

グラフィックスと GrWin

1 グラフィックスプログラミングと GrWin

コンピュータにおいてグラフィックス表示のプログラムを作成する場合、表示するウィンドウの管理は極めて複雑となる。Windows 環境の Microsoft Visual C++ 等では、グラフィックス表示の雑多な処理を自動生成することができる。

一方、「GrWin グラフィックス・ライブラリ」は Windows Xp/2000/NT/Me/9x 上で利用できる Fortran と C/C++ のためのフリーなグラフィックス・ライブラリで、フリーのコンパイラの MinGW や Cygwin などのプログラミング環境とともにこのライブラリを用いることで、Windows プログラミングについての知識や経験が全くない人でも Windows (x86) プラットフォーム上でのフリーなグラフィックス・プログラミングを手軽に楽しむことができる。

また、グラフィックスの表示は GUI(Graphical User Interface) に依存するため、OS が異なるとプログラムも違ってしまふ。OS に依存しないグラフィックスとしては、Java によるプログラミング演習も考えられる。

今後のグラフィックスの演習では、これまでの C 言語でのプログラミングの知識と、無償で利用できるグラフィックス環境として GrWin を使用する。GrWin は計算結果を簡単にグラフ化することを想定して作られているため、簡単な命令でグラフ作成などができる。

GrWin は、グラフィックス表示用の `grwnd.exe` というプログラムに、プログラムの描画命令を送ることで、グラフィックスデータを表示する。このため GUI の複雑な処理は、`grwnd` が行ってくれる。

1.1 プログラミング応用でのグラフィックスの目標

プログラミング応用の授業では、OS に依存したプログラミングテクニックの紹介は、技術変化に伴って陳腐化することを避けるために、座標変換や 3 次元表示のために必要な計算方法を中心に講義をすすめる。

GrWin では、グラフ表示領域の設定や画面サイズに応じた座標変換を行ってくれる。しかし、この講義では他のグラフィックス環境と同じ左上を原点とし、画面の 1 ドットを単位とする座標系を用いる。

2 GrWin の使い方

GrWin は、<http://spdg1.sci.shizuoka.ac.jp/grwinlib/> にて、公開されており、自分の利用しているコンパイラに応じて必要ファイルをダウンロード・インストールすれば良い。総合情報処理センターでは、あらかじめ GrWin はインストールされている。

インストール作業により、表示プログラム `grwnd.exe` や GrWin のためのヘッダファイル `GrWin.h` が、各コンパイラの include ディレクトリにコピーされる。

3 基本プログラム

基本的なサンプルプログラムを以下に示す。

```
#include <GrWin.h>

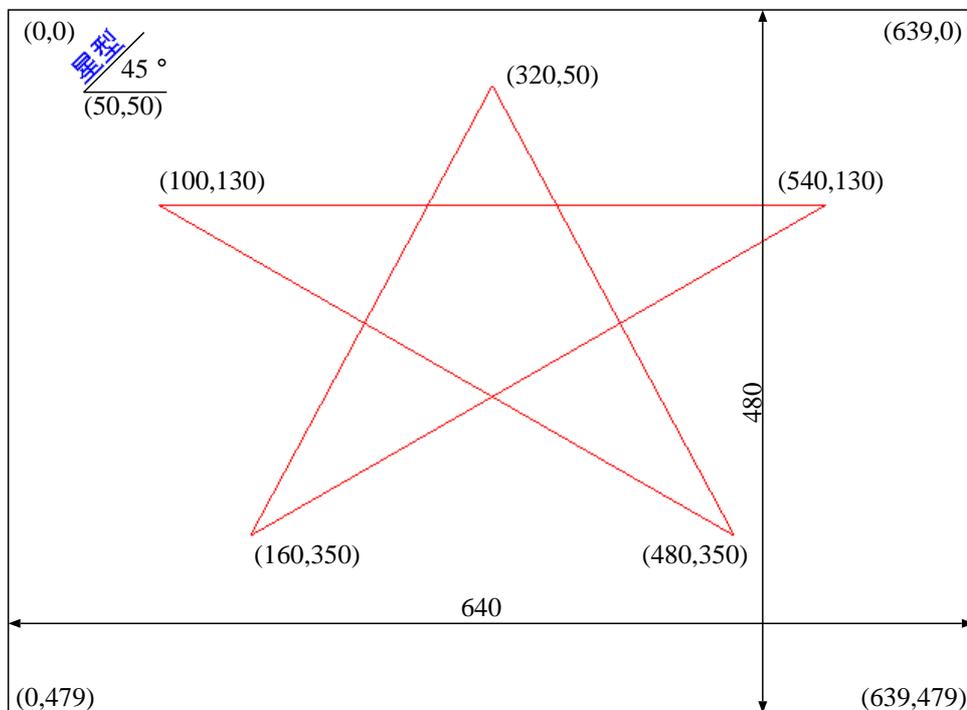
int main(void)
{
    /* 640x480 の左上原点でウィンドウを作成する */
    int width = 640, height = 480 ; /* ウィンドウサイズ 640 × 480 */
    GWopen( 0 ) ; /* ウィンドウのオープン */
    GWsize( -5 , &width , &height ) ; /* ウィンドウサイズ設定 */
    GWsize( -3 , NULL , NULL ) ; /* フレーム(枠)サイズ設定 */
    GWviewport( 0.0 , 0.0 , (float)width / (float)height , 1.0 ) ; /* ビューポート設定 */
    GWindow( 0.0 , (float)height - 1.0 , (float)width - 1.0 , 0.0 ) ; /* ワールド座標系設定(左上原点) */

    /* グラフィックスの描画 */
    GWclear( GWC_WHITE ) ;

    GWsetpen( GWC_YELLOW , GWL_SOLID , 1 , GWX_COPYPEN ) ;
    GWline( 320 , 50 , 160 , 350 ) ;
    GWline( 160 , 350 , 540 , 130 ) ;
    GWline( 540 , 130 , 100 , 130 ) ;
    GWline( 100 , 130 , 480 , 350 ) ;
    GWline( 480 , 350 , 320 , 50 ) ;

    GWsettxt( 20 , 45.0/360.0 , 1 , GWC_RED , -1 , "" ) ;
    GWputtxt( 50 , 50 , "星型" ) ;

    /* 終了処理 */
    GWquit();
    return 0;
}
```



