une grille en pointillés. Par défaut, cette grille ne correspond qu'aux graduations majeures des axes. Ici, nous avons décidé d'inclure les graduations mineures de l'axe des abscisses en rajoutant `ytics, xtics mxtics` en fin de commande.

Il faut enfin noter que nous avons modifié l'échelle d'affichage du graphe en rajoutant les options `[1950:2050][0:]` à la suite de la commande `plot`, ligne 12. Cette échelle est donnée sous la forme `[xmin:xmax][ymin:ymax]` (suivi de `z` dans le cas d'un graphe 3D), toute valeur manquante étant calculée par Gnuplot. Dans l'exemple de la figure 3, nous avons donc décidé de restreindre l'affichage des abscisses aux années 1950 à 2050 (par défaut Gnuplot ajoute un peu d'espace au début et à la fin des courbes), et de forcer la limite `ymin` à 0, de manière à mieux voir les variations absolues.

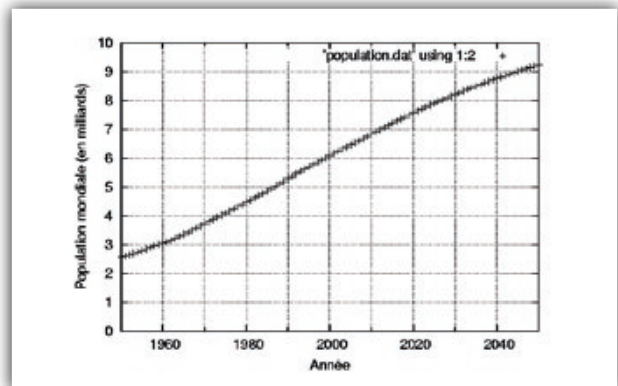


Fig 3 : Modification du style des axes.

### Finitions

Maintenant que les axes sont présentables, passons à la dernière étape de notre travail. Il nous reste à changer les polices du graphique (il est destiné à un support écrit, où une police à empattement est plus lisible), puis à mettre en forme la courbe elle-même.

```
1 # fichier Gnuplot pour tracer l'évolution de la population mondiale
2 # entre 1950 et 2050
3
4 set term post eps enhanced "Times-Roman" 18
5 set clip points
6 set style line 1 pointtype 9 pointsize 1
7 set style line 2 linewidth 2 linetype 2
8 set output "population.eps"
9 set encoding iso_8859_15
10 set xlabel "Années"
```



### Les polices (et leurs variantes) utilisables par défaut dans le terminal postscript

- ▶ Times-Roman, Times-Bold, Times-Italic, Times-BoldItalic ;
- ▶ Helvetica, Helvetica-Oblique, Helvetica-Bold, Helvetica-BoldOblique ;
- ▶ Courier, Courier-Oblique, Courier-Bold, Courier-BoldOblique ;
- ▶ Symbol ;
- ▶ AvantGarde-Book, AvantGarde-BookOblique, AvantGarde-Demi, AvantGarde-DemiOblique ;
- ▶ Bookman-Light, Bookman-LightItalic, Bookman-Demi, Bookman-Demitalic ;
- ▶ Helvetica-Narrow, Helvetica-Narrow-Oblique, Helvetica-Narrow-Bold, Helvetica-Narrow-BoldOblique ;
- ▶ NewCenturySchlbk-Roman, NewCenturySchlbk-Italic, NewCenturySchlbk-Bold, NewCenturySchlbk-BoldItalic ;
- ▶ Palatino-Roman, Palatino-Italic, Palatino-Bold, Palatino-BoldItalic ;
- ▶ ZapfChancery-MediumItalic ;
- ▶ ZapfDingbats.

```

11 set ylabel "Population mondiale (en milliards)"
12 set format y "%.0s"
13 set mxtics
14 set grid ytics xtics mxtics
15 set key bottom right
16 set label 1 "nous {^{e}_n} sommes {/Courier-Bold l\`a}" at 1980, 8.5e9
17 set arrow 1 from 2005,8e9 to 2005, 6.6e9 lw 2
18 plot [1950:2050][0:]'population.dat' using 1:($1=2005?$2:1/0) with lines \
21 title "prévisions" ls 2

