

专题：曼德布罗集

【背景说明】

(1) 曼德布罗集 (Mandelbrot Set) 是一个集合, 定义如下:

$M = \{C \mid \text{由迭代公式 } Z_{n+1} = Z_n^2 + C \text{ 推导的数列趋向收敛, 其中 } C \text{ 与 } Z_n \text{ 均为复数, 且 } Z_0 = 0\}$

对一个特定的复数 C , 初始 $Z_0=0$, 经过 $Z_{n+1}=Z_n^2+C$ 连续迭代后, 如果 Z_n 数列收敛于0附近 (不是发散到无穷远), 则 C 属于曼德布罗集。

(2) 由于复数可以对应到坐标平面上的点, 因此曼德布罗集可以直观地使用图形来展示, 如图所示, 图中黑色部分为曼德布罗集。

(3) 一般可以认为, 曼德布罗集中复数 C 的实部在-2.25至0.75之间, 虚部在-1.25至1.25之间, 在进行迭代计算中, 如果 Z_n 和原点0的距离超过3, 认为 Z_n 趋向发散, 如果连续经过256次迭代, 距离仍不超过3, 则认为 Z_n 趋向收敛 (即 C 属于曼德布罗集)。

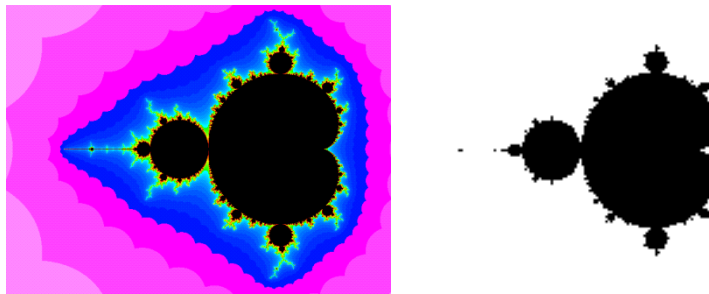


图1. 曼德布罗集的图形显示 (曼德布罗集是一种混沌图形, 局部和全局具有自相似性)

(4) 程序使用图像来显示曼德布罗集, 图像的分辨率一般有 640×480 (宽度上有640个像素, 高度上有480个像素)、 1920×1080 等, 每个像素由红绿兰3个颜色分量组成。本专题的任务是通过编程尽可能生成各种尺度下的曼德布罗集。

各问摘要:

第1问, 专题理解, 基础问题, 通过编程方式回答问题。

第2问, 程序调试, 迭代收敛判断程序, 结构体、收敛函数。

第3问, 资源使用, 基于自定义库函数及使用范例的图像生成程序。

第4问, 简单的曼德布罗集, 整合第2问第3问, 保持纵横比。

第5问, 彩色的曼德布罗集, 第4问基础上彩色着色。

第6问, 绚丽的曼德布罗集, 自由开放的着色方案。

【第1问, 专题理解, 编程回答基础问题】

求解曼德布罗集的第一步, 仔细阅读上述背景说明, 理解并回答以下基础问题。同时编写一个程序, 直接以printf语句输出问题的回答, 输出格式形如“(1)... (2)... (3)... (4)... (5)...”, 保存程序为C:\KS\Mand01.c。

(1) 由迭代公式, 当 $Z_0=0$ 且 $C=0$ 时, Z_n 趋向_____ (填写发散或收敛)。

(2) 设有复数 $z=3-4i$, 复数 z 与0的距离是_____ (填写数值, 保留一位小数)

(3) 观察提供的2个曼德布罗图形, 它的对称性是_____ (填写选项A/B/C/D)

A.上下对称 B.左右对称 C.中心对称 D.没有对称性

(4) 设某个图像的分辨率为 160×120 , 则该图像共有_____个像素 (填写整数)

(5) 图像的每个像素有3个分量, 分别为红色、绿色和_____ (填写颜色名称)