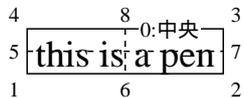
3.1 画面の初期化に関する関数

サンプルプログラム中の以下の関数は、画面の初期化などの関数である。通常は『決まり文句』的にコピーして利用すれば良い。詳細は GrWin のサイトを参照せよ。

- GWopen(): GrWin の初期化を行い、グラフィックウィンドウを開く。
- GWsize(): ウィンドウのサイズを変更する。-5 はピクセル値でウィンドウのサイズ指定。-3 はフレームサイズの指定。
- GWviewport(): ビューポート (描画命令で実際に表示される範囲) を指定する。
- GWwindow(): ワールド座標系 (描画命令で指定する座標の範囲) を指定する。
- GWquit(): ウィンドウを閉じる。

3.2 描画に関する関数

- GWclear(): 画面を指定色で塗りつぶして消す。色は、以下の色番号で指定する。

0:GWC_BLACK(黒)	1:GWC_DARKRED(暗赤)	2:GWC_DARKGREEN(暗緑)	3:GWC_DARKYELLOW(オリーブ)
4:GWC_DARKBLUE(暗紺)	5:GWC_DARKMAGENTA(紫)	6:GWC_DARKCYAN(緑青)	7:GWC_LIGHTGRAY(灰色)
8:GWC_MONEYGREEN(明緑)	9:GWC_SKYBLUE(薄水色)	10:GWC_CREAM(薄灰色)	11:GWC_MEDIUMGRAY(青灰色)
12:GWC_DARKGRAY(暗灰色)	13:GWC_RED(赤)	14:GWC_GREEN(緑)	15:GWC_YELLOW(黄)
16:GWC_BLUE(青)	17:GWC_MAGENTA(赤紫)	18:GWC_CYAN(水色)	19:GWC_WHITE(白)
- GWsetpen(): ペンを指定する。引数の意味は以下の通り
 1. 色番号
 2. 線の種類 0:GWL_NULL(透明), 1:GWL_SOLID(実線), 2:GWL_DASH(破線), 3:GWL_DOT(点線),... 等
 3. 線の太さ
 4. 混合モード (色々な描画方法が選べるが通常は以下を指定せよ)
GWX_COPYPEN(線データをコピーする)
- GWline(X1,Y1,X2,Y2): (X1,Y1) ~ (X2,Y2) をペンで結ぶ。
- GWsettxt(size, 傾き, 基準点, 色, 背景, フォント名)
 1. フォントの大きさ
 2. フォントの傾き (0 ~ 1.0 の範囲で反時計方向の角度)
 3. 基準点


