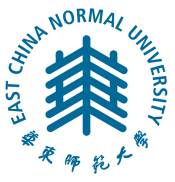
2020届研究生硕士学位论文

分 类 号：O241.6　　学校代码：10269

密　　级：　　学　　号：51170601007



華東師範大學

East China Normal University

硕士学位论文

MASTER'S DISSERTATION

论文题目：关于时间-空间分数阶扩散方程的预处理方法研究与探索

院　　系：数学系

专　　业：计算数学

研 究 方 向：数值代数

指 导 老 师：潘建瑜 教授

学位申请人：管国祥

2020 年 4 月

Dissertation for master degree in 2020 University Code: 10269

Student ID: 51170601007

East China Normal University

Title: Research and Exploration on Precondition of Time-Space Fractional Diffusion Equations

Department: Mathematics

Major: Computational Mathematics

Research Direction: Numerical Algebra

Supervisor: Jianyu Pan (Professor)

Candidate: GuanXiang Guan

April, 2020

华东师范大学学位论文原创性声明

郑重声明：本人呈交的学位论文《论文标题》，是在华东师范大学攻读硕士/博士（请勾选）学位期间，在导师的指导下进行的研究工作及取得的研究成果。除文中已经注明引用的内容外，本论文不包含其他个人已经发表或撰写过的研究成果。对本文的研究做出重要贡献的个人和集体，均已在文中作了明确说明并表示谢意。

作者签名：_____ **日　期：** 年　月　日

华东师范大学学位论文著作权使用声明

《论文标题》系本人在华东师范大学攻读学位期间在导师指导下完成的硕士/博士（请勾选）学位论文，本论文的著作权归本人所有。本人同意华东师范大学根据相关规定保留和使用此学位论文，并向主管部门和学校指定的相关机构送交学位论文的印刷版和电子版；允许学位论文进入华东师范大学图书馆及数据库被查阅、借阅；同意学校将学位论文加入全国博士、硕士学位论文共建单位数据库进行检索，将学位论文的标题和摘要汇编出版，采用影印、缩印或者其它方式合理复制学位论文。

本学位论文属于（请勾选）

(　　) 1. 经华东师范大学相关部门审查核定的“内部”或“涉密”学位论文 *，于　　年　　月　　日解密, 解密后适用上述授权。

(　　) 2. 不保密，适用上述授权。

导师签名：_____ **本人签名：**_____

年　　月　　日

*“涉密”学位论文应是已经华东师范大学学位评定委员会办公室或保密委员会审定过的学位论文（需附获批的《华东师范大学研究生申请学位论文“涉密”审批表》方为有效），未经上述部门审定的学位论文均为公开学位论文。此声明栏不填写的，默认为公开学位论文，均适用上述授权）。

管国祥 硕士学位论文答辩委员会成员名单

姓 名	职 称	单 位	备 注
XXX	教授	XXXXX 大学数学系	主席
XXX	教授	XXXXX 大学数学系	
XXX	教授	XXXXX 大学数学系	

摘　要

由于分数阶导数的非局部性质, 它可以很好地用来描述物理中的一些非正常扩散现象, 故而基于分数阶导数的模型近些年来被越来越多地应用于科学和工程计算领域. 但也正是因为分数阶导数的非局部性质, 离散后得到的代数方程组的系数矩阵往往是稠密的, 但问题规模较大时, 这给问题的数值求解带来了很大的困难. 传统整数阶微分方程的快速算法不再适用于分数阶微分方程. 近年来, 针对分数阶扩散方程的快速算法的研究受到了越来越多的关注.

本文的主要研究内容如下:

(1) 我们考虑了时间-空间分数阶扩散方程的预处理快速算法, 通过时间上的 Grünwald 差分方法离散和空间上的带平移 Grünwald 差分方法离散, 并经过整理, 原问题最终转化为一个大规模的代数方程组, 其系数矩阵可表示为两个 Kronecker 乘积之和.

(2) 基于系数矩阵的特殊结构, 我们结合 Toeplitz 矩阵求逆方法构造了一类块对角预处理子, 并进行了理论分析. 数值实验表明, 当时间分数阶导数较小时, 比如不超过 0.5 时, 该预处理子具有很好的加速效果.

(3) 观察到系数矩阵由两部分组成, 而且每一部分都具有特殊的结构, 因此我们提出了一类基于交替方向思想的矩阵分裂预处理子, 并对其进行了理论分析. 在具体实施时, 为了减少运算量, 我们分别考虑了用循环矩阵近似和 Toeplitz 矩阵求逆两种方法, 并进行了数值测试.

关键词: 时间-空间分数阶扩散方程, Toeplitz 矩阵求逆, 交替方向迭代, 预处理

ABSTRACT

Due to the non-local nature of fractional derivatives, the fractional diffusion equations can be used to describe some abnormal diffusion phenomena in physics. Therefore, models based on fractional derivatives have been increasingly used in the fields of scientific and engineering computing in recent years. But it is also because of the non-local nature of fractional derivatives, the coefficient matrix of the discrete linear system is often dense. When the scale of the problem is large, it brings great difficulty to compute the numerical solution of the problem. Fast algorithms for traditional integer-order differential equations are no longer applicable to fractional-order differential equations. In recent years, research on fast algorithms for fractional diffusion equations has received more and more attentions.

The main contributions of this thesis are as follows:

(1) We consider the numerical methods for the time-space fractional diffusion equations. Discretized by the Grünwald difference method in time and the shifted Grünwald difference method in space, the original problem was finally transformed into a large-scale system of algebraic linear equations. Its coefficient matrix can be expressed as a sum of two Kronecker products.

(2) Based on the special structure of the coefficient matrix, we combine inversion method of Toeplitz matrix and propose a class of block diagonal preconditioner. The properties of the preconditioned coefficient matrix are studied. Numerical experiments show that when the time fractional derivative is small, such as not exceeding 0.5, the preconditioner has a good acceleration effect.

(3) We observe that the coefficient matrix consists of two parts naturally, and each part has a special structure. Therefore, we propose a class of matrix splitting preconditioner based on the idea of alternating directions. Some theoretical results are presented. In order to reduce the amount of workload, in the real applications, we make use of the technique of circulant matrix approximation and inversion method of Toeplitz matrix. Numerical tests were carried out to show the performance of the two preconditioners.

Keywords: Time-space fractional diffusion equation, Inversion of Toeplitz matrix, Alternating direction iteration, Preconditioning

目 录

第一章　引言

1.1　问题介绍

本文主要研究如下的时间-空间分数阶扩散方程的初边值问题

$$\begin{cases} {}_{0}^{\mathrm{C}}\mathrm{D}_{t}^{\alpha}u(x,t) = d_{+}(x)\,{}_{a}\mathrm{D}_{x}^{\beta}u(x,t) + d_{-}(x)\,{}_{x}\mathrm{D}_{b}^{\beta}u(x,t) + f(x,t), \\ \qquad\qquad\qquad\qquad\qquad\qquad (x,t) \in (a,b)\times(0,T], \\ u(a,t) = u(b,t) = 0, \qquad t \in [0,T], \\ u(x,0) = 0, \qquad x \in [a,b], \end{cases}$$

其中 $\alpha \in (0,1)$, $\beta \in (1,2)$ 分别是时间分数阶导数和空间分数阶导数, 函数 $d_{\pm}(x) \geq 0$ 是扩散系数, $f(x,t)$ 是右端项, ${}_{0}^{\mathrm{C}}\mathrm{D}_{t}^{\alpha}$ 是 Caputo 分数阶导数, 定义如下:

$${}_{0}^{\mathrm{C}}\mathrm{D}_{t}^{\alpha}u(x,t) = \frac{1}{\Gamma(1-\alpha)}\int_{0}^{t}\frac{\partial u(x,s)}{\partial s}(t-s)^{-\alpha}\mathrm{d}s,$$

${}_{a}\mathrm{D}_{x}^{\beta}$ 和 ${}_{x}\mathrm{D}_{b}^{\beta}$ 分别是 Riemann-Liouville 左、右分数阶导数, 定义如下:

$${}_{a}\mathrm{D}_{x}^{\beta}u(x,t) = \frac{1}{\Gamma(2-\beta)}\frac{\partial^{2}}{\partial x^{2}}\int_{a}^{x}\frac{u(\xi,t)}{(x-\xi)^{\beta-1}}\mathrm{d}\xi,$$

$${}_{x}\mathrm{D}_{b}^{\beta}u(x,t) = \frac{1}{\Gamma(2-\beta)}\frac{\partial^{2}}{\partial x^{2}}\int_{x}^{b}\frac{u(\xi,t)}{(\xi-x)^{\beta-1}}\mathrm{d}\xi,$$

其中 $\Gamma(\cdot)$ 是 Gamma 函数.

1.2　研究现状

分数阶扩散方程作为整数阶微分方程的推广, 早期主要局限于纯数学理论的研究. 直到最近的几十年, 人们在很多领域发现了反常扩散现象, 比如流体力学, 材料力学, 图像处理, 生物学, 金融, 信号处理以及控制论等. 在传统的整数阶微分方程不能够很好地描述这些反常扩散现象的背景下, 分数阶微分方程由于其非局部性质, 非常适合描述这些反常扩散现象 [4, 16], 于是逐渐引起了学者的重视并广泛地被应用到了各个领域, 具体可参考 [1, 5, 21, 29, 36, 42]. 由于在大多数情形下, 求分数阶扩散方程解析解是不可行的, 故而分数阶扩散方程的数值求解方法研究得到了快速发展.

由于分数阶算子的非局部性质, 经过数值离散后, 我们得到的线性方程组的系数矩阵往往都是稠密的, 这就导致了分数阶扩散方程数值求解的时间复杂度和空间复杂度都很高. 考虑将分数阶扩散方程离散成时间向前的形式, 对于每一个时间步, 如果使用直接法, 则时间复杂度为 $\mathcal{O}(N^3)$, 空间复杂度为 $\mathcal{O}(N^2)$, 其中 N 为空间的剖分密度. 对于大规模问题, 这样的时间和空间复杂度显然是让人无法接受的. 幸运的是, 有学者研究发现, 使用特定的离散方法后, 比如带平移的 Grünwald 离散方法 [50], 系数矩阵虽然是稠密的, 但是具备一定的 Toeplitz 特殊结构 [51], 即若干对角矩阵与 Toeplitz 矩阵的组合. 这样我们只需要储存系数矩阵的部分元素, 如其中的 Toeplitz 矩阵只需存放其第一行和第一列即可, 于是我们的空间复杂度就能降低到了 $\mathcal{O}(N)$. 再借助快速 Fourier 变换, 系数矩阵和向量的乘积运算量也降低到了 $\mathcal{O}(N\log(N))$. 于是, 自然地可以考虑使用 Krolov 子空间迭代算法来求解该问题, 此时问题每一时间步的时间复杂度降低到了 $\mathcal{O}(\ell N\log(N))$, 其中 ℓ 是每个时间步求解线性方程组的迭代步数. 但是分数阶微分方程离散出来的系数矩阵往往是坏条件的, 而且特征值发布也不聚集, 这通常会导致我们的算法的迭代步数 ℓ 非常大, 也就意味着时间复杂度的增加. 因此 Krolov 子空间迭代算法的预处理技术就变得至关重要, 于是大量基于快速 Fourier 变换和预处理技术的快速算法被提出来 [3, 19, 28, 31, 34, 40, 41, 45, 49–51].

针对时间-空间分数阶微分方程, 目前的预处理方法按照系数矩阵离散出来的形式主要有传统的按时间步地针对每个时间步的系数矩阵的预处理方法, 比如 [24, 25, 31, 33, 41, 51**?** , 52]. 近些年也有将问题离散形式写成时间离散和空间离散联结起来的块下三角矩阵的形式. 比如 Ke et al.[30] 将块向前取代方法和分治法相结合来解决块下三角其中矩阵块为三对角矩阵的问题. 该方法的时间复杂度为 $\mathcal{O}(MN\log^2 M)$, 空间复杂度为 $\mathcal{O}(MN)$, 其中 M 是时间步数. Lu et al.[35] 针对块下三角 Toeplitz 矩阵 (其中矩阵块为三对角矩阵) 提出了一种快速的近似逆的方法, 时间复杂度为 $\mathcal{O}(MN\log(M))$, 空间复杂度为 $\mathcal{O}(MN)$. 这种方法主要思想是用块 ϵ-循环矩阵来近似系数矩阵. Huang 和 Lei [27] 将分治法和 Toeplitz 矩阵逆的循环和斜循环分解来求解非奇异的块下三角 Toeplitz 矩阵, 其中矩阵块为稠密的 Toeplitz 矩阵, 方法的时间复杂度为 $\mathcal{O}(MN\log M(\log M+\log N))$.

1.3 本文工作

本文主要考察的是时间-空间分数阶扩散方程的数值求解. 通过时间差分离散和空间差分离散的联结, 将原问题就转化为一个代数方程组, 也就是说, 将所有时间步整合在一起, 这样就不需要每个时间步都求解一个方程. 通过整理, 其系数矩阵可写为两个 Kronecker 乘积相加的形式. 本文的主要研究内容是结合系数矩阵的整体结构特征, 构造有效的预处理方法.

本文的具体安排如下:

在第二章, 我们给出本文所需要使用到的一些准备知识, 包括分数阶导数的定义, Toeplitz 矩阵和循环矩阵的性质, 快速 Fourier 变换, Toeplitz 矩阵直接求逆方法等.

在第三章, 我们给出时间-空间分数阶扩散方程的离散形式, 分别对时间分数阶导数和空间分数阶导数使用 Grünwald 和带平移的 Grünwald 差分离散方法, 给出代数方程组的系数矩阵结构.

