

本稿は [Linux Japan 誌](#) 2002 年 05 月号に掲載された記事に補筆修正したものです。

## C 言語で gnuplot を呼ぶ

筆者が原稿を書いている今、この春先は 20 年前から憂鬱な季節となりました。そう、杉花粉症のため、目は真っ赤、鼻ジルジル、息ができないので口を空けて寝るため喉もガラガラなのです。当然、頭もボーッとされていて、ネタが思いつきません。そこで苦しい時の gnuplot<sup>[1]</sup><sup>[W<sup>3</sup>]</sup> 頼みということで、今月もタイトルにあるごとく、gnuplot の活用を紹介しようと思います。本質はシェルのパイプ機能を使ったコマンドとデータの受渡しに過ぎないのですが、C 言語の `popen()` 関数を使うことで gnuplot の起動もソースに含めることができます。したがって、ライブラリ感覚で利用できるわけです。それならグラフィックの標準ライブラリが正道のはずじゃないかと思われた方、貴方は正しいです。実際 CGM (Computer Graphics Metafile - ISO8632)<sup>[2]</sup><sup>[W<sup>3</sup>]</sup> や GKS (Graphical Kernel System, ISO7942,8651,8806)<sup>[3]</sup><sup>[W<sup>3</sup>]</sup> などが有名です。しかし、これらのライブラリを使いこなすようになるにはかなりの時間と忍耐が必要です。また、一般にライブラリは関数ですから、それを用いたアプリケーションのコマンドとは全く様相が異なっています。したがって、アプリケーションを動かしてもライブラリ関数を覚えることはできません。しかし、今回の方法は gnuplot そのもの呼び出すので gnuplot の命令そのままを用いて記述できます。新たに覚える事柄はほとんどありません。非常に手軽です。

### `popen()` 関数

エッセンスは次のようなことに尽きます。

```
FILE *gid;
...

gid=popen("gnuplot","w");
...
fprintf(gid,"gnuplot のコマンド\n");
fflush(gid); <--- 必要に応じて
...
pclose(gid);
```

`gid` という FILE 番号を宣言して、gnuplot を実行するシェルに対するパイプを開くプロセスの FILE 番号を取得し `gid` に代入します。以降、`gid` に対して `fprintf()` により書き込まれた文字列は gnuplot に全て渡ります。適宜 `fflush()` を使ってバッファから強制的に吐きださないと、アニメーションを描く場合などにはぎくしゃくします。開いた FILE はきちんと `pclose()` で閉じるのが作法です。

具体的な例を示します。次の List 1 を次のようにコンパイルして実行してください。

```
gcc -o ex1 ex1.c -lm
./ex1
```

どうですか、波が右に進行しくアニメーションができた筈です。

【リスト 1】 ex1.c

```
#include <stdio.h>
#include <stdlib.h>
#include <math.h>

int main(int argc, char **argv)
{
    FILE *gid;
    int i;
    double x;

    gid=popen("gnuplot","w");
    for (i=0; i<100 ;i++){
        x = 0.03*i;
        fprintf(gid,"plot sin(x-2.370000*pi)\n",x);
    }
    pclose(gid);
}
```

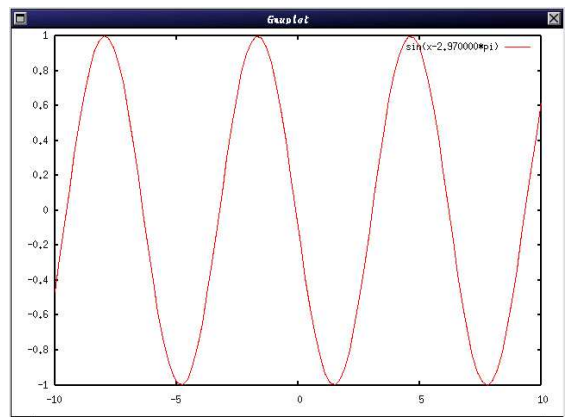


図 1 正弦波が右方向に進行

もう少し複雑なものを描いてみましょう。ex2 はまずバケツのデータを z 軸まわりの回転対称体で表現し、test.dat という名前で作成しておきます。それを、視点を変えながら 3 次元プロットコマンド `splot` で描いています。最後に 3 秒静止します。gnuplot の時間待ちコマンド `pause` を用いて、エンターキーが押されるまで待つには `-1` を引数に与えれば良いのですが、子プロセスで起動された場合にはこれが効きません、直に終了してしまいます。しばらく図を眺めたい場合には `pause` の引数に大きな値を与えるか、gnuplot 自身を `-persist` オプション付きで起動させる必要があります。

