



الجمهورية الجزائرية الديمقراطية الشعبية
People's Democratic Republic of Algeria
وزارة التعليم العالي والبحث العلمي
Ministry of Higher Education and Scientific Research
جامعة الشهيد عباس لغرور خنشلة
Abbes Laghrou- Khenchela University



Faculty of Sciences and Technology

Department of Mathematics

Serial number:.....

كلية العلوم والتكنولوجيا
قسم الرياضيات

Final Master's Thesis

Submitted in the Fulfilment of Requirements for the Master's Degree in Mathematics

Major :Mathematics

Specialization :Applied Mathematics

Titled by:

NUMERICAL CALCULATION OF A FRACTIONAL PROBLEM WITH INTEGRAL CONDITION

Presented by: Ghelloudj Abdelkader

Defended on: .. /05/2026

In the presence of the Examination Committee:

Fouzi Tebessi	MCB	Abbes Laghrou - Khenchela university	Chair
Brahimi Saadoune	MCB	Abbes Laghrou - Khenchela university	Supervisor
Mourad Boussaada	MAA	Abbes Laghrou - Khenchela university	Examiner

Academic year 2025/2026



Dedication

I dedicate this work, with love and gratitude, to my dear parents, whose prayers have been a light on my path in life.

To my brothers and sisters.

To all my family and relatives.

To all my friends, who stood by me and supported me in the most difficult moments.

Acknowledgements

*P*raise be to Allah, the Lord of the worlds, for His guidance and assistance.

I extend my sincere thanks and gratitude to my supervisor, **Dr. Brahimi Saadoune**, whose advice and guidance were of great help to me in accomplishing this work.

I also extend my thanks to the respected members of the examination committee for accepting to evaluate this work.

I cannot forget to thank all the professors who enriched us with their knowledge and experience, and who deserve the credit for our training and preparation for this stage.

Ghelloudj Abdelkader

Abstract

الملخص

في هذه المذكرة، قمنا بدراسة معادلة تفاضلية جزئية كسرية زمانية-مكانية من الرتبة α ($0 < \alpha < 1$) في الزمن (بكاوتو) والرتبة β ($1 < \beta < 2$) في المكان (بريمان-ليوفيل)، مع شروط حدية : شرط نيومان وشرط تكاملي. لحل هذه المسألة عددياً، اقترحنا مخططاً ضمناً للفروق المنتهية يجمع بين طريقة L1 لتقريب المشتقة الزمنية، وطريقة جرونوالد-ليتنيكوف للمُزاحة لتقريب المشتقة المكانية، وقاعدة شبه المنحرف لمعالجة الشرط التكاملي. قمنا بتحليل المخطط عددياً وأثبتنا:

• الاتساق: خطأ الاقتران المحلي من الرتبة $O(h_t + h)$.

• الاستقرار: استقرار مشروط بشرط:

$$h_t \leq C \cdot h^{\beta/\alpha}, \quad C = \left(\frac{1}{2\beta A \Gamma(2 - \alpha)} \right)^{1/\alpha}, \quad A = \max |a(x, t)|$$

• التقارب: تقارب المخطط بدقة من الرتبة الأولى في الزمن ($O(h_t)$) والرتبة الأولى في المكان ($O(h)$).

تم إجراء تجارب عددية باستخدام **MATLAB** تؤكد النتائج النظرية.

الكلمات المفتاحية: معادلات تفاضلية جزئية كسرية، طريقة L1، طريقة جرونوالد-ليتنيكوف المُزاحة، استقرار شرطي، تقارب من الرتبة الأولى.

Abstract

In this thesis, we studied a time-space fractional Partial differential equation of order α ($0 < \alpha < 1$) in time (Caputo) and order β ($1 < \beta < 2$) in space (Riemann-Liouville), with Neumann and integral boundary conditions. For the numerical solution, we proposed an implicit finite difference scheme combining the L1 method for the time derivative, the shifted Grünwald-Letnikov method for the space derivative, and the trapezoidal rule for the integral condition.

We analyzed the scheme and proved:

- **Consistency:** The local truncation error is of order $O(h_t + h)$.
- **Stability:** Conditional stability under the condition:

$$h_t \leq C \cdot h^{\beta/\alpha}, \quad C = \left(\frac{1}{2\beta A \Gamma(2 - \alpha)} \right)^{1/\alpha}, \quad A = \max |a(x, t)|$$

- **Convergence:** The scheme converges with first-order accuracy in time ($O(h_t)$) and first-order accuracy in space ($O(h)$).

Numerical experiments using **MATLAB** confirm the theoretical results.

Keywords: Fractional Partial differential equations, L1 method, shifted Grünwald-Letnikov method, conditional stability, first-order convergence.

Résumé

Dans ce mémoire, nous avons étudié une Équation aux dérivées partielles fractionnaire spatio-temporelle d'ordre α ($0 < \alpha < 1$) en temps (Caputo) et d'ordre β ($1 < \beta < 2$) en espace (Riemann-Liouville), avec des conditions aux limites de type Neumann et intégrale.

Pour la résolution numérique, nous avons proposé un schéma implicite aux différences finies combinant la méthode L1 pour la dérivée temporelle, la méthode de Grünwald-Letnikov décalée pour la dérivée spatiale, et la règle du trapèze pour la condition intégrale.

Nous avons analysé le schéma et prouvé :

- **Consistance :** L'erreur de troncature locale est d'ordre $O(h_t + h)$.
- **Stabilité :** Stabilité conditionnelle sous la condition :

$$h_t \leq C \cdot h^{\beta/\alpha}, \quad C = \left(\frac{1}{2\beta A \Gamma(2 - \alpha)} \right)^{1/\alpha}, \quad A = \max |a(x, t)|$$

- **Convergence :** Le schéma converge avec une précision d'ordre un en temps ($O(h_t)$) et d'ordre un en espace ($O(h)$).

Des expériences numériques avec **MATLAB** confirment les résultats théoriques.

Mots-clés : Équation aux dérivées partielles fractionnaire, méthode L1, méthode de Grünwald-Letnikov décalée, stabilité conditionnelle, convergence d'ordre un.

Contents

Preliminaries	11
0.1 The Gamma Function and Its Properties	11
0.1.1 The Gamma Function	11
0.2 The L^p Spaces	13
0.2.1 The Space $L^1(\mathbb{R})$	13
0.2.2 The Space $L^2(\mathbb{R})$	13
0.2.3 The Infinity Norm on a Bounded Domain	13
0.3 Fourier Transform	14
0.3.1 Basic Properties	14
0.3.2 Inverse Fourier Transform	14
0.4 Fractional Integration	15
0.4.1 Fractional Integral on an Interval $[a, b]$	15
0.4.2 Fractional Integral in the Riemann-Liouville Sense	17
0.5 Fractional Derivatives	19
0.5.1 Fractional Derivative in the Riemann-Liouville Sense	19
0.5.2 Grünwald-Letnikov fractional derivatives	21
0.5.3 Relationship Between RL and GL Definitions	22
0.5.4 Properties of the Grünwald-Letnikov Coefficients	25
0.5.5 Fractional Derivative in the Sense of Caputo	27
The Finite Difference method	28
1.1 Finite Difference Method in the Ordinary Case	28
1.1.1 Numerical Convergence	29
1.2 Finite Difference Method in the Fractional Case	31
1.2.1 The L1 Scheme	31
Numerical analysis of an EDPF problem with integral boundary conditions	34
2.1 Position of the problem	34
2.2 Discretization and Approximate Formulation	35
2.3 Numerical Scheme	37
2.3.1 Case $k = 0$ (First Time Step)	37
2.3.2 General Case ($k \geq 1$)	40

2.4	Solution of a Matrix System	45
2.5	Consistency Analysis	48
2.6	Stability Analysis	54
2.7	convergence Analysis	60
Applications		65
3.1	Example 1	65
3.2	Example 2	67
3.3	Application en 3D	69
3.3.1	EXAMPLE 1:	69
3.3.2	EXAMPLE 2:	70
3.4	Programme MATLAB	71
Conclusion		79
References		80

List of Symbols

$\mathbb{N}$:	set of natural numbers
$\mathbb{R}$:	set of real numbers
$\mathbb{C}$:	set of complex numbers
L^1 :	Lebesgue space of absolutely integrable functions
L^2 :	Lebesgue space of square-integrable functions
L^∞ :	Lebesgue space of essentially bounded functions
$\Gamma(\cdot)$:	Gamma function
α :	order of the Caputo time derivative ($0 < \alpha < 1$)
β :	order of the Riemann-Liouville space derivative ($1 < \beta < 2$)
${}^c D_t^\alpha u$:	Caputo fractional derivative of u with respect to t
${}^{RL} D_x^\beta u$:	Riemann-Liouville fractional derivative of u with respect to x
$\mathcal{D}_h^\beta u$:	shifted Grünwald-Letnikov approximation of the space derivative
$\mathcal{F}$:	Fourier transform
$\mathcal{F}^{-1}$:	inverse Fourier transform
$a(x, t)$:	diffusion coefficient
$\phi(x, t)$:	source term
$f(x)$:	initial condition: $u(x, 0) = f(x)$
$g_0(t)$:	Neumann boundary condition at $x = 0$
$g_1(t)$:	integral condition: $\int_0^1 u(x, t) dx = g_1(t)$
v_i^k :	numerical solution at the grid point (x_i, t_k)
e_i^k :	error: $e_i^k = u_i^k - v_i^k$
$\ E^k\ _\infty$:	infinity norm of the error vector
h :	space step size
h_t :	time step size
P^k :	system matrix at time level k

Introduction

Fractional differential equations are a powerful and effective mathematical tool for modelling many physical and technical phenomena characterised by memory and non-locality properties [5]. Unlike traditional integer-order derivatives, fractional derivatives allow for a more accurate description of processes that depend on the temporal history of the system or that exhibit anomalous diffusion behaviour [16]. These equations have proven their effectiveness in various fields such as mechanics, heat transfer in porous media, geophysical dynamics and signal processing [7].

Among the most commonly used formulations of fractional derivatives are the Caputo derivative and the Riemann-Liouville derivative [9]. The Caputo derivative is distinguished by its ability to handle traditional initial conditions with clear physical meaning, making it particularly suitable for time derivatives [5, 17]. In contrast, the Riemann-Liouville derivative is more appropriate for describing spatial non-locality, especially when combined with fractional boundary conditions representing non-local fluxes [16, 9]. This combination of the two derivatives forms a comprehensive mathematical framework for studying complex phenomena such as anomalous diffusion in memory materials [7]. The study of these equations is particularly important when they are coupled with non-traditional boundary conditions, such as fractional Neumann conditions and integral conditions [3]. Integral conditions represent a generalisation of traditional boundary conditions, linking the values of the solution across the entire domain rather than just at the boundary points. This type of condition occurs naturally in conservation laws and in models requiring knowledge of the overall behaviour of the system [17, 3].

Due to the complex nature of these problems and the lack of exact analytical solutions in most cases, numerical analysis becomes an essential tool for studying the behaviour of solutions and their approximation [9, 13]. Numerical methods for fractional derivatives rely on specialised approximation techniques, such as the L1 method for approximating the Caputo derivative [5] and the Grünwald-Letnikov formula for approximating the Riemann-Liouville derivative [16, 9]. The finite difference method is among the most widely used approaches in this field. It is based on discretising the domain into a grid of points and replacing derivatives with finite difference ratios [13, 8]. This method has proven its effectiveness in solving fractional differential equations thanks to its simplicity and ease of implementation, particularly when dealing with complex boundary conditions [8, 19]. This work focuses on the study of a fractional heat equation combining a Caputo time derivative of order $0 < \alpha < 1$ and a Riemann-Liouville spatial derivative of order $1 < \beta < 2$, with a variable diffusion coefficient $a(x, t)$ and an external source term $\phi(x, t)$. The equation is accompanied by traditional initial conditions, a fractional Neumann condition at the left boundary, and an integral condition over the entire spatial domain. This problem represents an important generalisation of the classical heat equation, allowing the simultaneous description of sub-diffusion phenomena and non-local effects [5, 16, 17].

The well-posedness of fractional differential equations with integral boundary conditions has been established in the literature. For the homogeneous case ($g_1(t) = 0$), [15] proved existence and uniqueness for parabolic fractional equations with integral conditions using energy inequality methods. Their work provides a theoretical foundation for problems with integral constraints. More recently, [2] extended this analysis to time-fractional advection-diffusion equations with purely integral conditions of the form $\int_0^1 v(x, t) dx$ and $\int_0^1 x^n v(x, t) dx$, establishing well-posedness for the case $1 < \alpha < 2$ using a priori estimates. Their work also offers a numerical treatment of integral boundary conditions via a combination of finite differences and numerical integration, demonstrating conditional stability and first-order convergence.

Although these studies focus on different ranges of the fractional order α , the methodologies for handling integral conditions and proving stability are directly applicable to our problem. These theoretical results provide a solid foundation for the numerical analysis presented in this work. This study differs from previous studies in several key aspects. First, unlike previous works that focused on the case $1 < \alpha < 2$ using the L2

scheme [2], we consider the case $0 < \alpha < 1$, which requires the L1 scheme for the Caputo derivative. Second, we investigate a fully fractional model coupling the Caputo time derivative with the Riemann-Liouville space derivative, moving beyond traditional hybrid models that combine fractional time derivatives with integer-order space derivatives. Third, we successfully combine three different conditions: an initial condition, a Neumann condition at the left boundary, and a non-local integral condition over the entire spatial domain. To the best of our knowledge and based on the literature review, this specific mathematical coupling and boundary configuration have not been studied before in this form. This thesis aims to construct and develop an efficient numerical scheme for solving this fractional problem using the finite difference method, with a detailed analysis of stability and convergence. The methodology adopted is based on approximating the time derivative using the L1 method and approximating the spatial derivative using the shifted Grünwald-Letnikov formula, with special treatment of boundary conditions and integrals [9, 3, 13]. A matrix system will be constructed that can be solved numerically at each time step, with proof that the resulting matrix is invertible under appropriate conditions on the discretisation steps [5, 9]. The consistency of the proposed scheme is proved using Taylor expansion, its conditional stability is established via norm estimates using properties of Grünwald-Letnikov coefficients, and its convergence is demonstrated using the Lax equivalence theorem. The scheme achieves first-order accuracy in both time and space, supported by numerical experiments documenting the efficiency and accuracy of the method [13, 8].

Preliminaries

In this chapter, we present the appropriate fundamental concepts that are used in the other chapters. [16]

0.1 The Gamma Function and Its Properties

0.1.1 The Gamma Function

One of the basic functions of fractional calculus is Euler's Gamma function.

Definition:

For $z \in \mathbb{C}$ such that $\text{Re}(z) > 0$. The Gamma function is defined by the following integral:

$$\Gamma(z) = \int_0^{\infty} e^{-t} t^{z-1} dt,$$

Properties

1/ An important property of the function $\Gamma(z)$ is the following recurrence relation for $\text{Re}(z) > 0$:

$$\Gamma(z+1) = z\Gamma(z)$$

This can be demonstrated through integration by parts :

$$\Gamma(z+1) = \int_0^{+\infty} e^{-t} t^z dt,$$

We take : $u = t^z$ and $v' = e^{-t}$ then $u' = z t^{z-1}$ and $v = -e^{-t}$ we find:

$$\begin{aligned} \Gamma(z+1) &= [-e^{-t} t^z]_0^{+\infty} + z \int_0^{+\infty} e^{-t} t^{z-1} dt \\ &= z \int_0^{+\infty} e^{-t} t^{z-1} dt \end{aligned}$$

therefore:

$$\Gamma(z+1) = z\Gamma(z)$$

2/ The factorial is generalized by Euler's Gamma function because :

$$\Gamma(n+1) = n!$$

We prove this by induction : $\forall n \in \mathbb{N}$, for $n = 0$ we obtain:

$$\Gamma(1) = 1$$

Suppose that the formula for n : $\Gamma(n+1) = n!$ holds, and consider $n+1$:

$$\begin{aligned}\Gamma(n+2) &= (n+1)\Gamma(n+1) \\ &= (n+1)n! \\ &= (n+1)!\end{aligned}$$

therefore :

$$\Gamma(n+2) = (n+1)!$$

for any $n \in \mathbb{N}$, The formula $\Gamma(n+1) = n!$ is true .

3/ We have the following formula:

$$\Gamma(1/2) = \int_0^{+\infty} e^{-t} t^{-1/2} dt,$$

By the substitution $t = \tau^2$, we get $\Gamma(1/2) = 2 \int_0^\infty e^{-\tau^2} d\tau$. The Gaussian integral evaluates to $\sqrt{\pi}/2$, hence $\Gamma(1/2) = \sqrt{\pi}$.

4/ We have the formula:

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

By induction, we will demonstrate that it is true $\forall n \in \mathbb{N}$.

for $n = 0$:

$$\Gamma\left(\frac{1}{2}\right) = \frac{(0)!\sqrt{\pi}}{4^0(0)!} = \sqrt{\pi}$$

Suppose the formula holds for n then:

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

and consider $n+1$, then:

$$\begin{aligned}\Gamma\left(n+1 + \frac{1}{2}\right) &= \left(n + \frac{1}{2}\right) \Gamma\left(n + \frac{1}{2}\right) \\ &= \frac{2n+1}{2} \cdot \frac{(2n)!\sqrt{\pi}}{4^n n!} \\ &= \frac{2n+1}{2 \cdot 2(n+1)} \cdot \frac{2(n+1)(2n)!\sqrt{\pi}}{4^n n!} \\ &= \frac{(2n+2)(2n+1)(2n)!\sqrt{\pi}}{2 \cdot 2(n+1) \cdot 4^n n!} \\ &= \frac{(2n+2)!\sqrt{\pi}}{4^{n+1}(n+1)!} \\ &= \frac{2(n+1)!\sqrt{\pi}}{4^{n+1}(n+1)!}\end{aligned}$$

Therefore: $(n+1/2)$ is verified for all $n \in \mathbb{N}$

0.2 The L^p Spaces

Let X be an arbitrary measure space with a positive measure μ .

General Definition : [18]

If $0 < p < \infty$ and if f is a complex measurable function on X , define

$$\|f\|_p = \left\{ \int_X |f|^p d\mu \right\}^{1/p}$$

and let $L^p(\mu)$ consist of all f for which $\|f\|_p < \infty$.

0.2.1 The Space $L^1(\mathbb{R})$

For $p = 1$ and $X = \mathbb{R}$ with the Lebesgue measure, $L^1(\mathbb{R})$ consists of measurable functions f such that

$$\|f\|_1 = \int_{\mathbb{R}} |f(x)| dx < \infty.$$

0.2.2 The Space $L^2(\mathbb{R})$

For $p = 2$ and $X = \mathbb{R}$ with the Lebesgue measure, $L^2(\mathbb{R})$ consists of measurable functions f such that

$$\|f\|_2 = \left(\int_{\mathbb{R}} |f(x)|^2 dx \right)^{1/2} < \infty.$$

0.2.3 The Infinity Norm on a Bounded Domain

For a continuous function f defined on the bounded interval $[0, 1]$, the infinity norm is defined as

$$\|f\|_{\infty} = \max_{x \in [0, 1]} |f(x)|.$$

In the discrete setting (on the grid points $\{x_1, x_2, \dots, x_m\}$), this becomes

$$\|f\|_{\infty} = \max_{1 \leq i \leq m} |f(x_i)|.$$

This norm satisfies the following properties:

- **Non-negativity:** $\|f\|_{\infty} \geq 0$, and $\|f\|_{\infty} = 0$ if and only if $f = 0$ on the domain (or on all grid points).
- **Homogeneity:** $\|\alpha f\|_{\infty} = |\alpha| \|f\|_{\infty}$ for any scalar α .
- **Triangle inequality:** $\|f + g\|_{\infty} \leq \|f\|_{\infty} + \|g\|_{\infty}$.

In the stability analysis, this norm provides a worst-case estimate of the error over the spatial domain.

0.3 Fourier Transform

Fourier transform methods are standard in the theoretical analysis of numerical schemes for fractional differential equations, particularly for proving consistency and establishing error estimates [11]. In this section, we recall the definition and main properties of the Fourier transform used in the consistency proof of the shifted Grünwald-Letnikov approximation, following the exposition in [18].

Definition:

For a function $f \in L^1(\mathbb{R})$, the Fourier transform $\mathcal{F}\{f\}$ is defined by

$$\hat{f}(\xi) = \mathcal{F}\{f(x)\}(\xi) = \int_{-\infty}^{\infty} f(x)e^{-i\xi x} dx, \quad \xi \in \mathbb{R}.$$

For functions in $L^2(\mathbb{R})$, the Fourier transform is defined by extension (Plancherel's theorem) .

0.3.1 Basic Properties

The following properties are used in our analysis. For a detailed exposition, we refer to [18].

- **Linearity:** For any $f, g \in L^1(\mathbb{R})$ and $\lambda, \mu \in \mathbb{C}$,

$$\mathcal{F}\{\lambda f + \mu g\} = \lambda \hat{f} + \mu \hat{g}.$$

- **Shift property:** For any $a \in \mathbb{R}$,

$$\mathcal{F}\{f(x - a)\}(\xi) = e^{-i\xi a} \hat{f}(\xi).$$

- **Differentiation:** If f and its derivatives decay sufficiently fast at infinity,

$$\mathcal{F}\{f^{(n)}(x)\}(\xi) = (i\xi)^n \hat{f}(\xi).$$

0.3.2 Inverse Fourier Transform

Definition:

The inverse Fourier transform is defined by

$$\mathcal{F}^{-1}\{\hat{f}\}(x) = \frac{1}{2\pi} \int_{-\infty}^{\infty} \hat{f}(\xi)e^{i\xi x} d\xi,$$

and satisfies $\mathcal{F}^{-1}\{\mathcal{F}\{f\}\} = f$ for sufficiently regular f .