在第四章, 我们针对系数矩阵的结构特征, 构造出不同的预处理方法.（1）我们首先讨论了简单循环预处理子, 数值结果表明, 该预处理子具有较好的加速效果.（2）其次, 我们结合 Toeplitz 矩阵求逆方法构造了一类块对角预处理子, 并进行了理论分析. 数值实验表明, 当时间分数阶导数较小时, 该预处理子具有更好的收敛效果.（3）最后, 观察到系数矩阵由两部分组成, 而且每一部分都具有特殊的结构, 因此我们提出了一类基于交替方向思想的矩阵分裂预处理子, 并对其进行了理论分析. 在具体实施时, 为了减少运算量, 我们分别考虑了用循环矩阵近似和 Toeplitz 矩阵求逆两种方法, 并进行了数值测试.

在第五章, 我们对本文的内容进行了总结, 并对未来可以进一步探讨的问题做了展望.

第二章　准备工作

为了更好地表述本文所研究的问题以及用到的方法, 本章主要介绍一些预备知识, 其中包括分数阶导数的定义, Toeplitz 矩阵和循环矩阵, 快速 Fourier 变换, Toeplitz 矩阵直接求逆等相关知识.

2.1　分数阶导数

为了方便理解分数阶积分, 我们先给出 Gamma 函数的定义.

定义 2.1　Gamma 函数定义为

$$\Gamma(x) \triangleq \int_0^{\infty} t^{x-1}e^{-t}\mathrm{d}t, \quad \mathrm{Re}(x) > 0,$$

其中 $t^{x-1} \triangleq e^{(x-1)\ln(t)}$.

下面我们给出 Gamma 函数的渐近表达式.

定理 2.1　**(Stirling Asymptotic Formula)** 设 $x \in \mathbb{R}$, 则

$$\Gamma(x) = \sqrt{2\pi}x^{x-\frac{1}{2}}e^{-x}\left(1+\mathcal{O}\left(\frac{1}{x}\right)\right), \quad x \to +\infty.$$

下面是 Gamma 函数的一些常见性质:

(1) 当 $z \to 0^+$ 时, $\Gamma(z) \to +\infty$;

(2) $\Gamma(z)\Gamma(z+\frac{1}{2}) = 2^{1-2z}\sqrt{\pi}\Gamma(2z)$;

(3) $\Gamma(z)\Gamma(z+\frac{1}{n})\Gamma(z+\frac{2}{n})\cdots\Gamma(z+\frac{n-1}{n}) = (2\pi)^{\frac{n-1}{2}}n^{\frac{1}{2}-nz}\Gamma(nz)$;

(4) $\Gamma(n+\frac{1}{2}) = \frac{(2n)!\sqrt{\pi}}{4^n n!}$.

本节我们将介绍常见的三种分数阶导数: Riemann-Liouville 分数阶导数, Caputo 分数阶导数以及 Grünwald-Letnikov 分数阶导数. 更多关于分数阶导数的定义可以参考 [43].

分数阶导数可以通过整数阶导数和分数阶积分相结合来表示. 我们首先给出分数阶积分的定义.

定义 2.2 设 $\alpha > 0$, 且有 $f(x) \in L_1[a,b]$. 则 $f(x)$ 的 α 阶积分定义为

$$ {}_a\mathrm{D}_x^{-\alpha} f(x) \triangleq \frac{1}{\Gamma(\alpha)} \int_a^x (x-t)^{\alpha-1} f(t) \mathrm{d}t. \tag{2.1} $$

注意到, 上面定义的分数阶积分实际上对任意正实数都有意义. 当我们将 α 取正整数时, 则 ${}_a\mathrm{D}_x^{-\alpha}$ 就是通常意义下的整数阶积分.

下面的引理表明分数阶积分算子具有可交换性.

引理 2.2 [44] 设 $\alpha > 0, \beta > 0, f(x) \in L_p[a,b]$, 其中 $1 \le p < \infty$, 则

$$ ({}_a\mathrm{D}_x^{-\beta}\, {}_a\mathrm{D}_x^{-\alpha}) f(x) = {}_a\mathrm{D}_x^{-(\beta+\alpha)} f(x) $$

在 $[a,b]$ 上几乎处处成立. 如果进一步有 $f(x) \in C[a,b]$ 或 $\alpha + \beta \ge 1$ 则该式对任意的 $x \in [a,b]$ 都成立.

2.1.1 Riemann-Liouville 分数阶导数

Riemann-Liouville 分数阶导数在历史上出现得较早, 目前关于该分数阶导数的理论研究得比较完善.

定义 2.3 设 α 是大于 0 的任意正实数, 且 n 是大于 α 的正整数, 满足 $n-1 \le \alpha < n$, 则 $f(x)$ 的 R-L 分数阶导数定义为

$$ {}^{\mathrm{RL}}_{a}\mathrm{D}_x^{\alpha} f(x) \triangleq \mathrm{D}^n({}_a\mathrm{D}_x^{\alpha-n} f(x)) = \frac{1}{\Gamma(n-\alpha)} \frac{\mathrm{d}^n}{\mathrm{d}x^n} \int_a^x \frac{f(t)}{(x-t)^{\alpha-n+1}} \mathrm{d}t. \tag{2.2} $$

即先求 $n-\alpha$ 阶积分, 然后再求 n 次导数.

特别地, 有

$$ {}^{\mathrm{RL}}_{a}\mathrm{D}_x^{0} f(x) = f(x), \qquad {}^{\mathrm{RL}}_{a}\mathrm{D}_x^{n} f(x) = f^{(n)}(x). $$

因为 $f(x)$ 在点 x 处的分数阶导数是通过 $f(x)$ 的左半区间 $[a,x]$ 来表示的. 所以 (2.2) 所定义的分数阶导数通常被称为 R-L 左分数阶导数. 相应地, 我们也可以通过 $f(x)$ 的右半区间 $[x,b]$ 来表示 R-L 右分数阶导数.

定义 2.4 (右分数阶积分) 设 $\alpha > 0$, 且有 $f(x) \in L_1[a,b]$. 则 $f(x)$ 的右 α 阶积分定义为

$$ {}_x\mathrm{D}_b^{-\alpha} f(x) \triangleq \frac{1}{\Gamma(\alpha)} \int_x^b (t-x)^{\alpha-1} f(t) \mathrm{d}t, $$

相应地, 我们可以得到关于 R-L 右分数阶导数的定义.

定义 2.5 设 α 是大于 0 的任意正实数, 且 n 是大于 α 的正整数, 满足 $n-1 \leq \alpha < n$, 则 $f(x)$ 的 R-L 右分数阶导数定义为

$$ {}^{\mathrm{RL}}_{x}\mathrm{D}^{\alpha}_{b} f(x) \triangleq \mathrm{D}^{n}({}_{x}\mathrm{D}^{\alpha-n}_{b} f(x)) = \frac{(-1)^n}{\Gamma(n-\alpha)} \frac{\mathrm{d}^n}{\mathrm{d}x^n} \int_x^b \frac{f(t)}{(t-x)^{\alpha-n+1}} \mathrm{d}t. $$

2.1.2 Caputo 分数阶导数

因为 R-L 分数阶导数在实际应用中的困难 [18], 于是便有学者提出了 Caputo 分数阶导数 [9, 10]. Caputo 分数阶导数和 R-L 分数阶导数很相似, 不过它们的微分和积分顺序相反, R-L 分数阶导数是先积分后微分, 而 Caputo 分数阶导数是先微分后积分. Caputo 左分数阶导数的定义如下:

定义 2.6 设 $\alpha > 0$ 且 n 是大于 α 的正整数, 满足 $n-1 \leq \alpha < n$, 则 $f(x)$ 的 Caputo 左分数阶导数定义为

$$ {}^{\mathrm{C}}_{a}\mathrm{D}^{\alpha}_{x} f(x) \triangleq \frac{1}{\Gamma(n-\alpha)} \int_a^x \frac{f^{(n)}(t)}{(x-t)^{\alpha+1-n}} \mathrm{d}t. $$

Caputo 分数阶导数一般会被用在时间导数上, 尤其会出现于初边值问题.

下面我们给出 Caputo 右分数阶导数的定义.

定义 2.7 设 $\alpha > 0$ 且 n 是大于 α 的正整数, 满足 $n-1 \leq \alpha < n$, 则 $f(x)$ 的 Caputo 右分数阶导数定义为

$$ {}^{\mathrm{C}}_{x}\mathrm{D}^{\alpha}_{b} f(x) \triangleq \frac{(-1)^n}{\Gamma(n-\alpha)} \int_x^b \frac{f^{(n)}(t)}{(t-x)^{\alpha+1-n}} \mathrm{d}t. $$

2.1.3 Grünwald-Letnikov 分数阶导数

我们给出一般导数的极限定义

$$ f'(x) = \lim_{h\to 0} \frac{f(x)-f(x-h)}{h} = \lim_{h\to 0} \frac{(I-T_h)f(x)}{h}, $$

其中的 T_h 是位移算子, 定义如下,

$$ T_h^k f = f(x-kh) $$

令 Δ_h 为步长为 h 的差分算子, 于是有

$$\begin{aligned} f^{(n)}(x) &= \lim_{h\to 0}\frac{\Delta_h^n f(x)}{h^n} \\ &= \lim_{h\to 0} h^{-n}(I-T_h)^n f(x) \\ &= \lim_{h\to 0} h^{-n}\sum_{k=0}^{n}\binom{n}{k}(-T_h)^k I^{n-k} f(x) \\ &= \lim_{h\to 0} h^{-n}\sum_{k=0}^{n}\binom{n}{k}(-1)^k f(x-kh). \end{aligned}$$

当 $k>n$ 时, 我们有 $\binom{n}{k}=0$, 于是

$$f^{(n)}(x) = \lim_{h\to 0} h^{-n}\sum_{k=0}^{\infty}\binom{n}{k}(-1)^k f(x-kh).$$

根据 Gamma 函数的性质我们可以得到

$$\begin{aligned} \binom{n}{k}(-1)^k &= (-1)^k\frac{n(n-1)\cdots(n-k+1)}{k!} \\ &= (-1)^k\frac{\Gamma(n+1)}{\Gamma(k+1)\Gamma(n-k+1)} \\ &= \frac{\Gamma(k-n)}{\Gamma(k+1)\Gamma(-n)}. \end{aligned}$$

将上面整数阶导数的极限定义推广到正实数的情形, 就得到 G-L 分数阶导数的定义

定义 2.8 设 $\alpha>0$, 则 G-L 左分数阶导数定义为

$$\begin{aligned} {}^{\mathrm{GL}}_{a}\mathrm{D}_x^{\alpha} &\triangleq \lim_{h\to 0}\frac{1}{h^{\alpha}}\sum_{k=0}^{\lfloor\frac{x-a}{h}\rfloor}\frac{(-1)^k\Gamma(\alpha+1)}{\Gamma(k+1)\Gamma(\alpha-k+1)}f(x-kh) \\ &= \lim_{h\to 0}\frac{1}{h^{\alpha}}\sum_{k=0}^{\lfloor\frac{x-\alpha}{h}\rfloor}\frac{\Gamma(k-\alpha)}{\Gamma(k+1)\Gamma(-\alpha)}f(x-kh). \end{aligned}$$

相应地, 我们给出 G-L 右分数阶导数的定义

定义 2.9 设 $\alpha>0$, 则 G-L 右分数阶导数定义为

$$\begin{aligned} {}^{\mathrm{GL}}_{x}\mathrm{D}_b^{\alpha} &\triangleq \lim_{h\to 0} h^{-\alpha}\sum_{k=0}^{\lfloor\frac{b-x}{h}\rfloor}\frac{(-1)^k\Gamma(\alpha+1)}{\Gamma(k+1)\Gamma(\alpha-k+1)}f(x+kh) \\ &= \lim_{h\to 0} h^{-\alpha}\sum_{k=0}^{\lfloor\frac{b-x}{h}\rfloor}\frac{\Gamma(k-\alpha)}{\Gamma(k+1)\Gamma(-\alpha)}f(x+kh). \end{aligned}$$

2.2 Toeplitz 矩阵和循环矩阵及其快速算法

2.2.1 Toeplitz 矩阵

给出 Toeplitz 矩阵形式如下:

$$T = \begin{bmatrix} t_0 & t_{-1} & \cdots & t_{-n+1} \\ t_1 & \ddots & \ddots & \vdots \\ \vdots & \ddots & \ddots & t_{-1} \\ t_{n-1} & \cdots & t_1 & t_0 \end{bmatrix} \in \mathbb{R}^{n\times n}.$$

注意到 Toeplitz 矩阵的对角线上的元素都相等, 因此在储存一个 Toeplitz 矩阵时, 只需要储存矩阵的第一列和第一行元素即可.

2.2.2 循环矩阵

给出循环矩阵的形式如下:

$$C = \begin{bmatrix} z_0 & z_{n-1} & \cdots & z_1 \\ z_1 & \ddots & \ddots & \vdots \\ \vdots & \ddots & \ddots & z_{n-1} \\ z_{n-1} & \cdots & z_1 & z_0 \end{bmatrix} \triangleq C(z). \tag{2.3}$$

其中 $z = [z_0, z_1, \ldots, z_{n-1}]^T$. 注意到, 循环矩阵其实是一类特殊的 Toeplitz 矩阵, 由它的第一列所决定的, 因此只需要储存第一列元素即可.

引理 2.3 设 $C = C(z)$ 是循环矩阵, 则

$$C = \sum_{k=0}^{n} z_k L^k,$$

其中 L 是 downshift 置换矩阵 [23], 即

$$L = \begin{bmatrix} 0 & 0 & \cdots & 0 & 1 \\ 1 & 0 & \cdots & 0 & 0 \\ 0 & 1 & \ddots & 0 & 0 \\ \vdots & & \ddots & \ddots & \vdots \\ 0 & 0 & \cdots & 1 & 0 \end{bmatrix}.$$

2.2.3 快速 Fourier 变换

定义 2.10 *我们称向量 $y = [y_0, y_1, \ldots, y_{n-1}] \in \mathbb{C}^n$ 是 $x = [x_0, x_1, \ldots, x_{n-1}] \in \mathbb{C}^n$ 的 DFT, 其中*

$$y_k = \sum_{j=0}^{n-1} x_j w_n^{kj}, k = 0, 1, \ldots, n-1. \tag{2.4}$$

其中

$$w_n = e^{-\frac{2\pi \mathrm{i}}{n}} = \cos\left(\frac{2\pi}{n}\right) - \mathrm{i}\sin\left(\frac{2\pi}{n}\right)$$

为 1 的一个 n 次根.

