

大作业——MyPlot 函数绘制工具

〇、声明

本文档旨在说明大作业基本要求。请勿外传。

设置大作业的目的是希望同学们活用课程学到的知识，锻炼解决实际问题的能力，体验团队协作过程，并从过程中学到更多课堂很难涉及的知识。

“程序设计基础”课程教学团队保留对本文档内容的最终解释权。

目录

〇、声明.....	1
一、任务描述.....	3
1.1、函数绘制.....	3
1.1.1、一元函数绘制	3
1.1.2、参数函数绘制	4
1.1.3、二元函数绘制	5
1.2、画布创建.....	6
1.2.1、创建二维坐标系画布.....	6
1.2.2、创建三维坐标系画布.....	7
1.3、画布保存.....	8
二、必做任务.....	9
2.1、直线函数.....	9
2.2、抛物线	9
2.3、笛卡尔心形线	10
2.4、其他心形线.....	10
2.5、多层心形线.....	10
2.6、墨西哥草帽	11
三、输入输出要求	12
3.1、输入.....	12
3.2、输出.....	12
四、组队、提交要求及计分说明	13
4.1、组队要求.....	13
4.2、提交要求.....	13
4.3、计分说明.....	14

一、任务描述

绘制函数图像是帮助我们理解函数性质的重要手段。让我们使用 C/C++ 语言，编写一个 MyPlot 工具，实现绘制已知函数的图像的功能吧！具体包括以下功能函数：

1.1、函数绘制

MyPlot 工具支持的函数绘制功能应包括以下三种：

1.1.1、一元函数绘制

已知一元函数方程，绘制函数图像。该功能函数的声明为：

```
void plot2D(  
    double (*func)(double),  
    double minX, double maxX,  
    int color);
```

其中：

func 为 C/C++ 语言实现的一元函数方程；

minX 和 maxX 表达定义域取值范围；

color 为颜色值。这里，颜色通过红绿蓝三原色的数值表达，每种颜色占用一个字节（即，取值范围 0-255），拼合成一个 int 整数表达一种颜色。例如，0x0000FF 表示红色，0x00FF00 表示绿色，0xFF0000 表示蓝色，0x00FFFF 表示黄色，0xFF00FF 表示亮紫色，等。

例如：

```
plot2D(func1, -7, 7, 0x00FF00);
```

将使用绿色绘制函数 $y = \text{func1}(x)$ 、且 x 取值范围 $[-7, 7]$ 的图

像。这时，如果 func1 的定义如下所示，则绘制的图像见任务 2.1。

```
double func1(double x) {  
    return x;  
}
```

1.1.2、参数函数绘制

已知一元参数方程，绘制函数图像。该功能函数的声明为：

```
void plotPara(  
    double (*funcX)(double), double (*funcY)(double),  
    double minT, double maxT,  
    int color);
```

其中：

funcX 和 funcY 分别为 C/C++语言实现的 X 和 Y 的参数方程；

minT 和 maxT 表达参数的取值范围；

color 为颜色值。

例如：

```
plotPara(func31, func32, 0, 2 * 3.1416, 0xFF0000);
```

将使用蓝色绘制由 $x = \text{func31}(t)$ 、 $y = \text{func32}(t)$ 联立形成、且 t 取值范围 $[0, 2 * 3.1416]$ 时的参数曲线。这时，如果 func31 和 func32 的定义如下所示，则绘制的图像见任务 2.3。

```
double func31(double t) {  
    return cos(t) * 2 - cos(t * 2);  
}  
double func32(double t) {  
    return sin(t) * 2 - sin(t * 2);  
}
```

1.1.3、二元函数绘制

已知二元函数方程，绘制函数图像（三维）。该功能函数的声明

为:

```
void plot3D(  
    double (*func)(double, double),  
    double minX, double maxX,  
    double minY, double maxY);
```