```

L'effet de ce script est visible figure 4. Par rapport au listing précédent, nous avons ajouté ou modifié les lignes 4 à 7 et 15 à 21. Ligne 4, nous avons simplement spécifié la police utilisée (*Times-Roman*, suivie de sa taille). Le nom peut être choisi parmi les 35 polices et variantes de base Postscript données dans l'encadré 1, mais il est possible d'en utiliser d'autres, comme *Computer modern*, qui est utilisée par défaut pour les documents LaTeX (voir `help postscript fontfile` pour plus de détails). Avec le changement d'échelle horizontale que nous avons effectué, certains points se retrouvent à cheval sur les limites du graphe (voir figure 3). Pour éviter cela, nous utilisons la commande `set clip points` (ligne 5) qui a pour effet de supprimer les points qui dépassent. Ligne 6 et 7, nous définissons deux styles de ligne (les types 1 et 2). Une ligne possède quatre caractéristiques : le type de point (`pointtype`), le type de trait (`linetype`), la taille du point (`pointsize`) et la largeur de ligne (`linewidth`). Comme vous pouvez le voir dans l'exemple, il est possible de ne définir qu'une partie de ces paramètres, les autres prenant alors leur valeur par défaut.

Pour obtenir la liste des types de traits et de points offerts par un terminal donné, ainsi que leurs numéros correspondants, vous pouvez utiliser le fichier script `ps_symbols.gpi`, livré avec Gnuplot [3]. Il génère, lorsque vous l'exécutez par `gnuplot ps_symbols.gpi`, un fichier Postscript semblable à celui de la figure 5. Vous pouvez bien entendu modifier ce script, destiné par défaut au terminal postscript monochrome, pour l'adapter aux terminaux qui vous intéressent. En langage Gnuplot, `key` est la légende. La ligne 15 permet de déplacer la légende en bas à droite du tracé (`bottom right`), de manière à ce qu'elle n'interfère pas avec la courbe. `help key` vous permet de connaître les autres mots-clés de positionnement, et si vous ne voulez pas de légende, il suffit d'utiliser la commande `unset key` ! Le contenu de la légende est défini plus bas.

Il est possible de rajouter des éléments graphiques sur le tracé. C'est ce que nous faisons avec les lignes 16 et 17 : ajout d'une zone de texte (`label`), et d'une flèche (`arrow`). Plusieurs remarques :

- ▶ Un numéro (optionnel, ici 1) est attribué à la zone de texte et à la flèche. Ce numéro sert d'identifiant et permet des manipulations ultérieures (cacher, déplacer, modifier les attributs de l'élément) ;
- ▶ La position des éléments (instruction `at` pour la zone de texte, `from` et `to` pour la flèche) peut se faire dans plusieurs systèmes de coordonnées. Par défaut, il s'agit des coordonnées affichées par les axes. C'est très pratique dans notre cas puisque si l'on modifie l'échelle de la figure, notre flèche pointera toujours au bon endroit ! Il est également possible d'utiliser des coordonnées relatives à la zone de tracé (et donc indépendantes des échelles) ou relatives à la figure complète (directives `graph` et `screen` respectivement). Comme d'habitude, tapez `help coordinates` pour plus de détails ;
- ▶ Bien d'autres paramètres peuvent être ajustés, comme la pointe de flèche, la justification du texte, etc. Ici, nous avons décidé d'utiliser un trait épais pour la flèche, et donc modifié le paramètre `linewidth` (`lw` en abrégé) en conséquence.

Les chaînes de caractères du graphe peuvent recevoir des directives de mise en forme. C'est le cas ici pour le `label` : nous avons localement changé la police du « l\`a » et placé « e<sub>n</sub> » en exposant. Les instructions de mise en forme sont encadrées par des accolades, et le nom de police est précédé d'un « / ». Le changement de police est surtout utile pour afficher les caractères grecs contenus dans la police *Symbol*.