DFT (2.4) 也可以表示成如下所示的矩阵与向量的乘积:

$$y = F_n x,$$

其中

$$F_n = [f_{kj}]_{n\times n} = \begin{bmatrix} 1 & 1 & 1 & \cdots & 1 \\ 1 & w_n & w_n^2 & \cdots & w_n^{n-1} \\ 1 & w_n^2 & w_n^4 & \cdots & w_n^{2(n-1)} \\ \vdots & \vdots & \vdots & \ddots & \vdots \\ 1 & w_n^{n-1} & w_n^{(n-1)2} & \cdots & w_n^{(n-1)(n-1)} \end{bmatrix} \tag{2.5}$$

可以看出

$$f_{kj} = w_n^{(k-1)(j-1)}, \qquad k, j = 1, 2, \ldots, n.$$

称 F_n 为 n 阶 DFT 矩阵 [23].

下面给出 DFT 矩阵的一条重要的性质:

引理 2.4 *设 F_n 是 n 阶 DFT 矩阵, 则有*

$$F_n^* F_n = nI, \quad F_n^{-1} = \frac{1}{n} F_n^* = \frac{1}{n}\bar{F}_n,$$

即 $\dfrac{1}{\sqrt{n}} F_n$ 是酉矩阵.

其实 DFT 就是矩阵和向量的乘积, 正常来说运算量为 $\mathcal{O}(n^2)$, 当 n 很大的时候, DFT 的计算量让人难以接受. Cooley 和 Tukey[17] 在 1965 年提出了快速 Fourier 变换, 通过巧妙地利用 F_n 中元素 w_n 的周期性和对称性, 避免了 DFT 中包含的大量重复的计算. 使得 DFT 的运算量由 $\mathcal{O}(n^2)$ 降低到 $\mathcal{O}(n \log n)$. 关于 FFT 更多的知识可以参考 [20, 48].

循环矩阵和向量乘积的快速算法

设 C 是由 (2.3) 所定义的循环矩阵, 则由引理 2.3 可知

$$C = \sum_{k=0}^{n} z_k L^k,$$

其中 L 是上面提及的 downshift 置换矩阵. 通过计算, 我们有:

$$F_n L = W F_n,$$

其中

$$W \triangleq \text{diag}(1, w_n, w_n^2, \ldots, w_n^{n-1}).$$

于是就有

$$L^k = (F_n^{-1} W F_n)^k = F_n^{-1} W^k F_n = \frac{1}{n} F_n^* W^k F_n.$$

所以有

$$C = \sum_{k=0}^{n} z_k L^k = \sum_{k=0}^{n} z_k \tilde{F}_n^* W^k \tilde{F}_n,$$

其中 $\tilde{F}_n \triangleq \frac{1}{\sqrt{n}} F_n$ 是酉矩阵, 故而我们有

$$\begin{aligned} C = \sum_{k=0}^{n} \tilde{F}_n^* z_k \tilde{F}_n \tilde{F}_n^* W^k \tilde{F}_n &= \sum_{k=0}^{n} \tilde{F}_n^* z_k W^k \tilde{F}_n \\ &= \tilde{F}_n^* \sum_{k=0}^{n} z_k W^k \tilde{F}_n \\ &= \tilde{F}_n^* \Lambda \tilde{F}_n \\ &= \tilde{F}_n^{-1} \Lambda \tilde{F}_n \\ &= (\frac{1}{\sqrt{n}} F_n)^{-1} \Lambda \frac{1}{\sqrt{n}} F_n \\ &= F_n^{-1} \Lambda F_n \end{aligned} \tag{2.7}$$

其中 $\Lambda = \sum_{k=0}^{n} z_k W^k \triangleq \text{diag}(\lambda_0, \lambda_1, \ldots, \lambda_{n-1})$ 是对角矩阵, 而且我们有

$$[\lambda_1, \lambda_2, \ldots, \lambda_n]^T = F_n z.$$

于是循环矩阵 C 和向量 v 的乘积可以表示为如下形式:

$$Cv = \tilde{F}_n^* \Lambda \tilde{F}_n v = F_n^{-1}(\Lambda(F_n v)). \tag{2.8}$$

Toeplitz 矩阵和向量乘积的快速算法

假设 $T \in \mathbb{R}^{n\times n}$ 是 n 阶 Toeplitz 矩阵, 我们将 Toeplitz 矩阵 T 嵌入到一个 $2n$ 阶的循环矩阵中, 构造如下:

$$C(z) = \begin{bmatrix} T & C_{12} \\ C_{21} & T \end{bmatrix} \in \mathbb{R}^{2n\times 2n}.$$

通过选取合适的 C_{12} 和 C_{21} 就可以将 C 构造成一个循环矩阵. 令 $C(z)$ 的第一列为

$$z = [t_0, t_1, \ldots, t_{n-1}, 0, t_{-n+1}, \ldots, t_{-2}, t-1]^T.$$

于是 Toeplitz 矩阵和向量的乘积就转化成了循环矩阵和向量的乘积, 如下所示:

$$C(z)\begin{bmatrix} v \\ 0 \end{bmatrix} = \begin{bmatrix} T & * \\ * & T \end{bmatrix}\begin{bmatrix} v \\ 0 \end{bmatrix} = \begin{bmatrix} Tv \\ * \end{bmatrix}.$$

2.3 Toeplitz 矩阵的循环矩阵近似

为了方便我们构造预处理算子, 我们有必要介绍一下常见的关于 Teoplitz 矩阵的 Strang 循环近似和 T. Chan 循环近似, 更多的参见 [11].

假设 A_n 是 n 阶的 Toeplitz 矩阵, 写成如下的形式:

$$A_n = \begin{bmatrix} a_0 & a_{-1} & \cdots & a_{2-n} & a_{1-n} \\ a_1 & a_0 & a_{-1} & \cdots & a_{2-n} \\ \vdots & a_1 & a_0 & \ddots & \vdots \\ a_{n-2} & \cdots & \ddots & \ddots & a_{-1} \\ a_{n-1} & a_{n-2} & \cdots & a_1 & a_0 \end{bmatrix} \tag{2.9}$$

我们研究代数方程组 $A_n x = b$ 的求解.

大多数的早期工作主要集中在对于 Toeplitz 问题的直接法求解. 我们知道, 直接地使用高斯消去法, 计算复杂度为 $\mathcal{O}(n^3)$. 但由于 Toeplitz 矩阵的特殊性, 人们设计出了许多快速算法, 将计算复杂度降低到了 $\mathcal{O}(n^2)$, 比如 Levinson(1946)[32], Baxter(1961)[6], Trench(1964)[47], 以及 Zohar(1974)[53] 等. 1980 年左右, 人们又提出了更快的求解方法, 将计算复杂度进一步降低到 $\mathcal{O}(n\log^2 n)$, 比如 Brent, Gustavson, and Yun(1980)[8], Bitmead and Anderson(1980)[7], Morf(1980)[39], de Hoog(1987)[26], 以及 Ammar and Gragg(1988)[2] 等.

最近研究的使用带预处理的 Krolov 子空间迭代算法用来求解 Toeplitz 问题引起了人们的关注. 最重要的结果之一是当我们取一个较合适的预处理子的时候, Toeplitz 问题的计算复杂度可以降低到 $\mathcal{O}(n\log n)$.

在本文中, 我们将用到用循环矩阵近似 Toeplitz 矩阵的方法, 更好地来针对时间-空间分数阶扩散方程来构造有效的预处理子.

Strang 预处理子

第一个循环预处理子是由 Strang[46] 在 1986 年首先提出, 取 A 的中间的对角线部分, 构成一个循环矩阵, 即 Strang 循环预处理子 $S=[s_{k-l}]_{0\le k,l<n}$ 的对角线 s_j 定义如下:

$$s_j=\begin{cases} a_j, & 0\le j\le \lfloor n/2\rfloor,\\ a_{j-n}, & \lfloor n/2\rfloor<j<n,\\ s_{n+j}, & 0<-j<n. \end{cases} \tag{2.10}$$

Strang 循环预处理子 S 的一个重要的性质是, 在所有的 Hermitian 循环矩阵 C 组成的集合中, 在 1-范数或 ∞-范数意义下, 它是距离 A 最近的, 即它是

$$\min\|C-A\|_1 \quad 和 \quad \min\|C-A\|_\infty$$

的解. 关于这个更多地知识可参见 [12].

定理 2.5 [12, 15] *令 f 是一个在 Wiener 类的正函数, 即它的 Fourier 系数是绝对有界的,*

$$\sum_{k=0}^{\infty}|a_k|<\infty.$$

其中 f 是 Toeplitz 矩阵 A 的生成函数. 则当 n 很大的时候, $S^{-1}A$ 的特征值集中在 1 附近.

T. Chan 循环预处理子

对于一个 n 阶 Toeplitz 矩阵 A, T. Chan 的循环预处理子 $c(A)$ 定义为所有的 $n\times n$ 的循环矩阵 C 中最小化

$$\|C-A\|_F \tag{2.11}$$

的那个循环矩阵 [13]. 这里的 $\|\cdot\|_F$ 指代 Frobenius 范数. 在 [13] 中, 矩阵 $c(A)$ 被称作是最优的循环预处理子, 因为它最小化了 (2.11). 易知, $c(A)$ 的第 j 个对角线

定义如下:

$$c_j = \begin{cases} \dfrac{(n-j)a_j + ja_{j-n}}{n}, & 0 \le j < n, \\ c_{n+j}, & 0 < -j < n, \end{cases} \tag{2.12}$$

对于 $c(A)$ 作为 Toeplitz 矩阵 A 的循环预处理子的表现, R.Chan [14] 证明了, 在前面定理 2.5 前提假设下, $c(A)^{-1}A$ 和 $S^{-1}A$ 的特征值随着 n 趋向于无穷大渐近相等, 即 $\lim_{n\to\infty} \|c(A)^{-1}A - S^{-1}A\|_2 = 0$.

2.4 Toeplitz 直接求逆

在我们构造预处理矩阵之前, 我们先介绍一种用来求解 Toeplitz 矩阵的逆的方法. 该方法在本文中为改进我们的预处理子起到了显著的作用.

对于 Toeplitz 矩阵, 除了矩阵向量的乘积, 它的逆也有被深入研究过. 在文献 [22] 中, Gohberg 和 Semencul 为 Toeplitz 矩阵 A_n 提出了 Gohberg-Semencul 方法 (GSF).

定理 2.6 设 $A = (a_{i-j})_{i,j=0}^{n-1}$ 是一个 Toeplitz 矩阵, 记方程组

$$Ax = e_0 \quad 和 \quad Ay = e_{n-1} \tag{2.13}$$

的解分别为 $x = (x_i)_{i=0}^{n-1}$ 和 $y = (y_i)_{i=0}^{n-1}$, 且 $x_0 \neq 0$, 其中 $e_0 = [1, 0, \cdots, 0]^T, e_{n-1} = [0, \cdots, 0, 1]^T$. 则 A 是可逆的, 并且可表示为

$$A^{-1} = \frac{1}{x_0}(X_n Y_n^T - \hat{Y}_n \hat{X}_n^T),$$

其中 $X_n, Y_n, \hat{X}_n$ 和 $\hat{Y}_n$ 是下三角的 Toeplitz 矩阵, 定义如下:

$$X_n = \begin{bmatrix} x_0 & 0 & \cdots & 0 \\ x_1 & x_0 & \cdots & 0 \\ \vdots & \vdots & \ddots & \vdots \\ x_{n-1} & x_{n-2} & \cdots & x_0 \end{bmatrix}, \quad Y_n = \begin{bmatrix} y_{n-1} & 0 & \cdots & 0 \\ y_{n-2} & y_{n-1} & \cdots & 0 \\ \vdots & \vdots & \ddots & \vdots \\ y_0 & y_1 & \cdots & y_{n-1} \end{bmatrix},$$

$$\hat{X}_n = \begin{bmatrix} 0 & \cdots & 0 & 0 \\ x_{n-1} & \cdots & 0 & 0 \\ \vdots & \ddots & \vdots & \vdots \\ x_1 & \cdots & x_{n-1} & 0 \end{bmatrix}, \quad \hat{Y}_n = \begin{bmatrix} 0 & \cdots & 0 & 0 \\ y_0 & \cdots & 0 & 0 \\ \vdots & \ddots & \vdots & \vdots \\ y_{n-2} & \cdots & y_0 & 0 \end{bmatrix}.$$

第三章　时间-空间分数阶扩散方程的数值离散

我们考虑下面的时间-空间分数阶扩散方程的初边值问题:

$$\begin{cases} {}_{0}^{\mathrm{C}}\mathrm{D}_{t}^{\alpha}u(x,t) = d_{+}(x)\,{}_{a}\mathrm{D}_{x}^{\beta}u(x,t) + d_{-}(x)\,{}_{x}\mathrm{D}_{b}^{\beta}u(x,t) + f(x,t), \\ \qquad\qquad\qquad\qquad\qquad\qquad (x,t)\in(a,b)\times(0,T], \\ u(a,t) = u(b,t) = 0, \qquad t\in[0,T], \\ u(x,0) = 0, \qquad x\in[a,b], \end{cases} \tag{3.1}$$