【第2问，程序调试，迭代收敛判断程序】

求解曼德布罗集的第二步，编写并验证迭代计算程序。某学生设计了复数结构体、连续迭代是否收敛的检测函数chkIteration、验证主函数main以及程序运行的预期运行结果，代码如下（参见Mand02Err.c），该程序**包含3个错误**，请调试该程序，使之正确运行。保存新程序为C:\KS\Mand02.c。

```
#include <stdio.h>
#include <math.h>
struct complex
{
    //定义复数结构体
    double real,    //实部
    double imag;    //虚部
};
int chkIteration(struct complex c, int n)
{
    //给定c，最多迭代n次，返回收敛或发散情况
    struct complex z1, z0={0,0};
    double d;
    int i;
    for (i=0; i<n; i++)
    {
        //每次迭代从z0计算得到z1，复数公式：z1=z0*z0+c
        z1.real = z0.real*z0.real - z0.imag*z0.imag + c.real;
        z1.imag = z0.real*z0.imag + z0.imag*z0.real + c.imag;
        printf("(%lg,%lg) ", z1.real, z1.imag); //中间结果，%lg格式符
        d = sqrt(z1.real*z1.real - z1.imag*z1.imag);
        //计算z1与原点(0,0)的距离，(实部的平方+虚部的平方)再开根
        if (d>3) //如果距离超过3，可以认为迭代结果为发散
            return i+1; //返回迭代次数
        z0 = z1; //计算得到的z1作为下次迭代的z0
    }
    return -1; //返回-1表示收敛
}
int main(void)
{
    struct complex c;
    int k;
    scanf("%lf %lf", &c.real, &c.imag);
    k = chkIteration(c,5); //为简化测试，暂时限制最大迭代5次
    printf("k=%d\n",k);
    return 0;
}
//输入：0 0，输出：(0,0) (0,0) (0,0) (0,0) (0,0) k=-1
//输入：2 -1，输出：(2,-1) (5,-5) k=2
//输入：1 0.5，输出：(1,0.5) (1.75,1.5) (1.8125,5.75) k=3
```

【第3问，资源使用，基于范例的图像生成程序】

求解曼德布罗集的第三步，编写一个图像生成程序。本专题提供了一个简单的图像生成库资源，并提供范例程序说明如何使用该资源，请参考范例，编写程序生成一个同心圆环的图像，并保存程序为C:\KS\Mand03.c。

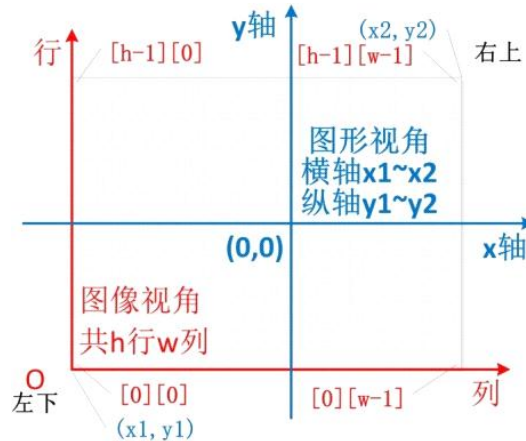


图2. 计算机图像与数学坐标平面的关系

方法概念：计算机将数学上图形转换为图像以方便处理，图像与图形的关系如上图所示，灰色部分为待处理的图形/图像。图示中兰色部分为图形视角，横坐标x轴，向右方向递增，纵坐标y轴，向上方向递增，图形左下角对应坐标(x1,y1)，右上角对应坐标(x2,y2)，坐标值可以为负数或小数。图示中红色部分为图像（.BMP格式）视角，图像将图形在横向上划分为w个像素（宽度上有0到w-1列），在图形纵向上划分h个像素（高度上有0到h-1行），图像左下角对应第0行第0列，图像右上角对应第h-1行第w-1列，图像中像素的行列值必须是非负整数，每个像素只能有一个颜色，颜色有红绿兰3个分量，一般取值0至255，全部为0时表示黑色，全部为255时表示白色。

