

本稿は [Linux Japan 誌](#) 2000 年 4 月号に掲載された記事に補筆修正したものです。

L^AT_EX と使う図面作成ツール

図やグラフやフローチャートは理工系の文章では非常に大切な構成要素です。ご存知の通り L^AT_EX には図を描くための `picture` 環境がありますが、機能は貧弱です。そこで、図版は外部ツールで描き、Unix 系の事実上の標準である PostScript 形式に変換して L^AT_EX に取り込むという方法が一般的となっています。PostScript ファイルは平テキストですから、その気になればエディタで編集をして... いくら CUI 好きの筆者でも、それは無謀と心得ています。

ところで、図を表現するには大きく 2 つの方法があります。一つは、図をまず構成するオブジェクトに分解し、オブジェクトを属性ベクターで表現するものです。いわゆるドローイング系のツールはこの考えに基づいて設計されています。ツールはそれぞれ固有の内部形式で保存しますが、一般に PostScript 形式、あるいは、ページの中の図形領域が指定できる EPS 形式で出力する機能が備わっています。拡大縮小が自由であり、ファイルサイズも概して小さいという利点がありますから、出来る限りこの方式で図を描くのが得策といえます。無数にあるツールの中から、筆者が普段使っている、Gnuplot、Xgraph、GNU Plotutility、Tgif、Pstoedit を取り上げて紹介します。もちろん、その全てを紹介するなどということは無理です。PostScript 形式出力の方法は必ず説明します。それと筆者の独断と偏見で面白いと思われる機能があれば紹介します。なお、たまたま使っているだけです。これが一番などと主張する気は毛頭ありません。FreeBSD ではなく、偶然 Linux を使うようになっていたというのと同じような話です。

もう一つは、図を点の集合 (=ビットマップ) として扱う方法で、いわゆるペイント系のツールが使われます。絵画や写真など、イメージと呼ばれるものは、図を構成するオブジェクトが点であるといえます。その場合には、わざわざ点オブジェクトに分解せずに、最初からビットマップとして保存すべきでしょう。筆者は絵の才能がないので最初から画像を書き起こすことはほとんどしません。写真に説明を入れたり、背景用の小さな Pixmap や Web の LOGO を作成するなど、素材を組み合わせるという場合がたまにあります。具体的には、古くからの定番 Xpaint(Xart)、Pixmap、また今や無視するわけにはいかない Gimp を使ったりしています。これらのツールについては筆者

が使い込んでないこともあり、名前だけの紹介でとどめます。さて、ビットマップは EPS 形式に変換する必要があります。画像変換ツール (群) として ImageMagick や Netpbm ははずせない逸品です。これらについてはバッチ処理的に使う場合があるので、少々説明します。

graphicx.sty の使い方

L^AT_EX への EPS ファイルの取り込み方について簡単に復習しておきましょう。L^AT_EX209 の場合、EPS ファイルを取り込むスタイルファイルは、DVI から PS への変換ツール (`dvi2ps`, `jdvi2kps`, `dvips`) との相性に応じて使い分けなければなりません。L^AT_EX2e では、`graphic` パッケージとして整理されて `graphics.sty` あるいはその拡張版 `graphicx.sty` を使います。プリアンブルに

```
\usepackage[dvips]{graphicx}
```

と宣言しておきます。そうして、図を入れたいところで、次のように記述します。

```
\includegraphics[options]{epsfile}
```

`options` には幅 (`width`)、高さ (`height`)、拡大率 (`scale`) に加えて、回転 (`angle`) も指定できるようになっています。図 1 は、“`angle=30`” を設定した場合の実行例です。



図 1 L^AT_EX2e の `graphicx` を使った図の回転: `angle=30` を指定

なお、DVI から PS への変換は宣言にあるように `dvips` を使しましょう。`dvi2ps` でも一応 PS に変換で

きますが、回転などは正常に処理されません。

Gnuplot

Gnuplot は多機能なグラフ描画ツールです。基本的には関数を 2 次元 3 次元プロットしますが、データを数値処理してプロットする機能もあります。また、スクリプトを用意して、バッチ处理的に使うことも可能ですから実験データをまとめてグラフ化する時など非常に便利です。出力形式が多様なことも大きな特徴で、もちろん PostScript 形式出力も備えています。Vine でも Plamo でもパッケージされていますからすぐに試すことができると思います。ソースが CTAN で配布されていますから、最新版の 3.7.1 を取ってきてコンパイルすることも簡単でしょう。