其中:

func 为 C/C++语言实现的二元函数方程;

minX 和 maxX 表达自变量 X 的取值范围;

minY 和 maxY 表达自变量 Y 的取值范围。

绘制二元函数时,通常不设置颜色值,而是根据函数值的大小设置不同的颜色值。

例如:

```
plot3D(func6, -8, 8, -8, 8);
```

将使用绘制 $z = \text{func6}(x, y)$ 、且 x 取值范围 $[-8, 8]$ 、 y 取值范围 $[-8, 8]$ 时的曲面。**绘制曲面可以由绘制多个小四边形实现。**这时,如果 func6 的定义如下所示,则绘制的图像见任务 2.6。

```
double func6(double x, double y) {  
    double r = sqrt(x * x + y * y);  
    if (r == 0.0)  
        return 1.0;  
    return sin(r) / r;  
}
```

1.2、画布创建

1.2.1、创建二维坐标系画布

```
void createCanvas(  
    int width, int height,  
    int origX, int origY,  
    int ratio,  
    bool axis);
```

其中：

width 和 height 分别为输出图像的宽和高；

origX 和 origY 表达坐标原点在图像中的位置，简化起见，我们规定 X 轴正方向向右，Y 轴正方向向上；

ratio 表示坐标单位对应于多少像素；

axis 表示是否绘制坐标轴。

例如：

```
createCanvas(800, 600, 400, 300, 10, true);
```

表示创建了一张 800x600 的图像，以其中的(400, 300)位置为坐标原点，每 10 个像素为一个单位，而且绘制坐标轴。即，图像的(410, 300)位置对应函数坐标系中的(1, 0)点，图像的(400, 290)位置对应函数坐标系中的(0, 1)点。

1.2.2、创建三维坐标系画布

```
void createCanvas3D(  
    int width, int height,  
    int origX, int origY,  
    double rxx, double rxy,  
    double ryx, double ryy,  
    double rzx, double rzy);
```

其中：

width 和 height 分别为输出图像的宽和高；

origX 和 origY 表达坐标原点在图像中的位置；

rxx、rxy、ryx、ryy、rzx、rzy 分别为三维坐标系的坐标对图像 x、y 坐标的变换参数。具体参考下面的公式，可以将三维坐标 (X, Y, Z) 变换为图像中的坐标 (imgX, imgY)：

```
imgX = origX + rxx * X + rxy * Y + rzx * Z  
imgY = origY + rxy * X + ryy * Y + rzy * Z
```

简单起见，三维坐标系画布不绘制坐标轴或坐标平面。

例如：

```
void createCanvas(  
    800, 600, 400, 300,  
    24.8438, -5.375, -16.175, -5.3, 0, -145);
```

