

微分方程数值解 Project1 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。在project目录下使用make命令即可编译整个项目并得到实验报告。

2 程序设计思路

所有实现样条计算的相关代码都在头文件BVP.h中，其中设计了一下几个类：

2.1 Class Function

其定义了(),diff_x,diff2_x,diff_y,diff2_y, laplace六个虚函数，分别用于函数求值，求导，求二阶导和求拉普拉斯算子。

2.2 Class point

定义了平面上的向量运算和求向量长度。

2.3 Class Circle

定义了圆与直线的求交算法

2.4 Class FD_Methods

有限差分法的虚类，定义了solve(),error_norm三个虚函数，分别用于求解，求值和求误差的范数。该基类有以下两个派生类：

- FD_regular 使用有限差分法在单位正方形上进行微分方程的求解
- FD_irregular 使用有限差分法在单位正方形内部去掉一个圆的区域上进行微分方程的求解

两者都可以输入以下参数：

- h: 网格粗细
- f: 计算的微分方程的真实解
- cond: 边值条件（1 表示 Dirichlet, 2 表示 Neumann, 3 表示混合条件）

FD_irregular 则需要一个额外的参数 C，表示去掉的圆的区域。

3 程序测试结果

我选择的测试用的函数为：

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

3.1 整体效果

由于单位正方形上的图像比较方便绘制，所以这里展示部分单位正方形上的求解结果。对第一个函数 $f(x, y) = e^{x+\sin(y)}$ 使用 Dirichlet 边值条件求解，结果如下：