Plancherel's Theorem :

For any $f \in L^2(\mathbb{R})$, the Fourier transform $\hat{f}$ belongs to $L^2(\mathbb{R})$ and the mapping $f \mapsto \hat{f}$ is an isometry up to a constant factor:

$$\|\hat{f}\|_{L^2} = \sqrt{2\pi} \|f\|_{L^2}.$$

Equivalently, the inverse Fourier transform is also an isometry:

$$\|f\|_{L^2} = \frac{1}{\sqrt{2\pi}} \|\hat{f}\|_{L^2}.$$

This theorem guarantees that the Fourier transform is a bounded linear operator on $L^2(\mathbb{R})$, and consequently, the inverse Fourier transform is also bounded and linear. For a detailed proof, see [18].

0.4 Fractional Integration

Fractional integration generalizes the classical integral operator to arbitrary orders, enabling the calculus of non-integer dimensions. This mathematical framework is particularly powerful for characterizing complex processes where the system's behavior is influenced by its past states, effectively capturing long-range correlations and memory dynamics.[14]

0.4.1 Fractional Integral on an Interval $[a, b]$

Definition:

Let $f : [a, b] \rightarrow \mathbb{R}$ be a continuous function on the interval $[a, b]$, we consider the integral:

$$(I_a^1 f)(t) = \int_a^t f(\tau) d\tau, \quad \forall t \in [a, b]$$

The second primitive of f is defined as :

$$(I_a^2 f)(t) = \int_a^t I_a^1 f(u) du = \int_a^t \left(\int_a^u f(\tau) d\tau \right) du$$

Fubini's Theorem [18]

Let f be a continuous function on $[a, b] \times [c, d]$ with values in $\mathbb{C}$, then:

$$\int_a^b \left(\int_c^d f(x, y) dy \right) dx = \int_c^d \left(\int_a^b f(x, y) dx \right) dy,$$

For : $a \leq \tau \leq u \leq t$.

According to Fubini's theorem :

$$(I_a^2 f)(t) = \int_a^t \left(\int_\tau^t du \right) f(\tau) d\tau = \int_a^t (t - \tau) f(\tau) d\tau$$

Second method we have:

$$I_a^2 f(t) = \int_a^t \left(\int_a^u f(\tau) d\tau \right) du$$

we set:

$$\int_a^u f(\tau) d\tau = g(u)$$

then:

$$(I_a^2 f)(t) = \int_a^t g(u) du$$

we integrate by parts

we set : $w = g(u)$ and $v' = 1$, then $w' = g'(u)$ and $v = u$, we find:

$$\begin{aligned}\int_a^t g(u)du &= [ug(u)]_a^t - \int_a^t ug'(u)du \\ &= tg(t) - ag(a) - \int_a^t ug'(u)du\end{aligned}$$

we have:

$$g(a) = \int_a^a f(\tau)d\tau = 0 \quad \text{and} \quad g(t) = \int_a^t f(\tau)d\tau \quad \text{and} \quad g(u) = \int_a^u f(\tau)d\tau$$

then we find:

$$\int_a^t g(u)du = \int_a^t tf(\tau)d\tau - \int_a^t uf(u)du = \int_a^t tf(\tau)d\tau - \int_a^t \tau f(\tau)d\tau = \int_a^t (t - \tau)f(\tau)d\tau$$

thus:

$$I_a^2 f(t) = \int_a^t (t - \tau)f(\tau)d\tau$$

The third primitive of f is defined as :

$$\begin{aligned}(I_a^3 f)(t) &= \int_a^t I_a^2 f(u)du \\ &= \int_a^t \left(\int_a^u I_a^1 f(v)dv \right) du \\ &= \int_a^t \left(\int_a^u \left(\int_a^v f(\tau)d\tau \right) dv \right) du\end{aligned}$$

by Fubini's theorem:

For : $a \leq \tau \leq v \leq u \leq t$.

$$\begin{aligned}(I_a^3 f)(t) &= \int_a^t \left(\int_\tau^t \int_v^t dudv \right) f(\tau)d\tau \\ &= \int_a^t \left(\int_\tau^t (t - v)dv \right) f(\tau)d\tau \\ &= \int_a^t \left[tv - \frac{v^2}{2} \right]_\tau^t f(\tau)d\tau \\ &= \int_a^t \left(t^2 - \frac{t^2}{2} - (t\tau - \frac{\tau^2}{2}) \right) f(\tau)d\tau \\ &= \frac{1}{2} \int_a^t (t^2 - 2t\tau + \tau^2) f(\tau)d\tau \\ &= \frac{1}{2} \int_a^t (t - \tau)^2 f(\tau)d\tau\end{aligned}$$

by Induction

Base Case: $n = 0$

$$(I_a^1 f)(t) = \int_a^t f(\tau) d\tau = \frac{1}{0!} \int_a^t (t - \tau)^0 f(\tau) d\tau$$

Assume the formula holds for some $n \in \mathbb{N}$:

$$(I_a^n f)(u) = \frac{1}{(n-1)!} \int_a^u (u - \tau)^{n-1} f(\tau) d\tau$$

We prove it for $n + 1$:

$$\begin{aligned} (I_a^{n+1} f)(t) &= \int_a^t (I_a^n f)(u) du \\ &= \int_a^t \left[\frac{1}{(n-1)!} \int_a^u (u - \tau)^{n-1} f(\tau) d\tau \right] du \\ &\stackrel{\text{Fubini}}{=} \frac{1}{(n-1)!} \int_{\tau=a}^t \left(\int_{u=\tau}^t (u - \tau)^{n-1} du \right) f(\tau) d\tau \\ &= \frac{1}{(n-1)!} \int_{\tau=a}^t \left[\frac{(u - \tau)^n}{n} \right]_{u=\tau}^t f(\tau) d\tau \\ &= \frac{1}{(n-1)!} \int_a^t \frac{(t - \tau)^n}{n} f(\tau) d\tau \\ &= \frac{1}{n \cdot (n-1)!} \int_a^t (t - \tau)^n f(\tau) d\tau \\ &= \frac{1}{n!} \int_a^t (t - \tau)^n f(\tau) d\tau \end{aligned}$$

Therefore, the formula holds for $n + 1$. By induction, it holds for all $n \in \mathbb{N}$.

0.4.2 Fractional Integral in the Riemann-Liouville Sense

The Riemann-Liouville integral is a powerful tool for generalizing the concept of integration to non-integer orders, with many properties and important applications in various mathematical fields.

Definition:

Suppose that $\alpha > 0$, $a < t$, with $\alpha, a, t \in \mathbb{R}$ and $f \in L^1([a, b])$. The Riemann-Liouville fractional integral of the function $f(t)$ for a non-integer order α is defined by:

Left-sided: $\forall t \in [a, b]$

$$I_{a+}^\alpha f(t) = \frac{1}{\Gamma(\alpha)} \int_a^t (t - \tau)^{\alpha-1} f(\tau) d\tau,$$

Right-sided: $\forall t \in [a, b]$

$$I_{b-}^\alpha f(t) = \frac{1}{\Gamma(\alpha)} \int_t^b (\tau - t)^{\alpha-1} f(\tau) d\tau,$$

Properties:

1/

$$I_{a+}^{\alpha}[I_{a+}^{\beta}f(t)] = I_{a+}^{\alpha+\beta}f(t), \quad \alpha, \beta > 0$$

Example:

Riemann-Liouville fractional integral of a constant:

$$I_{a+}^{\alpha}C = \frac{C}{\Gamma(\alpha+1)}(t-a)^{\alpha},$$

By definition we have:

$$\begin{aligned} I_{a+}^{\alpha}C &= \frac{1}{\Gamma(\alpha)} \int_a^t (t-\tau)^{\alpha-1} C \, d\tau \\ &= \frac{C}{\Gamma(\alpha)} \int_a^t (t-\tau)^{\alpha-1} d\tau \\ &= \frac{C}{\Gamma(\alpha)} \left[-\frac{(t-\tau)^{\alpha}}{\alpha} \right]_a^t \\ &= \frac{C}{\Gamma(\alpha)} \left(0 + \frac{(t-a)^{\alpha}}{\alpha} \right) \\ &= \frac{C}{\alpha\Gamma(\alpha)} (t-a)^{\alpha} \\ &= \frac{C}{\Gamma(\alpha+1)} (t-a)^{\alpha} \end{aligned}$$

hence:

$$I_{a+}^{\alpha}C = \frac{C}{\Gamma(\alpha+1)}(t-a)^{\alpha}$$

0.5 Fractional Derivatives

The fractional derivative is a generalisation of classical differential calculus to non-integer orders, which allows systems with ‘memory effects’ to be modelled. We will discuss three fundamental formulas: the Grunwald-Litnikov derivative, the Riemann-Liouville derivative, and the Caputo derivative. These definitions offer unique advantages for theoretical analysis and numerical approximation in the study of complex dynamical systems [16], [6].

0.5.1 Fractional Derivative in the Riemann-Liouville Sense

Definition:

Let $f \in L^1([a, b])$ be an integrable function on $[a, b]$, we have the relation between classical derivative and integral:

$$\frac{df}{dt} = \frac{d^2}{dt^2}(I_{a+}^1 f)$$

A fractional derivative of order α , $0 < \alpha < 1$, can be defined by:

$$\frac{d^\alpha f}{dt^\alpha} = \frac{d}{dt}(I_{a+}^{1-\alpha} f)$$

More generally, if $\alpha > 0$ and $n = \lfloor \alpha \rfloor + 1$, we can set:

$$\frac{d^\alpha f}{dt^\alpha} = \frac{d^n}{dt^n}(I_{a+}^{n-\alpha} f)$$

We obtain exactly the left-sided Riemann-Liouville derivative of the function f , of order $\alpha > 0$ denoted $D_{a+}^\alpha f$ is defined by:

$$\forall t \in [a, b]$$

$$\begin{aligned} D_{a+}^\alpha f(t) &= D^n I_{a+}^{n-\alpha} f(t) \\ &= \frac{1}{\Gamma(n-\alpha)} \left(\frac{d^n}{dt^n} \int_a^t (t-\tau)^{n-\alpha-1} f(\tau) d\tau \right) \end{aligned}$$

Propositions:

For $n-1 < \alpha < n$, $m-1 < \beta < m$.

1/ Let $\alpha > 0$, then for $f \in L^1([a, b])$:

$$D_{a+}^\alpha (I_{a+}^\alpha f(t)) = f(t), \quad t \in [a, b]$$

Proof:

$$\begin{aligned} D_{a+}^\alpha (I_{a+}^\alpha f(t)) &= D^n I_{a+}^{n-\alpha} (I_{a+}^\alpha f(t)) \\ &= D^n (I_{a+}^{n-\alpha+\alpha} f(t)) \\ &= D^n I_{a+}^n f(t) \\ &= f(t) \end{aligned}$$

2/ Let $\alpha > \beta > 0$, then for $f \in L^1([a, b])$:

$$D_{a+}^\beta (I_{a+}^\alpha f(t)) = I_{a+}^{\alpha-\beta} f(t), \quad t \in [a, b]$$

Proof:

$$\begin{aligned}
 D_{a+}^{\beta}(I_{a+}^{\alpha}f(t)) &= D_{a+}^m I_{a+}^{m-\beta}(I_{a+}^{\alpha}f(t)) \\
 &= D_{a+}^m I_{a+}^{m-\beta+\alpha}f(t) \\
 &= D_{a+}^m I_{a+}^m(I_{a+}^{\alpha-\beta}f(t)) \\
 &= I_{a+}^{\alpha-\beta}f(t)
 \end{aligned}$$

Example:

$$f(t) = t, \quad 0 < t < 1$$

We calculate the derivative of f in the R.L sense .

we set: $\alpha = \frac{1}{2}$

Then :

$$D_{0+}^{\frac{1}{2}}f(t) = \frac{d}{dt}I_{0+}^{\frac{1}{2}}f(t)$$

By Definition we have:

$$I_{0+}^{\frac{1}{2}}f(t) = \frac{1}{\Gamma(\frac{1}{2})} \int_0^t (t-\tau)^{-\frac{1}{2}} \tau d\tau$$

We integrate by parts:

we set $u = \tau$ and $v' = (t-\tau)^{-\frac{1}{2}}$ then $u' = 1$ and $v = -2(t-\tau)^{\frac{1}{2}}$

we find:

$$\begin{aligned}
 \int_0^t (t-\tau)^{-\frac{1}{2}} \tau d\tau &= \left[-2\tau(t-\tau)^{\frac{1}{2}} \right]_0^t - \int_0^t -2(t-\tau)^{\frac{1}{2}} d\tau \\
 &= 0 + 2 \int_0^t (t-\tau)^{\frac{1}{2}} d\tau = 2 \left[-\frac{2}{3}(t-\tau)^{\frac{3}{2}} \right]_0^t = \frac{4}{3}t^{\frac{3}{2}}
 \end{aligned}$$

therefore:

$$I_{0+}^{\frac{1}{2}}f(t) = \frac{1}{\sqrt{\pi}} \cdot \frac{4}{3}t^{\frac{3}{2}} = \frac{4}{3\sqrt{\pi}}t^{\frac{3}{2}}$$

we have:

$$D_{0+}^{\frac{1}{2}}f(t) = \frac{d}{dt}I_{0+}^{\frac{1}{2}}f(t) = \frac{d}{dt} \left(\frac{4}{3\sqrt{\pi}}t^{\frac{3}{2}} \right) = \frac{4}{3\sqrt{\pi}} \frac{d}{dt}t^{\frac{3}{2}} = \frac{4}{3\sqrt{\pi}} \left(\frac{3}{2}t^{\frac{1}{2}} \right)$$

then:

$$D_{0+}^{\frac{1}{2}}f(t) = \frac{2t^{\frac{1}{2}}}{\sqrt{\pi}}$$

0.5.2 Grünwald-Letnikov fractional derivatives

This definition is based on obtaining derivatives by finite differences.

Let $f : \mathbb{R} \rightarrow \mathbb{R}$, for $h > 0$ we have:

$$f'(t) = \lim_{h \rightarrow 0} \frac{1}{h} [f(t) - f(t-h)]$$

and the second derivative:

$$\begin{aligned} f''(t) &= \lim_{h \rightarrow 0} \frac{1}{h} [f'(t) - f'(t-h)] \\ &= \lim_{h \rightarrow 0} \frac{1}{h} \left[\frac{1}{h} (f(t) - f(t-h)) - \frac{1}{h} (f(t-h) - f(t-2h)) \right] \\ &= \lim_{h \rightarrow 0} \frac{1}{h^2} [f(t) - 2f(t-h) + f(t-2h)]. \end{aligned}$$

More generally, the n -th derivative of f is given by:

$$f^{(n)}(t) = \lim_{h \rightarrow 0} \frac{1}{h^n} \sum_{k=0}^n (-1)^k C_k^n f(t - kh), \quad (1.1)$$

where

$$C_k^n = \frac{n!}{k!(n-k)!} = \frac{n(n-1)\dots(n-(k-1))}{k!}.$$

It is possible to extend C_k^n to $k > n$, by setting $C_k^n = 0$.

Formula (1.1) then becomes

$$f^{(n)}(t) = \lim_{h \rightarrow 0} \frac{1}{h^n} \sum_{k=0}^{\infty} (-1)^k C_k^n f(t - kh).$$

Here again, we can generalize the right-hand term using the Gamma function, by setting for $\alpha \in \mathbb{R}^+ - \mathbb{N}$, and $k \in \mathbb{N}$,

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

Note this time that $C_k^\alpha \neq 0$ even if $k > n$.

Definition

Let $\alpha > 0$. The left Grünwald-Letnikov fractional derivative of order α of f is defined by:

$$\forall t \in \mathbb{R}, G^L \mathcal{D}_+^\alpha f(t) = \lim_{h \rightarrow 0} \frac{1}{h^\alpha} \sum_{k=0}^{\infty} (-1)^k C_k^\alpha f(t - kh).$$

Let us also define its right-sided analogue.

Definition

Let $\alpha > 0$. The right Grünwald-Letnikov fractional derivative of order α of f is defined by:

$$\forall t \in \mathbb{R}, G^L \mathcal{D}_-^\alpha f(t) = \lim_{h \rightarrow 0} \frac{1}{h^\alpha} \sum_{k=0}^{\infty} (-1)^k C_k^\alpha f(t + kh).$$

The Grünwald-Letnikov derivative has an obvious numerical interest. If h is small enough, the discrete evaluation of $\frac{1}{h^\alpha} \sum_{k=0}^{\infty} (-1)^k C_k^\alpha f(t - kh)$ allows approximating the (Liouville) fractional derivative on $\mathbb{R}$.

0.5.3 Relationship Between RL and GL Definitions

The Riemann–Liouville (RL) and Grünwald–Letnikov (GL) definitions of the fractional derivative of order p ($m - 1 \leq p < m$) are **equivalent** if the function $f(t)$ is $(m - 1)$ -**times continuously differentiable** and its derivative of order m is **integrable** on the interval $[a, T]$.

Under these conditions, the fractional derivative can be written in the unified form:

$${}_a D_t^p f(t) = \sum_{j=0}^{m-1} \frac{f^{(j)}(a)(t-a)^{j-p}}{\Gamma(1+j-p)} + \frac{1}{\Gamma(m-p)} \int_a^t \frac{f^{(m)}(\tau) d\tau}{(t-\tau)^{p-m+1}}$$

where Γ is the Gamma function. This expression combines terms depending on the values of the function and its derivatives at the point a (from the GL definition) and an integral depending on the subsequent behavior (from the RL definition). [16]

Special case: $1 \leq p < 2$

Required:

We start from the standard Riemann–Liouville definition :

$${}_a D_t^p f(t) = \frac{1}{\Gamma(2-p)} \left(\frac{d}{dt} \right)^2 \int_a^t (t-\tau)^{1-p} f(\tau) d\tau, \quad 1 \leq p < 2$$

and differentiate twice to obtain the analytical equivalent form of the Grünwald-Letnikov formula:

$${}_a D_t^p f(t) = \frac{f(a)(t-a)^{-p}}{\Gamma(1-p)} + \frac{f'(a)(t-a)^{1-p}}{\Gamma(2-p)} + \frac{1}{\Gamma(2-p)} \int_a^t (t-\tau)^{1-p} f''(\tau) d\tau$$

Proof:

We define

$$I(t) = \int_a^t (t-\tau)^{1-p} f(\tau) d\tau$$

First integration by parts:

We put $u = f(\tau)$ and $dv = (t-\tau)^{1-p} d\tau$.

We know that $\int (t - \tau)^{1-p} d\tau = -\frac{(t-\tau)^{2-p}}{2-p}$, thus:

$$I(t) = \left[-\frac{(t - \tau)^{2-p}}{2-p} f(\tau) \right]_{\tau=a}^{\tau=t} + \frac{1}{2-p} \int_a^t (t - \tau)^{2-p} f'(\tau) d\tau$$

At $\tau = t$, $t - \tau = 0$ so the term vanishes. Thus:

$$I(t) = \frac{f(a)(t-a)^{2-p}}{2-p} + \frac{1}{2-p} \int_a^t (t - \tau)^{2-p} f'(\tau) d\tau$$

Second integration by parts:

We apply the same method to $\int_a^t (t - \tau)^{2-p} f'(\tau) d\tau$:

We set $u = f'(\tau)$ and $dv = (t - \tau)^{2-p} d\tau$.

$\int (t - \tau)^{2-p} d\tau = -\frac{(t-\tau)^{3-p}}{3-p}$, thus:

$$\int_a^t (t - \tau)^{2-p} f'(\tau) d\tau = \left[-\frac{(t - \tau)^{3-p}}{3-p} f'(\tau) \right]_{\tau=a}^{\tau=t} + \frac{1}{3-p} \int_a^t (t - \tau)^{3-p} f''(\tau) d\tau$$

At $\tau = t$, the term vanishes. Thus:

$$\int_a^t (t - \tau)^{2-p} f'(\tau) d\tau = \frac{f'(a)(t-a)^{3-p}}{3-p} + \frac{1}{3-p} \int_a^t (t - \tau)^{3-p} f''(\tau) d\tau$$

Substitution into $I(t)$

$$I(t) = \frac{f(a)(t-a)^{2-p}}{2-p} + \frac{1}{2-p} \left[\frac{f'(a)(t-a)^{3-p}}{3-p} + \frac{1}{3-p} \int_a^t (t - \tau)^{3-p} f''(\tau) d\tau \right]$$

Thus:

$$I(t) = \frac{f(a)(t-a)^{2-p}}{2-p} + \frac{f'(a)(t-a)^{3-p}}{(2-p)(3-p)} + \frac{1}{(2-p)(3-p)} \int_a^t (t - \tau)^{3-p} f''(\tau) d\tau$$

Differentiating twice (since $m = 2$)

We know that ${}_a D_t^p f(t) = \frac{1}{\Gamma(2-p)} \cdot \frac{d^2}{dt^2} I(t)$.