表示创建了一张 800x600 的图像，以其中的 (400, 300) 位置为坐标原点。坐标变换参数可用来得到任务 2.6 所需的效果。

1.3、画布保存

```
void saveCanvas(char * filename);
```

其中：

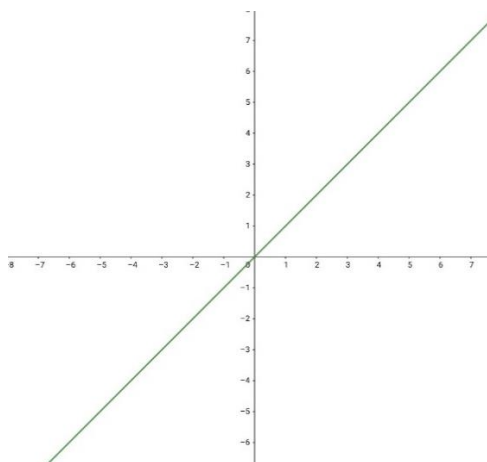
filename 表示输出的文件名。

二、必做任务

在完成上述功能函数的基础上，通过在主函数中调用这些函数，绘制出下列图像，具体的参数可根据审美自行设定。

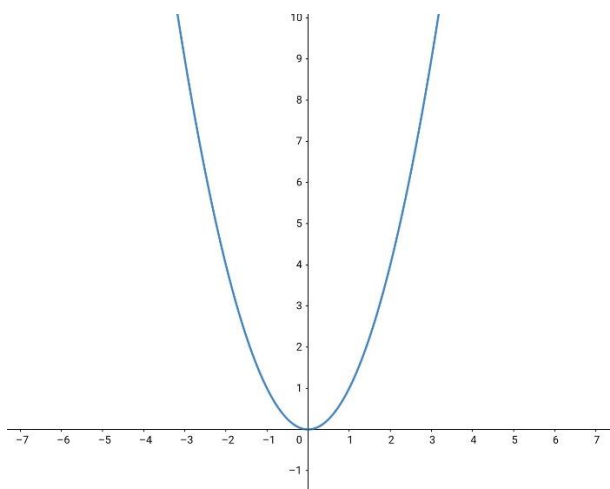
2.1、直线函数

绘制下面的图形。



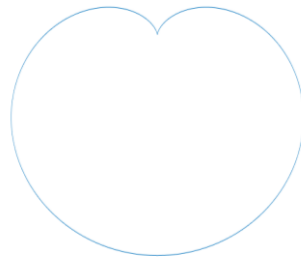
2.2、抛物线

绘制下面的图形。



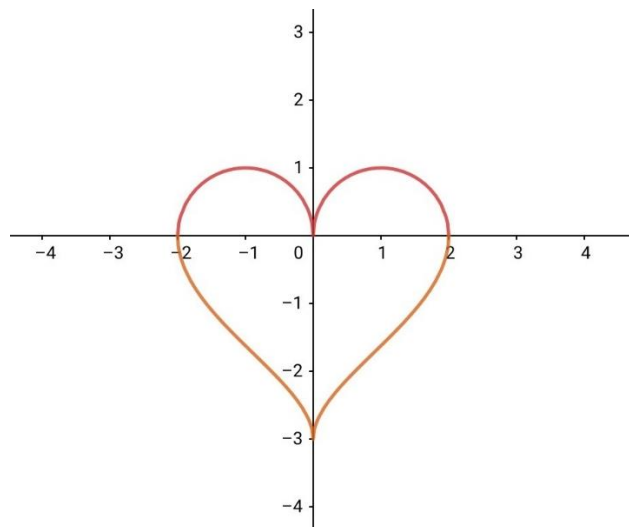
2.3、笛卡尔心形线

绘制下面的图形。提示，调用 `plotPara` 函数实现。



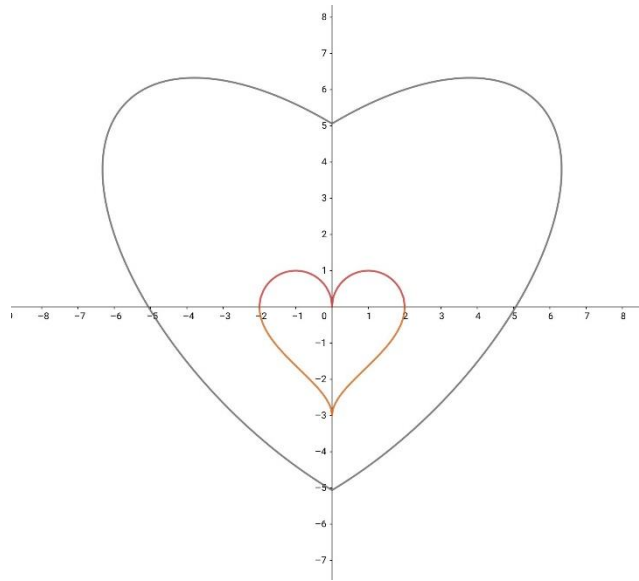
2.4、其他心形线

绘制下面的图形。提示，可通过多次调用 `plot2D` 函数实现。



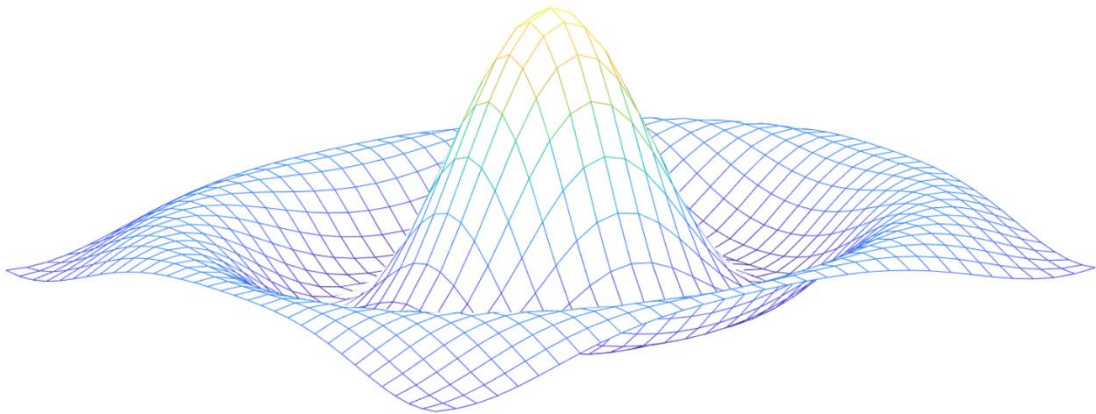
2.5、多层心形线

绘制下面的图形。提示，可多次调用 `plot2D` 和 `plotPara` 函数实现。



2.6、墨西哥草帽

绘制下面的图形。提示，调用 `plot3D` 和实现。



三、输入输出要求

3.1、输入

工具没有输入参数。但工程中应包括 `func1.cpp`、`func2.cpp`、`func3.cpp`、`func4.cpp`、`func5.cpp` 和 `func6.cpp`，6 个 `.cpp` 文件分别包含了任务 2.1-2.6 所绘制函数的 C/C++ 语言定义。

3.2、输出

输出为 6 个 `.bmp` 文件（`func1.bmp`、`func2.bmp`、`func3.bmp`、`func4.bmp`、`func5.bmp`、`func6.bmp`），分别对应任务 2.1-2.6 的绘制结果。