资源范例：本专题提供一个简单的图像生成资源（见文件mybmp.h），同时提供一个范例程序（见文件bmpSample.c，程序结构如下），该文件能正确编译运行，运行后生成图像文件mpSample.BMP（.BMP图像文件一般可以双击打开），图像效果如下左图所示。范例图像的分辨率为160列×120行，图形左下角坐标为(-2.5,-2)，右上角坐标为(1.5,1)，图形以坐标(0,0)为中心，半径0.8内，4个象限的颜色依次为红、绿、兰、紫，半径外为青色。

//图像生成范例程序bmpSample.c

```
.....
#include "mybmp.h"          //包含图像生成资源库
//生成图像文件fn，w列h行，坐标左界x1，右界x2，下界y1，上界y2
void drawPicture(char *fn, int w, int h, double x1, double x2, double y1, double y2)
{
    .....
    bmpInitial(w, h);      //调用资源函数，初始化图像宽度和高度信息
    for (i=0; i<h; i++)    //逐行循环，图像下边对应0行
    {
        y = y1 + i*(y2-y1)/h; //计算第i行对应的y坐标
        for (j=0; j<w; j++)  //逐列循环，图像左边对应0列
        {
            x = x1 + j*(x2-x1)/w; //计算第j列对应的x坐标
            if (.....)
                bmpSetPixel(i, j, 0, 255, 255); //青色
            //调用资源函数，设置第i行第j列的颜色（着色）
        }
    }
    else .....
```

```

        bmpSetPixel(i, j, 255, 0, 0);        //红色
        .....
    }
}
bmpWriteToFile(fn);        //调用资源函数，生成图像文件
}
int main(void)
{
    int w=160, h=120;
    double x1=-2.5, y1=-2, x2=1.5, y2=1;
    drawPicture("bmpSample.BMP", w, h, x1, x2, y1, y2);
    return 0;
}

```

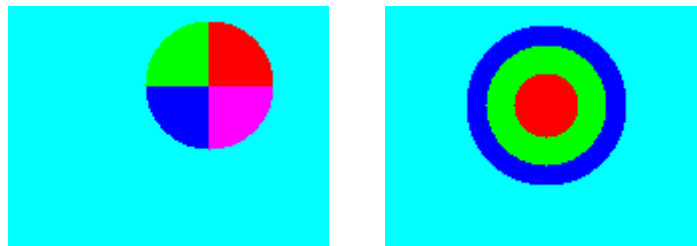


图3. 范例生成的图像与第3问需要生成的图像

编程功能：编写程序生成图像文件**Mand03.BMP**，图像分辨率由键盘输入，对应图形的坐标范围为左下(-2.5,-2.5)至右上(5.5,3.5)，以(1.5,1)为中心，半径0.8内红色，半径1.5内绿色，半径2.0内蓝色，其他青色。图像效果如上右图所示。保存新程序为**C:\KS\Mand03.c**。

运行示例：

输入：160 120

屏幕上输出（程序同时生成Mand03.BMP）：

fn=Mand03.BMP, w=160, h=120

x1=-2.5, x2=5.5, y1=-2.5, y2=3.5

【第4问，简单的曼德布罗集】

求解曼德布罗集的第四步，生成一个简单的黑白色曼德布罗图像。根据输入分辨率的不同，生成图像的细腻程度不同，如图所示。保存新程序为**C:\KS\Mand04.c**。