## Encadré 2 : Fichier permettant d'obtenir la figure 6

```

1 #fermeture d'un transistor MOS de puissance
2 set terminal postscript eps enhanced color solid "Times-Roman" 20 portrait
3 set grid
4 set style data lines
5
6 set mxtics 5
7 set mytics 5
8 set line style 1 lt 10 lw 3
9 set line style 2 lt 3 lw 3
10 set border 15 lt 7 lw 4
11
12 set arrow 1 from first 11e-6,graph 1 to first 11e-6,graph 0 ls 2 nohead
13 set label 1 "{t_0}" at first 10.95e-6,graph 0.9 center
14 set arrow 2 from first 11.04e-6,graph 1 to first 11.04e-6,graph 0 ls 2 nohead
15 set label 2 "{t_1}" at first 11.09e-6,graph 0.9 center
16 set arrow 3 from first 11.2e-6,graph 1 to first 11.2e-6,graph 0 ls 2 nohead
17 set label 3 "{t_2}" at first 11.25e-6,graph 0.9 center
18 set arrow 4 from first 11.44e-6,graph 1 to first 11.44e-6,graph 0 ls 2 nohead
19 set label 4 "{t_3}" at first 11.49e-6,graph 0.9 center
20
21 set output "multiplot.eps"
22 set multiplot
23     set bmargin 4
24     set size 1,0.4
25     set origin 0,0.6
26     set key bottom right
27     set format x ""
28     set format y "%3.0f"
29     unset xlabel
30     set ylabel "Tension grille source (V)"
31     plot [10.8e-6:12e-6][-3:20]"temporel.txt" using 1:3 title "simulation" ls 1
32
33     set origin 0,0.3
34     unset key
35     set ylabel "Tension drain source (V)"
36     plot [10.8e-6:12e-6][0:30]"temporel.txt" using 1:2 ls 1
37
38     set ylabel "Courant de drain (A)"
39     set format x "%3.1s%c"
40     set xlabel "Temps (s)"
41     set origin 0,0
42     plot [10.8e-6:12e-6][-10:130]"temporel.txt" using 1:(-$4) ls 1
43 unset multiplot

```

Continuons à détailler le script ci-dessus. Lignes 18 à 21, nous retrouvons la commande `plot`, qui a pris un peu d'embonpoint par rapport aux exemples précédents. L'objectif poursuivi ici est de tracer de manière différente les données issues de recensements (celles qui sont antérieures à 2005), et celles estimées (postérieures à 2005).

Gnuplot n'accepte qu'une seule occurrence de `plot` par graphe. Il est par contre possible, lorsque l'on veut tracer plusieurs courbes sur un même graphe, de les enchaîner sur une même ligne de la manière suivante :

```
plot 'fichier1.dat' [options], 'fichier2.dat' [options],...
```

C'est de cette façon que nous avons procédé dans l'exemple. Notez que lorsque l'on réutilise plusieurs fois le même fichier de données, il n'y a besoin de le spécifier qu'une fois, puis de remplacer les appels suivants par `'` (comme dans notre exemple). Il est également possible de mêler tracé de données et de fonctions mathématiques. Dans le cas qui nous intéresse, nous allons effectuer un petit traitement lors de la lecture du fichier de données de manière à obtenir deux jeux

de données, l'un antérieur à 2005, l'autre postérieur. Pour cela, nous utilisons la notation `using 1:($1)`. La présence de parenthèses indique une fonction à évaluer. À l'intérieur de ces parenthèses, les numéros de colonne du fichier doivent être précédés d'un « \$ ». Le point d'interrogation et les deux points correspondent à la structure `if/then/else`. La condition peut donc s'exprimer comme suit : si la valeur dans la colonne 1 est inférieure à 2005, on renvoie la valeur correspondante dans la colonne 2, sinon, on renvoie la valeur « I/O », qui correspond à une absence de valeur numérique (NaN, ou Not a Number).

Nous attribuons ensuite un des styles de ligne définis plus haut (lignes 6 et 7 du script) aux deux tracés de la commande `plot`, ainsi qu'un titre (qui apparaît dans la légende). Enfin, nous spécifions que la seconde courbe doit être tracée en utilisant une ligne (`with line`) plutôt qu'avec des points (valeur par défaut). Il est également possible de mêler ligne et points avec `with linespoints` et encore bien d'autres possibilités (la liste complète avec `help with`). Enfin, dernier raffinement, nous avons réduit le nombre de points utilisés pour tracer la première courbe, de manière à ce qu'ils ne se chevauchent pas. Pour cela, nous avons fait appel à la directive `every 3` (ligne 19) pour ne tracer qu'un point sur 3.