First derivative:

$$\frac{d}{dt} I(t) = \frac{f(a)}{2-p} \cdot \frac{d}{dt} (t-a)^{2-p} + \frac{f'(a)}{(2-p)(3-p)} \cdot \frac{d}{dt} (t-a)^{3-p} + \frac{1}{(2-p)(3-p)} \cdot \frac{d}{dt} \int_a^t (t - \tau)^{3-p} f''(\tau) d\tau$$

We know that: $-\frac{d}{dt} (t-a)^{2-p} = (2-p)(t-a)^{1-p}$ - $\frac{d}{dt} (t-a)^{3-p} = (3-p)(t-a)^{2-p}$

For the last integral, we use Leibniz rule:

$$\begin{aligned} \frac{d}{dt} \int_a^t (t - \tau)^{3-p} f''(\tau) d\tau &= \int_a^t \frac{\partial}{\partial t} (t - \tau)^{3-p} f''(\tau) d\tau + \underbrace{(t-t)^{3-p} f''(t) \cdot 1}_{=0} \\ &= (3-p) \int_a^t (t - \tau)^{2-p} f''(\tau) d\tau \end{aligned}$$

Thus:

$$\frac{d}{dt}I(t) = f(a)(t-a)^{1-p} + \frac{f'(a)(t-a)^{2-p}}{2-p} + \frac{1}{2-p} \int_a^t (t-\tau)^{2-p} f''(\tau) d\tau$$

Second derivative:

$$\frac{d^2}{dt^2}I(t) = f(a) \cdot \frac{d}{dt}(t-a)^{1-p} + \frac{f'(a)}{2-p} \cdot \frac{d}{dt}(t-a)^{2-p} + \frac{1}{2-p} \cdot \frac{d}{dt} \int_a^t (t-\tau)^{2-p} f''(\tau) d\tau$$

We know that: $-\frac{d}{dt}(t-a)^{1-p} = (1-p)(t-a)^{-p}$ - $\frac{d}{dt}(t-a)^{2-p} = (2-p)(t-a)^{1-p}$

For the last integral:

$$\begin{aligned} \frac{d}{dt} \int_a^t (t-\tau)^{2-p} f''(\tau) d\tau &= \int_a^t \frac{\partial}{\partial t} (t-\tau)^{2-p} f''(\tau) d\tau + \underbrace{(t-t)^{2-p} f''(t) \cdot 1}_{=0} \\ &= (2-p) \int_a^t (t-\tau)^{1-p} f''(\tau) d\tau \end{aligned}$$

Thus:

$$\frac{d^2}{dt^2}I(t) = f(a)(1-p)(t-a)^{-p} + \frac{f'(a)(2-p)(t-a)^{1-p}}{2-p} + \frac{1}{2-p} \cdot (2-p) \int_a^t (t-\tau)^{1-p} f''(\tau) d\tau$$

Simplification:

$$\frac{d^2}{dt^2}I(t) = f(a)(1-p)(t-a)^{-p} + f'(a)(t-a)^{1-p} + \int_a^t (t-\tau)^{1-p} f''(\tau) d\tau$$

Now we multiply by $\frac{1}{\Gamma(2-p)}$:

$${}_aD_t^p f(t) = \frac{1}{\Gamma(2-p)} \left[f(a)(1-p)(t-a)^{-p} + f'(a)(t-a)^{1-p} + \int_a^t (t-\tau)^{1-p} f''(\tau) d\tau \right]$$

We know that $\Gamma(2-p) = (1-p)\Gamma(1-p)$, and therefore:

$$(1-p) = \frac{\Gamma(2-p)}{\Gamma(1-p)}$$

Thus:

$$\frac{f(a)(1-p)(t-a)^{-p}}{\Gamma(2-p)} = \frac{f(a)}{\Gamma(2-p)} \cdot \frac{\Gamma(2-p)}{\Gamma(1-p)} \cdot (t-a)^{-p} = \frac{f(a)(t-a)^{-p}}{\Gamma(1-p)}$$

Also:

$$\frac{f'(a)(t-a)^{1-p}}{\Gamma(2-p)} = \frac{f'(a)(t-a)^{1-p}}{\Gamma(2-p)}$$

And:

$$\frac{1}{\Gamma(2-p)} \int_a^t (t-\tau)^{1-p} f''(\tau) d\tau = \frac{1}{\Gamma(2-p)} \int_a^t (t-\tau)^{1-p} f''(\tau) d\tau$$

Substituting:

$$\boxed{{}_aD_t^p f(t) = \frac{f(a)(t-a)^{-p}}{\Gamma(1-p)} + \frac{f'(a)(t-a)^{1-p}}{\Gamma(2-p)} + \frac{1}{\Gamma(2-p)} \int_a^t (t-\tau)^{1-p} f''(\tau) d\tau}$$

This is exactly the Grünwald-Letnikov formula for the case $1 \leq p < 2$.

0.5.4 Properties of the Grünwald-Letnikov Coefficients

The Grünwald-Letnikov coefficients are defined for $1 < \beta < 2$ as:

$$\omega_0 = 1, \quad \omega_j = (-1)^j \binom{\beta}{j}, \quad j \geq 1.$$

We now establish their fundamental properties.

1. Generating function (P1) : Using the generalized binomial theorem, for $|z| \leq 1$, we have :

$$(1 - z)^\beta = \sum_{j=0}^{\infty} \binom{\beta}{j} (-z)^j = \sum_{j=0}^{\infty} (-1)^j \binom{\beta}{j} z^j.$$

Thus, the coefficients satisfy :

$$\boxed{\sum_{j=0}^{\infty} \omega_j z^j = (1 - z)^\beta.} \quad (\text{P1})$$

2. Recurrence relation (P2) : For $j \geq 0$, we have :

$$\boxed{\omega_{j+1} = \left(1 - \frac{1 + \beta}{j + 1}\right) \omega_j.} \quad (\text{P2})$$

Proof :

We prove this by induction on j .

Base case $j = 0$:

$$\omega_1 = -\beta, \quad \left(1 - \frac{1 + \beta}{1}\right) \omega_0 = (1 - (1 + \beta)) \cdot 1 = -\beta.$$

Thus, (P2) holds for $j = 0$.

Inductive step : Assume (P2) holds for some $j \geq 0$. For $j + 1$, using the definition $\omega_{j+2} = (-1)^{j+2} \binom{\beta}{j+2}$ and the property $\binom{\beta}{j+2} = \frac{\beta - (j+1)}{j+2} \binom{\beta}{j+1}$, we obtain :

$$\omega_{j+2} = (-1)^{j+2} \cdot \frac{\beta - (j+1)}{j+2} \binom{\beta}{j+1} = \frac{\beta - (j+1)}{j+2} \cdot (-1) \omega_{j+1}.$$

Since $\frac{\beta - (j+1)}{j+2} = \frac{\beta+1}{j+2} - 1$, we get :

$$\omega_{j+2} = \left(1 - \frac{\beta + 1}{j + 2}\right) \omega_{j+1}.$$

This completes the induction .

3. Explicit formula in terms of the Gamma function (P3) : We prove by induction that

$$\omega_j = \frac{\Gamma(j - \beta)}{\Gamma(-\beta) \Gamma(j + 1)}, \quad j = 0, 1, 2, \dots \quad (\text{P3})$$

Base case $j = 0$:

$$\frac{\Gamma(0 - \beta)}{\Gamma(-\beta) \Gamma(1)} = \frac{\Gamma(-\beta)}{\Gamma(-\beta) \cdot 1} = 1 = \omega_0.$$

Inductive step : Assume the formula holds for some $j \geq 0$. Using the recurrence relation (P2)

$$\omega_{j+1} = \left(1 - \frac{1 + \beta}{j + 1}\right) \omega_j.$$

Substituting the induction hypothesis :

$$\omega_{j+1} = \frac{j - \beta}{j + 1} \cdot \frac{\Gamma(j - \beta)}{\Gamma(-\beta) \Gamma(j + 1)}.$$

Using the Gamma function property $\Gamma(z + 1) = z\Gamma(z)$, we have :

$$\Gamma(j - \beta + 1) = (j - \beta) \Gamma(j - \beta), \quad \Gamma(j + 2) = (j + 1) \Gamma(j + 1).$$

Thus :

$$\omega_{j+1} = \frac{\Gamma(j - \beta + 1)}{\Gamma(-\beta) \Gamma(j + 2)}.$$

This is exactly the formula for $j + 1$. Hence, by induction, (P3) holds for all $j \geq 0$.

4. First coefficients (P4) : From the definition, we directly compute

$$\omega_0 = 1, \quad \omega_1 = -\beta. \tag{P4}$$

5. Zero sum (P5) : Setting $z = 1$ in (P1) gives :

$$\sum_{j=0}^{\infty} \omega_j = (1 - 1)^\beta = 0^\beta = 0. \tag{P5}$$

6. Positivity of coefficients for $j \geq 2$ (P6) : For all $j \geq 2$, we have $\omega_j > 0$ (P6)

Proof :

We prove this by induction using the recurrence relation (P2). First, $\omega_2 = \frac{\beta(\beta-1)}{2} > 0$ because $\beta > 1$. Assume $\omega_j > 0$ for some $j \geq 2$. From (P2): $\omega_{j+1} = \left(1 - \frac{1+\beta}{j+1}\right) \omega_j$. For $j \geq 2$, the factor $\left(1 - \frac{1+\beta}{j+1}\right) = \frac{j-\beta}{j+1} > 0$ because $j \geq 2 > \beta$ (since $\beta < 2$). Thus, $\omega_{j+1} > 0$. By induction, $\omega_j > 0$ for all $j \geq 2$.

7. Sign of the partial sum (P7) : For every $n \geq 1$, define $S_n = \sum_{j=0}^n \omega_j$. Then :

$$S_n < 0 \quad \forall n \geq 1. \tag{P7}$$

Proof :

1. For $n = 1$: $S_1 = 1 - \beta < 0$ since $\beta > 1$.

2. For any $n \geq 1$, $S_{n+1} - S_n = \omega_{n+1}$. From Property (P6), $\omega_{n+1} > 0$ for $n+1 \geq 2$, i.e., for $n \geq 1$. Hence, $\{S_n\}_{n \geq 1}$ is strictly increasing.

3. From (P5), $\lim_{n \rightarrow \infty} S_n = \sum_{j=0}^{\infty} \omega_j = 0$.

4. Since $S_1 < 0$ and the sequence increases to 0, we conclude that $S_n < 0$ for all $n \geq 1$.

remark : The previous properties will be used throughout the consistency and stability analysis.

0.5.5 Fractional Derivative in the Sense of Caputo

The fractional derivative in the sense of Caputo is an extension of the classical derivative to non-integer orders, introduced by the mathematician Michele Caputo. This derivative allows generalizing the concept of differentiation to arbitrary orders. Unlike Riemann-Liouville's derivative, the derivative of Caputo is defined in a more direct and intuitive way, offering a different approach for modeling complex phenomena.

The introduction of the Caputo fractional derivative has sparked growing interest in various scientific fields, particularly in the mechanics of viscoelastic materials, physical modeling, and engineering. This derivative allows capturing memory and non-locality phenomena more precisely, thus offering new perspectives for the analysis and modeling of complex dynamic systems.[9]

Definition:

The derivative of order α in the sense of Caputo of the function $f \in C^{n+1}([a, b])$ is defined by:

$${}_a^C D_t^\alpha f(t) = I_{a+}^{n-\alpha} \frac{d^n}{dt^n} f(t) \quad (1)$$

$$= \frac{1}{\Gamma(n-\alpha)} \int_a^t (t-\tau)^{n-\alpha-1} \frac{d^n}{d\tau^n} f(\tau) d\tau \quad (2)$$

$$= \frac{1}{\Gamma(n-\alpha)} \int_a^t (t-\tau)^{n-\alpha-1} f^{(n)}(\tau) d\tau \quad (3)$$

Proposition:

1/ If $\alpha = 0$:

$${}_a^C D_t^0 f(t) = I_{a+}^{n-0} \frac{d^n}{dt^n} f(t) = I_{a+}^n \frac{d^n}{dt^n} f(t) = f(t),$$

2/ If $\alpha = n$, $n \in \mathbb{N}$:

$${}_a^C D_t^\alpha f(t) = I_{a+}^{n-n} \frac{d^n}{dt^n} f(t) = I_{a+}^0 \frac{d^n}{dt^n} f(t) = \frac{d^n}{dt^n} f(t),$$

The Finite Difference method

For solving partial differential equations (PDEs) there are several methods, and in this thesis we choose the Finite Difference method.

The finite difference method (FDM) is one of the most important numerical methods, and it is used for approximating derivatives and converting differential equations, both ordinary and fractional, into algebraic equations, which can be solved using computers. The fundamental idea of the FDM is based on the approximation of derivatives using the Taylor series expansions, and it is widely used for solving complicated problems arising in physics and engineering.[19] [12]

1.1 Finite Difference Method in the Ordinary Case

The method is mainly based on the principle of the Taylor expansion, where the derivatives of functions are replaced by discrete formulas.

Definition :

Let $u \in C^2([a, b])$, let $x_i \in (a, b)$ and let $h > 0$ be sufficiently small such that $x_i \pm h \in [a, b]$.

Forward Taylor Expansion:

$$u(x_i + h) = u(x_i) + u'(x_i)h + \frac{u''(\xi_+)}{2}h^2 \quad (1)$$

where $\xi_+ \in (x_i, x_i + h)$.

Backward Taylor Expansion:

$$u(x_i - h) = u(x_i) - u'(x_i)h + \frac{u''(\xi_-)}{2}h^2 \quad (2)$$

where $\xi_- \in (x_i - h, x_i)$.

From (1):

$$u(x_i + h) = u(x_i) + u'(x_i)h + \frac{u''(\xi_+)}{2}h^2 \quad \text{where } \xi_+ \in (x_i, x_i + h).$$

we find :

$$\frac{u_{i+1} - u_i}{h} = u'_i + O(h) \quad \text{where } \frac{u''(\xi_+)}{2}h = O(h)$$

hence:

$$u'_i = \frac{u_{i+1} - u_i}{h} + O(h)$$

We obtain the forward finite difference formula.

From (2):

$$u(x_i - h) = u(x_i) - u'(x_i)h + \frac{u''(\xi_-)}{2}h^2 \quad \text{where } \xi_- \in (x_i - h, x_i)$$

we find :

$$\frac{u_{i-1} - u_i}{-h} = u'_i + O(h) \quad \text{where } \frac{u''(\xi_-)}{2}h = O(h)$$

hence:

$$u'_i = \frac{u_i - u_{i-1}}{h} + O(h)$$

This is the backward finite difference formula.

For (1)-(2), we find:

$$u(x_i + h) - u(x_i - h) = 2hu'(x_i) + \frac{2h^3}{3!}u'''(\xi) \quad \text{where } \xi \in (x_i, x_{i+1})$$

Dividing by $2h$:

$$\frac{u_{i+1} - u_{i-1}}{2h} = u'_i + O(h^2) \quad \text{where } \frac{u'''(\xi)}{6}h^2 = O(h^2)$$

hence:

$$u'_i = \frac{u_{i+1} - u_{i-1}}{2h} + O(h^2)$$

This is the centered finite difference formula.

On the other hand:

for (1)+(2), we find:

$$u(x_i + h) + u(x_i - h) = 2u(x_i) + h^2u''(x_i) + \frac{u''''(\xi)}{12}h^4 \quad \text{where } \xi \in (x_i, x_{i+1})$$

hence:

$$u''_i = \frac{u_{i+1} - 2u_i + u_{i-1}}{h^2} + O(h^2) \quad \text{where } \frac{u''''(\xi)}{12}h^2 = O(h^2)$$

We obtain the second-order finite difference formula.

1.1.1 Numerical Convergence

A fundamental concept in numerical analysis is the numerical convergence of the approximate solution towards the exact solution. This convergence is based on the development of the numerical solution obtained by discretization techniques.

Numerical convergence is often studied in two parts: Consistency and Stability.

Definition:

A scheme is said to be convergent if it is consistent and stable.

Numerical Consistency Consistency evaluates how accurately the numerical scheme approximates the original equation (PDE) of the problem. It is verified by evaluating the truncation error, which is the difference between the differential operator of the PDE and its discrete approximation. A scheme is said to be consistent if this error tends towards zero as the discretisation steps $(\Delta t, \Delta x)$ tend towards zero.

Definition:

Consider the initial value problem consisting of the differential equation:

$$F(x, y, y') = 0 \quad (1)$$

subject to appropriate initial conditions. This problem is said to be well-posed if the existence and uniqueness of the solution are ensured.

Let $x \in [a, b] = I$, $y \in C^1$, the exact solution of (1) at the point x_n is $y(x_n)$ and the approximate solution of (1) at the point x_n is y_n . We write equation (1) at the point x_n , we find:

$$F(x, y, y') = 0 \rightarrow F(x, y(x); y'(x)) = 0$$

on the other hand we have:

$$F(x, y_n; y'_n) = 0 + \epsilon_n$$

with: ϵ_n the consistency error.

Then, we say that a numerical scheme is consistent with equation (1); if $\epsilon_n \rightarrow 0$ as $h \rightarrow 0$

Numerical Stability : The stability of a numerical scheme means that errors (rounding, discretisation) do not amplify over time. It ensures that the numerical solution remains bounded and does not diverge in an uncontrolled manner.

Definition:

A numerical method is said to be **stable** if there exists a constant $C > 0$, independent of the step size h and the number of iterations n , such that for any perturbation in the initial data, the difference between the numerical solution and the perturbed solution remains controlled .

We set :

- Y_n : exact numerical solution
- $\tilde{Y}_n$: perturbed numerical solution (resulting from a small perturbation in the initial condition)

We have :

$$Y_n - \tilde{Y}_n = e_n$$

We say that the method is stable if :

$$\|Y_n - \tilde{Y}_n\| = \|e_n\| \leq C \|Y_0 - \tilde{Y}_0\|$$

where $Y_0 - \tilde{Y}_0$ represents the initial perturbation .

1.2 Finite Difference Method in the Fractional Case

1.2.1 The L1 Scheme

We have the Caputo derivative:

$${}_0^C D_x^\alpha f(x) = \frac{1}{\Gamma(n-\alpha)} \int_0^x (x-t)^{n-\alpha-1} f^{(n)}(t) dt,$$

if $n = 1$, $0 < \alpha < 1$:

$${}_0^C D_x^\alpha f(x) = \frac{1}{\Gamma(1-\alpha)} \int_0^x (x-t)^{-\alpha} f'(t) dt,$$

We set:

$$x-t=z, \quad \text{then: } t=x-z, \quad \text{and } dt=-dz$$

By the change of variable, we find:

$$\begin{aligned} {}_0^C D_x^\alpha f(x) &= \frac{1}{\Gamma(1-\alpha)} \int_x^0 z^{-\alpha} f'(x-z)(-dz) \\ &= \frac{1}{\Gamma(1-\alpha)} \int_0^x \frac{f'(x-z)}{z^\alpha} dz \end{aligned}$$

then, we find:

$${}_0^C D_x^\alpha f(x) = \frac{1}{\Gamma(1-\alpha)} \int_0^x \frac{f'(x-z)}{z^\alpha} dz$$

For a discretization of the interval $[0, x]$

$$z_j = jh; \quad j = 0, 1, \dots, N$$

We find:

$${}_0^C D_x^\alpha f(x) = \frac{1}{\Gamma(1-\alpha)} \sum_{j=0}^{n-1} \int_{jh}^{(j+1)h} \frac{f'(x-jh)}{z^\alpha} dz$$

We have the backward finite difference formula:

$$f'_j = \frac{f_j - f_{j-1}}{h}$$

for a discretization:

$$f_j = f(x-jh), \quad j = 0, 1, \dots, N$$

therefore:

$$f'_j = \frac{f_j - f_{j-1}}{h} = \frac{f(x-jh) - f(x-(j+1)h)}{h} = \frac{f_j - f_{j+1}}{h}$$

We find:

$$\begin{aligned}
{}_0^C D_x^\alpha f(x) &= \frac{1}{\Gamma(1-\alpha)} \sum_{j=0}^{n-1} \int_{jh}^{(j+1)h} \frac{f_j - f_{j+1}}{hz^\alpha} dz \\
&= \frac{1}{\Gamma(1-\alpha)} \sum_{j=0}^{n-1} \frac{f_j - f_{j+1}}{h} \int_{jh}^{(j+1)h} \frac{dz}{z^\alpha} \\
&= \frac{1}{\Gamma(1-\alpha)} \sum_{j=0}^{n-1} \frac{f_j - f_{j+1}}{h} \left[\frac{z^{1-\alpha}}{1-\alpha} \right]_{jh}^{(j+1)h} \\
&= \frac{1}{(1-\alpha)\Gamma(1-\alpha)} \sum_{j=0}^{n-1} \frac{f_j - f_{j+1}}{h} [(j+1)^{1-\alpha} h^{1-\alpha} - j^{1-\alpha} h^{1-\alpha}] \\
&= \frac{1}{\Gamma(2-\alpha)} \sum_{j=0}^{n-1} (f_j - f_{j+1}) [(j+1)^{1-\alpha} h^{-\alpha} - j^{1-\alpha} h^{-\alpha}] \\
&= \frac{h^{-\alpha}}{\Gamma(2-\alpha)} \sum_{j=0}^{n-1} (f_j - f_{j+1}) d_j
\end{aligned}$$

Such that:

$$d_j = (j+1)^{1-\alpha} - j^{1-\alpha}$$

hence:

$${}_0^C D_x^\alpha f(x) = \frac{h^{-\alpha}}{\Gamma(2-\alpha)} \sum_{j=0}^{n-1} (f_j - f_{j+1}) d_j$$

Example: We have the function f defined by :

$$f(t) = t, \quad t \in [0, 1], \quad 0 < \alpha < 1$$

1/ Calculate the Caputo derivative:

$${}_0^C D_x^\alpha f(x) = \frac{1}{\Gamma(n-\alpha)} \int_0^x (x-t)^{n-\alpha-1} f^{(n)}(t) dt$$

for $n = 1$:

$$\begin{aligned}
{}_0^C D_x^\alpha f(x) &= \frac{1}{\Gamma(1-\alpha)} \int_0^x (x-t)^{-\alpha} f'(t) dt \\
&= \frac{1}{\Gamma(1-\alpha)} \int_0^x (x-t)^{-\alpha} \cdot 1 dt \\
&= \frac{1}{\Gamma(1-\alpha)} \left[-\frac{(x-t)^{1-\alpha}}{1-\alpha} \right]_0^x \\
&= \frac{1}{\Gamma(1-\alpha)} \cdot \frac{x^{1-\alpha}}{1-\alpha} \\
&= \frac{x^{1-\alpha}}{\Gamma(2-\alpha)}
\end{aligned}$$

For $\alpha = \frac{1}{2}$, we find:

$${}_0^C D_x^\alpha f(x) = \frac{1}{\Gamma(\frac{1}{2})} \int_0^x (x-t)^{-\frac{1}{2}} dt = \frac{1}{\sqrt{\pi}} \left[-2(x-t)^{\frac{1}{2}} \right]_0^x = \frac{2}{\sqrt{\pi}} x^{\frac{1}{2}} = \frac{2\sqrt{x}}{\sqrt{\pi}}$$

2/ Calculate the derivative by the Fractional F.D. method:

We have the L1 scheme:

$${}_0^C D_x^\alpha f(x) = \frac{h^{-\alpha}}{\Gamma(2-\alpha)} \sum_{j=0}^{n-1} (f_j - f_{j+1}) [(j+1)^{1-\alpha} - j^{1-\alpha}]$$

for:

$$n = 100, \quad t = 1, \quad h = \frac{1}{100}, \quad \Gamma\left(\frac{3}{2}\right) = \frac{\sqrt{\pi}}{2}$$

We find:

$$\begin{aligned} {}_0^C D_x^\alpha f(1) &= \frac{(0.01)^{-0.5}}{\Gamma(1.5)} \sum_{j=0}^{99} \left(\left(1 - \frac{j}{100}\right) - \left(1 - \frac{j+1}{100}\right) \right) ((j+1)^{0.5} - j^{0.5}) \\ &= \frac{10}{(\sqrt{\pi}/2)} \sum_{j=0}^{99} \frac{1}{100} ((j+1)^{0.5} - j^{0.5}) = \frac{20}{\sqrt{\pi}} \cdot \frac{1}{100} \cdot 100^{0.5} = \frac{2}{\sqrt{\pi}} \end{aligned}$$

and we have:

$${}_0^C D_x^\alpha f(x) = \frac{x^{1-\alpha}}{\Gamma(2-\alpha)}$$

taking $\alpha = 0.99$, $x = 1$:

$${}_0^C D_x^\alpha f(1) = \frac{1^{1-0.99}}{\Gamma(2-0.99)} = \frac{1}{\Gamma(1.01)} \approx 1.0057065285004$$

on the other hand:

$$\begin{aligned} {}_0^C D_x^\alpha f(1) &= \frac{(0.01)^{-0.99}}{\Gamma(2-0.99)} \sum_{j=0}^{99} \left(\left(1 - \frac{j}{100}\right) - \left(1 - \frac{j+1}{100}\right) \right) ((j+1)^{1-0.99} - j^{1-0.99}) \\ &= \frac{(0.01)^{-0.99}}{\Gamma(1.01)} \sum_{j=0}^{99} \frac{1}{100} ((j+1)^{0.01} - j^{0.01}) \approx 1.0057065285003 \end{aligned}$$

Numerical analysis of an EDPF problem with integral boundary conditions

The fractional finite difference method is an extension of the classical finite difference method, extending it to include fractional differential equations. This method allows for the numerical approximation of derivatives and integrals of fractional order that appear in these equations. The main steps of this method are:

- Dividing space and time into discrete points, as in the classical finite difference method.
- Approximating fractional derivatives and integrals using finite difference formulas of non-integer order, such as the Caputo or Riemann-Liouville formulas.
- Converting fractional differential equations into a system of algebraic equations that can be solved numerically.

