湘潭大学毕业论文

题　　目: 空间分数阶对流扩散方程的有限差分格式求解

院　　系: 数学与计算科学学院

专　　业: 信息与计算科学

学　　号: 2015750413

姓　　名: 周铁军

指导老师: 文立平　教授

完成日期: 2019 年 5 月 10 日

目　录

空间分数阶对流扩散方程的有限差分格式求解

摘要: 分数阶微分方程源于实际。区别于整数阶，分数阶微分方程可以更好地模拟一些动态过程与物理现象，在 Engineering Science、Physics、Finance、Hydrology 等范畴中起到着重要的作用。大多数分数阶微分方程的解析解有复杂的形式，不便于近似计算，因此，数值求解成了研究分数阶微分方程最重要的方法之一。本文主要研究一维空间分数阶对流-扩散方程的数值解法。通过离散 Riemann-Liouville 空间分数阶导数，构造 Lévy-Feller 对流—扩散方程的差分格式，利用有限差分方法求得数值解，并考虑加入干扰项，分析该格式的稳定性。

关键词:Lévy-Feller 对流-扩散方程 [2][1]；Riemann-Liouville 空间分数阶导数；有限差分方法；Grünwald-Letnikov 分数阶导数移位算子

Solution of finite difference scheme for spatial fractional convection-diffusion equations

Abstract: Fractional differential equations derive from reality. Different from integral order differential equation, fractional differential equation can better simulate some dynamic processes and physical phenomena, playing an important role in Engineering Science, Physics, Finance, Hydrology and other fields. Most analytic solutions of fractional differential equations have complex forms, which makes it difficult to approximate. Therefore, numerical solution has become one of the most important methods to study fractional differential equations. The numerical solution of fractional convection-diffusion equations in one-dimensional space is studied. The fractional derivative of Riemann-Liouville space is discretized to construct the difference scheme of the Lévy-Feller convection-diffusion equation.

Key words: The Lévy-Feller convection-diffusion equation; Riemann-Liouville space fractional derivative; Finite difference method; The Grünwald-Letnikov fractional derivative shift operator

第一章　引言

1.1　研究背景及现状

分数阶微积分作为整数阶微积分理论的推广，其历史可以上溯到莱布尼兹时代，当时它写给洛必达的信中就谈及了 $\frac{1}{2}$ 阶导数的意义及其相关内容。从那时起，Lagrange、Laplace、Fourier、Euler、Riemann、Liouville 等人为该领域做出了突出贡献，他们按照不同的基础与目的对分数阶导数做出了多种解释，从而得到了 Riemann-Liouville、Grünwald-Letnikov、Caputo、Riesz 等多种形式的分数阶微分方程形式。

分数阶微分方程是常被用来描述记忆特性和中间过程，它以分数阶微积分理论为支撑，是传统模型的延伸。在 Fractal theory, Random walk, Viscoelastic mechanics 等领域中起到了重要的作用，对于对流-扩散问题的数值计算方法的研究至关重要。它既补充微分方程数学理论的空白，又给出了研究 Physics、Biology、Chemistry、Economics 等范畴更好的数学模型，尤其是随着计算机的普及与成长，各类学者都可以借助程序语言实现算法，描述事物的物理意义，从而促成这些学科的蓬勃发展。

关于分数阶偏微分方程的数值求解，国内外学者提出了许多数值方法，其中有限差分方法发展最为完善。Meescharet 和 Tadjeran 等人首先提出利用一阶经典和移位 Grünwald-Letnikov 算子；Deng. 等人借助 WENO 格式，提出了空间精度为 6 阶的有限差分格式；Cui. 给出了紧致有限差分格式；Alikhanov 于分数阶导数提出了一种新的差分离散公式，并建立了相应的空间四阶和时间二阶的有限差分格式。Liu. 等人开创性地提出行方法，采用可变阶、可变步长的向后差分公式, 实现了将分数阶偏微分方程到常微分方程系统的转化 [2]。

迄今为止，空间分数阶微分方程大多是含有单侧分数阶导数或者对称的双侧导数，时间分数阶导数一般为 Caputo 导数，而空间分数阶导数则有很多种，包括单侧的分数阶导数——Riemann-Liouville 分数阶导数、Grünwald-Letnikov 分数阶导数、Caputo 分数阶导数、Riesz-Feller 分数阶导数等 [2]。文章中所研究的 Levy-Feller 对流-扩散方程就是利用的双侧 Riemann-Liouville 分数阶导数。

然而，因为分数阶微分方程的特殊性，其数值解法还不很成熟，依旧存在许多难而未决的问题。故如何得到求解分数阶微分方程的可行的数值方法，将会是一项富有价值的研究课题。此亦是本文研究的意义之所在。

本文研究的 Lévy-Feller 对流-扩散方程归纳于 Random walk 和一种随机过程的 Stable 分布，方程中含有非对称的 Riesz-Feller 分数阶导数，它含有双侧分数阶导数 ${}_hD_{\pm}^{\alpha}$ 及倾斜度参数 θ。

1.2 主要工作与安排

本文的主要工作如下:

文章主要研究 Riemann-Liouville 导数意义下的 Lévy-Feller 对流-扩散方程的定解问题。

本文共分四章, 具体安排如下：第一章，简单地阐述分数阶微积分的历史渊源、研究情况和数值解法的研究价值。

第二章，给出一些预备知识，包括 Riemann-Liouville 分数阶导数、Grünwald-Letnikov 分数阶导数移位算子 [2][1]、Toeplitz 矩阵及相关定理。

第三章，研究一维的有限区间上的 Lévy-Feller 对流-扩散微分方程的有限差分解法，并用 Matlab 语言编程实现上述算法，利用几个数值试验验证该格式的稳定性及收敛性进行分析，最终讨论其有效性、精确性和可靠性。

第四章，总结本文的主要研究工作及成果，并给出新的研究方向和着手点。

第二章　预备知识

本节主要给出一些分数阶微积分的定义和性质，为后续章节的研究提供理论支撑和根据。记 $\Omega = [a,b]$(a,b 为有限数或 ∞)。

2.1 Gamma 函数

定义 2.1.1. *Gamma 函数的积分形式定义如下 [8]*：

$$\Gamma(z) = \int_0^{\infty} e^{-t} t^{z-1} dt, \quad \mathrm{Re}(z) > 0 \tag{2.1.1}$$