其中 $\alpha \in (0,1)$, $\beta \in (1,2)$ 分别是时间分数阶导数和空间分数阶导数, 函数 $d_{\pm}(x) \geq 0$ 是扩散系数, $f(x,t)$ 是右端项, ${}_{0}^{\mathrm{C}}\mathrm{D}_{t}^{\alpha}$ 是 Caputo 分数阶导数, ${}_{a}\mathrm{D}_{x}^{\beta}$ 和 ${}_{x}\mathrm{D}_{b}^{\beta}$ 分别是 Riemann-Liouville 左、右分数阶导数, 即

$$\begin{aligned} {}_{0}^{\mathrm{C}}\mathrm{D}_{t}^{\alpha}u(x,t) &= \frac{1}{\Gamma(1-\alpha)}\int_{0}^{t}\frac{\partial u(x,s)}{\partial s}(t-s)^{-\alpha}\mathrm{d}s, \\ {}_{a}\mathrm{D}_{x}^{\beta}u(x,t) &= \frac{1}{\Gamma(2-\beta)}\frac{\partial^{2}}{\partial x^{2}}\int_{a}^{x}\frac{u(\xi,t)}{(x-\xi)^{\beta-1}}\mathrm{d}\xi, \\ {}_{x}\mathrm{D}_{b}^{\beta}u(x,t) &= \frac{1}{\Gamma(2-\beta)}\frac{\partial^{2}}{\partial x^{2}}\int_{x}^{b}\frac{u(\xi,t)}{(\xi-x)^{\beta-1}}\mathrm{d}\xi. \end{aligned}$$

3.1　有限差分离散

我们分别是对空间区间 $[a,b]$ 和时间区间 $[0,T]$ 进行等距剖分, 即取两个正整数 N 和 M, 定义空间网格和时间网格如下:

$$\begin{aligned} &x_i = a + i\Delta x, \quad \Delta x = \frac{b-a}{N+1}, \quad i = 0,1,\ldots,N+1, \\ &t_m = m\Delta t, \quad \Delta t = \frac{T}{M}, m = 0,1,\ldots,M. \end{aligned}$$

我们通过使用下面的带平移的 Grünwald 近似来离散空间上的分数阶导数 [37, 38]:

$$\begin{aligned} {}_{a}\mathrm{D}_{x}^{\beta}u(x_i,t) &= \frac{1}{\Delta x^{\beta}}\sum_{k=0}^{i+1}g_{k}^{(\beta)}u(x_{i-k+1},t) + \mathcal{O}(\Delta x), \\ {}_{x}\mathrm{D}_{b}^{\beta}u(x_i,t) &= \frac{1}{\Delta x^{\beta}}\sum_{k=0}^{N-i+2}g_{k}^{\beta}u(x_{i+k-1},t) + \mathcal{O}(\Delta x). \end{aligned} \tag{3.3}$$

于是我们有:

$$ {}_{\mathrm{C}}^{0}\mathrm{D}_{t}^{\alpha}u_i(t) = \frac{d_{+,i}}{\Delta x^{\beta}}\sum_{k=0}^{i+1} g_k^{(\beta)}u_{i-k+1}(t) + \frac{d_{-,i}}{\Delta x^{\beta}}\sum_{k=0}^{N-i+2} g_k^{\beta}u_{i+k-1}(t) + f_i(t), \quad i = 1,\dots,N, \tag{3.4} $$

其中 $u_i(t)$ 指代 $u(x_i,t)$ 的一个数值近似, $d_{\pm,i} = d_{\pm}(x_i), f_i(t) = f(x_i,t)$, 系数 $g_k^{(\beta)}$ 定义如下:

$$ g_k^{(\beta)} = \frac{\Gamma(k-\beta)}{\Gamma(-\beta)\Gamma(k+1)} = (-1)^k\binom{\beta}{k}, \tag{3.5} $$

为了计算的方便, 上面的形式可以写成下面迭代的形式 [42]:

$$ g_0^{(\beta)} = 1, \qquad g_k^{(\beta)} = \left(1 - \frac{\beta+1}{k}\right) g_{k-1}^{(\beta)}, $$

并且有性质

$$ \begin{cases} g_0^{(\beta)} = 1, \quad g_1^{(\beta)} = -\beta < 0, \quad 1 \geq g_2^{(\beta)} \geq g_3^{(\beta)} \geq \cdots \geq 0, \\ \sum\limits_{k=0}^{\infty} g_k^{(\beta)} = 0, \quad \sum\limits_{k=0}^{m} g_k^{(\beta)} \leq 0 \quad (m \geq 1). \end{cases} \tag{3.6} $$

考虑到齐次 Dirichlet 边界条件 $u_0(t) = u_{N+1}(t) = 0$, 于是, 我们可以将前面的半离散格式 (3.4) 写成如下的矩阵形式:

$$ {}_{\mathrm{C}}^{0}\mathrm{D}_{t}^{\alpha}u(t) + \Delta x^{-\beta}(D_+T_\beta + D_-T_\beta^T)u(t) = f(t), $$

其中

$$ u(t) = [u_1(t),\dots,u_N(t)]^T, f(t) = [f_1(t),\dots,f_N(t)]^T, D_{\pm} = \mathrm{diag}(\mathrm{d}_{\pm,1},\dots,\mathrm{d}_{\pm,N}), $$

且

$$ T_\beta = -\begin{bmatrix} g_1^{(\beta)} & g_0^{(\beta)} & 0 & \cdots & 0 & 0 \\ g_2^{(\beta)} & g_1^{(\beta)} & g_0^{(\beta)} & 0 & \cdots & 0 \\ \vdots & \ddots & \ddots & \ddots & \ddots & \vdots \\ \vdots & \ddots & \ddots & \ddots & \ddots & 0 \\ g_{N-1}^{(\beta)} & \ddots & \ddots & \ddots & g_1^{(\beta)} & g_0^{(\beta)} \\ g_N^{(\beta)} & g_{N-1}^{(\beta)} & \cdots & \cdots & g_2^{(\beta)} & g_1^{(\beta)} \end{bmatrix}, \tag{3.7} $$

是一个 N 阶的具有下 Hessenberg 形状的 Teoplitz 矩阵.

下面考虑时间上的离散, 我们采用 Grünwald 差分离散方法, 形式如下 [42]:

$$ {}_{\mathrm{C}}^{0}\mathrm{D}_{t}^{\alpha}u(t_m) = \frac{1}{\Delta t^{\alpha}}\sum_{k=0}^{m} g_k^{(\alpha)}u(t_{m-k}) + \mathcal{O}(\Delta t). $$

于是我们有下面的有限差分格式:

$$\frac{1}{\Delta t^{\alpha}}\sum_{k=0}^{m} g_k^{(\alpha)} u^{(m-k)} + \Delta x^{-\beta}(D_+T_\beta + D_-T_\beta^T)u^{(m)} = f^{(m)}, \qquad m = 1, \ldots, M, \tag{3.8}$$

其中 $u^{(m)}$ 是 $u(t_m)$ 的数值近似, 且有 $f^{(m)} = f(t_m)$. 注意到 $g_0^{(\alpha)} = 1$, 于是我们可以将上面的差分格式写成下面的按时间步迭代 (step-by-step) 的形式:

$$\big(I_N + \tau(D_+T_\beta + D_-T_\beta^T)\big)u^{(m)} = -\sum_{k=1}^{m} g_k^{(\alpha)} u^{(m-k)} + \Delta t^{\alpha} f(t_m), \qquad m = 1, \ldots, M,$$

其中 $\tau = \Delta t^{\alpha}\Delta x^{-\beta}$. 从这个时间-空间分数阶扩散方程差分格式中我们可以看出, 它的每一个时间步都有一个长长的尾巴, 这尾巴包含了所有先前时间步的额外项. 这就导致了当我们在一个很长的时间模型的时候, 会有很高的计算复杂度.

3.2 代数方程组的结构

不同于传统的按时间推进 (time-marching) 方法, 我们这里采用了另一种方式, 即将所有的时间步整合在一个时间-空间的线性系统中. 将初值条件 $u^{(0)} = 0$ 代入后, 通过整理, 我们可以将方程组 (3.8) 写成下面的形式:

$$(C \otimes I_N + I_M \otimes K)u = \Delta t^{\alpha} f, \tag{3.9}$$

其中 I_N 和 I_M 分别代表 N 阶和 M 阶单位矩阵,

$$u = [u^{(1)T}, \ldots, u^{(M)T}]^T \in \mathbb{R}^{NM}, \quad f = [f^{(1)T}, \ldots, f^{(M)T}]^T,$$

C 是一个下三角的 Toeplitz 矩阵:

$$C = \begin{bmatrix} g_0^{(\alpha)} & 0 & 0 & \cdots & 0 & 0 \\ g_1^{(\alpha)} & g_0^{(\alpha)} & 0 & 0 & \cdots & 0 \\ \vdots & \ddots & \ddots & \ddots & \ddots & \vdots \\ \vdots & \ddots & \ddots & \ddots & \ddots & 0 \\ g_{M-2}^{(\alpha)} & \ddots & \ddots & \ddots & g_0^{(\alpha)} & 0 \\ g_{M-1}^{(\alpha)} & g_{M-2}^{(\alpha)} & \cdots & \cdots & g_1^{(\alpha)} & g_0^{(\alpha)} \end{bmatrix}, \tag{3.10}$$

K 有下面的表达式

$$K = \tau(D_+T_\beta + D_-T_\beta^T), \tag{3.11}$$

也就是说, 方程组的系数矩阵可以写成两个 Kronecker 乘积之和, 而且每个 Kronecker 乘积都有 Teoplitz 矩阵, 单位矩阵和对角矩阵组成.

第四章　时间-空间分数阶扩散方程的预处理方法

4.1　循环矩阵近似预处理方法

在这一节, 我们讨论基于循环矩阵近似的预处理方法, 给出预处理子的构造方法和实施细节, 最后进行数值测试.

4.1.1　简单循环预处理子的构造

离散后的时间-空间分数阶扩散方程可写为

$$(C \otimes I_N + I_M \otimes K)u = \Delta t^{\alpha} f, \tag{4.1}$$

其中

$$K = \tau(D_+ T_\beta + D_- T_\beta^T),$$

其中 C 为 M 阶的下三角的 Toeplitz 矩阵, $D_\pm$ 是对角矩阵, T_β 是具有下 Hessenberg 形状的 Teoplitz 矩阵 (3.7).

为了讨论方便, 我们将系数矩阵记为

$$A \triangleq C \otimes I_N + I_M \otimes K. \tag{4.2}$$

在已知 C 和 T_β 是 Toeplitz 矩阵的情况下, 我们对 A 中出现的 Toeplitz 矩阵使用 Strang 循环近似, 并且为了后面的计算方便, 我们需要对扩散系数对角矩阵 $D_\pm$ 用一个常数的对角矩阵来近似

$$\bar{D}_\pm = \left(\frac{1}{N} \sum_{i=1}^{N} d_{\pm,i} \right) \cdot I_N \tag{4.3}$$

我们记 $s(C)$ 为下三角 Toeplitz 矩阵 C 的 Strang 循环近似, $s(T_\beta)$ 为下 Hessenberg Toeplitz 矩阵 T_β 的 Strang 循环近似, $\bar{D}_+$ 为扩散系数矩阵 D_+ 的常数化对角阵的近似, $\bar{D}_-$ 为扩散系数矩阵 D_- 的常数化对角矩阵的近似, 于是我们就得到下面的系数矩阵 A 的循环近似预处理子

$$P_1 = C_1 \otimes I_n + \tau I_M \otimes (\bar{D}_+ s(T_\beta) + \bar{D}_- s(T_\beta)^T), \tag{4.4}$$

我们记

$$\tilde{C}_2 = \tau(\bar{D}_+ s(T_\beta) + \bar{D}_- s(T_\beta)^T).$$

因为 $s(T_\beta)$ 是循环矩阵, 而 $\bar{D}_+, \bar{D}_-$ 是常数对角矩阵, 故而 $\tilde{C}_2$ 仍然是一个循环矩阵. 于是我们可以将 P_1 写成如下的形式

$$P_1 = C_1 \otimes I_N + I_M \otimes \tilde{C}_2 \tag{4.5}$$

我们知道循环矩阵可以写成 DFT 矩阵和对角阵的组合形式, 即

$$C_1 = F_M^{-1}\Lambda_1 F_M, \quad \tilde{C}_2 = F_N^{-1}\Lambda_2 F_N$$

其中

$$\Lambda_1 = F_M z_1, \quad \Lambda_2 = F_N z_2,$$

这里 z_1, z_2 分别代表循环矩阵 C_1 和 $\tilde{C}_2$ 的第一列. 于是我们可以进一步得到

$$\begin{aligned} P_1 &= F_M^{-1}\Lambda_1 F_M \otimes I_N + I_M \otimes F_N^{-1}\Lambda_2 F_N \\ &= (F_M^{-1}\Lambda_1 F_M) \otimes (F_N^{-1} I_N F_N) + (F_M^{-1} I_M F_M) \otimes (F_N^{-1}\Lambda_2 F_N) \\ &= (F_M^{-1} \otimes F_N^{-1})(\Lambda_1 \otimes I_N)(F_M \otimes F_N) + (F_M^{-1} \otimes F_N^{-1})(I_M \otimes \Lambda_2)(F_M \otimes F_N) \\ &= (F_M \otimes F_N)^{-1}(\Lambda_1 \otimes I_N + I_M \otimes \Lambda_2)(F_M \otimes F_N) \end{aligned}$$

将 P_1 写成这种形式之后, 当我们对问题使用右预处理的 GMRES 算法进行求解, 在计算预处理部分 $z = P_1^{-1}u$ 时, 可以使用直接法求解, 即

$$\begin{aligned} z &= P_1^{-1}u \\ &= (F_M \otimes F_N)^{-1}(\Lambda_1 \otimes I_N + I_M \otimes \Lambda_2)^{-1}(F_M \otimes F_N)u. \end{aligned}$$

借助使用快速 Fourier 变换, 可以将运算量降低到 $\mathcal{O}(MN\log N + NM\log M)$. 另外, 系数矩阵 A 与任意向量 u 的乘积

$$\begin{aligned} Au &= (C \otimes I_N + I_M \otimes K)u \\ &= (C \otimes I_N)u + (I_M \otimes K)u \\ &= \text{vec}(UC^T) + \text{vec}(KU) \end{aligned} \tag{4.7}$$

其中 u 是长度为 MN 的向量, 而 U 则是将向量 u 按列重新排列构成的 $N \times M$ 的矩阵, vec$(\cdot)$ 表示将矩阵按列排成一个列向量. 由于 C 是 Toeplitz 矩阵, K 由对角矩阵与 Toeplitz 矩阵组合而成, 因此该乘积可以通过快速 Fourier 变换来实现, 总运算量大约为 $\mathcal{O}(NM\log M + MN\log N)$. 可以看到, 我们的预处理部分所增加的运算量和系数矩阵同向量乘积部分的运算量处于同一个量级, 这说明我们的预处理子的选择是比较实用的.