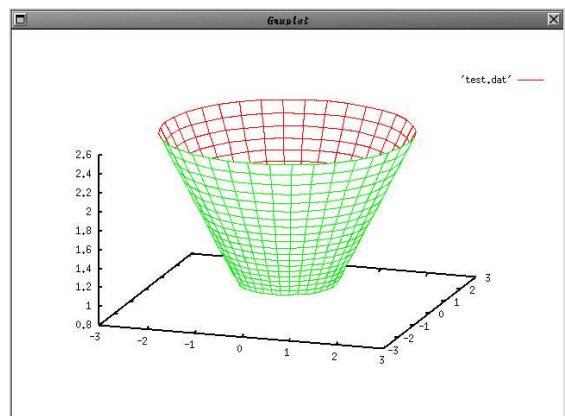


図 2 バケツが x, y 両方の軸のまわりに回転

【リスト 2】 ex2.c

```
#include <stdio.h>
#include <stdlib.h>
#include <math.h>

int main(int argc, char **argv)
{
    FILE *gid, *fid;
    int i,j;
    double x,y,z,t;
    char *data={"test.dat"};

    gid = popen("gnuplot","w");
    fid = fopen(data,"w");

    for (i=0; i<20; i++){
        z = exp(0.05*i);
        for(j=0; j<=36; j++){
            x = z*cos(M_PI/18*j);
            y = z*sin(M_PI/18*j);
            fprintf(fid,"%f %f %f\n",x,y,z);
        }
        fprintf(fid,"\n");
    }
    fclose(fid);

    fprintf(gid,"set hidden3d\n");
    fprintf(gid,"set ticslevel 0\n");
    for (i=0; i<=30 ;i++){
        fprintf(gid,"set view %d,%d,\n",30+2*i, i);
        fprintf(gid,"splot '%s' w lines \n", data);
        fflush(gid);
    }
    fprintf(gid,"pause 3\n");
    pclose(gid);
}
```

標準入力経由でデータを

さて、上記の方法では計算結果を別ファイル “test.dat” に保存し、gnuplot でそれを読み込むように指示しています。このような一時ファイルを残すことに不満なら、直接 gnuplot にデータを送出すべきです。次の例をご覧ください。gnuplot で標準入力を指定して

```
[s]plot '-' [オプション]
```

と記述することで読み込むことができます。行頭に 'e' があるとデータ入力が終了します。標準入力の記号 '-' と行頭の 'e' で終了は UNIX の慣習に従ったものです。