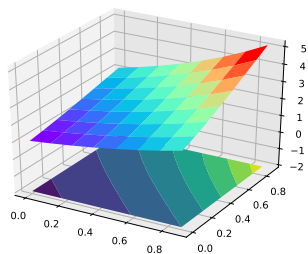


图 1: $n=8$

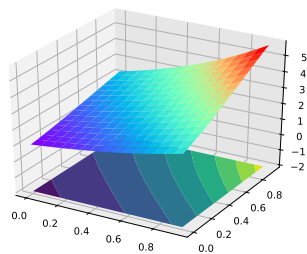


图 2: $n=16$

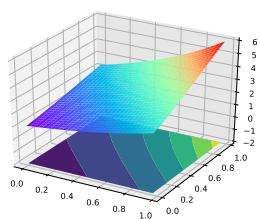


图 3: $n=32$

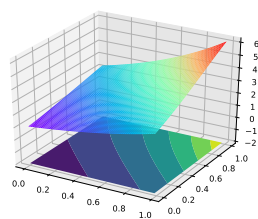


图 4: $n=64$

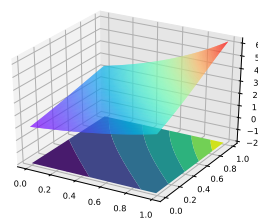


图 5: $n=128$

对第二个函数 $f(x, y) = \sin(3x + 3y)$ 使用 Neumann 边值条件求解，结果如下：

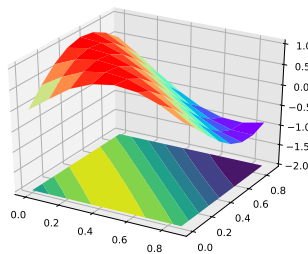


图 6: $n=8$

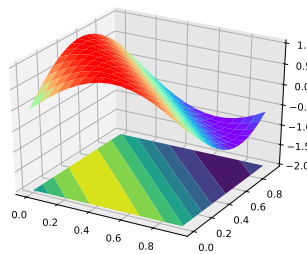


图 7: $n=16$

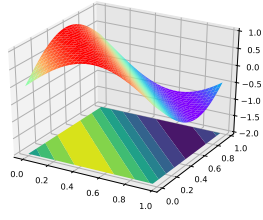


图 8: $n=32$

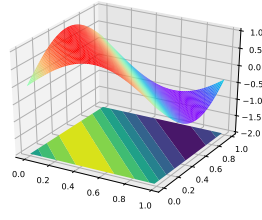


图 9: $n=64$

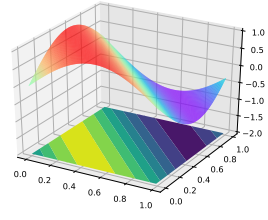


图 10: $n=128$

对第三个函数 $f(x, y) = e^{(x^3 + y^3)}$ 使用混合边值条件（一条边 Neumann 边值条件，三条边 Dirichlet 边值条件）求解，结果如下：

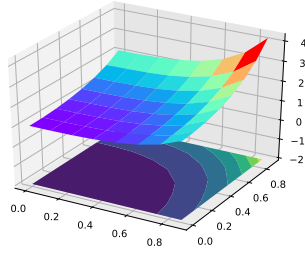


图 11: $n=8$

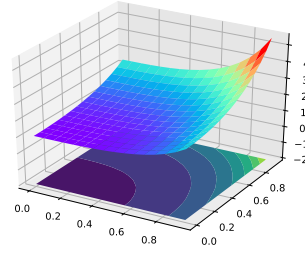


图 12: $n=16$

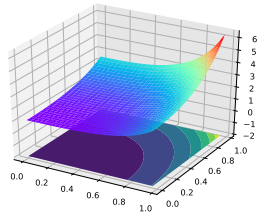


图 13: $n=32$

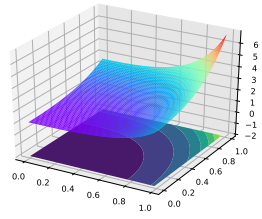


图 14: $n=64$

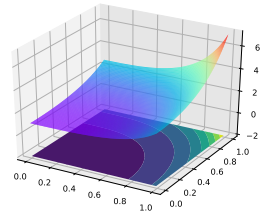


图 15: $n=128$

3.2 固定点收敛速率

我们取 $(0.25, 0.25)$, $(0.25, 0.75)$, $(0.75, 0.25)$, $(0.75, 0.75)$ 四个固定点，计算他们的平均误差，加细网格计算其误差变小的速率。结果如下：

对于正则化区域，效果如下：

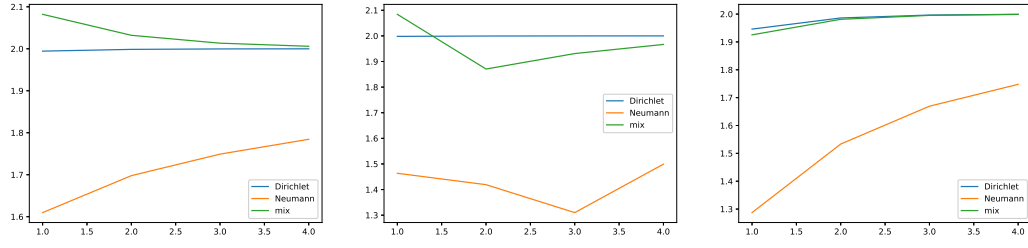


图 16: $f(x, y) = e^{(x^3 + y^3)}$ 图 17: $f(x, y) = \sin(3x+3y)$ 图 18: $f(x, y) = e^{(x^3 + y^3)}$

基本可以认为三个方程中三个边值条件的固定点误差收敛速度关于 h 都是 2 阶的，但可能由于纯 Neumann 条件给出的系数矩阵的条件数较大，导致了算法的不稳定性变大，其收敛的速度明显小于另外两个条件。

对于非正则化区域，我们设挖去的圆形圆心为 $(0.5, 0.5)$ ，半径为 $r = 0.2$ ，效果如下（由于纯 Neumann 条件的代码尚未完成，缺少对圆上的 Neumann 条件的处理，所以此处只有两条曲线）：