4.1.2 数值算例

在下面的数值算例中, 我们分别使用不带预处理的 GMERS 算法和使用 P_1 作为右预处理的 GMRES 算法进行比较, 主要比较二者的迭代步数和迭代终止所需要的运行时间.

在数值算例中, 我们以零向量作为初始迭代向量, 将算法的最大迭代步数设为 500. 记 r_0 为初始残量, r_k 为算法迭代 k 步之后的残量, 算法的迭代终止条件设为:

$$\frac{\|r_k\|_2}{\|r_0\|_2} < 10^{-6}.$$

例 4.1 我们考虑下面的时间-空间分数阶扩散方程

$$\begin{cases} {}_0^{\mathrm{C}}D_t^{\alpha}u(x,t) = d_+(x)\,{}_aD_x^{\beta}u(x,t) + d_-(x)\,{}_xD_b^{\beta}u(x,t) + f(x,t), \\ \qquad\qquad\qquad\qquad\qquad\qquad\qquad\qquad (x,t) \in (a,b)\times(0,T], \\ u(a,t) = u(b,t) = 0, \qquad t \in [0,T], \\ u(x,0) = 0, \qquad x \in [a,b], \end{cases} \tag{4.8}$$

其中扩散系数为

$$d_+(x) = \Gamma(3-\beta)x^{\beta}, \quad d_-(x) = \Gamma(3-\beta)(1-x)^{\beta}. \tag{4.9}$$

空间区间 $[a,b] = [0,1]$, 时间区间 $[0,T] = [0,1]$, 源项 $f(x,t)$ 形式如下:

$$\begin{aligned} f(x,t) = -32t\Bigg(x^2 + (1+x)^2 - \frac{6}{3-\beta}[x^3 + (1-x)^3] \\ + \frac{12}{(3-\beta)(4-\beta)}[x^4 + (1-x)^4]\Bigg) + \frac{16}{\Gamma(2-\alpha)}t^{1-\alpha}x^2(1-x)^2. \end{aligned}$$

通过计算可知, 该问题的精确解为

$$u(x,t) = 16tx^2(1-x)^2. \tag{4.10}$$

在数值实验中, 我们分别测试了空间分数阶导数 β 为 1.3, 1.5, 1.7 和时间分数阶导数 α 为 0.3, 0.5, 0.7 时的情形, 数值结果见表 4.1, 4.2 和 4.3. 在表格中, “GMRES”表示不带预处理的 GMRES 方法, “GMRES(P_1)”表示带预处理子 P_1 的预处理 GMRES 方法, “CPU”表示算法迭代终止所需要的时间, “Iter”表示算法迭代终止所需的迭代步数, 并用“$-$”表示算法不收敛或者迭代步数超过最大迭代步数.

表 4.1 数值结果

			GMRES		GMRES(P_1)	
α	β	$M=N$	Iter	CPU	Iter	CPU
0.3	1.3	2^7	110	0.43	41	0.13
		2^8	204	4.34	50	0.86
		2^9	391	97.44	59	5.64
		2^{10}	–	–	67	27.63
0.5	1.3	2^7	131	0.57	41	0.13
		2^8	246	5.60	51	0.90
		2^9	482	140.68	61	5.83
		2^{10}	–	–	71	29.85
0.7	1.3	2^7	184	0.94	40	0.13
		2^8	358	9.90	50	0.87
		2^9	–	–	61	5.85
		2^{10}	–	–	73	31.09

表 4.2 数值结果

			GMRES		GMRES(P_1)	
α	β	$M=N$	Iter	CPU	Iter	CPU
0.3	1.5	2^7	142	0.64	41	0.14
		2^8	262	6.19	50	0.87
		2^9	–	–	58	5.47
		2^{10}	–	–	67	28.52
0.5	1.5	2^7	190	0.98	43	0.14
		2^8	350	9.62	53	0.91
		2^9	–	–	63	6.12
		2^{10}	–	–	75	32.38
0.7	1.5	2^7	268	1.76	43	0.14
		2^8	–	–	54	0.94
		2^9	–	–	66	6.56
		2^{10}	–	–	79	35.48

表 4.3 数值结果

			GMRES		GMRES(P_1)	
α	β	$M=N$	Iter	CPU	Iter	CPU
0.3	1.7	2^7	195	1.03	43	0.14
		2^8	369	10.16	52	0.90
		2^9	–	–	61	5.83
		2^{10}	–	–	72	30.37
0.5	1.7	2^7	275	1.84	47	0.16
		2^8	–	–	57	1.00
		2^9	–	–	69	7.00
		2^{10}	–	–	81	35.86
0.7	1.7	2^7	413	3.64	48	0.16
		2^8	–	–	60	1.06
		2^9	–	–	74	7.67
		2^{10}	–	–	91	42.14

从上面的实验结果中可以看出, 无论 α 和 β 怎么取值, 如果直接用 GMRES 求解线性方程组的话, 收敛速递非常慢. 而使用了预处理子 P_1 后, GMRES 算法的收敛速度大幅增加, 所需时间也大幅减少, 这说明预处理子 P_1 具有非常好的数值加速效果.

4.2 基于 Toeplitz 求逆的块对角预处理

4.2.1 预处理子的构造

方程组的系数矩阵为

$$A = C \otimes I_N + I_M \otimes K,$$

其中

$$K = \tau(D_+ T_\beta + D_- T_\beta^T),$$

C 为 M 阶的下三角的 Toeplitz 矩阵, 见 (3.10), 扩散系数 $D_\pm$ 是对角矩阵, T_β 是 Toeplitz 矩阵, 见 (3.7).

观察系数矩阵 A 的块状结构, 即

$$A = \begin{bmatrix} g_0^{(\alpha)} I_N + K & 0 & \cdots & 0 \\ C_{21} & g_0^{(\alpha)} I_N + K & \cdots & 0 \\ \vdots & \ddots & \ddots & \vdots \\ C_{M1} & \cdots & C_{21} & g_0^{(\alpha)} I_N + K \end{bmatrix},$$

其中

$$C_{ij} = g_{i-j}^{(\alpha)} I_N.$$

由 $g_i^{(\alpha)}$ 的表达式可知 [37]

$$g_0^{(\alpha)} = 1, \quad g_i^{(\alpha)} < 0,\ i = 1, 2, \ldots \quad \text{且} \quad \sum_{i=0}^{M-1} g_i^{(\alpha)} \geq 0. \tag{4.11}$$

因此有

$$|g_0^{(\alpha)}| > \sum_{i=1}^{M-1} |g_i^{(\alpha)}|. \tag{4.12}$$

也就是说, C 是严格对角占优的, 因此我们可以选择其对角线部分作为它的近似, 于是就得到系数矩阵 A 的一个近似:

$$P_2 \triangleq g_0^{(\alpha)} I_M \otimes I_N + I_M \otimes K = I_M \otimes I_N + I_M \otimes K = I_M \otimes (I_N + K).$$

易知, P_2 其实就是 A 的块对角部分.

因为扩散系数 $d_{\pm}(x)$ 只与空间变量 x 有关, 所以 P_2 的对角块都是一样的. 如果我们将其作为原问题的预处理子, 则每次迭代时需求解 M 个以 $I_N + K$ 为系数矩阵的线性方程组. 由于 K 是由两个对角矩阵与 Toeplitz 矩阵的乘积之和组合而成, 据我们所知, 目前还不存在求解这类问题的快速直接解法, 所以求解代价会比较高. 考虑到预处理子的实用性, 我们需要对 K 做近似处理. 一个简单的方法就是将对角矩阵 $D_{\pm}$ 常量化, 即用 (4.3) 中的 $\bar{D}_{\pm}$ 来代替. 于是, 我们可以构造出如下的预处理矩阵:

$$\tilde{P}_2 = I_M \otimes I_N + I_M \otimes \bar{K} = I_M \otimes (I_N + \bar{K}), \tag{4.13}$$

其中

$$\bar{K} = \tau(\bar{D}_+ T_\beta + \bar{D}_- T_\beta^T), \quad \bar{D}_{\pm} = \left(\frac{1}{N} \sum_{i=1}^{N} d_{\pm,i} \right) \cdot I_N. \tag{4.14}$$

于是在每次迭代时, 我们需要求解 M 个以 $I_N + \bar{K}$ 为系数矩阵的线性方程组. 根据定理 2.6, 我们可以采用直接求逆的方法, 将 Toeplitz 矩阵 $I_N + \bar{K}$ 的逆先计算

出来, 然后再代入求解. 由于求逆过程只需计算一次, 因此总的运算量还是可以接受的.

根据定理 2.6, 在计算 $(I_N + \bar{K})^{-1}$ 过程中, 我们需要求解两个 Toeplitz 线性方程组

$$(I_N + \bar{K})x = e_0 \quad 和 \quad (I_N + \bar{K})y = e_{N-1},$$

其中

$$e_0 = [1, 0, \ldots, 0]^T, \quad e_{N-1} = [0, \ldots, 0, 1]^T.$$

我们可以采用快速直接法求解, 运算量大约为 $\mathcal{O}(N^2)$. 由于 $I_N + \bar{K}$ 的元素具有很好的非对角衰减性 (off-diagonal decay), 因此在实际计算中我们采用带循环预处理的 GMRES 迭代方法. 利用快速 Fourier 变换, 运算量可降低到 $\mathcal{O}(N \log N)$, 相较系数矩阵 A 和向量的乘积的运算量基本可以忽略不计.

因此, 整个预处理过程

$$\begin{aligned} \tilde{P}_2^{-1} u &= (I_M \otimes (\bar{K} + I_N)^{-1})u \\ &= (I_M \otimes T^{-1})u \\ &= \mathrm{vec}(T^{-1}U) \end{aligned}$$

的运算量大约为 $\mathcal{O}(MN \log N)$, 与系数矩阵 A 和向量乘积的运算量 $\mathcal{O}(NM \log M + MN \log N)$ 相当, 预处理子的实用性得到了保证.

4.2.2 预处理子的理论分析

由于 $g_0^{(\alpha)} = 1$, 因此 Toeplitz 矩阵 C 的对角线部分就是一个 M 阶的单位矩阵, 而且有

$$C - I_M = \begin{bmatrix} 0 & 0 & 0 & \cdots & 0 & 0 \\ g_1^{(\alpha)} & 0 & 0 & 0 & \cdots & 0 \\ \vdots & \ddots & \ddots & \ddots & \ddots & \vdots \\ \vdots & \ddots & \ddots & \ddots & \ddots & 0 \\ g_{M-2}^{(\alpha)} & \ddots & \ddots & \ddots & 0 & 0 \\ g_{M-1}^{(\alpha)} & g_{M-2}^{(\alpha)} & \cdots & \cdots & g_1^{(\alpha)} & 0 \end{bmatrix}.$$

根据 $g_i^{(\alpha)}$ 的性质 (4.11), 我们可得

$$\|C - I_M\|_1 = \max_{0 \le j \le M-1} \sum_{i=1}^{M-j-1} |g_i^{(\alpha)}| = \sum_{i=1}^{M-1} |g_i^{(\alpha)}| < 1. \tag{4.15}$$

下面分析预处理后的系数矩阵 $P_2^{-1}A$ 的特征值分布情况.

定理 4.1 设 $P_2 = I_M \otimes (I_N + K)$, 则 $P_2^{-1}A$ 的特征值全部为 1.

证明 因为 P_2 是矩阵 A 的块对角部分, 而且 P_2 是可逆的, 以及 A 本身是一个块下角的矩阵, 则 $P_2^{-1}A$ 为对角线为 1 的下三角矩阵, 故 $P_2^{-1}A$ 的特征值全部为 1.□

定理 4.2 设 $P_2 = I_M \otimes (I_N + K)$, 则

$$\|P_2 - A\|_1 \leq 1. \tag{4.16}$$

证明

$$\begin{aligned} \|P_2 - A\|_1 &= \|I_M \otimes I_N + I_M \otimes K - C \otimes I_N - I_M \otimes K\|_1 \\ &= \|I_M \otimes I_N - C \otimes I_N\|_1 \\ &= \|(C - I_M) \otimes I_N\|_1 \\ &= \|C - I_M\|_1 \end{aligned} \tag{4.18}$$

由 (4.15) 可知, 结论成立. □

由上面的理论分析结果可知, 用 P_2 做预处理子会有比较好的加速效果. 但在实际计算时, 考虑到预处理子的实用性, 我们对其中的对角矩阵做了常量化近似, 即实际实用的是预处理子 $\tilde{P}_2$. 如果扩散系数 $d_{\pm}(x)$ 的变化不是很大, 则具有较好的预处理效果. 当 $d_{\pm}(x)$ 的变化较大时, 我们可能需要考虑其他近似方法, 如 [40] 中提出的近似逆方法等.

4.2.3 数值算例

为了便于各个预处理子之间的比较, 我们依然考虑上一节中的例子 4.1. 所测试的参数也都一样, 数值结果见表 4.4, 4.5 和 4.6, 其中 “GMRES($\tilde{P}_2$)” 表示带预处理子 $\tilde{P}_2$ 的 GMRES 方法.