Gamma 函数亦可用极限形式表示：

$$\Gamma(z) = \lim_{n\to\infty} \frac{n! n^z}{z(z+1)\cdots(z+n)}, \quad \mathrm{Re}(z) > 0 \tag{2.1.2}$$

Gamma 函数具有如下性质 [6]：

$$\Gamma(z+1) = z\Gamma(z), \quad \forall z \in (0,+\infty) \tag{2.1.3}$$

$$\Gamma(n+1) = n!, \quad \forall n \in \mathbf{C} \tag{2.1.4}$$

$$\Gamma(z) \geq 0, \forall z \in (0,+\infty), \text{且} \quad \Gamma(1) = 1, \Gamma\left(\frac{1}{2}\right) = \sqrt{\pi} \tag{2.1.5}$$

2.2 Riemann-Liouville 分数阶导数

如今，最常用的分数阶导数定义有三种：Grünwald-Letnikov 分数阶导数、Riemann-Liouville 分数阶导数、Caputo 分数阶导数。文中，分数阶导数指的是 Riemann-Liouville 空间分数阶导数。

定义 2.2.1. *(左侧 Riemann-Liouville 分数阶导数) 设 u 是定义于 Ω 上的可积函数，$\alpha > 0$，设 n 为比 α 大的最小整数 $(n-1 \leq \alpha < n)$, 记 $\sigma = n - \alpha$，定义*

$$\frac{d^{\alpha}}{dx^{a}} u(x) = \frac{d^n}{dx^n}\left(\frac{d^{-\sigma}}{dx^{-\sigma}} u(x)\right) = \frac{d^n}{dx^n}\left(\frac{1}{\Gamma(\sigma)} \int_a^x (x-\xi)^{\sigma-1} u(\xi) d\xi\right) \tag{2.2.1}$$

为 $u(x)$ 的 α 阶左侧 Riemann-Liouville 分数阶导数。

定义 2.2.2. *(右侧 Riemann-Liouville 分数阶导数)* 设 u 是定义于 Ω 上的可积函数，$\alpha > 0$，设 n 为比 α 大的最小整数 $(n-1 \leq \alpha < n)$, 记 $\sigma = n - \alpha$, 定义

$$\frac{d^\alpha}{d(-x)^\alpha}u(x) = (-1)^n \frac{d^n}{dx^n}\left(\frac{d^{-\sigma}}{d(-x)^{-\sigma}}u(x)\right) = (-1)^n \left(\frac{d^n}{dx^n}\frac{1}{\Gamma(\sigma)}\int_x^b (\xi - x)^{\sigma-1}u(\xi)d\xi\right) \quad (2.2.2)$$

为 $u(x)$ 的 α 阶右侧 *Riemann-Liouville* 分数阶导数。

2.3 Grünwald-Letnikov 分数阶导数移位算子

定义 2.3.1. *(左侧 Grünwald-Letnikov 分数阶导数移位算子 [1])* 设 u 是定义于 Ω 上的可积函数，$\alpha > 0$，p 为正常数，定义

$${}_aD_z^\alpha u(x) = \lim_{n\to\infty}\frac{1}{h^\alpha}\sum_{k=0}^{n+p}(-1)^k \left(\begin{array}{c}\alpha\\ k\end{array}\right) u(x-(k-p)h), \quad h = \frac{x-a}{n} \quad (2.3.1)$$

为 $u(x)$ 的 α 阶左侧 *Grünwald-Letnikov* 分数阶导数的移位算子。

定义 2.3.2. *(右侧 Grünwald-Letnikov 分数阶导数移位算子)* 设 u 是定义于 Ω 上的可积函数，$\alpha > 0$，p 为正常数，定义

$${}_xD_b^\alpha u(x) = \lim_{n\to\infty}\frac{1}{h^a}\sum_{k=0}^{n+p}(-1)^k \left(\begin{array}{c}\alpha\\ k\end{array}\right) u(x+(k-p)h), \quad h = \frac{b-x}{n} \quad (2.3.2)$$

为 $u(x)$ 的 α 阶右侧 *Grünwald-Letnikov* 分数阶导数的移位算子。

其中 $k \geq 1$ 时

$$\left(\begin{array}{c}\alpha\\ k\end{array}\right) = \frac{\alpha(\alpha-1)\cdots(\alpha-k+1)}{k!} = \left(\frac{\alpha+1}{k}-1\right)\left(\begin{array}{c}\alpha\\ k-1\end{array}\right) \quad (2.3.3)$$

2.4 Toeplitz 矩阵与循环矩阵

定义 2.4.1. *Toeplitz[8]* 矩阵是任何一条与主对角线平行的对角线都取相同元素的矩阵。

一般而言，一个 n 阶的 Toeplitz 矩阵 Tn 可以由一组 2n-1 个数字 $t_{i\,i=1-n}^{\,n-1}$ 确定，矩阵元素 $T_n(i,j) = t_{j-i}$, 其中 i,j=1,2,...,n, 即:

$$T_n = \begin{bmatrix} t_0 & t_1 & t_2 & \cdots & t_{n-2} & t_{n-1} \\ t_{-1} & t_0 & t_1 & \cdots & t_{n-3} & t_{n-2} \\ t_{-2} & t_{-1} & t_0 & \ddots & \ddots & t_{n-3} \\ \vdots & \vdots & \ddots & \ddots & \ddots & \vdots \\ t_{2-n} & t_{3-n} & \ddots & \ddots & t_0 & t_1 \\ t_{1-n} & t_{2-n} & t_{3-n} & \cdots & t_{-1} & t_0 \end{bmatrix} \quad (2.4.1)$$

定义 2.4.2. 循环矩阵是一种特殊形式的 *Toeplitz* 矩阵, 它的行向量的每个元素都是前一个行向量各元素依次右移一个位置得到。

一般而言，一个 n 阶的循环矩阵 Cn 可以由一组 n 个数字 $c_i{}_{i=0}^{n-1}$ 完全确定，矩阵元素 $C_n(i,j) = c_{(j-i)mode \quad n}$, 其中 i,j=1,2,...,n, 即:

$$C_n = \begin{bmatrix} c_0 & c_1 & c_2 & \cdots & c_{n-2} & c_{n-1} \\ c_{n-1} & c_0 & c_1 & \cdots & c_{n-3} & c_{n-2} \\ c_{n-2} & c_{n-1} & c_0 & \ddots & \ddots & c_{n-3} \\ \vdots & \vdots & \ddots & \ddots & \ddots & \vdots \\ c_2 & c_3 & \ddots & \ddots & c_0 & c_1 \\ c_1 & c_2 & c_3 & \cdots & c_{n-1} & c_0 \end{bmatrix} \tag{2.4.2}$$