PS 形式での出力

出力は端末 (terminal) として扱われ、それに合った出力形式を設定するには、

```
set terminal terminal
```

とします。また、出力先設定は

```
set output "filename"
```

とします。結局 PostScript 形式で *psfile* に保存するには

```
set term postscript [color][eps]
set out "psfile"
```

と設定した後、描画命令 `plot`, `splot`, `replot` を実行することになります。なお、Gnuplot では認識できる限り、コマンド引数の省略形を使うことが許されていますから、`term`, `out` と記すことが多いです。

リスト 1 psdemo.gp

```
R=3*pi
set xrange[-R:R]
set yrange[-R:R]
set isosamples 30,30
set ticslevel 0
set hidden3d
splot sin(x)*sin(y)/x/y
set term postscript eps color
set out "psdemo.eps"
replot
!gv psdemo.eps
```

リスト 1 のようなスクリプトを

```
gnuplot psdemo.gp
```

と実行してみて確かめてください。11 行目で、gnuplot 内部から `gv` を起動して “psdemo.eps” という EPS ファイルが実際に出来上がっていることを確かめています。その他の項目については、gnuplot を起動して、

```
gnuplot> help 項目
```

のように `help` を使って調べてみてください。

Computer Graphics Metafile

Gnuplot は様々な端末出力形式を備えていますので、中には初めて知ったというものもあります。実は CGM という ANSI 標準の形式を筆者は扱ったことがありませんでした。公的な標準ですから、サポートするツールがあってもよさそうなのですが、非営利業界団 [1] [W³](#) は結成されているものの、フリーなものはなかなか見付かりません (なんと `help` によれば MS Word 6.0c が読み込めるとのこと)。一つだけ `gplot` [2] [W³](#) という古い (1991 年とあります) CGM のユティリティ (開発は停止しています) を探し出しました。x11 用のデバイスドライバ `x1,xws` を含むように Makefile を変更して `gplot` を作成します。こうして、カラーマップ機能がある x11 のデバイス `xws` をオプション `-d` で指定して CGM ファイルを表示させることができます。

```
gplot -dxws -t cgfile
```

`-t` は端末デバイスであることを指示するオプションで、これを指定しないとすぐに終了してしまいます。

また、出力デバイスとしてカラー PS (psc) が備わっていますから、配付に含まれる車の画像 “car.cgmb” を 90 度回転させて PS ファイルに変換してみましょう。なお、拡張子 `cgmb` は `egm` の圧縮バイナリ形式を表していて、このバージョンではデフォルトです。平テキストは拡張子 `cgmc` を付け、オプション `-C` を指定して読み込ませます。

```
gplot -dpssc -r90 car.cgmb > car.ps または
gplot -dpssc -r90 -C car.cgmc > car.ps
```