从数值结果可以看出, 当时间分数阶导数 α 比较小的时候, 比如 $\alpha = 0.3, 0.5$ 时, 预处理子 $\tilde{P}_2$ 具有较好的数值表现, 加速效果总体优于简单循环预处理子 P_1.

但我们同时也发现, 随着时间分数阶导数的增大, $\tilde{P}_2$ 的效果逐渐变差. 其实这也是不难想到的, 因为我们在构造预处理子 $\tilde{P}_2$ 时, 只使用了 C 的对角线部分. 而当 α 比较小时, C 的非对角线部分也比较小, 但随着 α 的增大, C 的非对角线部分的影响也会随之增大,

表 4.4 数值结果

			GMRES		GMRES(P_1)		GMRES($\tilde{P}_2$)	
α	β	$M=N$	Iter	CPU	Iter	CPU	Iter	CPU
0.3	1.3	2^7	110	0.43	41	0.13	23	0.22
		2^8	204	4.34	50	0.86	27	0.85
		2^9	391	97.44	59	5.64	31	3.82
		2^{10}	–	–	67	27.63	36	17.14
0.5	1.3	2^7	131	0.57	41	0.13	30	0.27
		2^8	246	5.60	51	0.90	39	1.26
		2^9	482	140.68	61	5.83	51	6.81
		2^{10}	–	–	71	29.85	66	35.49
0.7	1.3	2^7	184	0.94	40	0.13	55	0.51
		2^8	358	9.90	50	0.87	83	2.76
		2^9	–	–	61	5.85	126	21.47
		2^{10}	–	–	73	31.09	191	159.38

表 4.5 数值结果

			GMRES		GMRES(P_1)		GMRES($\tilde{P}_2$)	
α	β	$M=N$	Iter	CPU	Iter	CPU	Iter	CPU
0.3	1.5	2^7	110	0.43	41	0.14	17	0.17
		2^8	204	4.34	50	0.87	19	0.62
		2^9	391	97.44	58	5.47	21	2.59
		2^{10}	–	–	67	28.52	23	10.6
0.5	1.5	2^7	131	0.57	43	0.14	22	0.22
		2^8	246	5.60	53	0.91	27	0.86
		2^9	482	140.68	63	6.12	34	4.32
		2^{10}	–	–	75	32.38	43	20.94
0.7	1.5	2^7	184	0.94	43	0.14	43	0.41
		2^8	358	9.90	54	0.94	63	2.05
		2^9	–	–	66	6.56	93	14.38
		2^{10}	–	–	79	35.48	136	98.62

表 4.6 数值结果

			GMRES		GMRES(P_1)		GMRES($\tilde{P}_2$)	
α	β	$M=N$	Iter	CPU	Iter	CPU	Iter	CPU
0.3	1.7	2^7	110	0.43	43	0.14	13	0.12
		2^8	204	4.34	52	0.90	15	0.50
		2^9	391	97.44	61	5.83	16	1.99
		2^{10}	–	–	72	30.37	17	7.45
0.5	1.7	2^7	131	0.57	47	0.16	18	0.16
		2^8	246	5.60	57	1.00	22	0.72
		2^9	482	140.68	69	7.00	26	3.19
		2^{10}	–	–	81	35.86	32	14.72
0.7	1.7	2^7	184	0.94	48	0.16	37	0.35
		2^8	358	9.90	60	1.06	54	1.74
		2^9	–	–	74	7.67	77	11.21
		2^{10}	–	–	91	42.14	111	70.18

4.3 基于循环近似的交替方向预处理

4.3.1 预处理子的构造

考虑原问题 (3.9) 的系数矩阵

$$A = C \otimes I_N + I_M \otimes K,$$

其中

$$K = \tau(D_+ T_\beta + D_- T_\beta^T).$$

通过观察可以发现 A 自然地分裂成两个 Kronecker 乘积之和, 令 $\theta > 0$, 则可构造下面的交替方向分裂迭代格式:

$$\begin{cases} (\theta I_{MN} + C \otimes I_N) u^{k+1/2} = (\theta I_{MN} - I_M \otimes K) u^k + \Delta t^\alpha f, \\ (\theta I_{MN} + I_M \otimes K) u^{k+1} = (\theta I_{MN} - C \otimes I_N) u^{k+1/2} + \Delta t^\alpha f, \end{cases}$$

即

$$u^{k+1} = G(\theta) u^k + \Delta t^\alpha H(\theta) f, \tag{4.19}$$

其中

$$\begin{cases} G(\theta) = [(\theta I_M + C)^{-1}(\theta I_M - C)] \otimes [(\theta I_N + K)^{-1}(\theta I_N - K)], \\ H(\theta) = 2\theta(\theta I_M + C)^{-1} \otimes (\theta I_N + K)^{-1}. \end{cases} \tag{4.20}$$

其对应的矩阵分裂为:

$$A = P(\theta) - R(\theta),$$

其中

$$P(\theta) = \frac{1}{2\theta}(\theta I_M + C) \otimes (\theta I_N + K), \tag{4.21}$$

$$R(\theta) = \frac{1}{2\theta}(\theta I_M - C) \otimes (\theta I_N - K).$$

故可以使用 $P(\theta)$ 作为 A 的预处理矩阵. 但是计算 $P^{-1}(\theta)u$ 涉及到 $(\theta I_M + C)^{-1}$ 和 $(\theta I_N + K)^{-1}$, 计算成本较高. 为了降低运算量, 我们使用循环矩阵来近似 $P(\theta)$ 中的 Toeplitz 部分, 并对其中的对角矩阵 $D_{\pm}$ 做常量化处理, 于是得到预处理子

$$P_3 = \frac{1}{2\theta}(\theta I_M + s(C)) \otimes (\theta I_N + \tilde{K}), \tag{4.22}$$

其中

$$\tilde{K} = \tau\left(\bar{D}_+ s(T_\beta) + \bar{D}_- s(T_\beta)^T\right),$$

$s(C)$ 和 $s(T_\beta)$ 分别是 Toeplitz 矩阵 C 和 T_β 的 Strang 循环矩阵近似.

结合之前的分析, 我们可以看出该预处理方法的预处理部分的运算量大约为 $\mathcal{O}(MN\log N + NM\log M)$, 与系数矩阵 A 和向量乘积的运算量处于同一个量级, 所以该预处理子是比较实用的.

4.3.2 预处理子分析

由文献 [51] 中的结论可知, 矩阵 K 严格对角占优且对角线元素为正, 因此矩阵 K 的特征值有正的实部, 从而我们可以得到下面的定理.

定理 4.3 *对于任意的正常数 θ, 交替方向迭代法 (4.19) 的迭代矩阵 $G(\theta)$ 的谱半径 $\rho(G(\theta))$ 满足:*

$$\rho(G(\theta)) = \left|\frac{\theta - 1}{\theta + 1}\right| \max_{\mu\in\sigma(K)} \left|\frac{\theta - \mu}{\theta + \mu}\right| < 1,$$

其中 $\sigma(K)$ 表示矩阵 K 的所有特征值组成的集合.

证明 令 μ 和 ν 分别是矩阵 K 和 C 的特征值. 根据 $G(\theta)$ 的表达式 (4.20) 和 Kronecker 乘积的性质, 我们可知 $G(\theta)$ 的特征值 λ 可表示为:

$$\lambda = \frac{\theta - \nu}{\theta + \nu} \cdot \frac{\theta - \mu}{\theta + \mu}.$$

由于矩阵 C 是一个下三角的矩阵, 并且其对角线上的元素都是 $g_0^{(\alpha)} = 1$, 因此 C 的特征值均为 1, 所以迭代矩阵 $G(\theta)$ 的谱半径 $\rho(G(\theta))$ 满足:

$$\rho(G(\theta)) = \left|\frac{\theta - 1}{\theta + 1}\right| \max_{\mu \in \sigma(K)} \left|\frac{\theta - \mu}{\theta + \mu}\right|.$$

又因为 θ 是正的常数, 而且 μ 的实部均大于 0, 故而我们可以得到:

$$\left|\frac{\theta - \mu}{\theta + \mu}\right| < 1.$$

显然, 对于任意的正实数 θ 都有

$$\left|\frac{\theta - 1}{\theta + 1}\right| < 1.$$

所以

$$\rho(G(\theta)) < 1.$$

对于预处理后的系数矩阵, 我们有下面的结论.

定理 4.4 *对于任意的 $\theta > 0$, 预处理后的系数矩阵 $P^{-1}(\theta)A$ 的所有特征值都包含在以 $(1, 0)$ 为圆心, 半径小于 1 的圆盘里.*

证明 由于

$$G(\theta) = P^{-1}(\theta)R(\theta) = I - P^{-1}(\theta)A,$$

故而

$$P^{-1}(\theta)A = I - G(\theta).$$

根据定理 4.3, $G(\theta)$ 的谱半径小于 1, 所以定理的结论得证. □

注记 4.1 *从定理 4.3 的推导过程中, 我们发现当 $\theta = 1$ 时, 迭代矩阵 $G(\theta)$ 的谱半径最小.*

由于在实际计算时所使用的预处理子是 P_3, 下面我们讨论预处理子 P_3 和 $P(\theta)$ 的近似程度. 我们首先给出系数矩阵 A 中的两个 Toeplitz 矩阵 C 和 T_β 的相关性质.

从 C 的表达式可以看出, 它完全由 $g_k^{(\alpha)}(k=1,2,\ldots,M-1)$ 所决定. 根据 $g_k^{(\alpha)}$ 的定义, 我们有

$$g_k^{(\alpha)} = (-1)^k \binom{\alpha}{k} = \frac{\Gamma(k-\alpha)}{\Gamma(-\alpha)\Gamma(k+1)}.$$

由于 $0<\alpha<1$, 根据 Gamma 函数的性质, 可知 [37]

$$g_0^{(\alpha)} = 1, \quad g_1^{(\alpha)} = -\alpha, \quad 0 > g_2^{(\alpha)} > g_3^{(\alpha)} > \cdots > -1,$$

且

$$\sum_{k=0}^{\infty} g_k^{(\alpha)} = 0. \tag{4.23}$$

另外, 由定理 2.1 可知, $g_k^{(\alpha)}$ 还满足 [51]:

$$g_k^{(\alpha)} = \frac{1}{\Gamma(-\alpha)k^{\alpha+1}}\left(1+\mathcal{O}\left(\frac{1}{k}\right)\right),$$

即 $g_k^{(\alpha)}$ 以 $\alpha+1$ 的速率多项式衰减. 于是我们就可以得到下面的结论 [15].

定理 4.5 *设 $s(C)$ 是 Toeplitz 矩阵 C 的 Strang 循环近似, 则对任意给定的 $\epsilon>0$, 当 M 充分大时有*

$$C - s(C) = E_C + S_C,$$

其中 $\|E_C\|_1 < \epsilon$, $\mathrm{Rank}(S_C) \le \ell_C$, 这里 ℓ_C 是与 M 无关的正整数. 也就是说, $C-s(C)$ 可以写成一个小范数矩阵与一个低秩矩阵之和.

相类似地, 对于 Toeplitz 矩阵 T_β, 我们也可以得到下面的结论.

定理 4.6 *设 $s(T_\beta)$ 是 Toeplitz 矩阵 T_β 的 Strang 循环近似, 则对任意给定的 $\epsilon>0$, 当 N 充分大时有*

$$T_\beta - s(T_\beta) = E_T + S_T,$$

其中 $\|E_T\|_1<\epsilon$, $\|E_T\|_\infty<\epsilon$, $\mathrm{Rank}(S_T)\le\ell_T$, 这里 ℓ_T 是与 N 无关的正整数. 也就是说, $T_\beta - s(T_\beta)$ 可以写成一个小范数矩阵与一个低秩矩阵之和.

下面考虑 $\bar{D}_+T_\beta$ 和 D_+T_β 之间的差. 通过直接计算可得

$$\begin{aligned}\|\bar{D}_+T_\beta - D_+T_\beta\|_1 &\le \|\bar{D}_+ - D_+\|_1 \cdot \|T_\beta\|_1 \\ &\le \max_{1\le i\le N}\left|d_+(x_i) - \frac{1}{N}\sum_{i=1}^{N} d_+(x_i)\right| \cdot \sum_{k=0}^{N}|g_k^{(\beta)}|\end{aligned}$$

$$\leq 2\beta(d_{+,\max} - d_{+,\min}),$$

其中 $d_{+,\max}$ 和 $d_{+,\min}$ 分别表示 $d_+(x)$ 在 $[a,b]$ 上的最大值和最小值.