定理 2.4.1. *一个 n 阶 Toeplitz 矩阵 T_n 可以扩充为一个 $2n$ 阶的循环矩阵 C_{2n}。*

证明：. T_n 是定义 2.4.1 中的一个 n 阶 Toeplitz 矩阵，令 B_n 定义如下:

$$B_n = \begin{bmatrix} 0 & t_{1-n} & \cdots & \cdots & t_{-2} & t_{-1} \\ t_{n-1} & 0 & t_{1-n} & \ddots & \ddots & t_{-2} \\ t_{n-2} & t_{n-1} & 0 & \ddots & \ddots & t_{n-3} \\ \vdots & \vdots & \ddots & \ddots & \ddots & \vdots \\ t_2 & \vdots & \ddots & \ddots & 0 & t_{1-n} \\ t_1 & t_2 & \cdots & \cdots & t_{n-1} & 0 \end{bmatrix} \tag{2.4.3}$$

那么 $C_{2n} = \begin{bmatrix} T_n & B_n \\ B_n & T_n \end{bmatrix}$ 是一个每行都是 2n 个元素 $t_0, t_1, ..., t_{n-1}, 0, t_{1-n}, ..., t_{-1}$ 的排列的循环矩阵。 □

第三章　Lévy-Feller 对流-扩散方程的数值解法

1905 年，Einstein 根据布朗运动现象第一次提出扩散方程，而分数阶对流—扩散方程直接来源于这个结果的推广，并且断言其可以用来描述反常扩散过程。分数阶对流—扩散方程的解为一个 Stable 分布函数——方差无穷大的独立同分布的随机变量序列的极限分布函数。本章着重探讨的是 Lévy-Feller 对流-扩散方程。

3.1　Lévy-Feller 对流-扩散方程的有关知识

Lévy-Feller 对流-扩散方程表达的是满足某种稳定分布反常扩散的非对称空间分数阶对流-扩散方程 [2]。

3.1.1　Lévy-Feller 对流-扩散方程的表达式

首先引入 Lévy-Feller 对流—扩散微分方程：

$$\frac{\partial u(x,t)}{\partial t} = aD_\theta^\alpha u(x,t) - b\frac{\partial u(x,t)}{\partial x} \tag{3.1.1}$$

其中 a 为正常数，b 为常数，算子 D_θ^α 表示阶数为 α、倾斜度为 θ 的 Riesz-Feller 分数阶导数。Riemann-Liouville 分数阶导数以其 Fourier 变换形式给出：

$$\widehat{D}_\theta^\alpha = -|\kappa|^\alpha e^{i(\operatorname{sign}\kappa)\theta\pi/2} \tag{3.1.2}$$

由文献 [2] 中的内容可知：

$$D_\theta^\alpha = -\left[c_+(\alpha,\theta)\frac{d^\alpha}{dx^\alpha} + c_-(\alpha,\theta)\frac{d^\alpha}{d(-x)^\alpha}\right]$$

其中 $\frac{d^\alpha}{dx^\alpha}$ 和 $\frac{d^\alpha}{d(-x)^\alpha}$ 分别为左侧和右侧 Riemann-Liouville 分数阶导数算子。系数 c+、c-如下：

$$\begin{cases} c_+ = c_+(\alpha,\theta) := \dfrac{sin((\alpha-\theta)\pi/2)}{sin(\alpha\pi)} \\ c_- = c_-(\alpha,\theta) := \dfrac{sin((\alpha+\theta)\pi/2)}{sin(\alpha\pi)} \end{cases}$$

其中参数 α、θ 满足：

$$0 < \alpha \le 2(\alpha \ne 1); \quad |\theta| \le \begin{cases} \alpha, 0 < \alpha < 1 \\ 2-\alpha, 1 < \alpha \le 2. \end{cases}$$

在此给出参数 c_+ c_- 的性质:

$$c_{\pm}\begin{cases}\geq 0, 0<\alpha<1\\ \\ \leq 0, 1<\alpha\leq 2.\end{cases}$$

引理 3.1.1. *初边值条件如下：*

$$\begin{cases}u(x,0)=\varphi(x), \quad (x\in\mathbb{R})\\ \\ u(\pm\infty,t)=0, \quad (t>0)\end{cases}$$

的 Lévy-Feller 对流-扩散方程的解析解为

$$u(x,t)=\frac{1}{2\pi}\int_{-\infty}^{+\infty+\infty} e^{-i\kappa\xi}e^{t\left(-a|\kappa|^{\alpha}e^{i(sign\kappa)\theta\pi/2}+ib\kappa\right)}\varphi(x-\xi)d\kappa d\xi$$

详细证明请参考文献 [2]。

3.1.2 有限区间内的初边值问题的数值解法

给出初边值问题如下;

$$\begin{array}{ll}\frac{\partial u(x,t)}{\partial t}=aD_{\theta}^{\alpha}u(x,t)-b\frac{\partial u(x,t)}{\partial x}, & 0<x<R, \quad 0<t<T\\ u(x,0)=\varphi(x), & 0\leq x\leq R\\ u(0,t)=u(R,t)=0, & 0\leq t\leq T\end{array} \tag{3.1.3}$$

先进行网格剖分，将空间区间 [0,L] 作 M 等分，时间区间 [0,T] 作 N 等分。其中 h 和 τ 分别表示空间步长和时间步长，M 和 N 为给定正整数。

利用空间网格点

$$x_j=jh, h>0, j=0,\pm1,\pm2,...$$

和时间间隔

$$t_n=n\tau, \tau>0, n=0,1,2,...$$

离散空间和时间变量。引入 $y_j(t_n)$:

$$y_j\left(t_n\right)=\int_{x_j-h/2}^{x_j+h/2}u\left(x,t_n\right)dx\approx hu\left(x_j,t_n\right)$$

来离散因变量 u(x,t).