`.bmp` 文件格式参见百科资料（课堂上也会简单讲解）。

四、组队、提交要求及计分说明

4.1、组队要求

要求 2-4 人一组组队完成，推荐 3 人一组。每组选出一名队长。

4.2、提交要求

每组由队长在网络学堂提交一个.zip 压缩包。组员可以（但不必须）在网络学堂提交作业时说明队长姓名学号，但**不要提交附件**。发现组员在网络学堂提交附件的，将单独扣分。

压缩包内应包括但不限于：

1) 程序源文件。放置在 src 文件夹下，可以有多级目录，应包括全部工程编译所需文件。应包含 3.1 节中描述的 6 个.cpp 文件。

2) 说明文档。一个 Word 或 Pdf 文档，放置在压缩包根目录下，内容应包括但不限于：小组人员（姓名、学号、班级）、基本功能完成情况、扩展功能说明（可选）、分工情况（注意：这部分将作为小组内同学评分依据）。

3) 演示视频。放置在压缩包根目录下，内容应包括但不限于：小组人员展示、基本功能演示、扩展功能演示（可选）。建议长度不超过 3 分钟，大小不超过 30MB。

如果上述内容过大无法上传至网络学堂，可分别用一个内含有效链接的.txt 文件代替。

4.3、计分说明

1、要求主体使用 C/C++实现。完整实现第一、二、三节规定的全部任务就可以获得 100%的大作业分值。

2、允许调用其他库，允许调用其他语言编写的函数。但如这样做，应在说明文档中给出说明，并扣除由此节省的工作量所对等的分值。

3、本文档中，扩展功能指本文档中未说明的绘制函数功能（即，扩展本文档第一节内容）、更多的输入输出交互模式（即，扩展本文档第三节内容）；扩展任务指本文档中未说明的任务（即，扩展本文档第二节内容）。实现扩展功能或扩展任务有额外加分，分值视重要程度而定。

4、“**抑制内卷条款**”：实现扩展功能或扩展任务并不能获得超过 100%的大作业分值，但可以用来补足由于第 2 条规定导致的扣分。请各位同学在小组内充分沟通，根据小组同学的兴趣、能力和时间安排，选择是否实现扩展功能或扩展任务。