微分方程数值解法 第一次 实验报告

信息与计算科学
15336204
邢剑飞

一、问题

利用有限元方法解微分方程

$-\frac{d}{dx}(p\frac{du}{dx})+qu=f$

$where\ u(a)=0,\ u'(b)=0$

二、理论分析

为方便表示，用以下方程代替上述微分方程

$-(pu')'+qu=f$

两端同时乘v,化为变分形式

$\int_a^b(-p'u'-pu''+qu)vdx=\int_a^b fvdx$

$-\int_a^b p'u'vdx-\int_a^b pu''vdx+\int_a^b quvdx=\int_a^b fvdx$

利用分部积分公式，进一步化简

$-\int_a^b p'u'vdx-(u'pv|_a^b-\int_a^b(pv)'u'dx)+\int_a^b quvdx=\int_a^b fvdx$

$-\int_a^b p'u'vdx-u'pv|_a^b+\int_a^b p'u'vdx+\int_a^b pu'v'dx+\int_a^b quvdx=\int_a^b fvdx$

$-u'pv|_a^b+\int_a^b pu'v'dx+\int_a^b quvdx=\int_a^b fvdx$

代入边界条件得

$$\int_a^b pu'v'dx + \int_a^b quvdx = \int_a^b fvdx$$

由插值公式知道

$$u(x) = \Sigma_{i=1}^n u_i \varphi_i(x)$$

其中，$\varphi_i(x)$为插值基函数

将上式代入原式中，令$v(x) = \varphi(x)$，得到

$$\int_a^b p(\Sigma_{i=1}^n u_i \varphi_i(x))' \varphi_j'(x)dx + \int_a^b q\Sigma_{i=1}^n u_i \varphi_i(x)\varphi_j(x)dx = \int_a^b f\varphi_j(x)dx$$

整理得

$$\Sigma_{i=1}^n u_i(\int_a^b p\varphi_i'\varphi_j'dx + \int_a^b q\varphi_i\varphi_j dx) = \int_a^b f\varphi_j dx$$

因此，函数的解可以表示为

$$Au = F$$

其中，

$$u = (u_1, u_2, \cdots, u_n)^T$$

$$F = ((f, \varphi_1), (f, \varphi_2), \cdots, (f, \varphi_n))^T$$

$$A = A^1 + A^2$$

A_1为刚度矩阵，A_2为质量矩阵，表示为

$$A_{ji}^1 = \int_a^b p\varphi_i'\varphi_j'dx$$

$$A_{ji}^2 = \int_a^b q\varphi_i\varphi_j dx$$

A为三对角矩阵，一定是可逆的，计算时只要求解以下方程

$$u = A^{-1}F$$

便可以得到$u(x)$在x_i处的近似值

三、实验相关说明

1. 开发环境: Linux version 4.10.0-42-generic
2. 语言: Python >= 3.5.0
3. 目录树：

```
.
├── static/ # 静态文件目录
├── config.conf # 配置文件，包含具体参数，如目标端点值，步长等
├── config_parser.py # 配置文件驱动
├── function.py # 函数信息配置，运行时需要手动进行修改
├── index.py # 入口文件
├── integration.py # 功能：数值积分
├── README.md # 运行指南
└── requirement.txt # 依赖包记录
```

全部实验代码存放于

https://github.com/Jianfei2333/difference_equation

四、运行说明

运行前，可以手动调整各个函数及参数的值。函数存放于/function.py中，参数存放于/config.conf中。

其中函数配置部分包括待求函数真实表达式，其一阶、二阶导数表达式；$p(x)$, $q(x)$表达式及$p(x)$的一阶导数表达式。这些函数需要同步进行(手动)修改，如果不

匹配可能导致很糟糕的结果。其他部分通过调用方式使用上面的函数，不需要进行手动的修改。

由于版本的关系，一些依赖包和语法在Python2.7及以下不能正常运行，请确保在Python3.5及以上的环境下使用pip安装依赖和运行程序。

五、实验结果

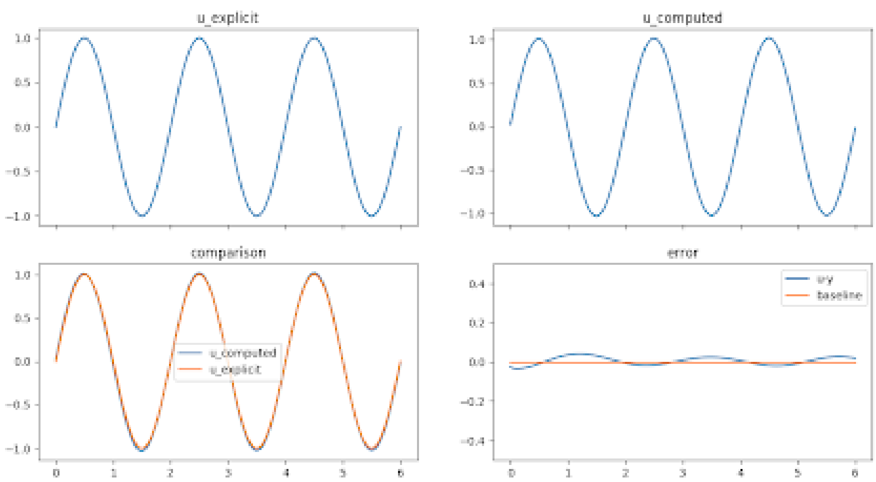
使用的测试函数为

$u(x) = sin(\pi x)$

$p(x) = x^2$

$q(x) = sin(e^x)$

得到结果如下图：



图左上：$u(x)$实际图像

图右上：经过解微分方程得到的$u(x)$图像

图左下：两个图像合并于同一坐标系下图像

图右下：误差图像($u - y$关于x)

拟合度较好，平均误差值约为0.0063

...一些题外话

Python环境真的难搭Orz，各种版本冲突语法规则冲突，如果实在不能运行的话可以发邮件给我2333，邮箱：xjf999999@hotmail.com

BTW虽然感觉有点紧张但是助教姐姐依然很努力地在讲课啦！其实讲得非常清楚了(虽然大家好像都没有什么反应...其实所有课都是这个状况2333 反正希望助教姐姐以后也一定要加油啦！(≧∇≦)ﾉ

2018年**03**月**31**日 星期六 **17:54:13 CST**