其次，利用一阶差商离散一阶导数 $\frac{\partial u}{\partial t}$ 和 $\frac{\partial u}{\partial x}$ 得到:

$$\frac{\partial u}{\partial t}=\frac{y_j\left(t_{n+1}\right)-y_j\left(t_n\right)}{\tau}+O(t) \tag{3.1.4}$$

$$\frac{\partial u}{\partial x}=\frac{y_j\left(t_n\right)-y_{j-1}\left(t_n\right)}{h}+O(h) \tag{3.1.5}$$

不妨假设方程的解有如下性质:一阶导数连续,二阶导数可积,则此函数在 Riemann-Liouville 和 Grünwald-Letnikov 意义下的 α 阶分数阶导数一致。利用这个性质，离散 Riemann-Liouville 分数阶导数，得到算子 D_{θ}^{α} 的离散格式 [6]: 离散所有变量，有如下差分格式:

$$\frac{y_j\left(t_{n+1}\right)-y_j\left(t_n\right)}{\tau}=a_hD_{\theta}^{\alpha}y_j\left(t_n\right)-b\frac{y_j\left(t_n\right)-y_{j-1}\left(t_n\right)}{h} \tag{3.1.6}$$

其中差分算子 ${}_hD_\theta^\alpha$:

$$ {}_hD_\theta^\alpha y_j(t_n) = -\left[c_+ {}_hD_+^\alpha y_j(t_n) + c_- {}_hD_-^\alpha y_j(t_n)\right] \tag{3.1.7}$$

我们把导数移位算子的差分格式表示:

$$ {}_hD_\pm^\alpha y_j(t_n) = \frac{1}{h^\alpha}\sum_{k=0}^{\infty}(-1)^k \begin{pmatrix} \alpha \\ k \end{pmatrix} y_{j\pm1\mp k}(t_n) \tag{3.1.8}$$

为了分析算子 ${}_hD_\theta^\alpha$ 差分格式的收敛性，引入如下引理。

引理 3.1.2. *若函数 $u \in L^1(R)$ 和 $H^{\alpha+1}(R)$,${}_hD_+^\alpha$ 为左侧 Grünwald-Letnikov 分数阶导数算子(或移位算子) 的离散：*

$$ {}_hD_+^\alpha f(x) = \frac{1}{h^\alpha}\sum_{k=0}^{\infty}(-1)^k \begin{pmatrix} \alpha \\ k \end{pmatrix} f(x-(k-p)h) \tag{3.1.9}$$

其中 p 是一个非负的整数 ($p=0$ 对应分数阶导数算子，$p>0$ 对应移位算子)，那么有：当 $h\to 0$ 时，在 $x\in R$ 一致地有：

$$ {}_hD_+^\alpha f(x) =_{-\infty} D_x^\alpha f(x) + O(h) \tag{3.1.10}$$

于是原方程最终可表示为:

$$\begin{aligned} \frac{\boldsymbol{u}_j^{n+1}-\boldsymbol{u}_j^n}{\tau} =& -\frac{\boldsymbol{a}}{\boldsymbol{h}^\alpha}\left[\boldsymbol{c}_+\sum_{k=0}^{j+1}(-1)^k\begin{pmatrix}\alpha\\ \boldsymbol{k}\end{pmatrix}\boldsymbol{u}_{j+1-k}^n + \boldsymbol{c}_-\sum_{k=0}^{N-j+1}(-1)^k\begin{pmatrix}\boldsymbol{u}\\ \boldsymbol{k}\end{pmatrix}\boldsymbol{u}_{j-1+k}^n\right] \\ & -\boldsymbol{b}\frac{\boldsymbol{u}_j^n-\boldsymbol{u}_{j-1}^n}{h} + O(t+h) \end{aligned} \tag{3.1.11}$$

下面我们作如下处理，将 n+1 阶与 n 阶分离:

$$u_j^{n+1} = u_j^n - \frac{a\tau}{h^\alpha}\left[c_+\sum_{k=0}^{j+1}(-1)^k\begin{pmatrix}\alpha\\ k\end{pmatrix}u_{j+1-k}^n + c_-\sum_{k=0}^{N-j+1}(-1)^k\begin{pmatrix}\alpha\\ k\end{pmatrix}u_{j-1+k}^n\right] - b\tau\frac{u_j^n-u_{j-1}^n}{h}$$

$$= \left[-\frac{\boldsymbol{a}\tau}{\boldsymbol{h}^\alpha}\boldsymbol{c}_-(-1)^{N-j+1}\begin{pmatrix}\alpha\\ \boldsymbol{N}-\boldsymbol{j}+1\end{pmatrix}\right]\boldsymbol{u}_N^n.$$

$$+ \ldots$$

$$+\left(-\frac{a\tau}{h^\alpha}\left[c_+(-1)^0\begin{pmatrix}\alpha\\ 0\end{pmatrix} + c_-(-1)^2\begin{pmatrix}\alpha\\ 2\end{pmatrix}\right]\right)u_{j+1}^n.$$

$$+\left(1-\frac{\boldsymbol{b}\tau}{\boldsymbol{h}}-\frac{\boldsymbol{a}\tau}{\boldsymbol{h}^\alpha}\left[\boldsymbol{c}_+(-1)\begin{pmatrix}\alpha\\ 0\end{pmatrix} + \boldsymbol{c}_-(-1)\begin{pmatrix}\alpha\\ 1\end{pmatrix}\right]\right)\boldsymbol{u}_j^n.$$

$$+ \ldots$$

$$+\left(-\frac{\boldsymbol{a}\tau}{\boldsymbol{h}^\alpha}\left[\boldsymbol{c}_+(-1)^{j+1}\begin{pmatrix}\alpha\\ \boldsymbol{j}+1\end{pmatrix} + \boldsymbol{c}_-(-1)^{\mathrm{I}-j}\begin{pmatrix}\alpha\\ 1-\boldsymbol{j}\end{pmatrix}\right]\right)\boldsymbol{u}_0^n.$$

