

## 「情報工学」CG 課題4

三浦憲二郎

令和4年10月26日(水曜日)

提出締め切り 令和4年11月1日(火曜日) 24:00

提出物：提出物は各課題(課題A、課題B、課題C)の最後の状態のプログラムのソースコードとする。

提出：watanabe.ryunosuke.18@gmail.com (三浦研渡辺龍之介君)

件名：情報工学課題4 学籍番号 名前

提出方法：添付ファイル、本文にも TA が管理しやすくなるので、学籍番号、名前を必ず記入すること。

### 曲線の描画と場合分け

グラフィック

- パラメータ
- 曲線の描画

---

#### サンプルプログラム

```
#include <GL/glut.h>
#include <math.h>
#define PI 3.141592653589793          /* 円周率  $\pi$  */
void display(void) {
    double x, y, x1, y1, t;
    glClearColor(GL_COLOR_BUFFER_BIT); /* 背景のクリア */
    glBegin(GL_LINES);                 /* 線分を描画する */
    for(t = 0.0; t < PI; t += 0.3) {
        x = cos(t);                    /* x座標を計算する */
        y = sin(t);                    /* y座標を計算する */
        if(t != 0.0) {
            glVertex2f(x, y); glVertex2f(x1, y1); /* 始点、終点の指定 */
        }
        x1 = x; y1 = y;                /* 座標を保存しておく */
    }
    glEnd();                            /* 線分の描画終了 */
    glFlush();                          /* 画面を再描画する */
}
```

```

}
void init(void) {
    glClearColor ( 0.0, 0.0, 0.0, 0.0 );          /* 背景色の指定          */
    glMatrixMode(GL_PROJECTION);                 /* 射影変換を指定        */
    glLoadIdentity();                            /* 射影行列の初期化      */
    glOrtho(-2.0, 2.0, -2.0, 2.0, -1.0, 1.0);    /* 描画のための座標系の指定 */
}
int main(int argc, char **argv) {
    glutInit(&argc, argv);                       /* GLUT の初期化          */
    glutInitDisplayMode(GLUT_SINGLE | GLUT_RGB); /* 表示モードの指定      */
    glutInitWindowSize(400, 400);                /* ウィンドウサイズの指定 */
    glutInitWindowPosition(100, 100);            /* ウィンドウの位置の指定 */
    glutCreateWindow("OpenGL Sample 3");         /* ウィンドウのオープン   */
    init();                                       /* 初期化処理            */
    glutDisplayFunc(display);                    /* 表示の関数の指定      */
    glutMainLoop();                             /* GLUT のメインループ    */
    return 0;
}

```

## 課題A

(a) きれいな1周の円を描画するように変更する.

Hint. パラメータ  $t$  を  $0 \sim \text{PI}(3,14\dots)$  まで変化させると半周の円が書けます. 円を1周させるには  $t$  を  $0 \sim \text{PI} * 2(6.28\dots)$  まで変化させるようにします. また  $t$  の増分を小さくすることで, 滑らかな曲線が書けるようになります. 例えば, 次のようにします.

$t$  を  $0 \sim 2 * \text{PI}$  まで変化させる.  $t$  が  $0.01$  ずつ増えるようにする.

```
for(t = 0.0; t < PI*2; t += 0.01) {
```

(b) 円の半径を指定できるように変更する.

Hint. 円の半径を保存しておく大域変数(double r;)を用意し, scanf 文を用いて半径を読み込みます. display() 関数内で変数 r の値を利用して円を書くようにします. 上のサンプルプログラムでは, 半径1の円を書くようになっているので, x, y 座標を r 倍することで半径 r の円を描画できるようことができます.

(c) n 重の円を描画できるように変更する.

Hint. 円を描画する箇所を for 文で囲み, n 回繰り返すようにします. また, 繰り返すごとに n を用いて半径を計算し, 円が小さくなっていくようにします.

(d) 楕円を n 重に描画するよう変更する.

Hint. x, y 座標を計算する部分を楕円を描画するように変更する.

楕円を描画する.

```
x = a * cos(t) * r;
```

```
y = b * sin(t) * r;
```

大域変数 `double a, b;` を用意しておき, `scanf` 文を用いて読み込むようにします. (ex. `a = 1.0, b = 0.5`).

**課題B** (時間に余裕のある人のために)

(a) エピトロコイドを書いてみましょう.

Hint. `x, y` 座標を計算する部分を次のように変更します.

エピトロコイドを描画する.

```
x = (a+b)*cos(t) - c*cos((a/b+1.0)*t);
```

```
y = (a+b)*sin(t) - c*sin((a/b+1.0)*t);
```

大域変数 `double a, b, c;` を用意しておき, `scanf` 文を用いて読み込むようにします. (ex. `a=1.0, b=0.2, c=0.3`).

(b) `c` の値を変化させて, エピトロコイドを重ね書きしてみましょう.

Hint. エピトロコイドを描画する箇所を `for` 文で囲い, `c` の値を変化させながら繰り返し描画を行います.

**課題C** (さらに時間に余裕のある人のために)

(a) ハイポトロコイドを書いてみましょう.

Hint. `x, y` 座標を計算する部分を次のように変更します.

ハイポトロコイドを描画する.

```
x = (a-b)*cos(t) + c*cos((a/b-1.0)*t);
```

```
y = (a-b)*sin(t) - c*sin((a/b-1.0)*t);
```

大域変数 `double a, b, c;` を用意しておき, `scanf` 文を用いて読み込むようにします. (ex. `a=1.0, b=0.2, c=0.3`).

(b) `c` の値を変化させて, ハイポトロコイドを重ね書きしてみましょう.

Hint. ハイポトロコイドを描画する箇所を `for` 文で囲い, `c` の値を変化させながら繰り返し描画を行います.