This method enables the numerical solution of a wide range of fractional differential equations, opening the door to extensive applications in fields such as physics, fluid mechanics, biology and electrochemistry, where the phenomena under study often exhibit memory-like behaviour.

2.1 Position of the problem

In the rectangular domain

$$Q = \{(x, t) \in \mathbb{R}^2 : 0 < x < 1, 0 < t < T\}, \text{ where } T > 0,$$

we consider the fractional differential equation:

$${}_0^C D_t^\alpha u(t) = a(x, t) {}_0^{RL} D_x^\beta u + \phi(x, t), \text{ where } 0 < \alpha < 1, 1 < \beta < 2 \quad (1)$$

to the equation (1), we associate the initial and Newman conditions:

$$u(x, 0) = f(x), \quad 0 < x < 1, \quad (2)$$

$$\frac{\partial u(0, t)}{\partial x} = g_0(t), \quad 0 < t < T, \quad (3)$$

and the purely integral condition

$$\int_0^1 u(x, t) dx = g_1(t), \quad 0 < t \leq T, \quad (4)$$

where ϕ, a, f, g_0, g_1 are known continuous functions, and a is positive function.

2.2 Discretization and Approximate Formulation

We consider a uniform subdivision of the intervals $[0, 1]$ and $[0, T]$ as follows

$$x_i = ih; i = 0, 1, \dots, N \quad \text{and} \quad t_k = kh_t; k = 0, 1, \dots, M.$$

$$u(ih, kh_t) = u_i^k = u_{i,k}$$

such that $u_{i,k}$ represent the value of the solution u in point (x_i, t_k) .

$$a_i^k = a(x_i, t_k), \quad \phi_i^k = \phi(x_i, t_k), \quad f_i^k = f(x_i, t_k), \quad g_1(t_k) = g_1^k, \quad g_0(t_k) = g_0^k$$

The Caputo derivative fractional of operator ∂_t^α at point (x_i, t_{k+1}) said L_1 with $0 < \alpha < 1$ is defined by (See [14].)

$$({}_0^C D_t^\alpha u(x_i, t_{k+1}))_{L_1} = ({}_0^C D_t^\alpha u)_i^{k+1} \simeq \mu \sum_{j=0}^k (u_i^{k-j+1} - u_i^{k-j}) d_j \quad (5)$$

where

$$\begin{cases} d_j = (j+1)^{1-\alpha} - j^{1-\alpha}, \\ d_0 = 1; k = 1, \dots, M \end{cases}, \quad \mu = \frac{h_t^{-\alpha}}{\Gamma(2-\alpha)}.$$

Grunwald approximation of the Riemann-Liouville fractional derivative of order $1 < \beta < 2$ at the point (x_i, t_{k+1}) :

$$({}^{RL} D_x^\beta u)_i^{k+1} \approx h^{-\beta} \sum_{j=0}^{i+1} \omega_j u(x_{i-j+1}, t_{k+1}) \quad (6)$$

Derivation of the shifted Grünwald:

The Riemann-Liouville fractional derivative is defined by the relation:

$$({}^{RL} D_x^\beta u)(x, t) = \frac{1}{\Gamma(2-\beta)} \frac{\partial^2}{\partial x^2} \int_a^x (x-s)^{1-\beta} u(s, t) ds$$

Grunwald introduced an approximate formula for this derivative based on extending the concept of finite differences. As proved in Chapter 1, the Grünwald-Letnikov fractional derivative is equivalent to the Riemann-Liouville derivative for $1 < \beta < 2$ under suitable smoothness conditions. Therefore, At the point (x_i, t_{k+1}) , we write:

$$({}^{RL} D_x^\beta u)_i^{k+1} = \lim_{h \rightarrow 0^+} h^{-\beta} \sum_{j=0}^{\lfloor x_i/h \rfloor} \omega_j u(x_i - jh, t_{k+1})$$

where ω_j are the Grunwald coefficients.

For numerical application, we replace the limit with a small value h representing the spatial step size, thus obtaining:

$$({}^{RL}D_x^\beta u)_i^{k+1} \approx h^{-\beta} \sum_{j=0}^i \omega_j u(x_i - jh, t_{k+1})$$

However, for $1 < \beta < 2$, the standard formula is numerically unstable. Therefore, we apply a **one-index shift**, replacing $x_i - jh$ by $x_i - (j-1)h$ and extending the sum to $j = i+1$:

$$({}^{RL}D_x^\beta u)_i^{k+1} \approx h^{-\beta} \sum_{j=0}^{i+1} \omega_j u(x_i - (j-1)h, t_{k+1})$$

We note that $x_i - (j-1)h = x_{i-j+1}$, Thus, we have obtained the required formula:

$$({}^{RL}D_x^\beta u)_i^{k+1} \approx h^{-\beta} \sum_{j=0}^{i+1} \omega_j u(x_{i-j+1}, t_{k+1})$$

where:

- h : spatial step size
- β : order of the derivative ($1 < \beta < 2$)
- $i = 1, 2, \dots, N-1$
- $k = 1, 2, \dots, M$
- ω_j : The Grunwald coefficients are given by the relation: (see P3)

$$\omega_j = \frac{\Gamma(j-\beta)}{\Gamma(-\beta)\Gamma(j+1)}, \quad \omega_0 = 1$$

equation (1) at point $(i, k+1)$ gives

$$\mu \sum_{j=0}^k \left(u_i^{k-j+1} - u_i^{k-j} \right) d_j = a_i^{k+1} h^{-\beta} \sum_{j=0}^{i+1} \omega_j u_{i-j+1}^{k+1} + \phi_i^{k+1} \quad (7)$$

where $k = 0, \dots, M-1$, $i = 1, \dots, N-1$

2.3 Numerical Scheme

2.3.1 Case $k = 0$ (First Time Step)

We start from equation (7) for the case $k = 0$:

$$\mu(u_i^1 - u_i^0) = a_i^1 h^{-\beta} \sum_{j=0}^{i+1} \omega_j u_{i-j+1}^1 + \phi_i^1$$

Multiply by h^β :

$$\mu h^\beta (u_i^1 - u_i^0) = a_i^1 \sum_{j=0}^{i+1} \omega_j u_{i-j+1}^1 + h^\beta \phi_i^1$$

Define $\lambda = \mu h^\beta$:

$$\lambda(u_i^1 - u_i^0) = a_i^1 \sum_{j=0}^{i+1} \omega_j u_{i-j+1}^1 + h^\beta \phi_i^1$$

Separate the sum:

$$\lambda u_i^1 - \lambda u_i^0 = a_i^1 \left(\sum_{j=0}^i \omega_j u_{i-j+1}^1 + \omega_{i+1} u_0^1 \right) + h^\beta \phi_i^1$$

Thus :

$$\lambda u_i^1 - a_i^1 \sum_{j=0}^i \omega_j u_{i-j+1}^1 - a_i^1 \omega_{i+1} u_0^1 = \lambda u_i^0 + h^\beta \phi_i^1$$

Define $A_i^1 = \lambda - a_i^1 \omega_1$, and separate the term $j = 1$ from the sum:

$$(\lambda - a_i^1 \omega_1) u_i^1 - a_i^1 \sum_{\substack{j=0 \\ j \neq 1}}^i \omega_j u_{i-j+1}^1 - a_i^1 \omega_{i+1} u_0^1 = \lambda u_i^0 + h^\beta \phi_i^1$$

we get :

$$\boxed{A_i^1 u_i^1 - a_i^1 \sum_{\substack{j=0 \\ j \neq 1}}^i \omega_j u_{i-j+1}^1 - a_i^1 \omega_{i+1} u_0^1 = \lambda u_i^0 + h^\beta \phi_i^1} \quad (8)$$

where:

- $A_i^1 = \lambda - a_i^1 \omega_1$
- $\lambda = \mu h^\beta = \frac{h_t^{-\alpha} h^\beta}{\Gamma(2 - \alpha)}$

Applying the Neumann condition: $u_0^1 = u_1^1 - h g_0^1$:

$$A_i^1 u_i^1 - a_i^1 \sum_{\substack{j=0 \\ j \neq 1}}^i \omega_j u_{i-j+1}^1 - a_i^1 \omega_{i+1} (u_1^1 - h g_0^1) = \lambda u_i^0 + h^\beta \phi_i^1$$

Expand:

$$A_i^1 u_i^1 - a_i^1 \sum_{\substack{j=0 \\ j \neq 1}}^i \omega_j u_{i-j+1}^1 - a_i^1 \omega_{i+1} u_1^1 + a_i^1 \omega_{i+1} h g_0^1 = \lambda u_i^0 + h^\beta \phi_i^1$$

we get :

$$\boxed{A_i^1 u_i^1 - a_i^1 \sum_{\substack{j=0 \\ j \neq 1}}^i \omega_j u_{i-j+1}^1 - a_i^1 \omega_{i+1} u_1^1 = \lambda u_i^0 + h^\beta \phi_i^1 - a_i^1 \omega_{i+1} h g_0^1} \quad (9)$$

For $i = N - 1$:

$$A_{N-1}^1 u_{N-1}^1 - a_{N-1}^1 \sum_{\substack{j=0 \\ j \neq 1}}^{N-1} \omega_j u_{N-j}^1 - a_{N-1}^1 \omega_N u_1^1 = \lambda u_{N-1}^0 + h^\beta \phi_{N-1}^1 - a_{N-1}^1 \omega_N h g_0^1$$

We have :

$$\sum_{\substack{j=0 \\ j \neq 1}}^{N-1} \omega_j u_{N-j}^1 = \omega_0 u_N^1 + \sum_{j=2}^{N-1} \omega_j u_{N-j}^1$$

Thus:

$$A_{N-1}^1 u_{N-1}^1 - a_{N-1}^1 \left(\omega_0 u_N^1 + \sum_{j=2}^{N-1} \omega_j u_{N-j}^1 \right) - a_{N-1}^1 \omega_N u_1^1 = \lambda u_{N-1}^0 + h^\beta \phi_{N-1}^1 - a_{N-1}^1 \omega_N h g_0^1$$

we get :

$$\boxed{A_{N-1}^1 u_{N-1}^1 - a_{N-1}^1 \omega_0 u_N^1 - a_{N-1}^1 \sum_{j=2}^{N-1} \omega_j u_{N-j}^1 - a_{N-1}^1 \omega_N u_1^1 = \lambda u_{N-1}^0 + h^\beta \phi_{N-1}^1 - a_{N-1}^1 \omega_N h g_0^1} \quad (10)$$

From the integral condition:

$$\int_0^1 u(x, t^1) dx = g_1^1$$

and applying the trapezoidal rule:

$$\frac{h}{2} \left[u_0^1 + 2 \sum_{i=1}^{N-1} u_i^1 + u_N^1 \right] = g_1^1$$

Substitute $u_0^1 = u_1^1 - h g_0^1$:

$$\frac{h}{2} \left[(u_1^1 - h g_0^1) + 2 \sum_{i=1}^{N-1} u_i^1 + u_N^1 \right] = g_1^1$$

Multiply by $2/h$:

$$u_1^1 - h g_0^1 + 2 \sum_{i=1}^{N-1} u_i^1 + u_N^1 = \frac{2}{h} g_1^1$$

S oneparate the sum: $\sum_{i=1}^{N-1} u_i^1 = u_1^1 + \sum_{i=2}^{N-1} u_i^1$:

$$u_1^1 - h g_0^1 + 2 u_1^1 + 2 \sum_{i=2}^{N-1} u_i^1 + u_N^1 = \frac{2}{h} g_1^1$$

$$3u_1^1 + 2 \sum_{i=2}^{N-1} u_i^1 + u_N^1 = \frac{2}{h}g_1^1 + hg_0^1$$

Then :

$$u_N^1 = -3u_1^1 - 2 \sum_{i=2}^{N-1} u_i^1 + \frac{2}{h}g_1^1 + hg_0^1 \quad (11)$$

Substitute into the equation (10):

$$A_{N-1}^1 u_{N-1}^1 - a_{N-1}^1 \omega_0 \left(-3u_1^1 - 2 \sum_{i=2}^{N-1} u_i^1 + \frac{2}{h}g_1^1 + hg_0^1 \right) - a_{N-1}^1 \sum_{j=2}^{N-1} \omega_j u_{N-j}^1 - a_{N-1}^1 \omega_N u_1^1 = \lambda u_{N-1}^0 + h^\beta \phi_{N-1}^1 - a_{N-1}^1 \omega_N hg_0^1$$

Expand the first term: $A_{N-1}^1 u_{N-1}^1 + a_{N-1}^1 \omega_0 \cdot 3u_1^1 + a_{N-1}^1 \omega_0 \cdot 2 \sum_{i=2}^{N-1} u_i^1 - a_{N-1}^1 \omega_0 \left(\frac{2}{h}g_1^1 + hg_0^1 \right) - a_{N-1}^1 \sum_{j=2}^{N-1} \omega_j u_{N-j}^1 - a_{N-1}^1 \omega_N u_1^1 = \lambda u_{N-1}^0 + h^\beta \phi_{N-1}^1 - a_{N-1}^1 \omega_N hg_0^1$

We get :

$$\begin{aligned} & a_{N-1}^1 (3\omega_0 - \omega_N) u_1^1 + a_{N-1}^1 \sum_{i=2}^{N-2} (2\omega_0 - \omega_{N-i}) u_i^1 + (A_{N-1}^1 + 2a_{N-1}^1 \omega_0) u_{N-1}^1 \\ & = \lambda u_{N-1}^0 + h^\beta \phi_{N-1}^1 - a_{N-1}^1 \omega_N hg_0^1 + a_{N-1}^1 \omega_0 \left(\frac{2}{h}g_1^1 + hg_0^1 \right) \end{aligned} \quad (12)$$

Matrix System for $k = 0$ (First Time Step)

Taking into account (9), (12) and defining:

$$U^k = (u_1^k, u_2^k, \dots, u_{N-1}^k)^T$$

$$U^0 = (u_1^0, u_2^0, \dots, u_{N-1}^0)^T$$

$$\Phi^k = (\phi_1^k, \phi_2^k, \dots, \phi_{N-1}^k)^T$$

$$W^1 = (-a_1^1 \omega_2 hg_0^1, -a_2^1 \omega_3 hg_0^1, \dots, -a_{N-2}^1 \omega_{N-1} hg_0^1, -a_{N-1}^1 \omega_N hg_0^1 + a_{N-1}^1 \omega_0 \left(\frac{2}{h}g_1^1 + hg_0^1 \right))^T$$

we obtain the following system:

$$P^1 U^1 = \lambda U^0 + h^\beta \Phi^1 + W^1 \quad (13)$$

where $P^1 = (p_{i,j})_{(N-1) \times (N-1)}$ is a matrix defined by:

$$p_{i,j} = \begin{cases} A_1^1 - a_1^1 \omega_2 & \text{if } i = 1, j = 1 \\ -a_1^1 & \text{if } i = 1, j = 2 \\ A_i^1 & \text{if } 2 \leq i \leq N-2, j = i \\ -a_i^1 & \text{if } 2 \leq i \leq N-2, j = i+1 \\ -a_i^1 \omega_{i+1} & \text{if } 2 \leq i \leq N-2, j = 1 \\ -a_i^1 \omega_{i-j+1} & \text{if } 2 \leq i \leq N-2, 2 \leq j \leq i-1 \\ a_{N-1}^1 (3\omega_0 - \omega_N) & \text{if } i = N-1, j = 1 \\ a_{N-1}^1 (2\omega_0 - \omega_{N-j}) & \text{if } i = N-1, 2 \leq j \leq N-2 \\ A_{N-1}^1 + 2a_{N-1}^1 \omega_0 & \text{if } i = N-1, j = N-1 \\ 0 & \text{otherwise} \end{cases} \quad (14)$$

2.3.2 General Case ($k \geq 1$)

For $k \geq 1$

Multiply equation (7) by h^β and define $\lambda = \mu h^\beta$:

$$\lambda \sum_{j=0}^k (u_i^{k-j+1} - u_i^{k-j}) d_j = a_i^{k+1} \sum_{j=0}^{i+1} \omega_j u_{i-j+1}^{k+1} + h^\beta \phi_i^{k+1}$$

Separate the term $j = 0$ from the time sum (noting that $d_0 = 1$):

$$\lambda(u_i^{k+1} - u_i^k) + \lambda \sum_{j=1}^k (u_i^{k-j+1} - u_i^{k-j}) d_j = a_i^{k+1} \sum_{j=0}^{i+1} \omega_j u_{i-j+1}^{k+1} + h^\beta \phi_i^{k+1}$$

we get :

$$\lambda u_i^{k+1} + \lambda \sum_{j=1}^k (u_i^{k-j+1} - u_i^{k-j}) d_j - a_i^{k+1} \sum_{j=0}^{i+1} \omega_j u_{i-j+1}^{k+1} = \lambda u_i^k + h^\beta \phi_i^{k+1} \quad (15)$$

Suppose that:

$$S = \sum_{j=1}^k (u_i^{k-j+1} - u_i^{k-j}) d_j$$

Expand the sum:

$$S = \sum_{j=1}^k u_i^{k-j+1} d_j - \sum_{j=1}^k u_i^{k-j} d_j$$

Set $m = j - 1$, i.e., $j = m + 1$:

- When $j = 1$: $m = 0$
- When $j = k$: $m = k - 1$

$$\sum_{j=1}^k u_i^{k-j+1} d_j = \sum_{m=0}^{k-1} u_i^{k-m} d_{m+1}$$

The second sum remains as is with an index change:

$$\sum_{j=1}^k u_i^{k-j} d_j = \sum_{m=1}^k u_i^{k-m} d_m$$

Thus:

$$S = \sum_{m=0}^{k-1} u_i^{k-m} d_{m+1} - \sum_{m=1}^k u_i^{k-m} d_m$$

Separate the sums:

$$S = u_i^k d_1 + \sum_{m=1}^{k-1} u_i^{k-m} d_{m+1} - \sum_{m=1}^{k-1} u_i^{k-m} d_m - u_i^0 d_k$$

Then

$$S = d_1 u_i^k - d_k u_i^0 + \sum_{m=1}^{k-1} u_i^{k-m} (d_{m+1} - d_m)$$

Define $\sigma_m = d_{m+1} - d_m$:

$$S = d_1 u_i^k - d_k u_i^0 + \sum_{m=1}^{k-1} \sigma_m u_i^{k-m}$$

where $\sigma_0 = d_1 - d_0 = d_1 - 1$.