利用边界条件 $u_0^n = u_N^n = 0$，可确定一个线性系统，矩阵形式如下：

$$U^{n+1} = AU^n \tag{3.1.12}$$

其中：

$$\boldsymbol{U}^{n+1} = \begin{pmatrix} \boldsymbol{u}_{N-1}^{n+1} & \boldsymbol{u}_{N-1}^{n+1} & \cdots & \boldsymbol{u}_1^{n+1} \end{pmatrix}^T$$

$$\boldsymbol{U}^{n} = \begin{pmatrix} \boldsymbol{u}_{N-1}^{n} & \boldsymbol{u}_{N-1}^{n} & \cdots & \boldsymbol{u}_1^{n} \end{pmatrix}^T$$

而系数矩阵 A=(aij) 为一个 Toeplitz 矩阵,

$$aij = \begin{cases} (-1)^{j-i}\dfrac{a\tau}{h^\alpha}c_-\begin{pmatrix} \alpha \\ j-i+1 \end{pmatrix}, j \geq i+2, i = 1,2,\ldots,N-3 \\ -\dfrac{aT}{h^\alpha}\left(c_+ + c_-\begin{pmatrix} \alpha \\ 2 \end{pmatrix}\right), j = i+1, i = 1,2,\ldots,N-2 \\ 1+\dfrac{a\tau}{h^\alpha}\begin{pmatrix} \alpha \\ 1 \end{pmatrix}(c_+ + c_-) - \dfrac{b\tau}{h}, j = i = 1,2,\cdots,N-1 \\ -\dfrac{a\tau}{h^\alpha}\left(c_+\begin{pmatrix} \alpha \\ 2 \end{pmatrix} + c_-\right) + \dfrac{b\tau}{h}, j = i-1, i = 2,3,\ldots,N-1 \\ (-1)^{i-j}\dfrac{a\tau}{h^\alpha}c_+\begin{pmatrix} \alpha \\ i-j+1 \end{pmatrix}, j \leq i-2, i = 3,4,\ldots,N-1 \end{cases}$$

运用以下算法进行 MATLAB 编程进行数值实验。

表 3.1: 算法初步

Step	Operation	Algorithm
one	网格剖分	$h = \frac{L}{M}, \tau = \frac{T}{N}.$ $x = linspace(0, L, M+1);$ $t = linspace(0, T, N+1);$
two	输入初边值	$u(1:end, 1) = \phi(x);$ $u(1, 1:end) = \psi_1(x);$ $u(end, 1:end) = \psi_2(x);$
three	计算	$A = Toeplitz(W, V)$ $for \quad n = 1:N$ $u(2:end-1, n+1) = A * u(2:end-1, n);$ end

3.2 数值试验

例 1：考虑如下 Lévy-Feller 对流-扩散微分方程初边值问题：

$$\begin{array}{ll} \frac{\partial u(x,t)}{\partial t} = aD_\theta^\alpha u(x,t) - b\frac{\partial u(x,t)}{\partial x}, & 0 < x < \pi, \quad 0 < t < T, 1-\alpha \leq 2, \\ u(x,0) = \varphi(x) = sinx, & 0 \leq x \leq \pi, \\ u(0,t) = u(R,t) = 0, & 0 \leq t \leq T, \end{array} \tag{3.2.1}$$

其中 α=1.7, θ=0.3, a=1.5, b=1.0.

数值分析

根据上述算法，我们利用 MATLAB 编程，得到如下数据，并通过 plot() 函数，直观地表现出 u(x,t) 的近似程度。

theta	-0.2	-0.1	0.1	0.2	0.3
	0.00000	0.00000	0.00000	0.00000	0.00000
	0.07645	0.06866	0.05777	0.05390	0.05070
	0.12416	0.11630	0.10498	0.10104	0.09789
	0.15662	0.15113	0.14383	0.14182	0.14066
	0.17797	0.17592	0.17497	0.17614	0.17827
	0.19041	0.19220	0.19876	0.20383	0.21011
	0.19547	0.20109	0.21550	0.22483	0.23573
alpha = 1.7 时	0.19442	0.20356	0.22556	0.23916	0.25483
	0.18832	0.20049	0.22935	0.24698	0.26728
u(x,t)的值	0.17816	0.19273	0.22737	0.24856	0.27306
	0.16484	0.18111	0.22016	0.24426	0.27235
	0.14918	0.16642	0.20835	0.23455	0.26540
	0.13193	0.14939	0.19259	0.22000	0.25263
	0.11375	0.13074	0.17356	0.20120	0.23453
	0.09525	0.11111	0.15196	0.17882	0.21167
	0.07692	0.09110	0.12844	0.15353	0.18469
	0.05920	0.07123	0.10367	0.12600	0.15420
	0.04245	0.05192	0.07822	0.09681	0.12080
	0.02691	0.03353	0.05254	0.06643	0.08484
	0.01275	0.01626	0.02678	0.03485	0.04598
	0.00000	0.00000	0.00000	0.00000	0.00000

图 3.1: θ 取不同值，$\alpha = 1.7$ 时最后一步的数值解 u(x,t)

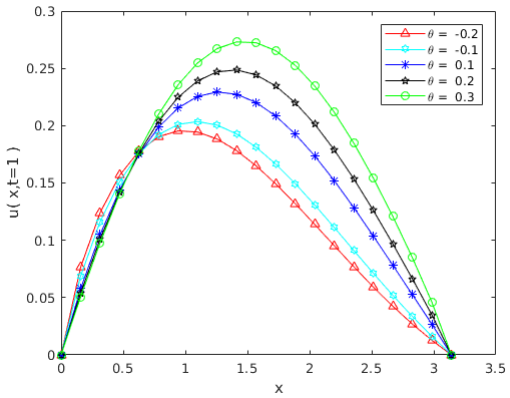

图 3.2: θ 取不同值，$\alpha = 1.7$ 时，最后一步数值解的图像

图 3.1、图 3.2 体现了带有不同倾斜度 θ 的对流-扩散过程的不同。其中 $\alpha = 1.7$,$\theta = \pm 0.2, \pm 0.1, 0.3$,a $= 1.5$,b $= 1.0$,$h = \pi/20$,$\tau = 0.0001$。