4 8 0:中央 3
5 this is a pen 7
1 6 2
 4. 色, 背景は、色番号を参照
 5. フォント名は "" を指定すればデフォルトを使用
- GWputtxt(X,Y, 文字列) GWsettxt で指定した方法で、(X,Y) 座標に文字列を描く。

4 関数のグラフを描く

通常関数のグラフを描く場合は、`GRline()` 命令を使い、`X` を微小に増加させながら `Y` の値を求め、細かい折れ線グラフを描くようにすれば良い。

```
#include <GrWin.h>
#include <math.h>

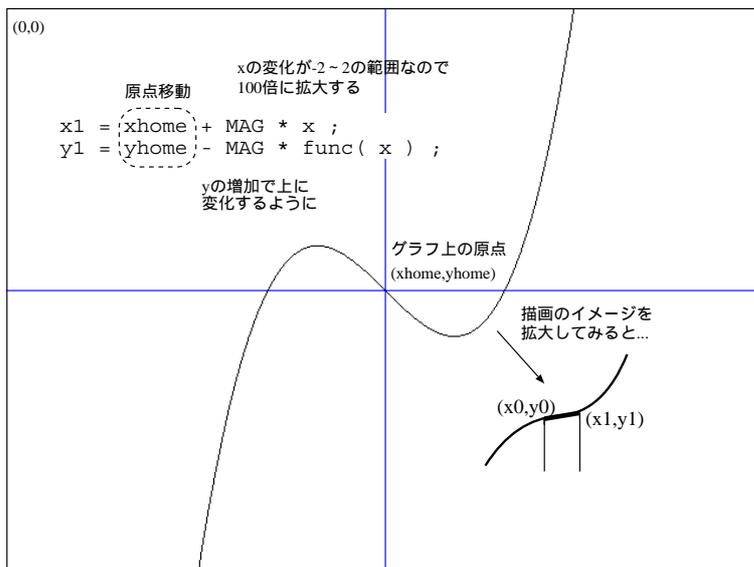
#define MAG 100
float func( float x ) /* f(x) = x^3 - x */
{
    return x*x*x - x ;
}
int main(void) {
    int width = 640 , height = 480 ; /* ウィンドウサイズ 640 x 400 */
    int xhome = width / 2 , yhome = height / 2 ;
    float x , x0 , y0 , x1 , y1 ;

    GWopen( 0 ) ;
    GWsize( -5 , &width , &height ) ;
    GWsize( -3 , NULL , NULL ) ;
    GWvport( 0.0 , 0.0 , (float)width / (float)height , 1.0 ) ;
    GWindow( 0.0 , (float)height - 1.0 , (float)width - 1.0 , 0.0 ) ;

    GWclear( GWC_WHITE ) ;
    GWsetpen( GWC_BLUE , GWL_SOLID , 1 , GWX_COPYPEN ) ;

    GWline( 0 , yhome , width , yhome ) ; /* 座標軸 */
    GWline( xhome , 0 , xhome , height ) ;

    GWsetpen( GWC_BLACK , GWL_SOLID , 1 , GWX_COPYPEN ) ;
    x0 = xhome + MAG * (-2.0) ; /* 最初の始点を計算 */
    y0 = yhome - MAG * func( -2.0 ) ;
    for( x = -2.0 ; x <= 2.0 ; x += 0.02 ) {
        x1 = xhome + MAG * x ;
        y1 = yhome - MAG * func( x ) ;
        GWline( x0 , y0 , x1 , y1 ) ; /* (x0,y0)-(x1,y1) を直線で */
        x0 = x1 ;
        y0 = y1 ; /* (x1,y1) を次の始点とするために代入 */
    }
    return 0 ;
}
```