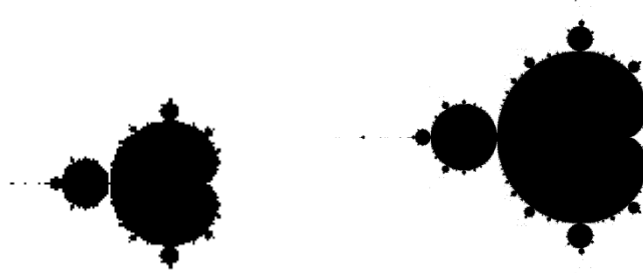


图4. 不同分辨率下的曼德布罗集

(1) 在**第3问**程序的基础上，增加**第2问**中的结构体定义和chkIteration函数，保存新程序为**C:\KS\Mand04.c**。

- (2) 优化函数chkIteration，删除不必要的输出语句，并省去sqrt调用，以提高运算速度。
- (3) 修改函数drawPicture。
- ① 双循环中计算出x,y坐标后，组成复数c，调用chkIteration (c,256)函数，如果函数返回结果为收敛，该像素着黑色，否则着白色。
- ② 统计图像中黑色的像素数量，以及黑色在所有像素中的占比并输出。
- ③ 为避免图像横纵两方向的比例失调，删除x2形参，由w:h = (x2-x1):(y2-y1) 计算比例因子 rxy=(y2-y1)/h，并重新计算坐标右界x2=x1+w*rxy。
- (4) 修改主函数，图像文件为Mand04.BMP，默认图像分辨率为160×120，默认坐标边界 x1=-2.25, y1=-1.25, y2=1.25。其中图像分辨率可以由键盘输入。
- (5) 运行示例：

输入：160 120，则输出：

fn=Mand04.BMP, w=160, h=120

x1=-2.25, x2=1.08333, y1=-1.25, y2=1.25, rxy=0.0208333

ct=3501:18.23%

【第5问，彩色的曼德布罗集】

彩色曼德布罗图像是在黑白图像的基础上，对图像中的发散区域进行着色，如图所示。根据chkIteration函数返回的迭代次数，使用不同的颜色加以标示，彩色的着色方案如表所示。在第4问的基础上修改，保存新程序为C:\KS\Mand05.c。

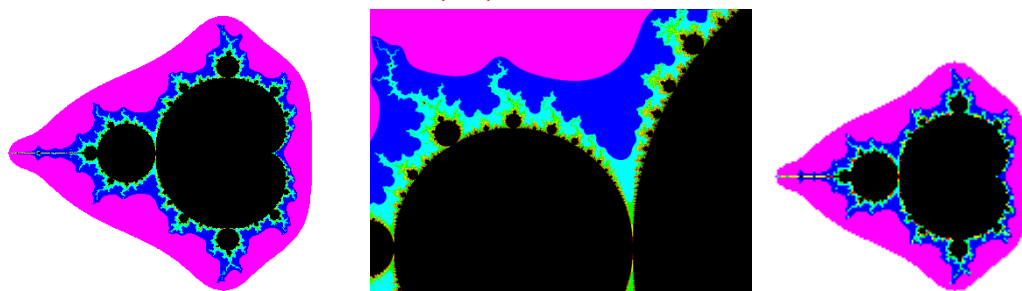


图5. 不同分辨率不同坐标位置的彩色曼德布罗集

序号	迭代次数	着色方案	红色分量	绿色分量	蓝色分量	备注
1	>256 收敛	黑	0	0	0	chkIteration 返回-1
2	>128	红	255	0	0	
3	>64	黄	255	255	0	黄=红+绿
4	>32	绿	0	255	0	
5	>16	青	0	255	255	青=绿+兰
6	>8	兰	0	0	255	
7	>4	紫	255	0	255	紫=红+兰
8	其他	白	255	255	255	白=红+绿+兰

(1) 修改drawPicture函数

- ① 在调用chkIteration函数后，根据返回值的大小，按上表进行着色。
- ② 统计图像中各种颜色出现的次数并输出。
- (2) 修改主函数，输出图像文件名称为Mand05.BMP，图像分辨率和坐标x1,y1,y2可以通过键盘输入。
- (3) 运行示例：

输入: 640 480 z (输入分辨率, 坐标使用默认)