alpha	1.1	1.3	1.5	1.7	1.9
	0.00000	0.00000	0.00000	0.00000	0.00000
	0.03584	0.05935	0.05841	0.05070	0.04090
	0.06696	0.10871	0.10902	0.09789	0.08227
	0.09397	0.15042	0.15299	0.14066	0.12156
	0.11700	0.18525	0.19048	0.17827	0.15732
	0.13604	0.21349	0.22139	0.21011	0.18849
	0.15104	0.23526	0.24562	0.23573	0.21429
theta=0.3	0.16195	0.25067	0.26313	0.25483	0.23419
	0.16878	0.25984	0.27394	0.26728	0.24784
u(x,t)	0.17158	0.26296	0.27820	0.27306	0.25507
	0.17047	0.26028	0.27613	0.27235	0.25592
	0.16562	0.25212	0.26808	0.26540	0.25057
	0.15724	0.23888	0.25446	0.25263	0.23934
	0.14562	0.22100	0.23577	0.23453	0.22268
	0.13106	0.19897	0.21255	0.21167	0.20117
	0.11391	0.17330	0.18540	0.18469	0.17543
	0.09453	0.14449	0.15488	0.15420	0.14613
	0.07325	0.11296	0.12147	0.12080	0.11395
	0.05036	0.07890	0.08541	0.08484	0.07941
	0.02602	0.04198	0.04617	0.04598	0.04252
	0.00000	0.00000	0.00000	0.00000	0.00000

图 3.3: α 取不同值，$\theta = 0.3$ 时最后一步的数值解 u(x,t)

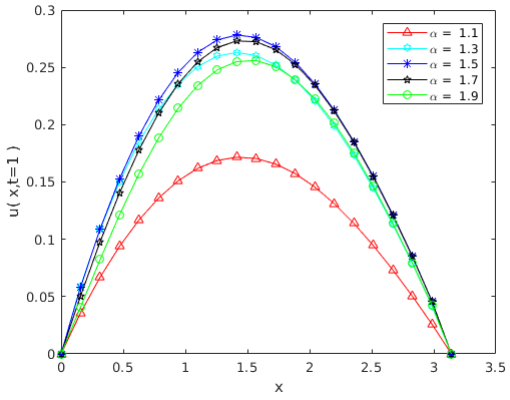

图 3.4: α 取不同值，$\theta = 0.3$ 时，最后一步数值解的图像

图 3.3、图 3.4 给出了不同阶 α 的对流-扩散过程的不同。其中 $\alpha = 1.1, 1.3, 1.5, 1.7, 1.9$，$\theta = 0.3$。结合以上图像可知，$\alpha$ 一定时，函数的振幅随 θ 的增大而增大，不同的是，θ 一定时，α=1.5 时振幅最大，并没有呈现出与变量 α 正相关的规律。

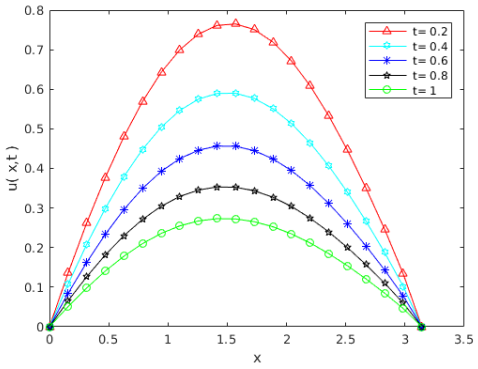

图 3.5: t 取不同值时，最后一步数值解的图像

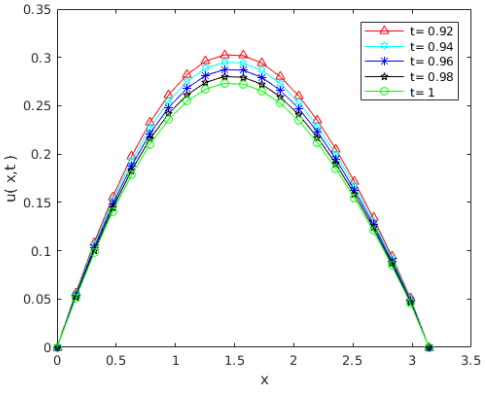

图 3.6: 细化 t 剖分，t 取不同值时，最后一步数值解的图像

图 3.5、图 3.6 描绘的是方程表示的对流-扩散过程。由图 3.5 可知，随着 t 的增大，u(x,t) 的振幅逐渐减小，且趋于平缓，于是我们细化 t 的剖分，令 $t = 0.92\ 0.94\ 0.96\ 0.98\ 1.0$, 结果如图 3.6 所示，u(x,t) 振幅越来越小，且函数图像逼近 $u(x, t = 1.0)$, 所以 u(x,t) 关于 t 是收敛的。