Substitute (S) into equation (15):

$$\lambda u_i^{k+1} + \lambda \left(d_1 u_i^k - d_k u_i^0 + \sum_{j=1}^{k-1} \sigma_j u_i^{k-j} \right) - a_i^{k+1} \sum_{j=0}^{i+1} \omega_j u_{i-j+1}^{k+1} = \lambda u_i^k + h^\beta \phi_i^{k+1}$$

Thus :

$$\lambda u_i^{k+1} + \lambda d_1 u_i^k - \lambda d_k u_i^0 + \lambda \sum_{j=1}^{k-1} \sigma_j u_i^{k-j} - \lambda u_i^k - a_i^{k+1} \sum_{j=0}^{i+1} \omega_j u_{i-j+1}^{k+1} = h^\beta \phi_i^{k+1}$$

Note that $\lambda d_1 u_i^k - \lambda u_i^k = \lambda(d_1 - 1)u_i^k$, and define $\sigma_0 = d_1 - 1$:

$$\lambda u_i^{k+1} + \lambda \sigma_0 u_i^k - \lambda d_k u_i^0 + \lambda \sum_{j=1}^{k-1} \sigma_j u_i^{k-j} - a_i^{k+1} \sum_{j=0}^{i+1} \omega_j u_{i-j+1}^{k+1} = h^\beta \phi_i^{k+1}$$

We get :

$$\lambda u_i^{k+1} - a_i^{k+1} \sum_{j=0}^{i+1} \omega_j u_{i-j+1}^{k+1} = -\lambda \sum_{j=0}^{k-1} \sigma_j u_i^{k-j} + \lambda d_k u_i^0 + h^\beta \phi_i^{k+1} \quad (16)$$

The Neumann condition gives us: $u_0^{k+1} = u_1^{k+1} - h g_0^{k+1}$

we have From the equation(15) :

$$\sum_{j=0}^{i+1} \omega_j u_{i-j+1}^{k+1} = \sum_{j=0}^i \omega_j u_{i-j+1}^{k+1} + \omega_{i+1} u_0^{k+1}$$

Substitute u_0^{k+1} from the Neumann condition:

$$\sum_{j=0}^{i+1} \omega_j u_{i-j+1}^{k+1} = \sum_{j=0}^i \omega_j u_{i-j+1}^{k+1} + \omega_{i+1} (u_1^{k+1} - h g_0^{k+1})$$

Substitute into equation (15):

$$\lambda u_i^{k+1} - a_i^{k+1} \left(\sum_{j=0}^i \omega_j u_{i-j+1}^{k+1} + \omega_{i+1} (u_1^{k+1} - h g_0^{k+1}) \right) = -\lambda \sum_{j=0}^{k-1} \sigma_j u_i^{k-j} + \lambda d_k u_i^0 + h^\beta \phi_i^{k+1}$$

Expand the bracket:

$$\lambda u_i^{k+1} - a_i^{k+1} \sum_{j=0}^i \omega_j u_{i-j+1}^{k+1} - a_i^{k+1} \omega_{i+1} u_1^{k+1} + a_i^{k+1} \omega_{i+1} h g_0^{k+1} = -\lambda \sum_{j=0}^{k-1} \sigma_j u_i^{k-j} + \lambda d_k u_i^0 + h^\beta \phi_i^{k+1}$$

Thus :

$$\lambda u_i^{k+1} - a_i^{k+1} \sum_{j=0}^i \omega_j u_{i-j+1}^{k+1} - a_i^{k+1} \omega_{i+1} u_1^{k+1} = -\lambda \sum_{j=0}^{k-1} \sigma_j u_i^{k-j} + \lambda d_k u_i^0 + h^\beta \phi_i^{k+1} - a_i^{k+1} \omega_{i+1} h g_0^{k+1}$$

We get :

$$\boxed{\lambda u_i^{k+1} - a_i^{k+1} \left(\sum_{j=0}^i \omega_j u_{i-j+1}^{k+1} + \omega_{i+1} u_1^{k+1} \right) = -\lambda \sum_{j=0}^{k-1} \sigma_j u_i^{k-j} + \lambda d_k u_i^0 + h^\beta \phi_i^{k+1} - a_i^{k+1} \omega_{i+1} h g_0^{k+1}} \quad (17)$$

In the sum $\sum_{j=0}^i \omega_j u_{i-j+1}^{k+1}$, separate the term $j = 1$ which gives $\omega_1 u_i^{k+1}$:

$$\sum_{j=0}^i \omega_j u_{i-j+1}^{k+1} = \omega_1 u_i^{k+1} + \sum_{\substack{j=0 \\ j \neq 1}}^i \omega_j u_{i-j+1}^{k+1}$$

Substitute into equation (17):

$$\lambda u_i^{k+1} - a_i^{k+1} \left(\omega_1 u_i^{k+1} + \sum_{\substack{j=0 \\ j \neq 1}}^i \omega_j u_{i-j+1}^{k+1} + \omega_{i+1} u_1^{k+1} \right) = -\lambda \sum_{j=0}^{k-1} \sigma_j u_i^{k-j} + \lambda d_k u_i^0 + h^\beta \phi_i^{k+1} - a_i^{k+1} \omega_{i+1} h g_0^{k+1}$$

Group the terms with u_i^{k+1} , then we get :

$$\boxed{A_i^{k+1} u_i^{k+1} - a_i^{k+1} \left(\sum_{\substack{j=0 \\ j \neq 1}}^i \omega_j u_{i-j+1}^{k+1} + \omega_{i+1} u_1^{k+1} \right) = -\lambda \sum_{j=0}^{k-1} \sigma_j u_i^{k-j} + \lambda d_k u_i^0 + h^\beta \phi_i^{k+1} - a_i^{k+1} \omega_{i+1} h g_0^{k+1}} \quad (18)$$

Where $A_i^{k+1} = \lambda - a_i^{k+1} \omega_1$

This equation is valid for $i = 1, 2, \dots, N-2$.

For $i = N - 1$:

$$A_{N-1}^{k+1}u_{N-1}^{k+1} - a_{N-1}^{k+1} \left(\sum_{\substack{j=0 \\ j \neq 1}}^{N-1} \omega_j u_{N-j}^{k+1} + \omega_N u_1^{k+1} \right) = -\lambda \sum_{j=0}^{k-1} \sigma_j u_{N-1}^{k-j} + \lambda d_k u_{N-1}^0 + h^\beta \phi_{N-1}^{k+1} - a_{N-1}^{k+1} \omega_N h g_0^{k+1}$$

We have :

$$\sum_{\substack{j=0 \\ j \neq 1}}^{N-1} \omega_j u_{N-j}^{k+1} = \omega_0 u_N^{k+1} + \sum_{j=2}^{N-1} \omega_j u_{N-j}^{k+1}$$

knowing that $\omega_0 = 1$, by Substitute we get :

$$A_{N-1}^{k+1}u_{N-1}^{k+1} - a_{N-1}^{k+1} \left(\omega_0 u_N^{k+1} + \sum_{j=2}^{N-1} \omega_j u_{N-j}^{k+1} + \omega_N u_1^{k+1} \right) = -\lambda \sum_{j=0}^{k-1} \sigma_j u_{N-1}^{k-j} + \lambda d_k u_{N-1}^0 + h^\beta \phi_{N-1}^{k+1} - a_{N-1}^{k+1} \omega_N h g_0^{k+1} \quad (19)$$

more clearly :

$$A_{N-1}^{k+1}u_{N-1}^{k+1} - a_{N-1}^{k+1} \omega_0 u_N^{k+1} - a_{N-1}^{k+1} \sum_{j=2}^{N-1} \omega_j u_{N-j}^{k+1} - a_{N-1}^{k+1} \omega_N u_1^{k+1} = -\lambda \sum_{j=0}^{k-1} \sigma_j u_{N-1}^{k-j} + \lambda d_k u_{N-1}^0 + h^\beta \phi_{N-1}^{k+1} - a_{N-1}^{k+1} \omega_N h g_0^{k+1}$$

From the integral condition (4) and the trapezoidal rule:

$$\int_0^1 u(x, t) dx = \frac{h}{2} \left[u_0^{k+1} + 2 \sum_{i=1}^{N-1} u_i^{k+1} + u_N^{k+1} \right] = g_1^{k+1}$$

Substitute the Neumann condition $u_0^{k+1} = u_1^{k+1} - h g_0^{k+1}$:

$$\begin{aligned} \frac{h}{2} \left[(u_1^{k+1} - h g_0^{k+1}) + 2 \sum_{i=1}^{N-1} u_i^{k+1} + u_N^{k+1} \right] &= g_1^{k+1} \\ \frac{h}{2} \left[u_1^{k+1} - h g_0^{k+1} + 2 u_1^{k+1} + 2 \sum_{i=2}^{N-1} u_i^{k+1} + u_N^{k+1} \right] &= g_1^{k+1} \\ \frac{h}{2} \left[3 u_1^{k+1} + 2 \sum_{i=2}^{N-1} u_i^{k+1} + u_N^{k+1} - h g_0^{k+1} \right] &= g_1^{k+1} \end{aligned}$$

Thus :

$$3 u_1^{k+1} + 2 \sum_{i=2}^{N-1} u_i^{k+1} + u_N^{k+1} - h g_0^{k+1} = \frac{2}{h} g_1^{k+1}$$

Isolate u_N^{k+1} :

$$u_N^{k+1} = -3u_1^{k+1} - 2 \sum_{i=2}^{N-1} u_i^{k+1} + \frac{2}{h} g_1^{k+1} + h g_0^{k+1} \quad (20)$$

Substitute into equation (19) :

$$A_{N-1}^{k+1} u_{N-1}^{k+1} - a_{N-1}^{k+1} \omega_0 \left(-3u_1^{k+1} - 2 \sum_{i=2}^{N-1} u_i^{k+1} + \frac{2}{h} g_1^{k+1} + h g_0^{k+1} \right) - a_{N-1}^{k+1} \sum_{j=2}^{N-1} \omega_j u_{N-j}^{k+1} - a_{N-1}^{k+1} \omega_N u_1^{k+1} = -\lambda \sum_{j=0}^{k-1} \sigma_j u_{N-1}^{k-j} + \lambda d_k u_{N-1}^0 + h^\beta \phi_{N-1}^{k+1} - a_{N-1}^{k+1} \omega_N h g_0^{k+1} \text{ then :}$$

$$\begin{aligned} & A_{N-1}^{k+1} u_{N-1}^{k+1} + 3a_{N-1}^{k+1} \omega_0 u_1^{k+1} + 2a_{N-1}^{k+1} \omega_0 \sum_{i=2}^{N-1} u_i^{k+1} - a_{N-1}^{k+1} \omega_0 \left(\frac{2}{h} g_1^{k+1} + h g_0^{k+1} \right) \\ & - a_{N-1}^{k+1} \sum_{j=2}^{N-1} \omega_j u_{N-j}^{k+1} - a_{N-1}^{k+1} \omega_N u_1^{k+1} = -\lambda \sum_{j=0}^{k-1} \sigma_j u_{N-1}^{k-j} + \lambda d_k u_{N-1}^0 + h^\beta \phi_{N-1}^{k+1} - a_{N-1}^{k+1} \omega_N h g_0^{k+1} \end{aligned}$$

Therefore

$$\begin{aligned} & a_{N-1}^{k+1} (3\omega_0 - \omega_N) u_1^{k+1} + a_{N-1}^{k+1} \sum_{i=2}^{N-2} (2\omega_0 - \omega_{N-i}) u_i^{k+1} + (A_{N-1}^{k+1} + 2a_{N-1}^{k+1} \omega_0) u_{N-1}^{k+1} \\ & = -\lambda \sum_{j=0}^{k-1} \sigma_j u_{N-1}^{k-j} + \lambda d_k u_{N-1}^0 + h^\beta \phi_{N-1}^{k+1} - a_{N-1}^{k+1} \omega_N h g_0^{k+1} + a_{N-1}^{k+1} \omega_0 \left(\frac{2}{h} g_1^{k+1} + h g_0^{k+1} \right) \end{aligned} \quad (21)$$

Matrix System for $k \geq 1$ (General Case)

For each interior point $i = 1, \dots, N-2$, we have equation (18):

$$A_i^{k+1} u_i^{k+1} - a_i^{k+1} \sum_{\substack{j=0 \\ j \neq 1}}^i \omega_j u_{i-j+1}^{k+1} - a_i^{k+1} \omega_{i+1} u_1^{k+1} = -\lambda \sum_{j=0}^{k-1} \sigma_j u_i^{k-j} + \lambda d_k u_i^0 + h^\beta \phi_i^{k+1} - a_i^{k+1} \omega_{i+1} h g_0^{k+1}$$

For point $i = N-1$, we have equation (21).

Define the vectors:

$$\begin{aligned} U^{k+1} &= (u_1^{k+1}, u_2^{k+1}, \dots, u_{N-1}^{k+1})^T \\ U^0 &= (u_1^0, u_2^0, \dots, u_{N-1}^0)^T \\ \Phi^{k+1} &= (\phi_1^{k+1}, \phi_2^{k+1}, \dots, \phi_{N-1}^{k+1})^T \end{aligned}$$

Define the boundary conditions vector W^{k+1} :

$$W^{k+1} = \left(-a_1^{k+1} \omega_2 h g_0^{k+1}, -a_2^{k+1} \omega_3 h g_0^{k+1}, \dots, -a_{N-2}^{k+1} \omega_{N-1} h g_0^{k+1}, -a_{N-1}^{k+1} \omega_N h g_0^{k+1} + a_{N-1}^{k+1} \omega_0 \left(\frac{2}{h} g_1^{k+1} + h g_0^{k+1} \right) \right)^T$$

Define the matrix P^{k+1} with elements:

$$p_{i,j}^{k+1} = \begin{cases} A_1^{k+1} - a_1^{k+1}\omega_2 & \text{if } i = 1, j = 1 \\ -a_1^{k+1} & \text{if } i = 1, j = 2 \\ A_i^{k+1} & \text{if } 2 \leq i \leq N-2, j = i \\ -a_i^{k+1} & \text{if } 2 \leq i \leq N-2, j = i+1 \\ -a_i^{k+1}\omega_{i+1} & \text{if } 2 \leq i \leq N-2, j = 1 \\ -a_i^{k+1}\omega_{i-j+1} & \text{if } 2 \leq i \leq N-2, 2 \leq j \leq i-1 \\ a_{N-1}^{k+1}(3\omega_0 - \omega_N) & \text{if } i = N-1, j = 1 \\ a_{N-1}^{k+1}(2\omega_0 - \omega_{N-j}) & \text{if } i = N-1, 2 \leq j \leq N-2 \\ A_{N-1}^{k+1} + 2a_{N-1}^{k+1}\omega_0 & \text{if } i = N-1, j = N-1 \\ 0 & \text{otherwise} \end{cases}$$

The final matrix system:

$$P^{k+1}U^{k+1} = -\lambda \sum_{j=0}^{k-1} \sigma_j U^{k-j} + \lambda d_k U^0 + h^\beta \Phi^{k+1} + W^{k+1} \quad (22)$$

2.4 Solution of a Matrix System

Let ρ be an eigenvalue of P^k and $V = (x_1, \dots, x_{N-1})^T$ a corresponding eigenvector.

Choose an index i such that $|x_i| = \max_{1 \leq j \leq N-1} |x_j|$.

From the i -th equation of $P^k V = \rho V$, we have $(\rho - p_{ii})x_i = \sum_{j \neq i} p_{ij}x_j$. Taking absolute values and using $|x_j| \leq |x_i|$, we obtain

$$|\rho - p_{ii}| \leq \sum_{j \neq i} |p_{ij}| = R_i. \quad (23)$$

This is precisely the statement of the Gershgorin circle theorem. We now apply it to our matrix P^k .

Elements of the matrix P^k

For the first row ($i = 1$):

$$p_{1,1} = A_1^k - a_1^k \omega_2, \quad p_{1,2} = -a_1^k, \quad p_{1,j} = 0 \quad (j \geq 3)$$

For interior rows ($2 \leq i \leq N-2$):

$$\begin{aligned} p_{i,i} &= A_i^k \\ p_{i,i+1} &= -a_i^k \\ p_{i,1} &= -a_i^k \omega_{i+1} \\ p_{i,i-j+1} &= -a_i^k \omega_j \quad \text{for } j = 2, 3, \dots, i-1 \end{aligned}$$

For the last row ($i = N-1$):

$$\begin{aligned} p_{N-1,1} &= a_{N-1}^k (3\omega_0 - \omega_N) \\ p_{N-1,j} &= a_{N-1}^k (2\omega_0 - \omega_{N-j}) \quad \text{for } j = 2, 3, \dots, N-2 \\ p_{N-1,N-1} &= A_{N-1}^k + 2a_{N-1}^k \omega_0 \end{aligned}$$

where:

- $A_i^k = \lambda - a_i^k \omega_1 = \lambda + a_i^k \beta$ (since $\omega_1 = -\beta$)
- $\lambda > 0, a_i^k > 0$
- $\omega_0 = 1, \omega_1 = -\beta, \omega_2 = \frac{\beta(\beta-1)}{2} > 0$

Case : $i = 1$

$$p_{1,1} = A_1^k - a_1^k \omega_2 = \lambda + a_1^k \beta - a_1^k \omega_2, \quad p_{1,2} = -a_1^k, \quad p_{1,j} = 0 \quad (j \geq 3).$$

$$R_1 = |p_{1,2}| = a_1^k.$$

The condition $0 \notin D_1$ is $p_{1,1} > R_1$, i.e.,

$$\lambda + a_1^k \beta - a_1^k \omega_2 > a_1^k \iff \lambda > a_1^k (1 + \omega_2 - \beta). \quad (24)$$

Since $1 < \beta < 2$ and $\omega_2 > 0$, we have $1 + \omega_2 - \beta < 0$. The right-hand side is therefore negative, so the inequality holds automatically for any $\lambda > 0$. Hence $0 \notin D_1$.

Case : $2 \leq i \leq N-2$

$$p_{i,i} = A_i^k = \lambda + a_i^k \beta.$$

The off-diagonal entries are:

$$p_{i,i+1} = -a_i^k, \quad p_{i,1} = -a_i^k \omega_{i+1}, \quad p_{i,i-j+1} = -a_i^k \omega_j \quad (j = 2, \dots, i-1).$$

Thus,

$$R_i = a_i^k \left(1 + \omega_{i+1} + \sum_{j=2}^{i-1} \omega_j \right) = a_i^k \left(1 + \sum_{j=2}^i \omega_j \right).$$

The condition $0 \notin D_i$ is $p_{i,i} > R_i$, i.e.,

$$\begin{aligned} \lambda + a_i^k \beta &> a_i^k \left(1 + \sum_{j=2}^i \omega_j \right), \\ \lambda &> a_i^k \left(1 + \sum_{j=2}^i \omega_j - \beta \right). \end{aligned}$$

Recall that $\sum_{j=2}^{\infty} \omega_j = \beta - 1$. Since $\sum_{j=2}^i \omega_j \leq \beta - 1$ (all ω_j are positive), we have

$$1 + \sum_{j=2}^i \omega_j - \beta \leq 1 + (\beta - 1) - \beta = 0. \quad (25)$$

The right-hand side is non-positive, so the inequality holds automatically for any $\lambda > 0$. Hence $0 \notin D_i$ for all interior rows.

Case : $i = N - 1$

$$p_{N-1,N-1} = \lambda + a_{N-1}^k(\beta + 2).$$

$$R_{N-1} = a_{N-1}^k(3 - \omega_N) + a_{N-1}^k \sum_{j=2}^{N-2} (2 - \omega_{N-j}).$$

The condition $0 \notin D_{N-1}$ is $p_{N-1,N-1} > R_{N-1}$, i.e.,

$$\lambda + a_{N-1}^k(\beta + 2) > a_{N-1}^k \left((3 - \omega_N) + \sum_{j=2}^{N-2} (2 - \omega_{N-j}) \right),$$

$$\lambda > a_{N-1}^k \left((3 - \omega_N) + \sum_{j=2}^{N-2} (2 - \omega_{N-j}) - (\beta + 2) \right). \quad (26)$$

We estimate the term in parentheses:

$$(3 - \omega_N) + \sum_{j=2}^{N-2} (2 - \omega_{N-j}) = 3 - \omega_N + 2(N - 3) - \sum_{j=2}^{N-2} \omega_{N-j}.$$

$$= 2N - 3 - \omega_N - \sum_{m=2}^{N-2} \omega_m.$$

Since $\omega_m > 0$, we have $\sum_{m=2}^{N-2} \omega_m \geq 0$, hence

$$(3 - \omega_N) + \sum_{j=2}^{N-2} (2 - \omega_{N-j}) \leq 2N - 3 - \omega_N \leq 2N - 3.$$

Thus,

$$(3 - \omega_N) + \sum_{j=2}^{N-2} (2 - \omega_{N-j}) - (\beta + 2) \leq 2N - 5 - \beta. \quad (27)$$

Consequently, a sufficient condition for (26) to hold is

$$\boxed{\lambda > a_{\max} \cdot (2N - 5 - \beta)}, \quad (28)$$

where $a_{\max} = \max_{i,k} a_i^{k+1}$.

Recall that

$$\lambda = \frac{h_t^{-\alpha} h^\beta}{\Gamma(2 - \alpha)}.$$

The condition $\lambda > a_{\max}(2N - 5 - \beta)$ can be satisfied by choosing h_t sufficiently small (since $\lambda \propto h_t^{-\alpha}$ with $\alpha > 0$). Such a choice is always possible and does not harm the study.

Conclusion

Under the condition that Δt is sufficiently small, all Gershgorin discs of the matrix P^k do not contain the origin 0. Therefore, 0 is not an eigenvalue of P^k and the matrix is invertible.

$$\boxed{P^k \text{ is invertible for } \Delta t \text{ sufficiently small.}}$$

2.5 Consistency Analysis

Let $u(x, t)$ be the exact solution of the original fractional differential equation:

$$\frac{\partial^\alpha u}{\partial t^\alpha}(x, t) = a(x, t) \frac{\partial^\beta u}{\partial x^\beta}(x, t) + \phi(x, t), \quad (x, t) \in (0, L) \times (0, T]$$

where $0 < \alpha < 1$ and $1 < \beta < 2$.

