

# 微分方程数值解 Project2 作业报告

褚朱钊恒

3200104144

## 1 运行说明

本项目需要调用jsoncpp与eigen3库，故请在运行此项目前安装好这两个包。您可以使用以下命令进行安装：

```
sudo apt-get install libeigen3-dev
sudo apt-get install libjsoncpp-dev
```

在project目录下使用make命令即可编译运行整个项目并得到实验报告。

## 2 程序设计思路

实现多重网格法计算的相关代码在头文件multigrid.h中，其中主要设计了 `template <int dim> class Multigrid_Method`

根据 `dim` 的取值（1 或 2），`Multigrid_Method` 类将被特化成求解一维或二维的边值问题的类。

### 2.1 函数

- 构造函数：需要边界条件，限制算子，插值算子，迭代算法，停止条件，网格粗细，初始解，停止条件参数，最粗网格作为参数
- `laplace`：对一个网格上的数值作拉普拉斯变换
- `restriction`：限制算子，实现了 `injection` 和 `full_weighting` 两种算法
- `interpolation`：插值算子，实现了 `linear` 和 `quadratic` 两种算法
- `jacobi`：带权的雅各比迭代，其中参数  $\omega = \frac{2}{3}$
- `Vcycle`与`FMG`：两种迭代算法
- `accuracy`：计算当前解的相对精度
- `error\_norm`：计算当前解的误差范数
- `residual\_norm`：计算当前解的残差范数
- `solve`：制定  $v_1$  与  $v_2$  并进行求解

### 2.2 参数与变量说明

#### 2.2.1 bound\_conditon

实现了三种边界条件：Dirichlet, Neumann, mixed。

其中，纯 Neumann 边界条件由于没有唯一解，计算时给定了区域中心点的具体点值使解唯一。

mixed 条件在一维时为左端点满足 Dirichlet 条件，右端点满足 Neumann 边界条件，二维时为上下边界满足 Dirichlet 条件，左右边界满足 Neumann 边界条件

### 2.2.2 stopping\_criteria

实现了两种条件: max\_iteration, rela\_accuracy.

前者为限制最大迭代次数为 st\_parm, 后者为迭代至相对误差小于 st\_parm。

但为了避免死循环和减少不必要的迭代, 当迭代时误差的收敛速度小于 1.01 时, 也会停止迭代。

## 3 程序测试结果

我选择的测试用的函数为:

- $f(x, y) = e^{x+\sin(y)}$
- $f(x, y) = \sin(3x + 3y)$
- $f(x, y) = e^{(x^3 + y^3)}$