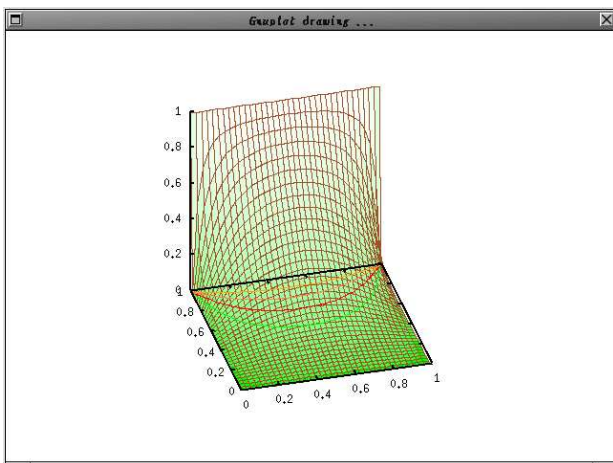


図 3 Relaxation 法による静電場：計算途中

【リスト 3】 ex3.c

```
#include <stdio.h>
#include <stdlib.h>
#include <math.h>

#define DIV 71
#define INTERVAL 30

main(int argc, char **argv)
{
    FILE *gid;
    int i,j, gout;
    double cmax, corr, EPS=1e-5, u[DIV][DIV];
    double MESH=1.0/DIV;

    if (argc>2) EPS = atof(argv[1]);

    /* 境界条件 */
    for (i = 0; i < DIV; i++){
        for (j = 0; j < DIV; j++){
            u[i][j]=0.0;
        }
    }
    for (i = 0; i < DIV; i++) u[i][DIV-1]=1.0;
    /* for (j = 0; j < DIV; j++) u[DIV-1][j]=1.0; */

#ifdef V38I
    gid = popen("gnuplot38i","w");
    fprintf(gid,"set pm3d\n");
    fprintf(gid,"unset colorbox\n");
    fprintf(gid,"set palette rgb 8,2,8\n");
    fprintf(gid,"set size square\n");
#else
    gid = popen("gnuplot37i","w");
    fprintf(gid,"set size 0.7,1.1\n");
#endif

    fprintf(gid,"set view 60,345\n");
    fprintf(gid,"set xtics 0.2 \n");
    fprintf(gid,"set ytics 0.2 \n");
    fprintf(gid,"set ticslevel 0\n");
    fprintf(gid,"set contour\n");
    fprintf(gid,"set nokey\n");
    fprintf(gid,"set size square\n");

    /* Relaxation 法による反復解法 */
    while(1) {
        gout++;
        cmax=0.0;
        for (i = 1; i < DIV-1; i++){
            for (j = 1; j < DIV-1; j++){
                corr = (u[i+1][j]+u[i-1][j]+u[i][j+1]
                    +u[i][j-1])/4.0 - u[i][j];
                if ( fabs(corr) > cmax ) cmax = corr;
                u[i][j] = u[i][j]+corr;
            }
        }
        if ( cmax < EPS ) break;
        if (gout % INTERVAL != 0) continue;

        fprintf(gid,"splot '-' w lines"); /* 注目 */
        for (i = 0; i < DIV; i+=2){
            fprintf(gid,"\n");
            for (j = 0; j < DIV; j+=2){
                fprintf(gid,"%e %e \n",i*MESH,j*MESH,u[i][j]);
            }
        }
        fprintf(gid,"e\n"); /* データ入力終了の印 */
    }

    /* 等高線の出力 gnuplot にお任せ*/
    fprintf(gid,"set cntrparam levels incre 0,0.05,1\n");
    fprintf(gid,"set nosurface\n");
    fprintf(gid,"set view 0,0\n");
    fprintf(gid,"set noclabel\n");
    fprintf(gid,"splot '-' w lines lt 2\n");
    for (i = 0; i < DIV; i++){
        fprintf(gid,"\n");
        for (j = 0; j < DIV; j++){
            fprintf(gid,"%e %e %e\n",i*MESH,j*MESH,u[i][j]);
        }
    }
    fprintf(gid,"e\n");
    fprintf(gid,"pause 60\n");
    pclose(gid);
}
```

内容は、2次元正方領域内の静電場を Relaxation 法で求めてい

ます。真空中の静電場は次のように、ポアソン方程式という2階の偏微分方程式を適当な境界条件を与えれば解くことが可能です。

$$\Delta\phi(r) = 0, \quad \phi(\partial\Omega) : \text{given}$$

この例では正方領域の1つの辺上で  $\phi = 1$  残りの辺上では  $\phi = 0$  という強引な境界条件を課しています。数値解放の Relaxation 法とはポアソン方程式を差分に変換すると、一つの格子点上の関数値は隣接の4つの格子点の平均として

$$\phi_{i,j} = \frac{1}{4} \left( \phi_{i-1,j-1} + \phi_{i+1,j-1} + \phi_{i-1,j+1} + \phi_{i+1,j+1} \right)$$

で与えられることを利用して、実際にそうなるまですべての点上の値を計算し続けるというものです。Excel などの表計算ソフトでも実現できる簡単なアルゴリズムです。もちろんある許容誤差内で一致するまでという収束判定をします。電位  $\phi(x,y)$  を  $z$  軸にとった俯瞰図が現れ、正しい分布に近づいていく様子が判り、筆者は気に入っています。等電位線をメイン側で計算することも可能ですが、gnuplot に contour 機能がありますから、それに任せることにしました。図3, 4 は開発バージョン 3.8i でカラーマップ pm3d を用いた場合のものです。3.8i では splot に対しても set size square が有効になっています。