We define the **local truncation error** at the point (x_i, t_{k+1}) as the difference between the numerical scheme (7) and the exact solution substituted into this scheme :

$$\tau_i^{k+1} = \mu \sum_{j=0}^k (u(x_i, t_{k-j+1}) - u(x_i, t_{k-j})) d_j - a_i^{k+1} h^{-\beta} \sum_{j=0}^{i+1} \omega_j u(x_{i-j+1}, t_{k+1}) - \phi_i^{k+1} \quad (29)$$

where: $u(x_i, t_k)$ represents the exact solution at the point (x_i, t_k)

Error Estimate of the L1 Method : Let $\frac{\partial^\alpha u}{\partial t^\alpha}$ be the Caputo derivative of order α ($0 < \alpha < 1$). Then the L1 approximation gives the following relation :

$$\frac{\partial^\alpha u}{\partial t^\alpha}(x_i, t_{k+1}) = \mu \sum_{j=0}^k (u(x_i, t_{k-j+1}) - u(x_i, t_{k-j})) d_j + \varepsilon_t \quad (30)$$

where ε_t represents the approximation error and is estimated by:

$$|\varepsilon_t| \leq C_t h_t$$

with C_t a constant independent of h_t .

proof

The truncation error ε_t is defined as the difference between the exact derivative and its approximation:

$$\varepsilon_t = \frac{\partial^\alpha u}{\partial t^\alpha}(x_i, t_{k+1}) - \mu \sum_{j=0}^k (u_i^{k-j+1} - u_i^{k-j}) d_j.$$

We now express the exact derivative in a form similar to the approximation :

$$\frac{\partial^\alpha u}{\partial t^\alpha}(x_i, t_{k+1}) = \frac{1}{\Gamma(1-\alpha)} \sum_{j=0}^k \int_{t_j}^{t_{j+1}} \frac{\partial u(x_i, \tau)}{\partial \tau} (t_{k+1} - \tau)^{-\alpha} d\tau.$$

Applying the mean value theorem for integrals to each subinterval, there exists $\xi_j \in (t_j, t_{j+1})$ such that:

$$\int_{t_j}^{t_{j+1}} \frac{\partial u(x_i, \tau)}{\partial \tau} (t_{k+1} - \tau)^{-\alpha} d\tau = \frac{\partial u(x_i, \xi_j)}{\partial \tau} \int_{t_j}^{t_{j+1}} (t_{k+1} - \tau)^{-\alpha} d\tau.$$

This is valid because the function $(t_{k+1} - \tau)^{-\alpha}$ does not change sign on the interval (it is positive).

Subtracting the approximation from the exact derivative gives :

$$\varepsilon_t = \frac{1}{\Gamma(1-\alpha)} \sum_{j=0}^k \left[\frac{\partial u(x_i, \xi_j)}{\partial \tau} - \frac{u_i^{j+1} - u_i^j}{h_t} \right] \int_{t_j}^{t_{j+1}} (t_{k+1} - \tau)^{-\alpha} d\tau.$$

To estimate the difference between the derivatives, we use Taylor expansions. First, expanding u_i^{j+1} around t_j :

$$u_i^{j+1} = u_i^j + h_t \frac{\partial u(x_i, t_j)}{\partial t} + \frac{h_t^2}{2} \frac{\partial^2 u(x_i, \eta_j)}{\partial t^2},$$

where $\eta_j \in (t_j, t_{j+1})$. From this, we obtain:

$$\frac{u_i^{j+1} - u_i^j}{h_t} = \frac{\partial u(x_i, t_j)}{\partial t} + \frac{h_t}{2} \frac{\partial^2 u(x_i, \eta_j)}{\partial t^2}.$$

Similarly, expanding $\frac{\partial u(x_i, \xi_j)}{\partial t}$ around t_j yields:

$$\frac{\partial u(x_i, \xi_j)}{\partial t} = \frac{\partial u(x_i, t_j)}{\partial t} + (\xi_j - t_j) \frac{\partial^2 u(x_i, \zeta_j)}{\partial t^2},$$

with $\zeta_j \in (t_j, \xi_j)$. Taking the absolute difference and using the triangle inequality, we get:

$$\left| \frac{\partial u(x_i, \xi_j)}{\partial t} - \frac{u_i^{j+1} - u_i^j}{h_t} \right| \leq \left(|\xi_j - t_j| + \frac{h_t}{2} \right) \max_{[t_j, t_{j+1}]} \left| \frac{\partial^2 u}{\partial t^2} \right|.$$

Since $\xi_j \in (t_j, t_{j+1})$, we have $|\xi_j - t_j| \leq h_t$. Let $M_2 = \max_{0 \leq \tau \leq t_{k+1}} \left| \frac{\partial^2 u}{\partial t^2}(x_i, \tau) \right|$. Then:

$$\left| \frac{\partial u(x_i, \xi_j)}{\partial t} - \frac{u_i^{j+1} - u_i^j}{h_t} \right| \leq \frac{3h_t}{2} M_2.$$

The integral in each subinterval can be bounded as:

$$\int_{t_j}^{t_{j+1}} (t_{k+1} - \tau)^{-\alpha} d\tau = \frac{h_t^{1-\alpha}}{1-\alpha} [(k+1-j)^{1-\alpha} - (k-j)^{1-\alpha}].$$

Substituting these estimates into the expression for ε_t gives:

$$|\varepsilon_t| \leq \frac{1}{\Gamma(1-\alpha)} \sum_{j=0}^k \frac{3h_t}{2} M_2 \cdot \frac{h_t^{1-\alpha}}{1-\alpha} [(k+1-j)^{1-\alpha} - (k-j)^{1-\alpha}].$$

Simplifying the constants:

$$|\varepsilon_t| \leq \frac{3M_2}{2\Gamma(1-\alpha)(1-\alpha)} h_t^{2-\alpha} \sum_{j=0}^k [(k+1-j)^{1-\alpha} - (k-j)^{1-\alpha}].$$

The sum is telescoping:

$$\sum_{j=0}^k [(k+1-j)^{1-\alpha} - (k-j)^{1-\alpha}] = (k+1)^{1-\alpha} - 0^{1-\alpha} = (k+1)^{1-\alpha}$$

since $1-\alpha > 0$ implies $0^{1-\alpha} = 0$.

Therefore:

$$|\varepsilon_t| \leq \frac{3M_2}{2\Gamma(1-\alpha)(1-\alpha)} h_t^{2-\alpha} (k+1)^{1-\alpha}.$$

Using the relation $t_{k+1} = (k+1)h_t$, we have $(k+1)^{1-\alpha} = t_{k+1}^{1-\alpha} h_t^{\alpha-1}$. Substituting:

$$|\varepsilon_t| \leq \frac{3M_2}{2\Gamma(1-\alpha)(1-\alpha)} h_t^{2-\alpha} \cdot t_{k+1}^{1-\alpha} h_t^{\alpha-1} = \frac{3M_2}{2\Gamma(1-\alpha)(1-\alpha)} t_{k+1}^{1-\alpha} h_t.$$

Finally, if T is the final time, then $t_{k+1} \leq T$ and we obtain the desired estimate:

$$|\varepsilon_t| \leq C_t h_t,$$

where

$$C_t = \frac{3M_2 T^{1-\alpha}}{2\Gamma(1-\alpha)(1-\alpha)}.$$

This completes the proof that the truncation error of the L1 approximation is of first order in time.

Error Estimate of the Shifted Grünwald-Letnikov Approximation

Let $\frac{\partial^\beta u}{\partial x^\beta}$ be the Riemann-Liouville derivative of order β ($1 < \beta < 2$). The shifted Grünwald-Letnikov approximation is defined as

$$\mathcal{D}_h^\beta u(x) = h^{-\beta} \sum_{j=0}^{\infty} \omega_j^{(\beta)} u(x - (j-1)h), \quad (31)$$

where the Grünwald-Letnikov coefficients are given by (see P3)

$$\omega_j^{(\beta)} = (-1)^j \binom{\beta}{j} = \frac{\Gamma(j-\beta)}{\Gamma(-\beta)\Gamma(j+1)}, \quad j = 0, 1, 2, \dots \quad (32)$$

These coefficients satisfy the generating function (see P1)

$$\sum_{j=0}^{\infty} \omega_j^{(\beta)} z^j = (1-z)^\beta, \quad |z| \leq 1. \quad (33)$$

Theorem. For a sufficiently smooth function u that decays at infinity (or is extended by zero outside $[0, 1]$), the shifted Grünwald-Letnikov approximation satisfies

$$\mathcal{D}_h^\beta u(x) = {}_{RL}D_x^\beta u(x) + \varepsilon_x, \quad (34)$$

with the truncation error estimate

$$|\varepsilon_x| \leq C_x h, \quad (35)$$

where C_x is a constant independent of h .

Proof (following [11]).

Fourier transform. We extend u by zero outside the interval $[0, 1]$ so that its Fourier transform is well-defined on $\mathbb{R}$. Let $\hat{u}(\xi) = \int_{-\infty}^{\infty} u(x)e^{-i\xi x}dx$ be the Fourier transform of u .

Using the shift property $\mathcal{F}\{u(x-a)\}(\xi) = e^{-i\xi a}\hat{u}(\xi)$, we obtain

$$\mathcal{F}\{u(x - (j-1)h)\}(\xi) = e^{-i\xi(j-1)h}\hat{u}(\xi).$$

Taking the Fourier transform of (31) and using the linearity of the Fourier transform, we get

$$\widehat{\mathcal{D}_h^\beta u}(\xi) = h^{-\beta} \sum_{j=0}^{\infty} \omega_j e^{-i\xi(j-1)h} \hat{u}(\xi).$$

Since $\hat{u}(\xi)$ does not depend on the summation index j , it can be factored out:

$$\widehat{\mathcal{D}_h^\beta u}(\xi) = h^{-\beta} \left(\sum_{j=0}^{\infty} \omega_j e^{-i\xi(j-1)h} \right) \hat{u}(\xi).$$

We separate the factor $e^{i\xi h}$ from the sum:

$$\sum_{j=0}^{\infty} \omega_j e^{-i\xi(j-1)h} = e^{i\xi h} \sum_{j=0}^{\infty} \omega_j e^{-i\xi j h}.$$

Now, using the generating function (33) with $z = e^{-i\xi h}$, we obtain

$$\sum_{j=0}^{\infty} \omega_j e^{-i\xi j h} = (1 - e^{-i\xi h})^\beta.$$

Therefore, the Fourier transform of the shifted GL approximation is

$$\widehat{\mathcal{D}_h^\beta u}(\xi) = h^{-\beta} e^{i\xi h} (1 - e^{-i\xi h})^\beta \hat{u}(\xi). \quad (36)$$

Fourier transform of the exact derivative. The Fourier transform of the Riemann-Liouville derivative is (see [16])

$$\widehat{{}_{RL}D_x^\beta u}(\xi) = (i\xi)^\beta \hat{u}(\xi). \quad (37)$$

Expansion of $1 - e^{-i\xi h}$ for small h . We expand the exponential function $e^{-i\xi h}$ in Taylor series around $h = 0$:

$$e^{-i\xi h} = 1 - i\xi h - \frac{(\xi h)^2}{2} + i \frac{(\xi h)^3}{6} + O(h^4).$$

Thus,

$$1 - e^{-i\xi h} = i\xi h + \frac{(\xi h)^2}{2} - i \frac{(\xi h)^3}{6} + O(h^4).$$

Factor $i\xi h$:

$$1 - e^{-i\xi h} = i\xi h \left[1 - \frac{i\xi h}{2} + O(h^2) \right].$$

Raising to the power β .

$$(1 - e^{-i\xi h})^\beta = (i\xi h)^\beta \left[1 - \frac{i\xi h}{2} + O(h^2) \right]^\beta.$$

Using the generalized binomial expansion $(1 + X)^\beta = 1 + \beta X + \frac{\beta(\beta-1)}{2} X^2 + \dots$, valid for small X , with $X = -\frac{i\xi h}{2} + O(h^2)$, we get

$$\left[1 - \frac{i\xi h}{2} + O(h^2) \right]^\beta = 1 - \frac{\beta i\xi h}{2} + O(h^2).$$

Therefore,

$$(1 - e^{-i\xi h})^\beta = (i\xi h)^\beta \left(1 - \frac{\beta i\xi h}{2} + O(h^2) \right). \quad (38)$$

Substituting into the Fourier transform. Substitute (38) into (36):

$$\widehat{\mathcal{D}_h^\beta u}(\xi) = h^{-\beta} e^{i\xi h} (i\xi h)^\beta \left(1 - \frac{\beta i\xi h}{2} + O(h^2) \right) \hat{u}(\xi).$$

Simplify $h^{-\beta} (i\xi h)^\beta = (i\xi)^\beta$:

$$\widehat{\mathcal{D}_h^\beta u}(\xi) = (i\xi)^\beta e^{i\xi h} \left(1 - \frac{\beta i\xi h}{2} + O(h^2) \right) \hat{u}(\xi). \quad (39)$$

Expanding $e^{i\xi h}$. Expand $e^{i\xi h}$ in Taylor series:

$$e^{i\xi h} = 1 + i\xi h + O(h^2).$$

Substitute this into (39):

$$\widehat{\mathcal{D}_h^\beta u}(\xi) = (i\xi)^\beta (1 + i\xi h + O(h^2)) \left(1 - \frac{\beta i\xi h}{2} + O(h^2) \right) \hat{u}(\xi).$$

Multiply the two parentheses, keeping terms up to $O(h)$:

$$(1 + i\xi h) \left(1 - \frac{\beta i\xi h}{2} \right) = 1 + i\xi h - \frac{\beta i\xi h}{2} + O(h^2) = 1 + i\xi h \left(1 - \frac{\beta}{2} \right) + O(h^2).$$

Thus,

$$\widehat{\mathcal{D}_h^\beta u}(\xi) = (i\xi)^\beta \left(1 + i\xi h \left(1 - \frac{\beta}{2} \right) + O(h^2) \right) \hat{u}(\xi). \quad (40)$$

Computing the difference between the symbols. The Fourier transform of the exact Riemann-Liouville derivative is given by (37). Subtracting (37) from (40), we obtain

$$\widehat{\mathcal{D}_h^\beta u}(\xi) - \widehat{{}_{RL}D_x^\beta u}(\xi) = (i\xi)^\beta \cdot i\xi h \left(1 - \frac{\beta}{2}\right) \hat{u}(\xi) + O(h^2).$$

Hence,

$$\widehat{\mathcal{D}_h^\beta u}(\xi) - \widehat{{}_{RL}D_x^\beta u}(\xi) = h \left(1 - \frac{\beta}{2}\right) (i\xi)^{\beta+1} \hat{u}(\xi) + O(h^2). \quad (41)$$

Returning to the spatial domain. Apply the inverse Fourier transform to (41). Since the inverse Fourier transform is a linear and continuous operator on $L^2(\mathbb{R})$, we obtain

$$\mathcal{D}_h^\beta u(x) - {}_{RL}D_x^\beta u(x) = h \left(1 - \frac{\beta}{2}\right) \cdot {}_{RL}D_x^{\beta+1} u(x) + O(h^2). \quad (42)$$

Estimate of the truncation error ε_x . From (42), taking absolute values and using the boundedness of the operator ${}_{RL}D_x^{\beta+1}u$, we get

$$|\mathcal{D}_h^\beta u(x) - {}_{RL}D_x^\beta u(x)| \leq h \left|1 - \frac{\beta}{2}\right| \cdot \|{}_{RL}D_x^{\beta+1}u\|_\infty + C_2 h^2.$$

For sufficiently small h (e.g., $0 < h \leq 1$), we have $h^2 \leq h$. Therefore,

$$|\varepsilon_x| \leq h \left|1 - \frac{\beta}{2}\right| \cdot \|{}_{RL}D_x^{\beta+1}u\|_\infty + C_2 h^2 \leq h \left(\left|1 - \frac{\beta}{2}\right| \cdot \|{}_{RL}D_x^{\beta+1}u\|_\infty + C_2 \right).$$

Define the constant

$$C_x = \left|1 - \frac{\beta}{2}\right| \cdot \|{}_{RL}D_x^{\beta+1}u\|_\infty + C_2,$$

which is independent of h . Consequently,

$$|\varepsilon_x| \leq C_x h.$$

This completes the proof.

Summary :

The shifted Grünwald-Letnikov approximation achieves first-order accuracy $O(h)$ for the Riemann-Liouville fractional derivative of order $\beta \in (1, 2)$. This result justifies the statement that ε_x is estimated by $|\varepsilon_x| \leq C_x h$.

We now substitute the exact solution u into the numerical scheme. Using (30) and (34), we express the numerical approximations in terms of the exact derivatives:

$$\tau_i^{k+1} = \left(\frac{\partial^\alpha u}{\partial t^\alpha}(x_i, t_{k+1}) - \varepsilon_t \right) - a_i^{k+1} \left(\frac{\partial^\beta u}{\partial x^\beta}(x_i, t_{k+1}) - \varepsilon_x \right) - \psi_i^{k+1}.$$

Rearranging the terms gives:

$$\tau_i^{k+1} = \left(\frac{\partial^\alpha u}{\partial t^\alpha}(x_i, t_{k+1}) - a_i^{k+1} \frac{\partial^\beta u}{\partial x^\beta}(x_i, t_{k+1}) - \psi_i^{k+1} \right) - \varepsilon_t + a_i^{k+1} \varepsilon_x.$$

The exact solution u satisfies the original differential equation, therefore the expression in parentheses vanishes:

$$\frac{\partial^\alpha u}{\partial t^\alpha}(x_i, t_{k+1}) - a_i^{k+1} \frac{\partial^\beta u}{\partial x^\beta}(x_i, t_{k+1}) - \psi_i^{k+1} = 0.$$

Thus we are left with:

$$\tau_i^{k+1} = -\varepsilon_t + a_i^{k+1} \varepsilon_x \quad (43)$$

From the estimates in (30) and (34) we have:

$$|\varepsilon_t| \leq C_t h_t, \quad |\varepsilon_x| \leq C_x h.$$

Since the function $a(x, t)$ is continuous on the closed domain $[0, L] \times [0, T]$, it is bounded. Let

$$A = \max_{(x,t) \in [0,L] \times [0,T]} |a(x, t)|.$$

Applying the triangle inequality to (43) yields:

$$|\tau_i^{k+1}| \leq |\varepsilon_t| + A|\varepsilon_x| \leq C_t h_t + AC_x h.$$

Setting $C = \max(C_t, AC_x)$, we obtain the final estimate:

$$|\tau_i^{k+1}| \leq C (h_t + h) \quad (44)$$

where the constant C is independent of h_t , h , and the indices i, k . From (44) we conclude that

$$\lim_{h_t, h \rightarrow 0} \tau_i^{k+1} = 0,$$

which means that **the numerical scheme is consistent** with the original fractional differential equation. Moreover, the truncation error is of **first order** in both time and space:

$$\tau_i^{k+1} = O(h_t) + O(h).$$

2.6 Stability Analysis

Let v_i^k be the numerical solution obtained from the scheme (18) with exact initial data.

Let $\tilde{v}_i^k$ be the numerical solution obtained from the same scheme but with a perturbed initial condition:

$$\tilde{v}_i^0 = v_i^0 + \varepsilon_i^0,$$

where ε_i^0 is a small perturbation.

We define the perturbation error at point (x_i, t_k) as:

$$e_i^k = \tilde{v}_i^k - v_i^k.$$

The infinity norm of the perturbation error is defined as:

$$\|E^k\|_\infty = \max_{1 \leq i \leq N-1} |e_i^k|, \quad \text{where } E^k = (e_1^k, e_2^k, \dots, e_{N-1}^k)^T.$$

For the first time step $k = 0$, the scheme (18) applied to the unperturbed solution v gives:

$$A_i^1 v_i^1 - a_i^1 \sum_{\substack{j=0 \\ j \neq 1}}^i \omega_j v_{i-j+1}^1 - a_i^1 \omega_{i+1} v_1^1 = \lambda v_i^0,$$

where $A_i^1 = \lambda - a_i^1 \omega_1$.

The same scheme applied to the perturbed solution $\tilde{v}$ gives:

$$A_i^1 \tilde{v}_i^1 - a_i^1 \sum_{\substack{j=0 \\ j \neq 1}}^i \omega_j \tilde{v}_{i-j+1}^1 - a_i^1 \omega_{i+1} \tilde{v}_1^1 = \lambda \tilde{v}_i^0.$$

Subtracting the first equation from the second, we obtain the error equation for the first time step:

$$A_i^1 e_i^1 - a_i^1 \sum_{\substack{j=0 \\ j \neq 1}}^i \omega_j e_{i-j+1}^1 - a_i^1 \omega_{i+1} e_1^1 = \lambda e_i^0. \quad (45)$$

Substituting $A_i^1 = \lambda - a_i^1 \omega_1$ into equation (45):

$$\begin{aligned} (\lambda - a_i^1 \omega_1) e_i^1 - a_i^1 \sum_{\substack{j=0 \\ j \neq 1}}^i \omega_j e_{i-j+1}^1 - a_i^1 \omega_{i+1} e_1^1 &= \lambda e_i^0 \\ \lambda e_i^1 - a_i^1 \left(\omega_1 e_i^1 + \sum_{\substack{j=0 \\ j \neq 1}}^i \omega_j e_{i-j+1}^1 + \omega_{i+1} e_1^1 \right) &= \lambda e_i^0 \\ \lambda e_i^1 - a_i^1 \left(\sum_{j=0}^i \omega_j e_{i-j+1}^1 + \omega_{i+1} e_1^1 \right) &= \lambda e_i^0 \end{aligned} \quad (46)$$

Equation (46) is the fundamental error equation for the stability analysis. It describes how the initial perturbation e_i^0 propagates to the first time step.

Taking the absolute value of both sides:

$$\left| \lambda e_i^1 - a_i^1 \left(\sum_{j=0}^i \omega_j e_{i-j+1}^1 + \omega_{i+1} e_1^1 \right) \right| = \lambda |e_i^0|$$

Using the reverse triangle inequality:

$$\lambda|e_i^1| - a_i^1 \left(\sum_{j=0}^i |\omega_j| |e_{i-j+1}^1| + |\omega_{i+1}| |e_1^1| \right) \leq \lambda|e_i^0| \quad (47)$$

We choose i such that $|e_i^1| = \|E^1\|_\infty$. Since $|e_{i-j+1}^1| \leq \|E^1\|_\infty$ and $|e_1^1| \leq \|E^1\|_\infty$, we obtain from (47):

$$\begin{aligned} \lambda \|E^1\|_\infty - a_i^1 \left(\sum_{j=0}^i |\omega_j| \|E^1\|_\infty + |\omega_{i+1}| \|E^1\|_\infty \right) &\leq \lambda \|E^0\|_\infty \\ \left(\lambda - a_i^1 \left(\sum_{j=0}^i |\omega_j| + |\omega_{i+1}| \right) \right) \|E^1\|_\infty &\leq \lambda \|E^0\|_\infty \end{aligned} \quad (48)$$

From the properties of Grünwald coefficients, we have: (see P7)

$$\sum_{j=0}^i \omega_j + \omega_{i+1} < 0 \quad \text{for all } i$$

For $1 < \beta < 2$, the Grünwald coefficients satisfy $|\omega_j| \geq |\omega_{j+1}|$ for $j \geq 1$, and the series $\sum_{j=0}^\infty |\omega_j|$ converges . Hence :

$$\sum_{j=0}^i |\omega_j| + |\omega_{i+1}| \leq C$$

where C is a constant. However, for stability, we need the coefficient $\lambda - a_i^1 (\sum_{j=0}^i |\omega_j| + |\omega_{i+1}|)$ to be positive.