図 2 は `gplot` で変換した PS 画像を実際に取り込んだものです。

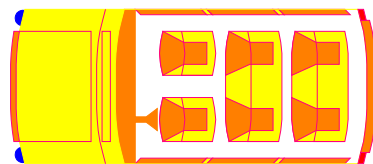


図 2 `gplot` で PS に変換した車の画像

Tektronix グラフィック端末

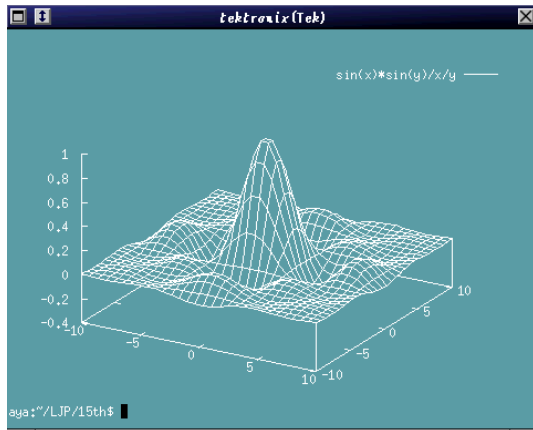


図 3 xterm の Tektronix グラフィックモード

tek** という端末名を見て思い出話がしたくなったのでつき合ってください。筆者が昔勤めていた研究所は資金が潤沢だったので、外国製の高級 (=高信頼性) 計測器を使っていました。で、シンクロスコープと言えば、もう Sony-Tektronix で決まりでした。筆者は使ったことはないのですが、グラフィック端末市場でも Tektronix は業界標準だったのでしょう。xterm で VT options メニューに “Show Tek Window”, “Switch to Tek Mode” という項目があり、その名残りを知ることができます (Kterm では残念ながら? コンパイル時のデフォルトオプションからはずれたようです)。Gnuplot で terminal を tek40xx などに設定して、直接 Tek 端末に書き出せばグラフが端末自身に表示されます。これでちょっと遊んでみましょう。まず xterm をオプション -t により、Tektronix グラフィック端末モードで起動します。画面サイズと背景色と小さめのフォントを使うオプションもついでに指定しましょう。

```
xterm -t %480x360 -fg white -bg cadetblue
      -xrm "*tek4014.fontLarge:6x13"
```

そして、リスト 2 のような gnuplot のスクリプトを Tek 端末上で

```
gnuplot tekdemo.gp
```

のように実行すれば (gnuplot から端末に直接書き出しています) 図 3 のような結果が得られます。

リスト 2 tekdemo.gp

```
set isosamples 30,30
set hidden3d
set ticslevel 0
set term tek40xx
set out "|cat "
splot sin(x)*sin(y)/x/y
```

vt102 の端末エミュレータ rxvt のマニュアルに、この Tektronix エミュレーションを省くことで、xterm に比べて軽くなったと記載されています。ほとんど使うことが無くなったのですね。

Dumb 端末

話は PostScript からどんどんそれますが、ダム (dumb) 端末も懐かしいです。キャラクター端末にグラフを表示させるので、かなり無理がありますが、学生の頃の計算機実習を思い出します。リスト 3 の gnuplot スクリプトを xterm や rxvt 上で実行させてみましょう。

リスト 3 dumbdemo.gp

```
set term dumb
set out "|cat "
plot sin(x)
```

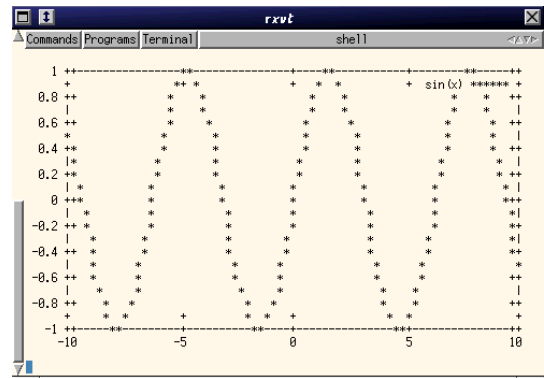


図 4 ダム端末での無理矢理グラフ表示

図 4 に実行例を示します。もの悲しい雰囲気は漂っていいですね。このような図でも定性的な理解には十分だった気がします。

パイプ接続

ところで、データ列をパイプで流し込んでグラフを表示させるという Unix 流の使い方を Gnuplot で行おうとすると、ちょっと工夫が要ります。gnuplot では、標準入力からのデータ読み込みを

```
plot "-" あるいは splot "-"
```

で行うことができます。が、パイプで接続して使うような場合には、表示を保ったままにするコマンド pause -1 が効かなくなるので、すぐに終了してしまうのです。

そこで、起動時オプション `-persist` を指定し、表示を保たせなければなりません。 x^2 を出力する awk スクリプト `xg1.awk`(リスト 3) を作成して、実行ビットを立てましょう。

```
chmod +x xg1.awk
```

そして、bash の group command 機能を利用して、次のように実行します。

```
(echo 'plot "-" ; xg1.awk) |gnuplot -persist
```

`xg1.awk` の中にコマンド “`plot "-"`” を埋め込むこともできますが、それでは gnuplot に特化することになりますから、勧められません。