### J'veux de la couleur !

Nous nous sommes limités jusqu'ici à la génération de figures monochromes, comme la plupart de celles que l'on utilise

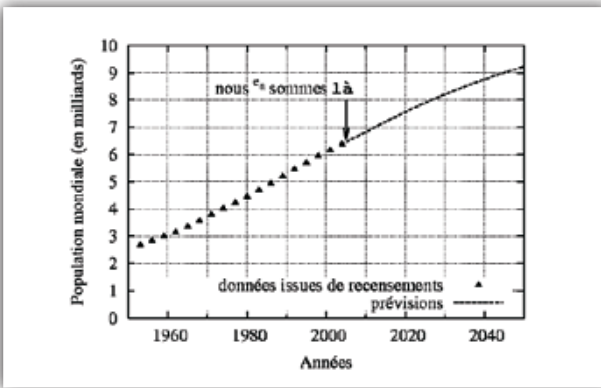


Fig. 4 : Notre figure terminée.



Fig. 5 : Les types de traits et de points du terminal postscript.

sur un document papier. Dans certains cas, l'utilisation de la couleur peut être intéressante (pour un affichage écran ou une présentation au vidéo-projecteur). Il suffit, pour en bénéficier dans l'exemple précédent, de modifier le terminal (ligne 4) comme suit :

```
set term post eps enhanced color solid "Times-Roman" 18
```

La directive `solid` est utilisée pour que les styles de ligne fassent appel à la couleur plutôt qu'aux pointillés. C'est en effet à notre sens la plus grande faiblesse de Gnuplot : les couleurs de ligne et les styles de pointillés ne sont pas indépendants. Il n'est pas possible de spécifier « trait bleu en pointillés » et « trait rouge continu ». La solution la plus simple consiste à n'utiliser les pointillés qu'en terminal monochrome et ne jouer que sur les couleurs de traits en terminal couleur. Même si elle est peu satisfaisante, cette solution se révèle en général peu contraignante.

### Tracé plus complexe

Pour finir ce tour d'horizon de Gnuplot, nous vous proposons de décortiquer la figure 6, qui regroupe plusieurs concepts importants. Cette figure représente l'évolution de trois grandeurs électriques d'un transistor MOS durant sa fermeture. Ces trois grandeurs étant différentes (deux tensions et un courant) mais synchronisées, il est intéressant de les tracer dans trois graphes superposés. Nous allons pour cela utiliser l'environnement `multiplot` qui permet de faire appel plusieurs fois à la commande `plot` dans une même figure. Attention, cette commande sert à tracer plusieurs graphes (qui peuvent se superposer partiellement ou totalement) dans une même

figure. Pour tracer plusieurs courbes dans un seul graphe, il faut utiliser la commande `plot` (voir lignes 18 à 21 dans l'exemple précédent). Le code permettant d'obtenir la figure 6 est donné dans l'encadré n°2. Regardons-le de plus près. Ligne 2, nous faisons appel au terminal postscript couleur, et nous définissons une utilisation verticale (`portrait`) de la figure. La ligne 4 permet de spécifier de manière globale que les données seront tracées par des lignes. Il n'y aura donc pas besoin de spécifier `with lines` lors de chaque appel à `plot` comme dans les exemples précédents.

Avec les lignes 6 et 7, nous affichons les graduations secondaires des axes d'abscisse et d'ordonnée, et nous en définissons le nombre entre les graduations principales (5). Nous définissons ensuite (lignes 8 et 9) deux types de lignes de tracé (la propriété `lt` étant la version abrégée de `linetype`). La ligne 10 permet de modifier le style de la bordure des graphes. L'utilisation de la commande `set border` est un peu spécifique, car elle est suivie d'un nombre définissant les côtés auxquels elle s'applique (15 pour tous les côtés, `help border` pour la liste complète de ces nombres). Le reste de la commande est simple, avec le choix d'un type (7 pour le noir dans le terminal postscript couleur), et d'une largeur de trait. Lignes 12 à 19, nous définissons quatre flèches et zones de texte (correspondant aux traits bleus et aux étiquettes «  $t_x$  » sur la figure 6). Nous avons utilisé ici un mélange de coordonnées : les coordonnées horizontales sont prises relativement à l'échelle des graphes (mot-clef `first`), alors que les coordonnées verticales sont prises relativement au graphe lui-même (mot-clef `graph`). De cette manière, les traits resteront toujours synchronisés aux tracés (puisque'ils utilisent la même échelle horizontale), et ils couvriront la totalité de la hauteur du graphe (puisque leurs coordonnées verticales sont liées à ce dernier). Notez également la directive `nohead` qui permet de tracer des flèches sans pointes (pour faire des traits !).

Nous passons ensuite dans l'environnement `multiplot` (lignes 21 à 44).