We assume that λ is sufficiently large such that :

$$\lambda \geq a_i^1 \left(\sum_{j=0}^i |\omega_j| + |\omega_{i+1}| \right) \quad \text{for all } i \quad (49)$$

Recall that from the numerical scheme, λ is defined as :

$$\lambda = \mu h^\beta, \quad \mu = \frac{1}{\Gamma(2-\alpha)(h_t)^\alpha}.$$

Substituting this into condition (49) gives :

$$\frac{h^\beta}{\Gamma(2-\alpha)(h_t)^\alpha} \geq a_i^1 \left(\sum_{j=0}^i |\omega_j| + |\omega_{i+1}| \right) \quad \text{for all } i.$$

Using the estimate $\sum_{j=0}^i |\omega_j| + |\omega_{i+1}| \leq 2\beta$ and taking the maximum over i of $|a_i^1|$, we obtain :

$$\frac{h^\beta}{\Gamma(2-\alpha)(\Delta t)^\alpha} \geq A \cdot 2\beta, \quad A = \max_i |a_i^1|.$$

Solving for h_t :

$$(h_t)^\alpha \leq \frac{h^\beta}{2^\beta A \Gamma(2 - \alpha)}.$$

Hence, we must have :

$$\boxed{h_t \leq C_{\text{stab}} \cdot h^{\beta/\alpha}}, \quad C_{\text{stab}} = \left(\frac{1}{2^\beta A \Gamma(2 - \alpha)} \right)^{1/\alpha}.$$

From (48) and under condition (49), we have :

$$\left(\lambda - a_i^1 \left(\sum_{j=0}^i |\omega_j| + |\omega_{i+1}| \right) \right) \|E^1\|_\infty \leq \lambda \|E^0\|_\infty$$

Since the coefficient is positive, we can divide by it :

$$\|E^1\|_\infty \leq \frac{\lambda}{\lambda - a_i^1 \left(\sum_{j=0}^i |\omega_j| + |\omega_{i+1}| \right)} \|E^0\|_\infty$$

If λ is sufficiently large, the denominator approaches λ , thus :

$$\|E^1\|_\infty \leq K \|E^0\|_\infty$$

where K is a constant close to 1. In the limiting case where λ is sufficiently large, we obtain:

$$\|E^1\|_\infty \leq \|E^0\|_\infty \tag{50}$$

From (50) we conclude that:

$$|e^1| \leq |e^0|$$

where $|e^k| = \|E^k\|_\infty$. This means that the error does not grow when moving from time step $t = 0$ to t^1 .

Therefore, the method is **conditionally stable** provided that condition (49) holds .

Lemma 1

For $k \geq 1$ the scheme (18) is stable and we have

$$|e^{k+1}| \leq C |e^0|, \quad C > 0, \text{ for all } k \geq 1$$

Proof. We use mathematical induction.

Assume that $|e^j| \leq c_j |e^0|$ for $j = 1, 2, \dots, k$ where $c_j > 0$.

From the equation (18) for the error, we have:

$$A_i^{k+1}e_i^{k+1} - a_i^{k+1} \left(\sum_{\substack{j=0 \\ j \neq 1}}^i \omega_j e_{i-j+1}^{k+1} + \omega_{i+1} e_1^{k+1} \right) = -\lambda \sum_{j=0}^{k-1} \sigma_j e_i^{k-j} + \lambda d_k e_i^0 \quad (51)$$

where $A_i^{k+1} = \lambda - a_i^{k+1} \omega_1$ and $\sigma_j = d_{j+1} - d_j$.

Let i be such that $|e_i^{k+1}| = |e^{k+1}| = \max_{1 \leq m \leq N-1} |e_m^{k+1}|$.

From equation (51), we take absolute values and use the triangle inequality:

$$\left| A_i^{k+1} e_i^{k+1} - a_i^{k+1} \left(\sum_{\substack{j=0 \\ j \neq 1}}^i \omega_j e_{i-j+1}^{k+1} + \omega_{i+1} e_1^{k+1} \right) \right| = \left| -\lambda \sum_{j=0}^{k-1} \sigma_j e_i^{k-j} + \lambda d_k e_i^0 \right|$$

Using the reverse triangle inequality on the left side:

$$A_i^{k+1} |e_i^{k+1}| - a_i^{k+1} \left(\sum_{\substack{j=0 \\ j \neq 1}}^i |\omega_j| |e_{i-j+1}^{k+1}| + |\omega_{i+1}| |e_1^{k+1}| \right) \leq \lambda \sum_{j=0}^{k-1} |\sigma_j| |e_i^{k-j}| + \lambda d_k |e_i^0|$$

Since $|e_{i-j+1}^{k+1}| \leq |e^{k+1}|$, $|e_1^{k+1}| \leq |e^{k+1}|$, and $|e_i^{k-j}| \leq |e^{k-j}|$, we obtain:

$$\begin{aligned} A_i^{k+1} |e^{k+1}| - a_i^{k+1} \left(\sum_{\substack{j=0 \\ j \neq 1}}^i |\omega_j| + |\omega_{i+1}| \right) |e^{k+1}| &\leq \lambda \sum_{j=0}^{k-1} |\sigma_j| |e^{k-j}| + \lambda d_k |e^0| \\ \left(A_i^{k+1} - a_i^{k+1} \left(\sum_{\substack{j=0 \\ j \neq 1}}^i |\omega_j| + |\omega_{i+1}| \right) \right) |e^{k+1}| &\leq \lambda \sum_{j=0}^{k-1} |\sigma_j| |e^{k-j}| + \lambda d_k |e^0| \end{aligned} \quad (52)$$

From the induction hypothesis, $|e^{k-j}| \leq c_{k-j} |e^0|$ for $j = 0, 1, \dots, k-1$. Substituting into (52):

$$\left(A_i^{k+1} - a_i^{k+1} \left(\sum_{\substack{j=0 \\ j \neq 1}}^i |\omega_j| + |\omega_{i+1}| \right) \right) |e^{k+1}| \leq \lambda \left(\sum_{j=0}^{k-1} |\sigma_j| c_{k-j} + d_k \right) |e^0|$$

Let $M_k = \max_{0 \leq j \leq k-1} c_{k-j}$. Then:

$$\left(A_i^{k+1} - a_i^{k+1} \left(\sum_{\substack{j=0 \\ j \neq 1}}^i |\omega_j| + |\omega_{i+1}| \right) \right) |e^{k+1}| \leq \lambda \left(M_k \sum_{j=0}^{k-1} |\sigma_j| + d_k \right) |e^0| \quad (53)$$

Recall the signs of Grünwald coefficients for $1 < \beta < 2$: (see P4 and P6)

- $\omega_0 = 1 > 0$

- $\omega_1 = -\beta < 0$
- $\omega_j > 0$ for all $j \geq 2$

Therefore:

$$\sum_{\substack{j=0 \\ j \neq 1}}^i |\omega_j| + |\omega_{i+1}| = \omega_0 + \sum_{j=2}^i \omega_j + \omega_{i+1} = 1 + \sum_{j=2}^i \omega_j + \omega_{i+1}$$

From the properties of Grünwald coefficients, we have $\sum_{j=2}^{\infty} \omega_j = \beta - 1$. Hence:

$$\sum_{j=2}^i \omega_j + \omega_{i+1} \leq \beta - 1$$

Thus:

$$\sum_{\substack{j=0 \\ j \neq 1}}^i |\omega_j| + |\omega_{i+1}| \leq 1 + (\beta - 1) = \beta$$

Consequently:

$$A_i^{k+1} - a_i^{k+1} \left(\sum_{\substack{j=0 \\ j \neq 1}}^i |\omega_j| + |\omega_{i+1}| \right) \geq \lambda + a_i^{k+1} \beta - a_i^{k+1} \beta = \lambda$$

Therefore, from (53):

$$\lambda |e^{k+1}| \leq \left(A_i^{k+1} - a_i^{k+1} \left(\sum_{\substack{j=0 \\ j \neq 1}}^i |\omega_j| + |\omega_{i+1}| \right) \right) |e^{k+1}| \leq \lambda \left(M_k \sum_{j=0}^{k-1} |\sigma_j| + d_k \right) |e^0|$$

Dividing both sides by $\lambda > 0$:

$$|e^{k+1}| \leq \left(M_k \sum_{j=0}^{k-1} |\sigma_j| + d_k \right) |e^0| \quad (54)$$

Recall that $\sigma_j = d_{j+1} - d_j$ and $d_j = (j+1)^{1-\alpha} - j^{1-\alpha}$ with $d_0 = 1$.

Since d_j is positive and decreasing, we have $d_{j+1} - d_j < 0$, so $|\sigma_j| = d_j - d_{j+1}$.

Thus:

$$\sum_{j=0}^{k-1} |\sigma_j| = \sum_{j=0}^{k-1} (d_j - d_{j+1}) = d_0 - d_k = 1 - d_k$$

Also, $d_k \leq d_0 = 1$, so $1 - d_k \geq 0$.

Substituting into (54):

$$|e^{k+1}| \leq (M_k(1 - d_k) + d_k) |e^0| = (M_k + d_k(1 - M_k)) |e^0|$$

Since $0 < d_k < 1$, we have:

$$|e^{k+1}| \leq \max(M_k, 1)|e^0|$$

Let $C = \max_{1 \leq j \leq k} c_j$. Then $M_k = \max_{0 \leq j \leq k-1} c_{k-j} \leq C$.

Therefore:

$$|e^{k+1}| \leq (C + 1)|e^0| \quad \text{or simply} \quad |e^{k+1}| \leq \tilde{C}|e^0|$$

where $\tilde{C}$ is a constant independent of k .

By mathematical induction, we have shown that for all $k \geq 1$:

$$|e^{k+1}| \leq C|e^0|, \quad C > 0$$

This proves that the method is stable. Since the constant C may depend on the choice of Δt and Δx , the method is **conditionally stable**.

2.7 convergence Analysis

Lemma 2 : The Caputo fractional derivative of order α ($0 < \alpha < 1$) using the L1 method satisfies:

$$\begin{aligned} \frac{\partial^\alpha u}{\partial t^\alpha} &= \left(\frac{\partial^\alpha u}{\partial t^\alpha} \right)_{L_1} + O(h_t) \\ \left| \frac{\partial^\alpha u}{\partial t^\alpha} - \left(\frac{\partial^\alpha u}{\partial t^\alpha} \right)_{L_1} \right| &\leq C_t h_t \end{aligned}$$

Lemma 3 : The Riemann-Liouville fractional derivative of order β ($1 < \beta < 2$) using the Grünwald-Letnikov method satisfies:

$$\begin{aligned} \frac{\partial^\beta u}{\partial x^\beta} &= h^{-\beta} \sum_{j=0}^{i+1} \omega_j u_{i-j+1}^{k+1} + O(h) \\ \left| \frac{\partial^\beta u}{\partial x^\beta} - h^{-\beta} \sum_{j=0}^{i+1} \omega_j u_{i-j+1}^{k+1} \right| &\leq C_x h \end{aligned}$$

Proposition 2

The numerical scheme (7) is convergent and satisfies

$$\|E^{k+1}\|_\infty \leq C\|E^0\|_\infty + O(h_t + h)$$

The Proof

Let u_i^{k+1} be the exact solution of problem (1) at point (x_i, t_{k+1}) . Let v_i^{k+1} be the numerical solution obtained from scheme (7).

We define the error:

$$e_i^{k+1} = u_i^{k+1} - v_i^{k+1}$$

We define the error vector:

$$E^{k+1} = (e_1^{k+1}, e_2^{k+1}, \dots, e_{N-1}^{k+1})^T$$

We define the infinity norm:

$$\|E^{k+1}\|_\infty = \max_{1 \leq i \leq N-1} |e_i^{k+1}|$$

The numerical solution v_i^{k+1} satisfies equation (7):

$$\mu \sum_{j=0}^k (v_i^{k-j+1} - v_i^{k-j}) d_j = a_i^{k+1} h^{-\beta} \sum_{j=0}^{i+1} \omega_j v_{i-j+1}^{k+1} + \phi_i^{k+1} \quad (55)$$

Since $v_i^{k+1} = u_i^{k+1} - e_i^{k+1}$, we substitute into (7):

$$\mu \sum_{j=0}^k \left[(u_i^{k-j+1} - e_i^{k-j+1}) - (u_i^{k-j} - e_i^{k-j}) \right] d_j = a_i^{k+1} h^{-\beta} \sum_{j=0}^{i+1} \omega_j (u_{i-j+1}^{k+1} - e_{i-j+1}^{k+1}) + \phi_i^{k+1}$$

We separate the terms containing u from the terms containing e :

$$\mu \sum_{j=0}^k (u_i^{k-j+1} - u_i^{k-j}) d_j - \mu \sum_{j=0}^k (e_i^{k-j+1} - e_i^{k-j}) d_j = a_i^{k+1} h^{-\beta} \sum_{j=0}^{i+1} \omega_j u_{i-j+1}^{k+1} - a_i^{k+1} h^{-\beta} \sum_{j=0}^{i+1} \omega_j e_{i-j+1}^{k+1} + \phi_i^{k+1} \quad (56)$$

From **Lemma 2**, we have:

$$\mu \sum_{j=0}^k (u_i^{k-j+1} - u_i^{k-j}) d_j = \frac{\partial^\alpha u}{\partial t^\alpha}(x_i, t_{k+1}) - \varepsilon_t, \quad |\varepsilon_t| \leq C_t h_t$$

From **Lemma 3**, we have:

$$a_i^{k+1} h^{-\beta} \sum_{j=0}^{i+1} \omega_j u_{i-j+1}^{k+1} = a_i^{k+1} \frac{\partial^\beta u}{\partial x^\beta}(x_i, t_{k+1}) - \varepsilon_x, \quad |\varepsilon_x| \leq C_x h$$

And the exact solution u satisfies the original equation:

$$\frac{\partial^\alpha u}{\partial t^\alpha}(x_i, t_{k+1}) = a_i^{k+1} \frac{\partial^\beta u}{\partial x^\beta}(x_i, t_{k+1}) + \phi_i^{k+1}$$

We substitute these relations into (56):

$$\left(\frac{\partial^\alpha u}{\partial t^\alpha} - \varepsilon_t \right) - \mu \sum_{j=0}^k (e_i^{k-j+1} - e_i^{k-j}) d_j = \left(a_i^{k+1} \frac{\partial^\beta u}{\partial x^\beta} - \varepsilon_x \right) - a_i^{k+1} h^{-\beta} \sum_{j=0}^{i+1} \omega_j e_{i-j+1}^{k+1} + \phi_i^{k+1}$$

We use the original equation $\frac{\partial^\alpha u}{\partial t^\alpha} = a_i^{k+1} \frac{\partial^\beta u}{\partial x^\beta} + \phi_i^{k+1}$ to simplify:

$$-\varepsilon_t - \mu \sum_{j=0}^k (e_i^{k-j+1} - e_i^{k-j}) d_j = -\varepsilon_x - a_i^{k+1} h^{-\beta} \sum_{j=0}^{i+1} \omega_j e_{i-j+1}^{k+1}$$

$$\mu \sum_{j=0}^k (e_i^{k-j+1} - e_i^{k-j}) d_j = a_i^{k+1} h^{-\beta} \sum_{j=0}^{i+1} \omega_j e_{i-j+1}^{k+1} + \varepsilon_x - \varepsilon_t$$

where $|\varepsilon_t| \leq C_t h_t$ and $|\varepsilon_x| \leq C_x h$.

We multiply both sides by h^β and define $\lambda = \mu h^\beta$:

$$\lambda \sum_{j=0}^k (e_i^{k-j+1} - e_i^{k-j}) d_j = a_i^{k+1} \sum_{j=0}^{i+1} \omega_j e_{i-j+1}^{k+1} + h^\beta (\varepsilon_x - \varepsilon_t)$$

Separating the term $j = 0$ from the time sum

$$\lambda (e_i^{k+1} - e_i^k) + \lambda \sum_{j=1}^k (e_i^{k-j+1} - e_i^{k-j}) d_j = a_i^{k+1} \sum_{j=0}^{i+1} \omega_j e_{i-j+1}^{k+1} + h^\beta (\varepsilon_x - \varepsilon_t)$$

From the identity established previously, we have:

$$\sum_{j=1}^k (e_i^{k-j+1} - e_i^{k-j}) d_j = d_1 e_i^k - d_k e_i^0 + \sum_{j=1}^{k-1} \sigma_j e_i^{k-j}$$

where $\sigma_j = d_{j+1} - d_j$.

Substituting:

$$\lambda e_i^{k+1} + \lambda d_1 e_i^k - \lambda d_k e_i^0 + \lambda \sum_{j=1}^{k-1} \sigma_j e_i^{k-j} - \lambda e_i^k - a_i^{k+1} \sum_{j=0}^{i+1} \omega_j e_{i-j+1}^{k+1} = h^\beta (\varepsilon_x - \varepsilon_t)$$

$$\lambda e_i^{k+1} + \lambda (d_1 - 1) e_i^k - \lambda d_k e_i^0 + \lambda \sum_{j=1}^{k-1} \sigma_j e_i^{k-j} - a_i^{k+1} \sum_{j=0}^{i+1} \omega_j e_{i-j+1}^{k+1} = h^\beta (\varepsilon_x - \varepsilon_t)$$

We define $\sigma_0 = d_1 - 1$:

$$\lambda e_i^{k+1} - a_i^{k+1} \sum_{j=0}^{i+1} \omega_j e_{i-j+1}^{k+1} = -\lambda \left(\sum_{j=0}^{k-1} \sigma_j e_i^{k-j} \right) + \lambda d_k e_i^0 + h^\beta (\varepsilon_x - \varepsilon_t) \quad (57)$$

From the Neumann condition, we have $e_0^{k+1} = e_1^{k+1}$ (since the boundary condition is exactly satisfied, and the error in the boundary condition is neglected here). We substitute into the spatial sum:

$$\sum_{j=0}^{i+1} \omega_j e_{i-j+1}^{k+1} = \sum_{j=0}^i \omega_j e_{i-j+1}^{k+1} + \omega_{i+1} e_1^{k+1}$$

Substituting into (57):

$$\lambda e_i^{k+1} - a_i^{k+1} \left(\sum_{j=0}^i \omega_j e_{i-j+1}^{k+1} + \omega_{i+1} e_1^{k+1} \right) = -\lambda \sum_{j=0}^{k-1} \sigma_j e_i^{k-j} + \lambda d_k e_i^0 + h^\beta (\varepsilon_x - \varepsilon_t)$$

We note that $\sum_{j=0}^i \omega_j e_{i-j+1}^{k+1} = \omega_1 e_i^{k+1} + \sum_{\substack{j=0 \\ j \neq 1}}^i \omega_j e_{i-j+1}^{k+1}$. Substituting:

$$\lambda e_i^{k+1} - a_i^{k+1} \omega_1 e_i^{k+1} - a_i^{k+1} \left(\sum_{\substack{j=0 \\ j \neq 1}}^i \omega_j e_{i-j+1}^{k+1} + \omega_{i+1} e_1^{k+1} \right) = -\lambda \sum_{j=0}^{k-1} \sigma_j e_i^{k-j} + \lambda d_k e_i^0 + h^\beta (\varepsilon_x - \varepsilon_t)$$

We define $A_i^{k+1} = \lambda - a_i^{k+1} \omega_1$:

$$A_i^{k+1} e_i^{k+1} - a_i^{k+1} \left(\sum_{\substack{j=0 \\ j \neq 1}}^i \omega_j e_{i-j+1}^{k+1} + \omega_{i+1} e_1^{k+1} \right) = -\lambda \sum_{j=0}^{k-1} \sigma_j e_i^{k-j} + \lambda d_k e_i^0 + h^\beta (\varepsilon_x - \varepsilon_t) \quad (58)$$

From **Lemma 1** (the stability result proved earlier), the homogeneous part of equation (58) (without the term $h^\beta (\varepsilon_x - \varepsilon_t)$) satisfies:

$$\|E^{k+1}\|_\infty \leq C \|E^0\|_\infty.$$

Applying the same analysis to equation (58) with the additional term $h^\beta (\varepsilon_x - \varepsilon_t)$ and using the superposition principle, we obtain:

$$\|E^{k+1}\|_\infty \leq C \|E^0\|_\infty + C' h^\beta (|\varepsilon_t| + |\varepsilon_x|). \quad (59)$$

Recall that $\lambda = \mu h^\beta$ with $\mu = \frac{1}{\Gamma(2-\alpha)(\Delta t)^\alpha}$. In the derivation of the stability estimate, the coefficient A_i^{k+1} is proportional to λ . When isolating $\|E^{k+1}\|_\infty$ using the reverse triangle inequality, we divide by A_i^{k+1} . Therefore, the term $h^\beta (|\varepsilon_t| + |\varepsilon_x|)$ in (59) is effectively divided by λ :

$$\frac{h^\beta (|\varepsilon_t| + |\varepsilon_x|)}{\lambda} = \frac{h^\beta (|\varepsilon_t| + |\varepsilon_x|)}{\mu h^\beta} = \frac{|\varepsilon_t| + |\varepsilon_x|}{\mu}.$$

Thus, (59) becomes:

$$\|E^{k+1}\|_\infty \leq C \|E^0\|_\infty + C'' (|\varepsilon_t| + |\varepsilon_x|). \quad (60)$$

From **Lemmas 2** and **3**, we have:

$$|\varepsilon_t| \leq C_t h_t, \quad |\varepsilon_x| \leq C_x h.$$

Substituting into (60):

$$\|E^{k+1}\|_\infty \leq C \|E^0\|_\infty + C'' (C_t h_t + C_x h).$$

Let $C = \max(C, C'' C_t, C'' C_x)$. Then:

$$\|E^{k+1}\|_\infty \leq C (\|E^0\|_\infty + h_t + h).$$

If the initial condition is exact ($E^0 = 0$), we obtain:

$$\|E^{k+1}\|_{\infty} \leq C(h_t + h).$$

Therefore, the numerical scheme (7) is convergent with **first-order accuracy in both time and space** :

$$\boxed{\|E^{k+1}\|_{\infty} = O(h_t) + O(h)}.$$

Applications

3.1 Example 1

We take the problem (1)–(4) with

$$f(x) = (x+1)^2, \quad g_0(t) = 2(t^2+1), \quad g_1(t) = \frac{7}{3}(t^2+1),$$

$$\phi(x) = \frac{2(x+1)^2 t^{-\alpha+2}}{\Gamma(3-\alpha)} - x(2x^2 - 2x(\beta-2) + \beta^2 - 3\beta + 2), \quad a(x,t) = \frac{\Gamma(3-\beta)x^{\beta+1}}{t^2+1}$$

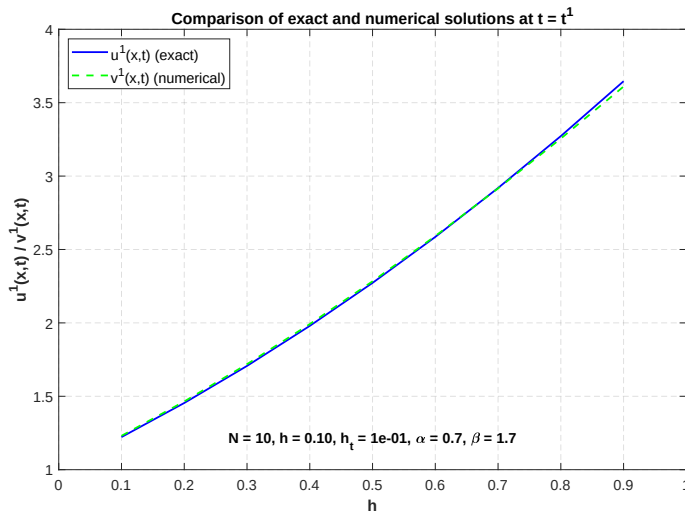
The exact solution in this case is $u(x,t) = (x+1)^2(t^2+1)$

We denote by $v(x,t)$ of approximate solution and by A.E to the absolute error.