リスト 3 `xg1.awk`

```
#!/usr/bin/awk -f
BEGIN{
  for (i=1; i <=100; i++){
    x = i*0.01;
    printf "%f\t%f\n",x,x*x;
  }
}
```

まだまだあるぞ gnuplot

筆者は gnuplot がいたく気に入っているのので、拙著『Linux 活用術』で 1 章を割いて紹介しています。また、Web サイト [3][W³] も開設していますので、ご覧ください。また本誌 2 月号 p.166 にも紹介されていますが、『gnuplot パーフェクト・マニュアル』という書籍も出ています。解説が一冊書けるくらい多機能なのです。

Xgraph

Xgraph は Unix の教科書には必ず顔をだす老舗の 2 次元グラフ作図ツールです。 x - y データの並びを標準入力から読み取って(ここが Unix らしい)、最大値最小値を求めて作図範囲を決定し、図を描きます(従って、データ全てを読み終えてから描画します)。作者は David Harrison(davidh@ic.Berkeley.EDU) 氏とありますが、Christos Zoulas(christos@ee.cornell.edu) 氏の `tgif` 追加の patch を基に、William C. Cheng 氏(Tgif の作者) [4][W³] が、EPS 出力を改良した X11R6 用の patch を作成してますから、それを使いましょう。オプションがいろいろあるので、一度

```
xgraph --help
```

とでもして眺めてください(`-help` というオプションはないのですが、`unknown` と判断されて詳細な Usage が

表示されます)。さて、PS ファイルの作成方法ですが、2 通りの方法があります。通常は図 5 のように、`xgraph` の画面にグラフを描いて確認し、上部左にあるメニューから `Hardcopy` を選択し、`Hardcopy` のダイアログで必要な設定をしてファイルに出力します。もう一つの方法は、コマンドライン上で

```
xgraph -device ps -o epsfile
```

と処理する方法です。

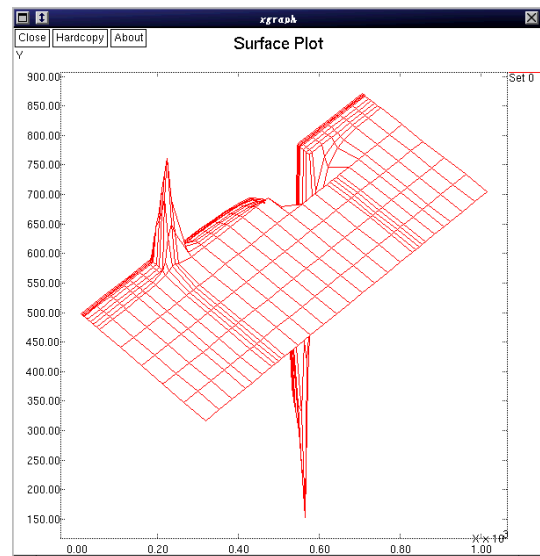


図 5 `xgraph` の実行画面：配付 `examples` の中の `surface.xg`

`xgraph` の文法のうち、2 つほど覚えておくと良い事柄があります。まず、空行でデータがブロック化されるという事、また線で結ばずにカレントポイントを移動する “`move x y`” 命令がある事です。これだけで随分と作図の幅が広がります。まあ、もっともこれは大概のプロッターに共通の基本的な事柄でしょうが、また、`xgraph` ではマウス左ボタンを使って画面の一部を選択し、拡大表示することができます。これはかなり使える便利な機能です。

そうですね、せっかくですから awk スクリプトを作成して遊んでみましょう。先程の `xg1.awk` に加えて \sqrt{x} を出力する `xg2.awk`(リスト 4) を作成し、忘れずに実行ビットを立てましょう。この 2 つのグラフを `xgraph` で一つの画面に表示するには、またまた bash の group command 機能を利用して、

```
(xg1.awk; echo; xg2.awk) |xgraph -P
```

とします。`echo` の役割は重要で、空行を入れてデータをブロック化しています。なお、`-P` はデータ点を描くオプションです。

リスト 4 xg2.awk

```
#!/usr/bin/awk -f
BEGIN{
  for (i=1; i <=100; i++){
    x = 0.01*i;
    printf "%f\t%f\n",x,sqrt(x);
  }
}
```