Pour tracer nos trois graphes sur la même figure, il faut bien entendu réduire leur taille. C'est le rôle de `set size`, ligne 24. Attention, les valeurs numériques définissant la taille sont normalisées par rapport à la figure complète. Il faut donc lire la ligne 24 comme « La largeur d'un graphe sera égale à la largeur de la figure (1), et sa hauteur sera de 0,4\*la hauteur totale de la figure ».



### Si vous avez du mal avec les positions et les positions des graphes...

Lorsque vous allez commencer à jouer avec les tailles ou les positions des graphes, il vous arrivera fréquemment de « sortir du cadre », c'est-à-dire que certains éléments du graphe dépasseront et seront coupés à l'affichage. S'il est préférable de comprendre le fonctionnement de Gnuplot, dans certains cas désespérés, vous pouvez faire appel à `ps2eps`. Véritable caisse à outils du Postscript, cet utilitaire (en ligne de commande) permet – entre autres – de recalculer les limites d'une figure (pour que tous les éléments qu'elle contient soient visibles). Pour recalculer ces limites (appelées « Bounding Box »), il faut utiliser l'option `-B:ps2eps -B ma_figure.ps` (un fichier eps peut également être utilisé).

Il ne suffit pas que les graphes aient la même taille pour qu'il soit possible de les aligner. En effet, pour Gnuplot, la taille que nous avons spécifiée correspond à un rectangle englobant tous les éléments du graphe, c'est-à-dire la zone de tracé, mais également les échelles, les graduations (si elles sont à l'extérieur de la zone de tracé), les titres d'axes et de graphe. En fait, Gnuplot considère que la taille du graphe est égale à la taille de la zone de tracé plus des marges de chaque côté. Pour pouvoir aligner les graphes, il faut donc qu'ils aient la même taille et les mêmes marges. Par défaut, les marges sont calculées automatiquement par Gnuplot. Si nous prenons l'exemple de la marge gauche des trois graphes, nous voyons que sa taille est définie par la présence d'un titre d'axe et des valeurs numériques d'échelles. Cependant, les trois échelles ne sont pas identiques. Les deux premières n'ont que deux chiffres significatifs, alors que la troisième en a trois. Pour forcer Gnuplot à considérer le même espace pour chacune des trois échelles, nous avons spécifié un format d'affichage à trois chiffres (ligne 28 dans l'encadré n°2). De cette manière, l'espace nécessaire est réservé par Gnuplot, et les marges gauches ont toutes la même largeur.

La marge basse présente un autre problème, puisque le graphe du bas possède un titre d'axe et une d'échelle, ce qui n'est pas le cas des deux autres. Dans ce cas, la méthode la plus simple consiste à forcer la largeur de la marge. C'est le but de la commande `set bmargin 4` (ligne 23). On définit ainsi une valeur de marge du bas (`bmargin` pour « bottom margin », `help margin` pour la liste) de 4 caractères dans la police courante. La valeur de 4 a été choisie par essais successifs. Cette méthode n'est cependant pas très robuste, puisqu'elle dépend de la taille de police choisie pour le graphe. Enfin, dernière étape pour aligner les graphes, il faut les positionner dans la figure. Nous définissons pour cela l'origine de chaque graphe (`set origin` avant chaque appel à `plot`). Comme pour `set size`, les valeurs des coordonnées sont normalisées par rapport à la figure, avec le point 0,0 situé en bas à gauche.

Dans cette figure, nous avons également utilisé la commande `unset`, qui permet d'annuler l'action d'un `set`. On retrouve `unset` pour terminer le `multiplot` (ligne 44), mais aussi pour masquer la légende (ligne 34).

En effet, lorsqu'un élément est modifié ou créé par `set`, il reste en mémoire jusqu'à ce qu'il soit supprimé (`unset`) ou modifié (autre `set` sur le même élément). C'est ce qui se passe avec les quatre traits et zones de texte, définis lignes 12 à 19, et qui apparaissent sur les trois graphes. En ce qui concerne la légende, on la fait apparaître sur le premier graphe (ligne 26), puis on la supprime (ligne 34), pour ne pas la voir sur les deux autres graphes. On aurait aussi pu utiliser ce comportement de la commande `set` pour les échelles des graphes :

- ▶ L'échelle sur l'axe des abscisses étant la même pour tous, on aurait pu ne l'indiquer qu'une fois à l'aide de la commande `set xrange [10.8e-6:12e-6]` ;
- ▶ On aurait pu ensuite indiquer l'échelle des ordonnées de chaque tracé en insérant la commande `set yrange [...]` avant chaque appel à la commande `plot`.