输出:

fn=Mand05.BMP, w=640, h=480

x1=-2.25, x2=1.08333, y1=-1.25, y2=1.25, rxy=0.00520833

像素计数: 黑56112, 红597, 黄1300, 绿2588, 青6266, 兰17255, 紫55399, 白167683

输入: 1024 768 -1.3 -0.1 0.5 (输入分辨率和坐标值)

输出:

fn=Mand05.BMP, w=1024, h=768

x1=-1.3, x2=-0.5, y1=-0.1, y2=0.5, rxy=0.00078125

像素计数: 黑456276, 红5597, 黄11195, 绿23193, 青51885, 兰121501, 紫116785, 白0

输入: z

(所有输入使用默认数据)

输出:

fn=Mand05.BMP, w=160, h=120

x1=-2.25, x2=1.08333, y1=-1.25, y2=1.25, rxy=0.0208333

像素计数: 黑3501, 红39, 黄95, 绿173, 青390, 兰1060, 紫3465, 白10477

【第6问, 绚丽的曼德布罗集】

尽可能生成更加美丽细腻的曼德布罗图像, 如图所示。至少从以下方面进行优化, 在**第5问**的基础上修改, 保存新程序为**C:\KS\Mand06.c**, 生成的新图像文件形如**Mand6*.BMP** (每个文件不超过1M字节)。

- (1) 各种颜色要有更多梯度 (总颜色至少64种以上), 使层次感更加强烈。
- (2) 预调试4组以上坐标数据 (x_1, y_1, y_2), 逐步放大得到更细微的图像。
- (3) 输入分辨率和数据组号 (取值1至4), 组号不是1至4时, 继续输入坐标数据。
- (4) 生成的图像文件为Mand06_#.BMP (#为组号) 或Mand6.BMP (组号不是1至4时)。
- (5) 任何其他方面优化, 请在程序中以注释方式说明。

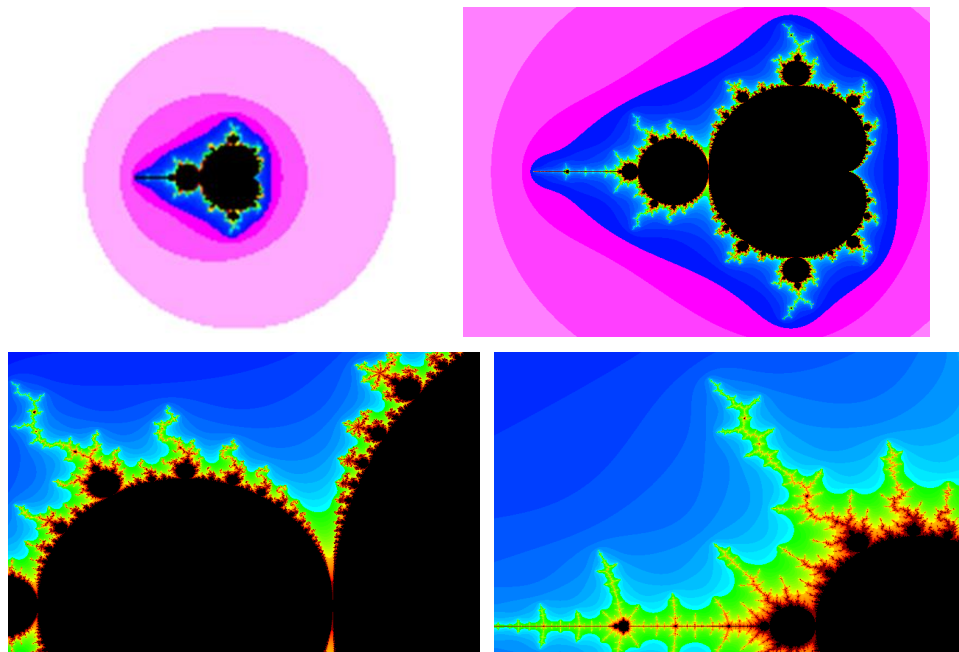


图6. 各种绚丽的曼德布罗集