4.1 媒介変数の関数グラフ

以下のような θ を媒介変数とする関数のグラフを表示するプログラムの下線部 (A) ~ (D) を埋めよ。

$$\begin{cases} x = \cos 2\theta \\ y = \sin 4\theta \end{cases}$$

```
#include <GrWin.h>
#include _____(A)
void func( float* x , float* y , float th )
{
    *x = cos( 2 * th ) ;
    *y = sin( 4 * th ) ;
}
#define MAG 150

int main(void) {
    int width = 640, height = 480;
    int xhome = width/2 , yhome = height/2 ;
    float x , y , th , x0 , y0 , x1 , y1 ;

    GWopen( 0 ) ;
    GWsize( -5 , &width , &height ) ;
    GWsize( -3 , NULL , NULL ) ;
    GWvport( 0.0 , 0.0 , (float)width / (float)height , 1.0 ) ;
    GWindow( 0.0 , (float)height - 1.0 , (float)width - 1.0 , 0.0 ) ;

    GWclear( GWC_WHITE ) ;
    GWsetpen( GWC_BLUE , GWL_SOLID , 1 , GWX_COPYPEN ) ;
    GWline( 0 , yhome , width , yhome ) ;
    GWline( xhome , 0 , xhome , height ) ;

    GWsetpen( GWC_BLACK , GWL_SOLID , 1 , GWX_COPYPEN ) ;
    func( _____ , _____ , _____ ) ;
    _____(B)
    x0 = xhome + MAG * x ;
    y0 = yhome - MAG * y ;
    for( th = 0 ; th <= 360.0 / 180.0 * 3.14159 ; th += 5.0 / 180.0 * 3.141592 ) {
        _____(E) _____(F)
        func( &x , &y , th ) ;
        x1 = xhome + MAG * _____ ;
        _____(C)
        y1 = yhome - MAG * _____ ;
        _____(D)
        GWline( x0 , y0 , x1 , y1 ) ;
        x0 = x1 ;
        y0 = y1 ;
    }
    return 0;
}
```

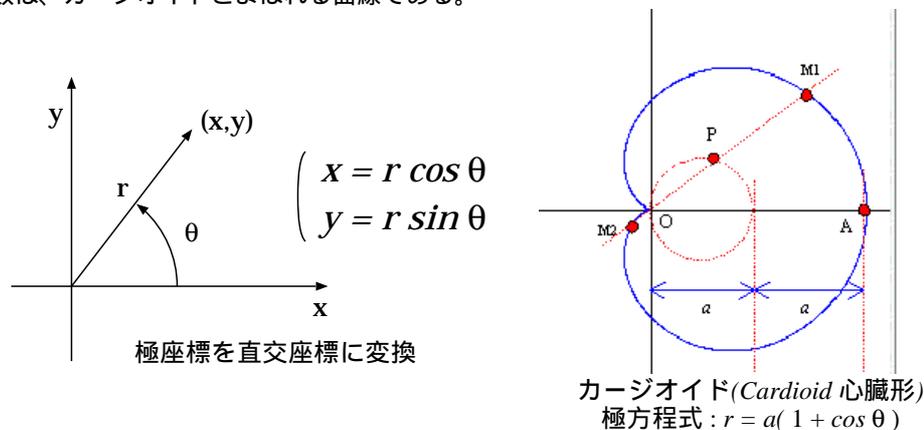
質問

1. 下線部 (E) の条件判断を、『 $th \neq 360.0 / 180.0 * 3.141592$ 』と書くと、プログラムが無限ループで終了しない。なぜ？
2. 下線部 (F) の部分を『 $th += 5 / 180 * 3.141592$ 』と書くと、これもまたプログラムが無限ループとなる。なぜ？

4.2 極座標のグラフ

媒介変数のグラフ表示のプログラムを参考に、極座標形式の方程式 $r = a(1 + \cos \theta)$ のグラフを表示するプログラムを作成せよ。

この関数は、カージオイドとよばれる曲線である。



カージオイド(Cardioid 心臓形)
極方程式: $r = a(1 + \cos \theta)$

他にも、インボリュート曲線は機械系のギアの断面に使われる、工学上重要な極方程式がある。興味のある人は、課題にて取り組むことを期待する。

5 レポート課題

以下のテーマの中から自分の理解度に応じて、選択しレポートとして提出せよ。

1. $y = \sin x$, $y = \tan x$
2. 一つの画面内に、複数の関数グラフをまとめて表示せよ。
3. 数学的にグラフの形が美しい曲線を探し、その関数のグラフを GrWin を使って表示せよ。

なお、グラフィックス演習では、複数回のレポートを予定している。このため説明・考察などはある程度省略しても良いものとする。

参考資料

GrWin の利用にあたって、参考となるページの URL を以下に示す。

1. <http://spdg1.sci.shizuoka.ac.jp/grwinlib/> 配布元
(静岡大学理学部:Tsuguhiro TAMARIBUCHI 氏)
2. <http://www005.upp.so-net.ne.jp/h-masuda/grwin/>
増田@大阪府立成城高等学校 さんによる「C 言語でグラフィックス」

以下の資料では、上記ページの内容を参考にしています。