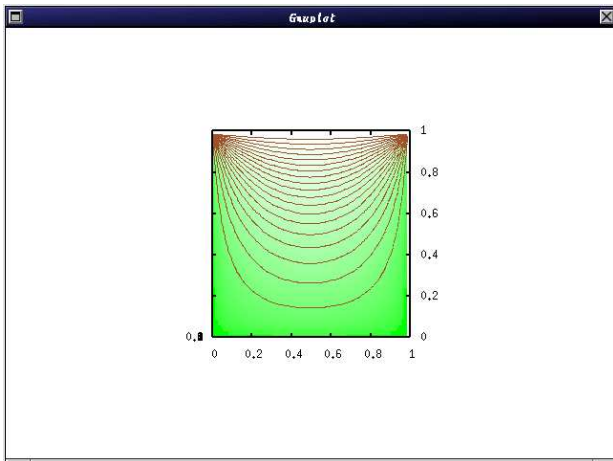


図 4 Relaxation 法による静電場：最終結果の等電位線

### 複数のデータを1つの図に

最後にデータを複数入力する場合について説明します。その場合には標準入力を複数並べます。gnuplot で標準入力を使う場合には入力データは保存されませんから、その都度データをメイン側から送らなければなりません。簡単な例をリスト4に示します。この機能は正式公開バージョン 3.7.1 では使えないようです。次期公開バージョン 4.0 にむけての開発バージョン（現在 3.8i）を入手して確かめてください [4][W<sup>3</sup>]。以前紹介しましたが、手軽さはそのままに、マウスと pm3d というカラーマップという大きな機能追加がますます充実しています。

【リスト4】 ex4.c

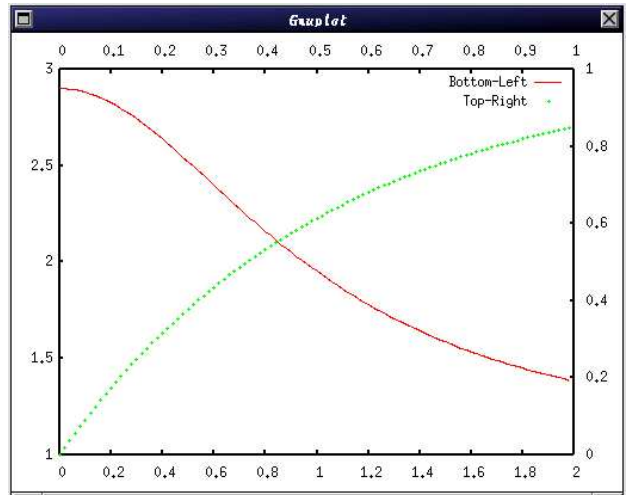


図 5 2つの独立なデータの同時表示

```
#include <stdio.h>
#include <stdlib.h>
#include <math.h>

int main(int argc, char **argv)
{
    FILE *gid;
    int i,j;
    double x;

    gid = popen("gnuplot -geometry 512x384","w");
    fprintf(gid, "set yrange[1:3]\n");
    fprintf(gid, "set y2range[0:1]\n");
    fprintf(gid, "set y2tics\n");
    fprintf(gid, "set x2tics\n");
    fprintf(gid, "set xtics nomirror\n");
    fprintf(gid, "set ytics nomirror\n");

    for (i=0; i<20; i++){
        fprintf(gid, "plot '-' t 'Bottom-Left' w lines, \
                '-' t 'Top-Right' axis x2y2\n");
        for (j=0; j<100; j++){ /* 1番目のデータ */
            x = 0.02*j;
            fprintf(gid, "%e %e\n", x, 1+i*0.1/(1+x*x));
        }
        fprintf(gid, "e\n"); /* データ入力終了 */

        for (j=0; j<100; j++){ /* 2番目のデータ */
            x = 0.01*j;
            fprintf(gid, "%e %e\n", x, 1-exp(-(i*0.1)*x));
        }
        fprintf(gid, "e\n"); /* データ入力終了 */
    }
    fprintf(gid, "pause 60\n");
    pclose(gid);
}
```

### 参考文献

- [1] Gnuplot の公式サイト. [W<sup>3</sup>]  
http://www.gnuplot.info/
- [2] NIST にある CGM の規定. [W<sup>3</sup>]  
http://www.itl.nist.gov/div897/ctg/graphics/cgm\_std.htm
- [3] Sourceforge の XGKS プロジェクト. [W<sup>3</sup>]  
http://sourceforge.net/projects/xgks/

- [4] 開発版は SourceForge の CVS レポジトリから. [W<sup>3</sup>](#)  
[http://sourceforge.net/cvs/?group\\_id=2055](http://sourceforge.net/cvs/?group_id=2055)  
gnuplot 以下にソースが展開されてますから, まずおまじないの  
./prepare を実行してください. あとは configure, make する  
だけです. もし version.c 関係でエラーが出たら, version.c の  
ソースを適当に編集してください.