稳定性分析

我们给初值条件 sin(x) 加入一个微小干扰，即令 $o(h) = \pi/20$, 考虑初值条件为 sin(x+o(h)) 时的数值解，得到的图象如下：

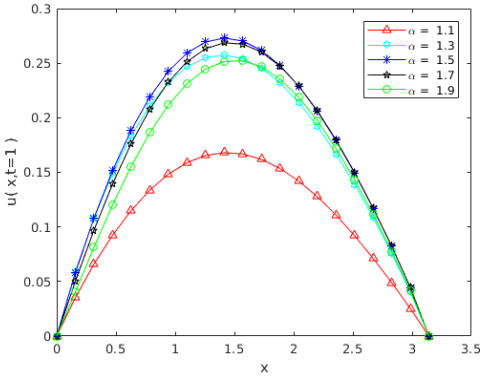

图 3.7: x+o(h) 后,α 取不同值，$\theta = 0.3$ 时，最后一步数值解的图像

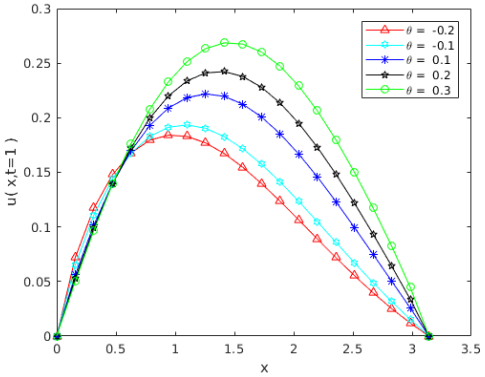

图 3.8: x+o(h) 后,θ 取不同值，$\alpha = 1.7$ 时，最后一步数值解的图像

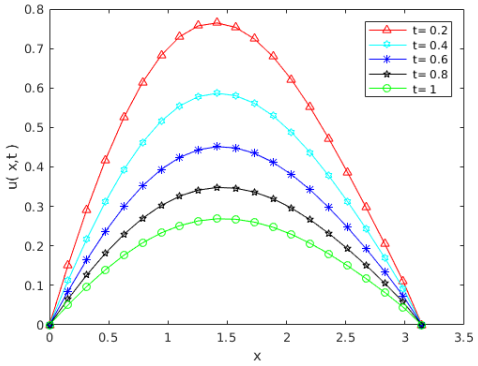

图 3.9: x+o(h) 后,t 取不同值时，最后一步数值解的图像

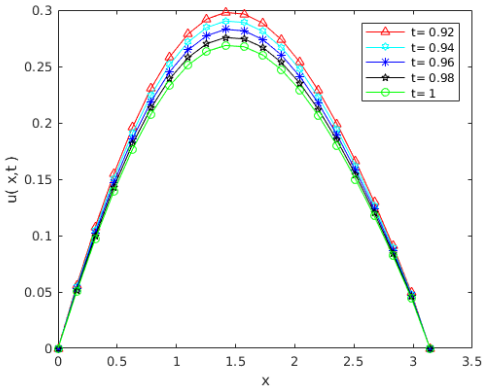

图 3.10: 细化 t 剖分，t 取不同值时，最后一步数值解的图像

通过与初值条件为 sin(x) 时的图像比较，我们直观地看到其结果的变化是微小的，故我们认为该格式是稳定的。

第四章　总结与展望

本文的主要工作是研究的有限区间上的 Lévy-Feller 对流—扩散方程的初边值问题，利用 Grünwald-Letnikov 算子和 Riemann-Liouville 分数阶导数，进过一系列运算与转化，得到离散的有限差分格式，并利用 Matlab 对算法进行数值模拟, 得到了数值解 u(x,t) 与参数 θ,α 和变量 t 的关系。

本文主要讨论的算例，鉴于其解析解的特殊性，我们只深入探讨了其数值解的影响因素。即对于不同的 θ,α,t, 方程的解表现出怎样的形式及特点。

其次，我们数值实验针对的仅仅是特定的 Lévy-Feller 对流-扩散微分方程，而差分方法在分数阶微分方程上的应用需要我们进一步探讨。

求解此类分数阶偏微分方程，除了文中讨论的差分方法外，刘发旺教授曾首次成功应用了另一种方法——行方法，上述方程的行方法格式为:

$$\frac{du_l}{dt} = -\frac{a}{h^\alpha}\left(c_+\sum_{k=0}^{l+1}(-1)^k\begin{pmatrix}\alpha\\k\end{pmatrix}u_{l+1-k} + c_-\sum_{k=0}^{N-l+1}(-1)^k\begin{pmatrix}\alpha\\k\end{pmatrix}u_{l-1+k}\right) - b\frac{u_l - u_{l-1}}{h}$$

故我们还可以比较行方法和差分方法所得到的数值解，以及及两种方法对于不同问题或不同条件的有效性和优劣性。

参考文献

[1] Q. Liu, F. Liu, I. Turner, and V. Anh. Approximation of the lévy–feller advection–dispersion process by random walk and finite difference method. *Journal of Computational Physics*, 222(1):57–70, 2007.

[2] 刘青霞. 空间分数阶对流—扩散方程的数值解及其应用. PhD thesis, 2007.

[3] 张贤达. 矩阵分析与应用 *[M]*. 北京：清华大学出版社, 2004.

[4] 李寿佛. 刚性常微分方程及刚性泛函微分方程数值分析 *[M]*. 湘潭：湘潭大学出版社, 2010.

[5] 李荣华. 偏微分方程数值解法 *[M]*. 北京：科学出版社, 2005.

[6] 林世敏，许传炬. 分数阶微分方程的理论和数值方法研究. 计算数学, 38(1):9–12, 2016.

[7] 林然，刘发旺. 分数阶常微分方程初值问题的高阶近似 [j]. 厦门大学学报, 43(1):25–30, 2004.

[8] 王朵. 双边空间分数阶对流扩散方程的几种数值解法. 2016.

[9] 章红梅，刘发旺. 数值求解 lévy-feller 扩散方程 *. 高等学校计算数学学报, 27:239–241, 2005.

[10] 黄云清，舒适，陈艳萍，金继承，文立平. 数值计算方法 *[M]*. 北京：科学出版社, 2009.

致谢

惊风飘白日兮，光景西驰流，搁管城于案台兮，识大学之将逝。立平逵而回望兮，恨流光之不怜，目苍茫而神定兮，知余生之所求。既凝辞以骋怀兮，陈赋言于其下：

一谢文教授立平兮，诲余之不倦，选题之用心兮，讲解之入微。指归处于迷途兮，授我之以渔，开余研究之思维兮，将所学以致用。得其三阅而指漏兮，乃知文之不足，幸其数审而费神兮，余方能成此文。

二谢余之家人兮，为余之秉甲，若鹓鹭之梧桐兮，若鲲之南溟。三谢挚友与导师兮，助我以编程与算法，不以余之不敏兮，教余而不厌。四谢爱余之人兮，又谢余之所爱，非其之不弃兮，余不能悦于此。

五谢审余文之教授兮，再谢与答辩之老师，劳诸君之心力兮，实感激之无极。

路迢迢之不迩兮，余将求索而不止，青春将不老兮，未来非不可期。

附录：

一. Run_main.m

```
%%  空间分数阶对流-扩散方程数值模拟
%   时间：10-May-2019                                    0
%   版本：8.3.0.532 (R2014a)
%   M：x 方向区间剖分数
%   N：y 方向区间剖分数
%   left_condation：左边值条件
%   right_condation：右边值条件
%   initial_condation：初值条件
%   show.image_theta_together: theta 取不同值时，最后一步数值解的图形
%   show.image_alph_together: alph 取不同值时，最后一步数值解的图形
%   show.image_time_together: 取定 alph 时，最后一步数值解的图形
clear;clc;
Lx = 0; Rx = pi; Lt = 0; Rt = 1;
M = 20; N = 10000; alph = 1.7;
theta = 0.3;


left_condation = @( t ) zeros(1,N+1);
right_condation = @( t ) zeros(1,N+1);
initial_condation = @( x ) sin(x);
show = show_solution( );
SCDE = model_date( Lx,Rx,Lt,Rt,left_condation,right_condation,...
    initial_condation);
```

```
[ x,t,u ] = crank_weighted_method( M,N,SCDE,alph,theta);
show.image_alph_together( M,N,SCDE,[1.1,1.3,1.5,1.7,1.9],theta);
show.image_theta_together( M,N,SCDE,alph,[-0.2,-0.1,0.1,0.2,0.3])
show.image_time_together( x,t,u,[0.2,0.4,0.6,0.8,1.0]);
```