GNU Plotutils [5] [W3](#)

GNU plotutilities の初期の歴史についての文書 ONEWS によれば, GNU のものは Unix のユーティリティを真似て作成されています. Unix では graph, plot, spline という作図ツールが存在し, デバイス依存性を吸収した標準ライブラリ libplot も開発されていました. 初めて実装された表示デバイスは Tektronix 611 というストレージスコープ (残像型のオシロスコープのこと) だったそうです. 1989年に, 初めての GNU バージョンの graph, plot, tek2plot, spline, double が書かれました. その後, 何度かの全面書き直しを経, 常微分方程式の解法ツール ode などを加えて, 1997年 バージョン 1.1 がリリースされたのです. 最新は 2.3 であり, デバイスは, tek(Tektronix), X, pnm, gif, ai(Adobe Illustrator), ps, fig(xfig), pcl, hpgl, cgm, meta(GNU graphics metafile) がサポートされています.

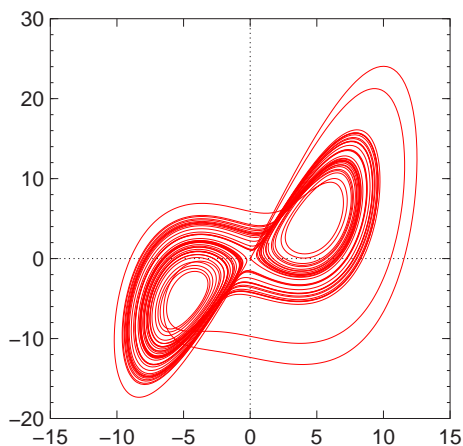


図 6 GNU plotutility の ode と graph による Lorenz 方程式の数値解

中心的なコマンド graph は,

```
graph -T X
```

のように端末名を指定して起動します. ode の examples からカオス解を示すことで有名な Lorenz 方程式

を例にして図を書かせてみましょう. ode による数値解を lorenz.ode という ode のソースファイルを用いて解き, その出力 (x-y データ列) をパイプで graph に渡します. すなわち,

```
ode <lorenz.ode |graph -T ps -C >lorenz.ps
```

のように実行して PS ファイルに変換できます (図 6). -C は color を使うオプション指定です (デフォルトは monochrome).

Xgraph のところで作成した awk スクリプトにより出力されるデータを graph に描かせることももちろんできます.

```
(xg1.awk; echo; xg2.awk) |graph -T X -C
```

Tektronix と CGM

ちょっとそれますが, graph では, 端末が xterm 互換の場合に Tektronix 端末モードの補助窓を開いてそこに描画します. したがって, 予め Tektronix モードに切替える必要はありません. VT102 モードの xterm 上で

```
echo 0 0 1 1 2 0 3 1 |spline |graph -T tek
```

として, spline も一緒に使って確かめてください. ついでながら, tek2plot という Tektronix 端末形式のファイルを変換するツールも含まれていて, /usr/local/share/tek2plot にサンプルがインストールされます. usmap.tek は米国の地図データでした.

また, CGM はバージョン 3 のフォーマットで出力されます. 開発の停止している gplot はバージョン 1 までしか扱えません. graph が吐き出す CGM をバージョン 1 に制限するには, 環境変数 CGM_MAX_VERSION を 1 に設定します.

Gnuplot の table 形式

graph には入力データ形式を指定するオプション “-I [a|f|d|i|g]” があります. デフォルトの ascii 文字列データ (a) 以外にも, 実数 (f,d) や整数 (i) のバイナリデータを読むことができます. 都合の良いことに, Gnuplot の数値データ出力形式 table (g) もサポートされています. すなわち, “-I g” とします.

```
set term table
set out "fraun.table"
plot (sin(x)/x)**2
!graph -TX -Ilg < fraun.table
```

を gnuplot で実行して確かめてください。

ドキュメント

man ではなく info ドキュメントがインストールされますから、端末上で

```
info plotutils
```

により情報を得てください。また、GNU お約束の起動時オプション `--help` も有効ですが、項目が多いので、やはり info のお世話になるのが一番でしょう。

Real Time?