同理可得

$$\begin{aligned}
\|\bar{D}_- T_\beta^T - D_- T_\beta^T\|_1 &\leq \|\bar{D}_- - D_-\|_1 \cdot \|T_\beta^T\|_1 \\
&= \|\bar{D}_- - D_-\|_1 \cdot \|T_\beta\|_\infty \\
&\leq \max_{1\leq i\leq N} \left| d_-(x_i) - \frac{1}{N}\sum_{i=1}^{N} d_-(x_i) \right| \cdot \sum_{k=0}^{N} \left| g_k^{(\beta)} \right| \\
&\leq 2\beta(d_{-,\max} - d_{-,\min}),
\end{aligned}$$

其中 $d_{-,\max}$ 和 $d_{-,\min}$ 分别表示 $d_-(x)$ 在 $[a,b]$ 上的最大值和最小值. 于是

$$\begin{aligned}
K - \tilde{K} &= \tau\Big(D_+ T_\beta - \bar{D}_+ s(T_\beta) + D_- T_\beta^T - \bar{D}_- s(T_\beta)^T \Big) \\
&= \tau\Big(D_+ T_\beta - \bar{D}_+ T_\beta + \bar{D}_+ T_\beta - \bar{D}_+ s(T_\beta) \\
&\qquad + D_- T_\beta^T - \bar{D}_- T_\beta^T + \bar{D}_- T_\beta^T + \bar{D}_- s(T_\beta)^T \Big) \\
&= \tau\Big(D_+ T_\beta - \bar{D}_+ T_\beta + \bar{D}_+ E_T + \bar{D}_+ S_T \\
&\qquad + D_- T_\beta^T - \bar{D}_- T_\beta^T + \bar{D}_- E_T^T + \bar{D}_- S_T^T) \Big) \\
&\triangleq E_K + S_K,
\end{aligned}$$

其中

$$\begin{aligned}
E_K &= \tau\Big(D_+ T_\beta - \bar{D}_+ T_\beta + \bar{D}_+ E_T + D_- T_\beta^T - \bar{D}_- T_\beta^T + \bar{D}_- E_T^T) \Big), \\
S_K &= \tau\Big(\bar{D}_+ S_T + \bar{D}_- S_T^T) \Big).
\end{aligned}$$

由定理 4.5 和定理 4.6 可知

$$\mathrm{Rank}(S_K) \leq 2\ell_T$$

且

$$\begin{aligned}
\|E_K\|_1 &\leq \tau\Big(\|D_+ T_\beta - \bar{D}_+ T_\beta\|_1 + \|\bar{D}_+\|_1 \epsilon + \|D_- T_\beta^T - \bar{D}_- T_\beta^T\|_1 + \|\bar{D}_-\|_1 \epsilon \Big) \\
&\leq 2\beta\tau(d_{+,\max} - d_{+,\min} + d_{-,\max} - d_{-,\min}) + \tau(d_{+,\max} + d_{-,\max})\epsilon.
\end{aligned}$$

定理 4.7 设 $P(\theta)$ 和 P_3 分别由 (4.21) 和 (4.22) 所定义, 且 $d_\pm(x) > 0$ 在 $[a,b]$ 上连续, 则对任意 $\epsilon > 0$, 存在常数 c_1, 使得当 M, N 充分大时有

$$P(\theta) - P_3 = E_P + S_P,$$

其中 $\text{Rank}(S_P) \le N\ell_C + 2M\ell_T$,

$$\|E_P\|_1 \le c_1\epsilon + 2\beta\tau(\theta+2)(d_{+,\max} - d_{+,\min} + d_{-,\max} - d_{-,\min}).$$

证明 根据 $P(\theta)$ 和 P_3 的定义可知

$$\begin{aligned} P(\theta) - P_3 &= \frac{1}{2\theta}(\theta I_M + C) \otimes (\theta I_N + K) - \frac{1}{2\theta}(\theta I_M + s(C)) \otimes (\theta I_N + \tilde{K}) \\ &= \frac{1}{2\theta}\Big((C - s(C)) \otimes (\theta I_N + K) + (\theta I_M + s(C)) \otimes (K - \tilde{K})\Big) \\ &\triangleq E_P + S_P \end{aligned}$$

其中

$$\begin{aligned} E_P &= \frac{1}{2\theta}\Big(E_C \otimes (\theta I_N + K) + (\theta I_M + s(C)) \otimes E_K\Big), \\ S_P &= \frac{1}{2\theta}\Big(S_C \otimes (\theta I_N + K) + (\theta I_M + s(C)) \otimes S_K\Big). \end{aligned}$$

由于

$$\begin{aligned} \|\theta I_N + K\|_1 &= \|\theta I_N + \tau(D_+T_\beta + D_-T_\beta^T)\|_1 \\ &\le \theta + \tau(\|D_+\|_1 \cdot \|T_\beta\|_1 + \|D_-\|_1 \cdot \|T_\beta^T\|_1) \\ &\le \theta + 2\beta\tau(d_{+,\max} + d_{-,\max}), \\ \|\theta I_M + s(C)\|_1 &\le \theta + \sum_{k=0}^{M-1}\left|g_k^{(\alpha)}\right| \le \theta + 2. \end{aligned}$$

所以

$$\begin{aligned} \|E_P\|_1 &\le \frac{1}{2\theta}\Big(\|E_C\|_1 \cdot \|\theta I_N + K\|_1 + \|\theta I_M + s(C)\|_1 \cdot \|E_K\|_1\Big) \\ &\le c_1\epsilon + 2\beta\tau(\theta+2)(d_{+,\max} - d_{+,\min} + d_{-,\max} - d_{-,\min}), \end{aligned}$$

其中

$$c_1 = \frac{1}{2\theta}(d_{+,\max} + d_{-,\max})\big((\theta + 2\beta\tau) + \tau(\theta+2)\big).$$

又 $\text{Rank}(S_C) \le \ell_C$, $\text{Rank}(S_K) \le 2\ell_T$, 所以

$$\text{Rank}(S_P) \le N\ell_C + 2M\ell_T.$$

定理结论成立. □

如果扩散系数是常数, 则 $d_{+,\max} - d_{+,\min} = 0$, $d_{-,\max} - d_{-,\min} = 0$, 因此我们有下面的结论.

推论 4.8 设 $d_{\pm}(x) > 0$ 是常函数, 则对任意 $\epsilon > 0$, 存在常数 c_1, 使得当 M, N 充分大时有

$$P(\theta) - P_3 = E_P + S_P,$$

其中 $\text{Rank}(S_P) \leq N\ell_C + 2M\ell_T$, $\|E_P\|_1 \leq c_1\epsilon$.

4.3.3 数值算例

考虑数值算例 4.1, 所有测试参数都一样, 数值结果见表 4.7, 4.8 和 4.9, 其中“GMRES(P_3)”表示带预处理子 P_3 的 GMRES 方法.

从数值结果可以看出, 预处理子 P_3 具有一定的加速效果, 但并不理想, 数值表现不如预处理子 P_1 和 $\tilde{P}_2$. 理论分析表明, $P(\theta)$ 具有较好的预处理效果, 但在实际使用时我们对其中的 Toeplitz 矩阵 C 和 T_β 做了循环矩阵近似, 可能由此导致预处理效果的差强人意. 在下一节中, 我们考虑 Toeplitz 矩阵直接求逆的方式.

表 4.7 数值结果

			GMRES		GMRES(P_1)		GMRES($\tilde{P}_2$)		GMRES(P_3)	
α	β	$M = N$	Iter	CPU	Iter	CPU	Iter	CPU	Iter	CPU
0.3	1.3	2^7	110	0.43	41	0.13	23	0.22	57	0.25
		2^8	204	4.34	50	0.86	27	0.85	70	1.45
		2^9	391	97.44	59	5.64	31	3.82	84	9.60
		2^{10}	–	–	67	27.63	36	17.14	98	48.22
0.5	1.3	2^7	131	0.57	41	0.13	30	0.27	54	0.23
		2^8	246	5.60	51	0.90	39	1.26	73	1.56
		2^9	482	140.68	61	5.83	51	6.81	101	12.29
		2^{10}	–	–	71	29.85	66	35.49	139	81.75
0.7	1.3	2^7	184	0.94	40	0.13	55	0.51	67	0.30
		2^8	358	9.90	50	0.87	83	2.76	103	2.24
		2^9	–	–	61	5.85	126	21.47	163	25.26
		2^{10}	–	–	73	31.09	191	159.38	260	224.56

表 4.8 数值结果

			GMRES		GMRES(P_1)		GMRES($\tilde{P}_2$)		GMRES(P_3)	
α	β	$M=N$	Iter	CPU	Iter	CPU	Iter	CPU	Iter	CPU
0.3	1.5	2^7	110	0.43	41	0.14	17	0.17	59	0.25
		2^8	204	4.34	50	0.87	19	0.62	71	1.50
		2^9	391	97.44	58	5.47	21	2.59	84	9.74
		2^{10}	–	–	67	28.52	23	10.6	96	47.94
0.5	1.5	2^7	131	0.57	43	0.14	22	0.22	55	0.23
		2^8	246	5.60	53	0.91	27	0.86	74	1.57
		2^9	482	140.68	63	6.12	34	4.32	101	12.24
		2^{10}	–	–	75	32.38	43	20.94	137	80.93
0.7	1.5	2^7	184	0.94	43	0.14	43	0.41	70	0.32
		2^8	358	9.90	54	0.94	63	2.05	108	2.37
		2^9	–	–	66	6.56	93	14.38	169	26.44
		2^{10}	–	–	79	35.48	136	98.62	269	241.11

表 4.9 数值结果

			GMRES		GMRES(P_1)		GMRES($\tilde{P}_2$)		GMRES(P_3)	
α	β	$M=N$	Iter	CPU	Iter	CPU	Iter	CPU	Iter	CPU
0.3	1.7	2^7	110	0.43	43	0.14	13	0.12	59	0.26
		2^8	204	4.34	52	0.90	15	0.50	70	1.50
		2^9	391	97.44	61	5.83	16	1.99	83	9.53
		2^{10}	–	–	72	30.37	17	7.45	96	46.67
0.5	1.7	2^7	131	0.57	47	0.16	18	0.16	58	0.26
		2^8	246	5.60	57	1.00	22	0.72	79	1.69
		2^9	482	140.68	69	7.00	26	3.19	107	13.33
		2^{10}	–	–	81	35.86	32	14.72	143	85.56
0.7	1.7	2^7	184	0.94	48	0.16	37	0.35	78	0.36
		2^8	358	9.90	60	1.06	54	1.74	120	2.68
		2^9	–	–	74	7.67	77	11.21	189	31.56
		2^{10}	–	–	91	42.14	111	70.18	300	285.33

4.4 基于 Teoplitz 直接求逆的交替方向预处理

4.4.1 预处理子的构造

考虑上一节中提出的交替方向矩阵分裂预处理子

$$P(\theta)=\frac{1}{2\theta}(\theta I_M+C)\otimes(\theta I_N+K).$$

在实际应用时需要计算 $(\theta I_M+C)^{-1}$ 和 $(\theta I_N+K)^{-1}$. 由于 $(\theta I_M+C)^{-1}$ 是 Toeplitz 矩阵, 我们可以采用 Gohberg-Semencul 方法来计算. 但 $(\theta I_N+K)^{-1}$ 不是 Toeplitz 矩阵, 直接计算成本较高. 因此我们需要对其做近似. 这里我们用 $\bar{K}=\tau(\bar{D}_+T_\beta+\bar{D}_-T_\beta^T)$ 来近似其中的 K, 于是我们就得到新的预处理子

$$\frac{1}{2\theta}(\theta I_M+C)\otimes(\theta I_N+\bar{K}).$$

此时, θI_M+C 和 $\theta I_N+\bar{K}$ 都是 Toeplitz 矩阵, 因此在实际计算时, 我们都可以采用 Gohberg-Semencul 直接求逆的方法. 预处理部分所需的运算量大约为 $\mathcal{O}(MN\log N+NM\log M)$, 与不带预处理的 GMRES 方法的每一个迭代步的运算量相当.

4.4.2 数值算例

为了便于各个预处理子之间的比较, 我们还是考虑数值算例 4.1. 采用相同的测试数据, 数值结果见表 4.10, 4.11 和 4.12, 其中“GMRES(P_4)”表示带预处理子 P_4 的 GMRES 方法. 由于不带预处理的 GMRES 方法收敛非常慢, 因此我们只列出带预处理的 GMRES 方法的数值结果.

从数值结果可以看出, 预处理子 P_4 的总体表现要优于 P_3. 当时间分数阶导数 α 较小时, P_4 的数值效果与 $\tilde{P}_2$ 相当, 收敛速度都比 P_1 快. 而当 α 较大时, P_4 的数值效果要优于 $\tilde{P}_2$.

表 4.10 数值结果

			GMRES(P_1)		GMRES($\tilde{P}_2$)		GMRES(P_3)		GMRES(P_4)	
α	β	$M=N$	Iter	CPU	Iter	CPU	Iter	CPU	Iter	CPU
0.3	1.3	2^7	41	0.13	23	0.22	57	0.25	26	0.41
		2^8	50	0.86	27	0.85	70	1.45	30	1.50
		2^9	59	5.64	31	3.82	84	9.60	33	5.98
		2^{10}	67	27.63	36	17.14	98	48.22	36	23.02
0.5	1.3	2^7	41	0.13	30	0.27	54	0.23	32	0.50
		2^8	51	0.90	39	1.26	73	1.56	42	2.07
		2^9	61	5.83	51	6.81	101	12.29	55	10.29
		2^{10}	71	29.85	66	35.49	139	81.75	73	52.35
0.7	1.3	2^7	40	0.13	55	0.51	67	0.30	43	0.68
		2^8	50	0.87	83	2.76	103	2.24	66	3.35
		2^9	61	5.85	126	21.47	163	25.26	102	21.45
		2^{10}	73	31.09	191	159.38	260	224.56	159	144.42

表 4.11 数值结果

			GMRES(P_1)		GMRES($\tilde{P}_2$)		GMRES(P_3)		GMRES(P_4)	
α	β	$M=N$	Iter	CPU	Iter	CPU	Iter	CPU	Iter	CPU
0.3	1.5	2^7	41	0.14	17	0.17	59	0.25	19	0.32
		2^8	50	0.87	19	0.62	71	1.50	20	1.07
		2^9	58	5.47	21	2.59	84	9.74	21	3.73
		2^{10}	67	28.52	23	10.6	96	47.94	22	13.51
0.5	1.5	2^7	43	0.14	22	0.22	55	0.23	23	0.36
		2^8	53	0.91	27	0.86	74	1.57	29	1.43
		2^9	63	6.12	34	4.32	101	12.24	35	6.32
		2^{10}	75	32.38	43	20.94	137	80.93	44	28.88
0.7	1.5	2^7	43	0.14	43	0.41	70	0.32	33	0.52
		2^8	54	0.94	63	2.05	108	2.37	46	2.28
		2^9	66	6.56	93	14.38	169	26.44	67	13.02
		2^{10}	79	35.48	136	98.62	269	241.11	99	77.10

表 4.12 数值结果

			GMRES(P_1)		GMRES($\tilde{P}_2$)		GMRES(P_3)		GMRES(P_4)	
α	β	$M=N$	Iter	CPU	Iter	CPU	Iter	CPU	Iter	CPU
0.3	1.7	2^7	43	0.14	13	0.12	59	0.26	14	0.24
		2^8	52	0.90	15	0.50	70	1.50	15	0.76
		2^9	61	5.83	16	1.99	83	9.53	15	2.61
		2^{10}	72	30.37	17	7.45	96	46.67	16	9.65
0.5	1.7	2^7	47	0.16	18	0.16	58	0.26	18	0.29
		2^8	57	1.00	22	0.72	79	1.69	21	1.03
		2^9	69	7.00	26	3.19	107	13.33	25	4.41
		2^{10}	81	35.86	32	14.72	143	85.56	30	18.94
0.7	1.7	2^7	48	0.16	37	0.35	78	0.36	25	0.40
		2^8	60	1.06	54	1.74	120	2.68	34	1.68
		2^9	74	7.67	77	11.21	189	31.56	48	9.04
		2^{10}	91	42.14	111	70.18	300	285.33	68	48.15

第五章　总结与展望

本文主要研究的问题是时间-空间分数阶扩散方程预处理方法. 由于传统的按时间步的求解方法在计算当前时间步的解时, 需要计算先前所有时间步的解, 这就导致了当我们在很长的时间上建模, 问题的求解变得非常的冗长. 为了解决这样的问题, 有人就提出了将问题的时间离散和空间离散后的形式联结起来, 写成一个用 Kronecker 积组合的形式. 这样的矩阵形式简洁明了, 并且只需要求解该问题一次即可. 但问题是, 该形式的系数矩阵规模非常的大, 并且不具有明显的 Toeplitz 形式.