二. Crank_weighted_method

```
function [ x,t,u ] = crank_weighted_method( M,N,SCDE,alph,thet)
%CRANK_WEIGHTED 算 法 摘 要
W=zeros(1,M-1);
V=zeros(1,M-1);
u = zeros(M+1,N+1);

[ x,h ] = SCDE.space_grid( M );
[ t,tau ] = SCDE.time_grid( N );

u(1:M+1,1) = SCDE.initial_value( x );

u(1,1:N+1) = SCDE.left_boundary( t );
u(M+1,1:N+1) = SCDE.right_boundary( t );

c1=sin((alph-thet)*pi/2)/sin(alph*pi);
c2=sin((alph+thet)*pi/2)/sin(alph*pi);
%Grünwald 系数
g=zeros(1,M+1);  g(1) = alph;
for k=2:M
```

```
    g(k) = ( -1 + (alph + 1) / k ) * g( k-1 );
end

    for k = 1:M-1
        W(k) = w(k);
    end

    for k = 1:M-1
        V(k) = v(k);
    end

A = toeplitz(W,V);

    function wk = w(k)
        if k==1
            wk = 1+1.5*tau/(h^alph)*g(k)*(c1+c2)-1.0*tau/h;
        elseif k==2
            wk = -1.5*tau/(h^alph)* (c2*g(k) + c1);
        else
            wk = ((-1)^(k-1))*1.5*tau/(h^alph)*c2*g(k);
        end
    end

    function vl = v(l)
        if l==1
```

```
            v1 = 1+1.5*tau/(h^alph)*g(1)*(c1+c2)-1.0*tau/h;
        elseif l==2
            v1 = -1.5*tau/(h^alph)* (c1*g(1) + c2)+1.0*tau/h;
        else
            v1 = ((-1)^(1-1))*1.5*tau/(h^alph)*c1*g(1);
        end
    end

%%计算 u(n+1)=A*u(n)
    for n = 1:N
        u( 2:M,n+1 ) = A*u(2:M,n);
    end
end
```

三. Model_date

```
function SCDE = model_date( Lx,Rx,Lt,Rt,left_condation,right_condation,...
    initial_condation )
%%  MODEL_DATE : 空间分数阶对流—扩散方程

%结构体
SCDE = struct('space_grid',@space_grid,'time_grid',@time_grid,'left_boundary'...
    ,@left_boundary,'right_boundary',@right_boundary,'initial_value',@initial_value,...
 'Newton_iterator',@Newton_iterator);

 %空间步长
    function    [ x,h ] = space_grid( M )
```

```
        h = ( Rx - Lx ) / M;
        x = linspace( Lx,Rx,M+1 );
    end

%时间步长
    function    [ t,tau ] = time_grid( N )
        tau = ( Rt - Lt ) / N;
        t = linspace( Lt,Rt,N+1 );
    end

  %初值
    function u = initial_value( x )
        u = initial_condation( x );
    end

%边值
    function u = left_boundary( t )
        u = left_condation( t );
    end

    function u = right_boundary( t )
        u = right_condation( t );
    end
end
```

四. show_solution

```
function show = show_solution(  )

%% SHOW_SOLUTION 可视化控制

%% 测量观测阶，可视化数值解图像

show =
struct('image_time_together',@image_time_together,'image_alph_together',@image_alph_to
gether,...

    'image_theta_together',@image_theta_together);


%   show.image_time_together: 取定 alph 时，最后一步数值解的图形

    function image_time_together( x,t,u,time )

        color =  {'-r^';'-ch';'-b*';'-kp';'-go'};

        figure

        for k = 1: length( time )

        plot( x,u( 1:end,t == time(k) ),char(color( k ) ) );

            hold on;

        end

        xlabel('x');ylabel( ['u( x,','t )'] );

%         title(['\alpha',' = ',num2str(alph),' 时的数值解图像']);

        legend( [' t= ',num2str(time(1))],[' t= ',num2str(time(2))],...

                [' t= ',num2str(time(3))], [' t= ',num2str(time(4))],...

                [' t= ',num2str(time(5))]);


    end



%%   show.image_alph_together: alph 取不同值时，最后一步数值解的图形
```

```
    function image_alph_together( M,N,SCDE,alph,theta)
        color =  {'-r^';'-ch';'-b*';'-kp';'-go'};
        figure
        for i = 1:length( alph )
            [ x,~,u ] = crank_weighted_method( M,N,SCDE,alph(i),theta);
            plot( x,u( 1:end,end ),char(color( i ) ) );
            hold on;
            xlabel('x');ylabel( ['u( x,','t=1 )'] );
        end
        legend( ['\alpha',' =  ',num2str(alph(1))],['\alpha',' =
',num2str(alph(2))],...
                ['\alpha',' =  ',num2str(alph(3))],['\alpha',' =
',num2str(alph(4))],...
                ['\alpha',' =  ',num2str(alph(5))]);

    end
%%   show.image_theta_together: theta 取不同值时，最后一步数值解的图形
    function image_theta_together( M,N,SCDE,alph,theta)
        color =  {'-r^';'-ch';'-b*';'-kp';'-go'};
        figure
        for i = 1:length( theta )
            [ x,~,u ] = crank_weighted_method( M,N,SCDE,alph,theta(i));
            plot( x,u( 1:end,end ),char(color( i ) ) );
            hold on;
            xlabel('x');ylabel( ['u( x,','t=1 )'] );
        end
```

```
        legend( ['\theta',' =  ',num2str(theta(1))],['\theta',' =
',num2str(theta(2))],...

                ['\theta',' =  ',num2str(theta(3))],['\theta',' =
',num2str(theta(4))],...

                ['\theta',' =  ',num2str(theta(5))]);


    end

end
```