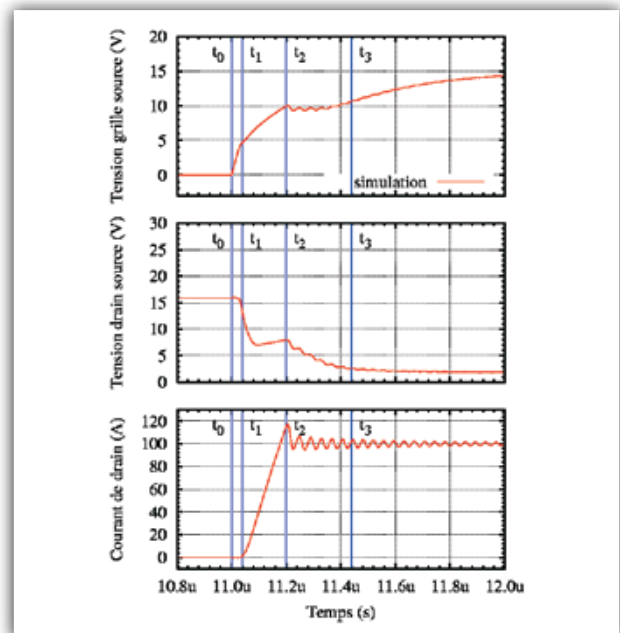


Fig. 6 : Une figure faisant appel à `multiplot`.

### Toujours plus loin...

Nous arrivons au terme de cette introduction. Le but était ici de voir comment utiliser Gnuplot pour obtenir des figures à inclure dans un document LaTeX, et nous avons donc mis



l'accent sur la mise en forme des graphes. Gnuplot offre encore bien d'autres possibilités, comme les graphes 3-D, les graphes bâtons, l'ajout de barres d'erreurs, des fonctions de « fitting » de données...

Nous espérons qu'à l'issue de cet article, vous serez armés pour affronter la documentation de Gnuplot qui est très complète, mais un peu rebutante au premier abord. Elle est disponible sous trois formes (avec le même contenu) : en tapant `help` après avoir lancé Gnuplot, sous forme de fichier PDF (livré avec Gnuplot, ou disponible sur [www.gnuplot.info](http://www.gnuplot.info)), et au format HTML sur le même site.

## Bibliographie

Le site de Gnuplot, [www.gnuplot.info](http://www.gnuplot.info), propose la documentation complète, ainsi qu'un grand nombre d'exemples.

De nombreux autres exemples sur un site très complet (même s'il ne s'agit pas de la dernière version de Gnuplot) : <http://t16web.lanl.gov/Kawano/Gnuplot/index-e.html>

Vous trouverez dans Gnuplot plusieurs fichiers de documentation, comme le manuel, la liste des styles de ligne et de points disponibles [3], mais aussi un guide très court des commandes typographiques : `ps_guide.ps`.

Enfin, pour les cas désespérés, vous pouvez poser des questions sur le newsgroup Usenet `comp.graphics.apps.Gnuplot` (accessible par exemple via Google groups ou un

lecteur dédié). Sur ce groupe de discussion anglophone, les développeurs de Gnuplot sont très actifs.



## NOTES

- ▶ [1] Des interfaces graphiques existent pour les allergiques à la ligne de commande. Certaines sont recensées sur la page <http://www.Gnuplot.info/links.html> à la rubrique **front-ends**.
- ▶ [2] Le paquetage LaTeX `psfrag` permet de remplacer des chaînes de caractères d'une figure Postscript. Il est donc également possible de modifier la typographie d'une figure générée par le terminal postscript.
- ▶ [3] Pour une Mandrake 10, ce fichier se trouve dans `/usr/share/doc/Gnuplot-4.0.0/psdoc/ps_symbols.gpi`.

Cyril Buttay, Florent Morel,

# 2 SITES INCONTOURNABLES

Toute l'actualité du magazine sur :

[www.gnulinuxmag.com](http://www.gnulinuxmag.com)



Abonnements et anciens numéros en vente sur :

[www.ed-diamond.com](http://www.ed-diamond.com)