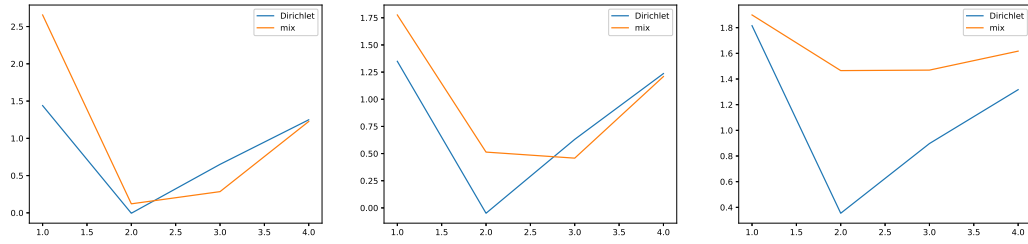


图 19: $f(x, y) = e^{(x^3 + y^3)}$ 图 20: $f(x, y) = \sin(3x+3y)$ 图 21: $f(x, y) = e^{(x^3 + y^3)}$

从整体趋势来看，收敛速度应该也是二阶的，但在误差比较大，可能是对于圆附近的 ghost 点的处理不太优美（目前的处理办法是在与坐标轴平行一个方向上找一个网格上的点，再圆上的点与网格上的点的值线性差值得到 ghost 点的约束条件）。

3.3 误差范数及其收敛分析

3.3.1 正则化区域上的结果

分别对三个方程使用 Dirichlet, Neumann, mix 条件求解，其误差范数收敛速度如下：

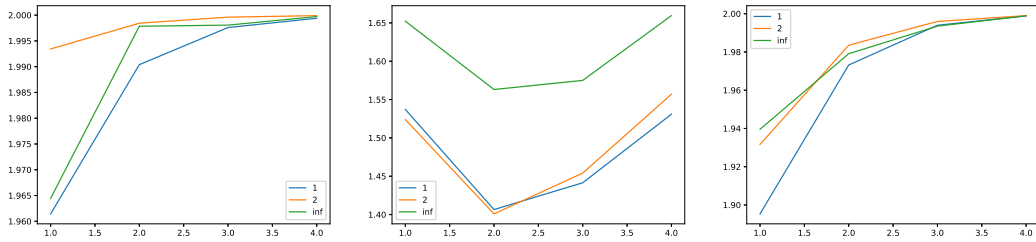


图 22: $f(x, y) = e^{(x^3 + y^3)}$ 图 23: $f(x, y) = \sin(3x+3y)$ 图 24: $f(x, y) = e^{(x^3 + y^3)}$

3.3.2 非正则化区域上的结果

分别对三个方程使用 mix 条件求解，其误差范数收敛速度如下：

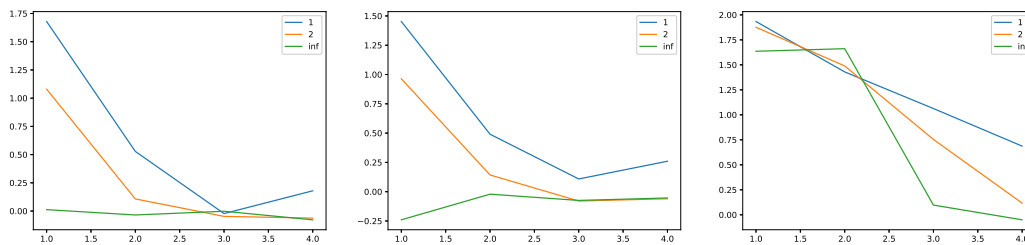


图 25: $f(x, y) = e^{(x^3 + y^3)}$ 图 26: $f(x, y) = \sin(3x+3y)$ 图 27: $f(x, y) = e^{(x^3 + y^3)}$

发现收敛速度一开始是解决二阶的，后来逐渐变的不收敛，可能是计算机的浮点误差导致的影响。