针对这种形式的系数矩阵, 我们主要讨论了四种预处理方法. 一个是简单地将系数矩阵中的 Toeplitz 矩阵用循环矩阵来近似, 得到一个简单循环预处理子. 一个是根据系数矩阵的结构特点, 构造出比较直观的基于 Toeplitz 直接求逆的块对角预处理子. 数值算例显示在时间分数阶导数 α 较小的时候 (比如小于 0.5), 我们的基于 Toeplitz 直接求逆的块对角预处理子是优于我们的循环预处理子的, 但是在时间分数阶导数 α 较大时 (比如大于 0.5), 我们的基于 Toeplitz 直接求逆的块对角预处理子表现得不是很好. 借鉴交替方向迭代思想, 并利用循环矩阵近似 Toeplitz 矩阵和 Toeplitz 矩阵直接求逆的技术, 我们构造两个交替方向矩阵分裂预处理子, 数值算例显示出了我们的基于 Toeplitz 直接求逆的交替方向预处理子具有较大的优越性.

考虑到预处理子的实用性, 我们在构造预处理子时采取了较多的近似方法, 这使得我们的预处理子在时间分数阶较小时, 具有很好的数值效果. 对于时间分数阶较大, 如何构造有效的预处理子, 需要在今后的工作中做进一步的研究.

参考文献

[1] O.P. Agrawal, A general formulation and solution scheme for fractional optimal control problems, *Nonlinear Dyn.*, 38 (2004), 191–206.

[2] G. Ammar and W. Gragg, Superfast solution of real positive definite Toeplitz systems, *SIAM J. Matrix Anal. Appl.*, 9 (1988), pp. 61–76.

[3] Z.Z. Bai, K.Y. Lu, and J.Y. Pan, Diagonal and Toeplitz splitting iteration methods for diagonal-plus-Toeplitz linear systems from spatial fractional diffusion equations, *Numer. Linear. Algebra. Appl.*, 24 (2017), e2093.

[4] D. Benson, S.W. Wheatcraft and M.M. Meerschaert, The fractional-order governing equation of Lévy motion, *Water Resour. Res.*, 36 (2000), 1413–1423.

[5] D.A. Benson, S.W. Wheatcraft, M.M. Meerschaert, Application of a fractional advection-dispersion equation, *Nonlinear Anal.*, 36 (2006), 1403–1412.

[6] G. Baxter, Polynomials defined by a difference system, *J. Math. Anal. Appl., 2(1961)*, pp. 223–263.

[7] R. Bitmead and B. Anderson, Asymptotically fast solution of Toeplitz and related systems of linear equations, *Linear Algebra Appl.*, 34 (1980), pp. 103–116.

[8] R. Brent, F. Gustavson, and D. Yun, Fast solution of Toeplitz systems of equations and computation of Padé approximants, *J. Algorithms*, 1 (1980), pp. 259–295.

[9] M. Caputo, Linear model of dissipation whose Q is almost frequency independent–II, *Geophys. J. R. Astr. Soc.*, 13 (1967), 529–539.

[10] M. Caputo, Elasticità e Dissipazione, *Zanichelli, Bologna*, 1969.

[11] R.H. Chan and Michael K.NG, Conjugate gradient methods for Toeplitz systems, *SIAM Review*, 38 (1996), pp. 427–482.

[12] R. Chan, Circulant preconditioners for Hermitian Toeplitz systems, *SIAM J. Matrix Anal. Appl.*, 10 (1989), pp. 542–550.

[13] T. Chan, An optimal circulant preconditioner for Toeplitz systems, *SIAM J. Sci. Stat. Comput.*, 9 (1988), pp. 766–771.

[14] R. Chan, Circulant preconditioners for Hermitian Toeplitz systems, *SIAM J. Matrix Anal. Appl.*, 10 (1989), pp. 542–550.

[15] R. Chan and G. Strang, Toeplitz equations by conjugate gradients with circulant preconditioner, *SIAM J. Sci. Stat. Comput.*, 10 (1989), pp. 104–119.

[16] S. Chen, F. Liu, X. Jiang, I. Turner and V. Anh, A fast semi-implicit difference method for a nonlinear two-sided space-fractional diffusion equation with variable

diffusivity coefficients, *Appl. Math. Comput.*, 257 (2015), 591–601.

[17] J.W. Cooley and J.W. Tukey, Analgorithm for the machine calculation of complex Fourier series, *Math. Comput.*, 19 (1965), 297–301.

[18] K. Diethelm, *The Analysis of Fractional Differential Equations －An Application-Oriented Exposition Using Differential Operators of Caputo Type*, Springer,, 2010.

[19] M. Donatelli, M. Mazza, and S. Serra-Capizzano, Spectral analysis and structure preserving preconditioners for fractional diffusion equations, *J. Comput. Phys.*, 307 (2016), 262–279.

[20] P. Duhamel and M. Vetterli, Fast fourier transform: A tutorial review and a state of the art, *Signal Processing*, (19) 1990, 259–299.

[21] V. Gafiychuk, B. Datsko, and V. Meleshko, Mathematical modeling of time fractional reaction diffusion systems, *J. Math. Anal. Appl.*, 220 (2008), 215–225.

[22] I. Gohberg and A. Semencul, On the inversion of finite Toeplitz matrices and their continuous analogs, *Mat. Issled.*, 2 (1972), pp. 201–233.

[23] G. H. Golub and C. F. Van Loan, *Matrix Computations*, The 4th Editon, The Johns Hopkins University Press, Baltimore, MD, 2013.

[24] X.-M. Gu, T.-Z. Huang, C.-C. Ji, B. Carpentieri, and A.A. Alikhanov, Fast iterative method with a second-order implicit difference scheme for time-space fractional convection－diffusion equation, *J. Sci. Comput.*, 72 (2017), 957–985.

[25] X.-M. Gu, T.-Z. Huang, X.-L. Zhao, H.-B. Li, and L. Li, Strang-type preconditioners for solving fractional diffusion equations by boundary value methods, *J. Comput. Appl. Math.*, 277 (2015), 73–86.

[26] F. De Hoog, A new algorithm for solving Toeplitz systems of equations, *Linear Algebra Appl.*, 88/89 (1987), pp. 123–138.

[27] Y.-C. Huang and S.-L. Lei, A fast numerical method for block lower triangular Toeplitz with dense toeplitz blocks system with applications to time-space fractional diffusion equations, *Numer. Algorithms*, 76 (2017), 605–616.

[28] X.Q. Jin, F.R. Lin, and Z. Zhao, Preconditioned iterative methods for two-dimensional space-fractional diffusion equations, *Commun. Comput. Phys.*, 18 (2015), 469–488.

[29] D. Kusnezov, A. Bulgac, and G.D. Dang, Quantum Lévy processes and fractional kinetics, *Phys. Rev. Lett.*, 82 (1999), 1136–1139.

[30] R. Ke, M.K. Ng, and H.-W. Sun, A fast direct method for block triangular Toeplitz-like with tri-diagonal block systems from time-fractional partial differential equations, *J. Comput. Phys.*, 303 (2015), 203–211.

[31] S.L. Lei and H.W. Sun, A circulant preconditioner for fractional diffusion equa-

tions, *J. Comput. Phys.*, 242 (2013), 715–725.

[32] N. Levinson, The Wiener RMS (root mean square) error criterion in filter design and prediction, *J. Math. Phys.*, 25 (1946), pp. 261-278.

[33] M. Li, X.-M. Gu, C. Huang, M. Fei, and G. Zhang, A fast linearized conservative finite element method for the strongly coupled nonlinear fractional Schrödinger equations, *J. Comput. Phys.*, 358 (2018), 256–282.

[34] R.R. Lin, S.W. Yang, and X.Q. Jin, Preconditioned iterative methods for fractional diffusion equation, *J. Comput. Phys.*, 256 (2014), 109–117.

[35] X. Lu, H.-K. Pang, and H.-W. Sun, Fast apprioximate inversion of a block triangular Toeplitz matrix with applications to fractional sub-diffusion equations, *Numer. Linear Algebra Appl.*, 22 (2015), 866–882.

[36] M.M. Meerschaert and E. Scalas, Coupled continuous time random walks in finance, *Phys. A*, 390 (2006), 114–118.

[37] M.M. Meerschaert and C. Tadjeran, Finite difference approximations for fractional advection-dispersion flow equation, *J. Comput. Appl. Math.*, 172 (2004).

[38] M.M. Meerschaert and C. Tadjeran, Finite difference approximations for two-sided space-fractional partial differential equations, *Appl. Numer. Math.*, 56 (1) (2006), 80-90.

[39] M. Morf, Doubling algorithms for Toeplitz and related equations, in *Proc. IEEE Intl. Conf.* on *Acoust.Speech and Signal Process.*, 3 (1980), pp. 954–959.

[40] J. Pan, R. Ke, M. Ng, and H. Sun, Preconditioning techniques for diagonal-times-Toeplitz matrices in fractional diffusion equations, *SIAM J. Sci. Comput.*, 36 (2014), 2698–2719.

[41] H. Pang and H.W. Sun, Multigrid method for fractional diffusion equations, *J. Comput. Phys.*, 231 (2012), 693–703.

[42] I. Podlubny, *Fractional Differential Equations,* Academic Press, San Diego, 1999.

[43] E. C. de Oliveira and J. A. Tenreiro Machado, A review of definitions for fractional derivatives and integral Mathematical Problems in Engineering, 2014 (2014), Article ID 238459.

[44] S.G. Samko, A.A. Kilbas, and O.I. Marichev, *Fractional Integrals and Derivatives: Theory and Applications*, Gordon and Breach Science Publishers, Switzerland,, 1993.

[45] A. Simmons, Q.Q. Yang, and T. Moroney, A preconditioned numerical solver for stiff nonlinear reaction-diffusion equations with fractional Laplacians that avoids dense matrices, *J. Comput. Phys.*, 287 (2015), 254–268.

[46] G. Strang, A proposal for Toeplitz matrix calculations, *Stud. Appl. Math.*, 74

(1986), pp. 171–176.

[47] W. Trench, An algorithm for the inversion of finite Toeplitz matrices, *SIAM J. Appl. Math.*, 12 (1964), pp. 515–522.

[48] C. Van Loan, *Computational Frameworks for the Fast Fourier Transform*, SIAM, Philadelphia, 1992.

[49] H. Wang and N. Du, A fast finite difference method for three-dimensional time-dependent space-fractional diffusion equations and its efficient implementation, *J. Comput. Phys.*, 253 (2013), 50–63.

[50] H. Wang and K. Wang, An $\mathcal{O}(N \log^2 N)$ alternating-direction finite difference method for two-dimensional fractional diffusion equations, *J. Comput. Phys.*, 230 (2011), pp.7830–7839.

[51] H. Wang, K. Wang, and T. Sircar, A direct $\mathcal{O}(n \log^2 n)$ finite difference method for fractional diffusion equations, *J. Comput. Phys.*, 229 (2010), 8095–8104.

[52] Y.-L. Zhao, P.-Y. Zhu, and W.-H. Luo, A fast second-order implicit scheme for non-linear time-space fractional diffusion equation with time delay and drift term, *Appl. Math. Comput.*, 336 (2018), 231–248.

[53] S. Zohar, The solution of a Toeplitz set of linear equations, *Journal of the ACM*, 21 (1974), pp. 272–276.

致谢

研究生三年的生涯转瞬间就要结束了, 这三年的时光我经历了很多, 也收获了很多.

首先, 我非常感谢我的导师潘建瑜教授, 在论文的撰写过程中, 潘老师给了我很大的帮助. 在知识积累阶段离不开潘老师的细心指导, 在论文的难点攻克阶段离不开潘老师的热心帮助. 在平时的学习过程中, 潘老师端正了我的学习态度, 使我养成了脚踏实地的学习态度以及生活态度. 潘老师的严谨的学术风格深刻地影响着我的学习态度, 并且激励着我无论是在学习中, 还是生活中, 都奋勇向前.

同时也感谢我的同门冯凯玥在平时抽出时间和我探讨论文中的问题, 加深了我对所研究问题的理解.

还有, 正是我家里人对我的支持和理解, 我才能全身心地投入到论文的撰写过程中.

最后, 感谢参加本文评审工作的答辩委员会的老师们能在百忙之中抽空参加我的论文答辩, 感谢老师们的工作与付出.

管国祥

2020 年 3 月于华东师范大学