graph の info には、『graph -T X 等には他のプロットングプログラムにない特徴がある。すなわち、パイプでデータを読み込み real time で表示することができる』という一文があります。パイプ接続は紹介しましたが、real time というのは確かに珍しい機能なので確かめてみましょう。まず real time に描画するには、x, y 領域が決定されていなければなりません。次のように描画範囲を指示をして描かせてみますと、

```
ode <lorenz.ode |graph -T X -x -15 15 -y -20 30
```

なるほど、少しずつ曲線を描きます。が、どうもギクシャクしています。どこかでバッファリングしている感じです。info で Plot Option を調べると、どうやら `--max-line-length` が折れ線のデータ点をバッファリングする上限を定めているらしいので、これを 1 にしてみましょう。

```
ode <lorenz.ode |graph -T X -x -15 15 -y -20 30
-C --max-line-length 1
```

いい感じです。この機能（ソースでは filter モードと呼んでいるようです）は、データを real time に表示したい場合には非常に便利ですね。データを発生させて模擬実験したくなります。xgl.awk に手を加えたリスト 5 の内容の awk スクリプトを書いて、

```
gr.awk -v WAIT=100000|graph -TX -x 0 1 -y 0 1
-C --max-line-length 1 -W 0.007
```

と走らせてみましょう。オプション `-W` で線を少し小さくしました、0.07 はグラフの大きさに対する比率です。gr.awk では 8 行目の バッファに溜ったデータを出力

先に吐き出すコマンド `fflush()` が大切です。C 言語にも同じ関数がありますね。7 行目の待時間のためのループはどうもいただけませんが、awk には `sleep()` みたいな関数がないので、ご勘弁を。

リスト 5 gr.awk

```
#!/usr/bin/awk -f
BEGIN{
  for (i=1; i <=100; i++){
    x = i*0.01;
    printf "%f\t%f\n",x,x*x;
    for (j=1; j<=WAIT; j++){
      fflush();
    }
  }
}
```

線種と記号

折れ線を描かないようにするには、線種を指定するオプション `-m arg` で負の整数を指定します。もちろん、その場合には記号を指定するオプション `-S [arg(s)]` を指定して何か描くようにしないとはいけません。どんな線種と記号（標準で）があるかは、データファイル中に `#m=arg,S=arg` で指示が出せますから、リスト 6 の awk スクリプトを作成して次のように走らせると一覧表示されます [6]。

```
linetype.awk |graph -T X -C
```

リスト 6 linetype.awk

```
#!/usr/bin/awk -f
BEGIN{
  for (j=0; j<=4; j++){
    for (i=1; i<=25; i++){
      printf "#m=%d,S=%d\n",i+j*25, i+j*25;
      printf "%f %f\n",j*0.2+0.05,1.0-i/26;
      printf "%f %f\n",j*0.2+0.15,1.0-i/26;
      printf "\n";
    }
  }
}
```

次回は

いやー、このような奥深い？ ツールになると、筆者は異常に凝ってしまうので、話が尽きません。途中でですが、Tgif 以降は次回です。

参考文献

- [1] CGM に関する非営利協議会 . [W³](http://www.cgmopen.org)
<http://www.cgmopen.org>
- [2] CGM のビューア gplot . [W³](ftp://ftp.psc.edu/pub/gplot)
<ftp://ftp.psc.edu/pub/gplot>
- [3] 筆者の「グラフは Gnuplot におまかせ」ページ . [W³](http://ayapin.film.s.dendai.ac.jp/~matuda/Gnuplot/gnuplot.html)
<http://ayapin.film.s.dendai.ac.jp/~matuda/Gnuplot/gnuplot.html>
- [4] William C. Cheng さんの tgif パッチがあたった Xgraph . [W³](ftp://bourbon.cs.umd.edu/pub/tgif/tools/xgraph/)
<ftp://bourbon.cs.umd.edu/pub/tgif/tools/xgraph/>
- [5] GNU Plotutils のページ . [W³](http://www.gnu.org/software/plotutils/plotutils.html)
<http://www.gnu.org/software/plotutils/plotutils.html>
- [6] 雑誌には載せられなかった GNU Plotutils で描ける線種の一覧です .