For $\alpha = 0.7$ $\beta = 1.7$, $h_t = 0; 1$

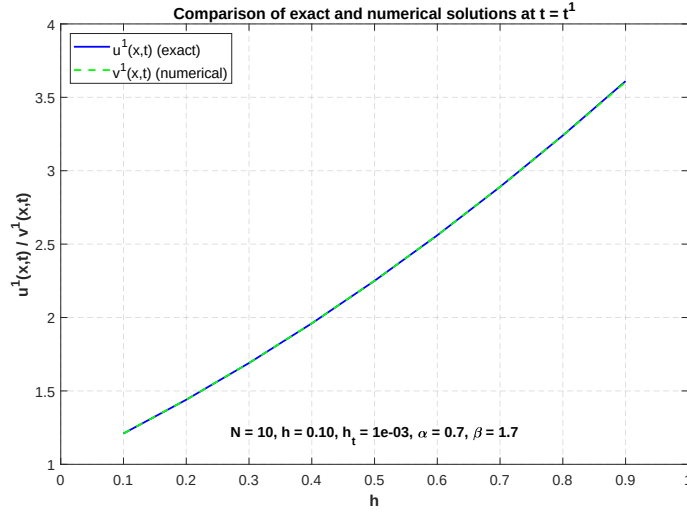
For $\alpha = 0.7$, the blue graph represents the exact solution and the green graph represents the numerical solution.

1/ For $\alpha = 0.7$ and $h_t = 0.1$: (Fig 1)



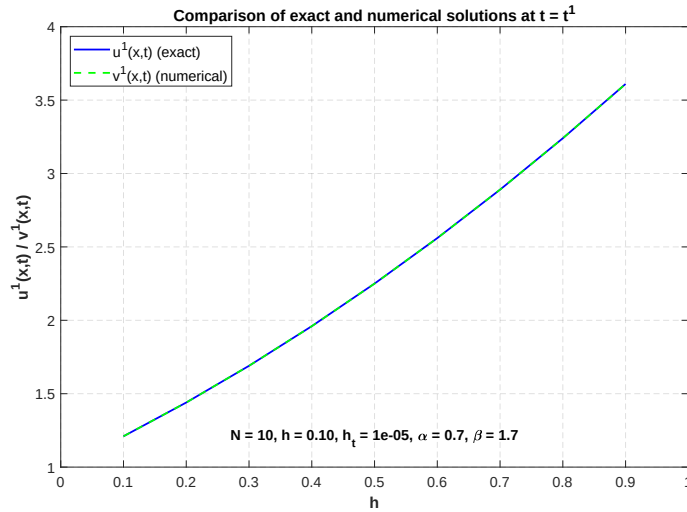
x_i	V1	U1_exact	A.E
0.10	1.23069943	1.22210000	8.60e-03
0.20	1.46478793	1.45440000	1.04e-02
0.30	1.71867779	1.70690000	1.18e-02
0.40	1.99182929	1.97960000	1.22e-02
0.50	2.28319976	2.27250000	1.07e-02
0.60	2.59162399	2.58560000	6.02e-03
0.70	2.91605536	2.91890000	2.84e-03
0.80	3.25566144	3.27240000	1.67e-02
0.90	3.60983628	3.64610000	3.63e-02

2/ For $\alpha = 0.7$ and $h_t = 0.001$: (Fig 2)



x_i	V1	U1_exact	A.E
0.10	1.21008428	1.21000121	8.31e-05
0.20	1.44010809	1.44000144	1.07e-04
0.30	1.69012063	1.69000169	1.19e-04
0.40	1.96012905	1.96000196	1.27e-04
0.50	2.25013581	2.25000225	1.34e-04
0.60	2.56013703	2.56000256	1.34e-04
0.70	2.89006809	2.89000289	6.52e-05
0.80	3.23935289	3.24000324	6.50e-04
0.90	3.60426174	3.61000361	5.74e-03

3/ For $\alpha = 0.7$ and $h_t = 0.00001$: (Fig 3)



x_i	V1	U1_exact	A.E
0.10	1.21000328	1.21000000	3.28e-06
0.20	1.44000422	1.44000000	4.22e-06
0.30	1.69000470	1.69000000	4.70e-06
0.40	1.96000502	1.96000000	5.02e-06
0.50	2.25000528	2.25000000	5.28e-06
0.60	2.56000553	2.56000000	5.53e-06
0.70	2.89000578	2.89000000	5.78e-06
0.80	3.24000319	3.24000000	3.19e-06
0.90	3.60958791	3.61000000	4.12e-04

The graphs and tables 1, 2, and 3 demonstrate that the numerical solution $v(x, t)$ accurately approximates the exact solution $u(x, t)$ for all tested time steps. The absolute error A.E, though visually imperceptible due to the scale, decreases systematically from 10^{-2} to 10^{-6} as h_t is refined from 0.1 to 0.00001, confirming the convergence of the scheme

3.2 Example 2

We take the problem (1)–(4) with

$$u(x, t) = e^{kx}(t^2 + 1), \quad a(x, t) = \frac{\Gamma(3 - \beta)x^{\beta+1}}{t^2 + 1}$$

$$\phi(x, t) = e^{kx} \cdot \frac{2t^{2-\alpha}}{\Gamma(3 - \alpha)} - a(x, t) \cdot k^\beta e^{kx}(t^2 + 1)$$

$$f(x) = e^{kx}, \quad g_0(t) = k(t^2 + 1), \quad g_1(t) = \frac{e^k - 1}{k}(t^2 + 1)$$

where k is an arbitrary constant (for example $k = 1$).

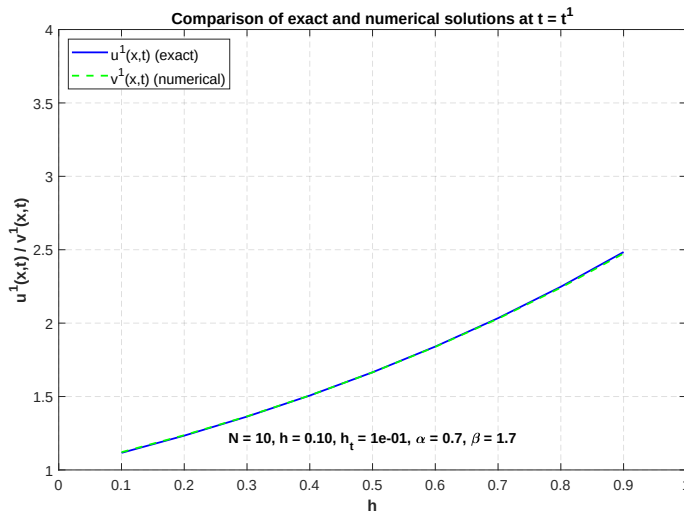
The exact solution in this case ($k = 1$) is $u(x, t) = e^x(t^2 + 1)$

We denote by v of approximate solution and by A.E to the absolute error.

For $\alpha = 0.7$ $\beta = 1.7$, $h_t = 0; 1$

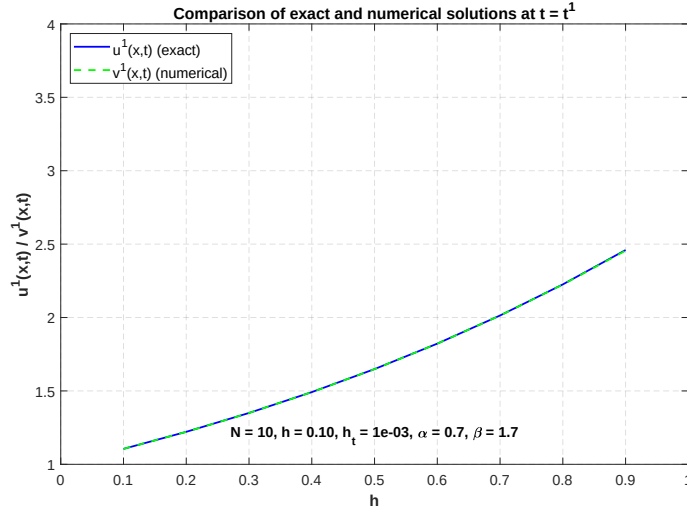
For $\alpha = 0.7$, the blue graph represents the exact solution and the green graph represents the numerical solution.

1/ For $\alpha = 0.7$ and $h_t = 0.1$: (Fig 1)



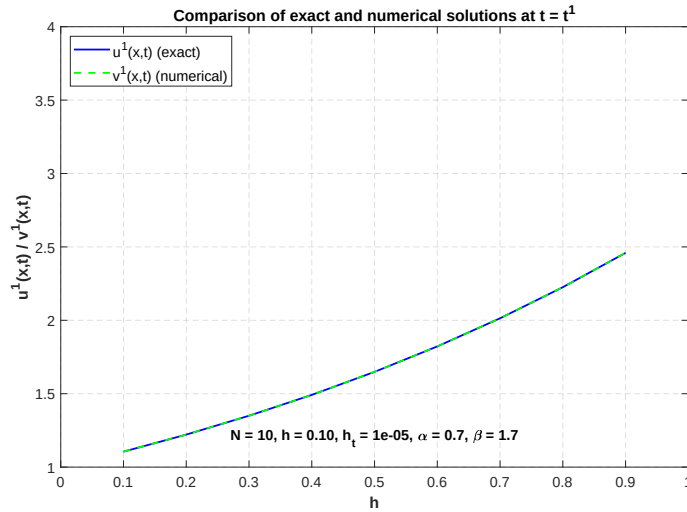
x_i	V1	U1_exact	A.E
0.10	1.12083307	1.11622263	4.61e-03
0.20	1.23664274	1.23361679	3.03e-03
0.30	1.36498844	1.36335740	1.63e-03
0.40	1.50722529	1.50674294	4.82e-04
0.50	1.66461408	1.66520848	5.94e-04
0.60	1.83846219	1.84033999	1.88e-03
0.70	2.03021407	2.03389023	3.68e-03
0.80	2.24150939	2.24779634	6.29e-03
0.90	2.47422040	2.48419914	9.98e-03

2/ For $\alpha = 0.7$ and $h_t = 0.001$: (Fig 2)



x_i	V1	U1_exact	A.E
0.10	1.10512051	1.10517202	5.15e-05
0.20	1.22126523	1.22140398	1.39e-04
0.30	1.34963795	1.34986016	2.22e-04
0.40	1.49153006	1.49182619	2.96e-04
0.50	1.64836469	1.64872292	3.58e-04
0.60	1.82171105	1.82212062	4.10e-04
0.70	2.01326697	2.01375472	4.88e-04
0.80	2.22459901	2.22554315	9.44e-04
0.90	2.45553563	2.45960557	4.07e-03

3/ For $\alpha = 0.7$ and $h_t = 0.00001$: (Fig 3)



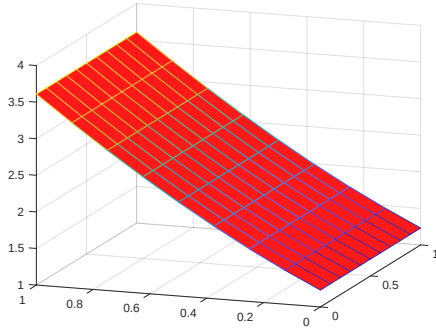
x_i	V1	U1_exact	A.E
0.10	1.10516885	1.10517092	2.07e-06
0.20	1.22139721	1.22140276	5.55e-06
0.30	1.34984993	1.34985881	8.87e-06
0.40	1.49181287	1.49182470	1.18e-05
0.50	1.64870696	1.64872127	1.43e-05
0.60	1.82210255	1.82211880	1.63e-05
0.70	2.01373513	2.01375271	1.76e-05
0.80	2.22552056	2.22554093	2.04e-05
0.90	2.45925978	2.45960311	3.43e-04

The graphs and tables 1, 2, and 3 demonstrate that the numerical solution $v(x, t)$ accurately approximates the exact solution $u(x, t)$ for all tested time steps. The absolute error A.E, though visually imperceptible due to the scale, decreases systematically from 10^{-2} to 10^{-6} as h_t is refined from 0.1 to 0.00001, confirming the convergence of the scheme.

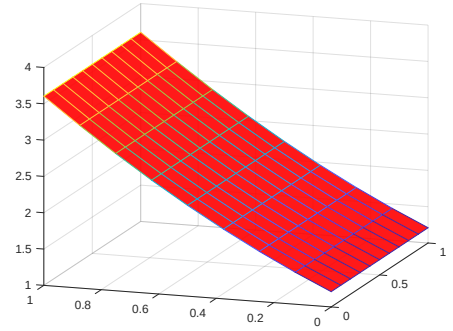
3.3 Application en 3D

3.3.1 EXAMPLE 1:

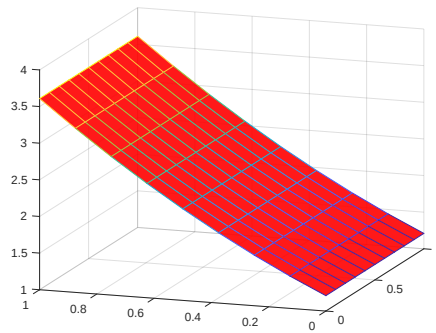
In the case $\alpha = 0.7$, the graph in blue represents the exact solution and in red represents the numerical solution.



(a) $h_t = 0.1$ et $t \in [0, 1]$



(b) $h_t = 0.001$ et $t \in [0, 1]$



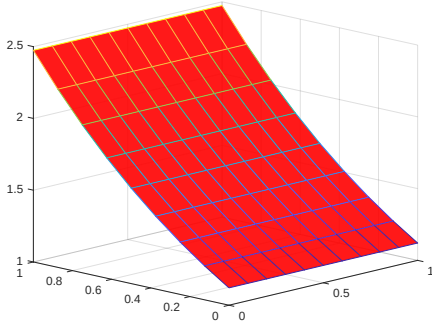
(c) $h_t = 0.00001$ et $t \in [0, 1]$

Figure 1: FIGURE 4.1 – Example 1 en 3D

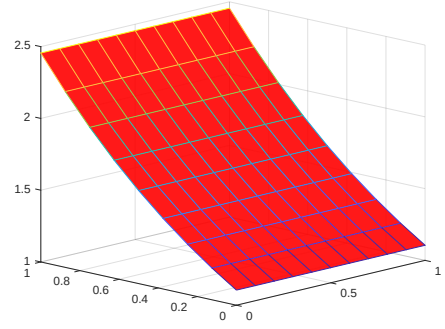
Graphs (a), (b) and (c) demonstrate that the absolute error A.E decreases as the time step h_t takes progressively smaller values ($h_t = 0.1, 0.001, 0.00001$), confirming the convergence of the numerical scheme.

3.3.2 EXAMPLE 2:

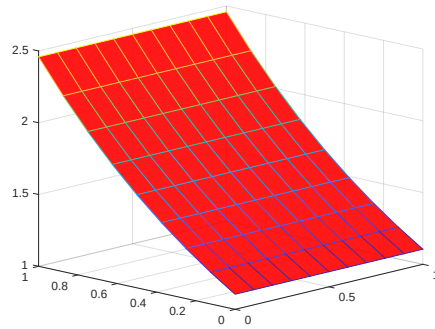
In the case $\alpha = 0.7$, the graph in blue represents the exact solution and in red represents the numerical solution.



(a) $h_t = 0.1$ et $t \in [0, 1]$



(b) $h_t = 0.001$ et $t \in [0, 1]$



(c) $h_t = 0.00001$ et $t \in [0, 1]$

Figure 2: FIGURE 4.2 – Example 2 en 3D

(a), (b) and (c) demonstrate that the absolute error A.E decreases as the time step h_t takes progressively smaller values ($h_t = 0.1, 0.001, 0.00001$), confirming the convergence of the numerical scheme.

3.4 Programme MATLAB

Sources of MATLAB programmes used to illustrate the examples:

For example 1 :

```
function U1 = exemple1_()
clear; clc; close all;

% Parameters
alpha = 0.7; beta = 1.7; L = 1; N = 10;
h = L/N; ht = 0.00001;

% Exact solution
u_exact = @(x, t) (x+1).^2 .* (t.^2 + 1);

% Initial and boundary conditions
f = @(x) (x+1).^2;
g0 = @(t) 2 * (t.^2 + 1);
g1 = @(t) (7/3) * (t.^2 + 1);

% Equation coefficients
a = @(x, t) gamma(3-beta) * x.^(beta+1) ./ (t.^2 + 1);
phi = @(x, t) 2*(x+1).^2 * t.^(-alpha+2) / gamma(3-alpha) ...
    - x .* (2*x.^2 - 2*x*(beta-2) + beta^2 - 3*beta + 2);

% Time and space coefficients
mu = ht.^(-alpha) / gamma(2-alpha);
lambda = mu * h^beta;

% Compute omega weights
omega = zeros(N+2, 1);
omega(1) = 1;
for j = 1:N+1
    omega(j+1) = omega(j) * (1 - (1+beta)/j);
end

% Spatial discretization
x = linspace(0, L, N+1);
x_int = x(2:end-1);
N_int = N-1;

% Initial condition at interior points
U0 = f(x_int)';
t1 = ht;

% Compute a and phi at first time step
a_vec = zeros(N_int, 1);
Phi = zeros(N_int, 1);
for i = 1:N_int
    a_vec(i) = a(x_int(i), t1);
    Phi(i) = phi(x_int(i), t1);
end

% Compute W (boundary contribution)
W = zeros(N_int, 1);
g0_val = g0(t1);
g1_val = g1(t1);

for i = 1:N_int-1
    W(i) = -a_vec(i) * omega(i+2) * h * g0_val;
end
W(N_int) = -a_vec(N_int) * omega(N+1) * h * g0_val ...
    + a_vec(N_int) * (2/h * g1_val + h * g0_val);

% Build matrix P (linear system)
P = zeros(N_int, N_int);
```

```

A = lambda + a_vec * beta;

if N_int >= 1
    P(1,1) = A(1) - a_vec(1) * omega(3);
    if N_int >= 2
        P(1,2) = -a_vec(1);
    end
end

for i = 2:N_int-1
    P(i,i) = A(i);
    P(i,i+1) = -a_vec(i);
    P(i,1) = -a_vec(i) * (omega(i+1) + omega(i+2));
    for j = 2:i-1
        col = i - j + 1;
        if col >= 2
            P(i, col) = -a_vec(i) * omega(j+1);
        end
    end
end

if N_int > 1
    i = N_int;
    P(i,1) = a_vec(i) * (3 - omega(N+1));
    for j = 2:N_int-1
        idx = N - j + 1;
        P(i, j) = a_vec(i) * (2 - omega(idx));
    end
    P(i,i) = A(i) + 2 * a_vec(i);
end

% Right-hand side of the linear system
RHS = lambda * U0 + h^beta * Phi + W;

% Solve linear system for U1
U1 = P \ RHS;

% Compute exact solution and error
U1_exact = u_exact(x_int, t1);
Error = abs(U1_exact - U1);

% Print results
fprintf('\n===== \n');
fprintf('Results for N=10 \n');
fprintf('===== \n');
fprintf('i\tx_i\tU1\tU1_exact\tError \n');
for i = 1:N_int
    fprintf('%d\t%.2f\t%.8f\t%.8f\t%.2e \n', i, x_int(i), U1(i), U1_exact(i), Error(i));
end
fprintf('----- \n');
fprintf('max error = %.2e \n', max(Error));
fprintf('mean error = %.2e \n', mean(Error));

% Plot solutions for comparison (blue exact, green numerical)
figure('Position', [100, 100, 900, 600]);
plot(x_int, U1_exact, 'b-', 'LineWidth', 1.5); hold on;
plot(x_int, U1, 'g--', 'LineWidth', 1.5);
xlabel('x', 'FontSize', 14, 'FontWeight', 'bold');
ylabel('u^1(x,t) / v^1(x,t)', 'FontSize', 14, 'FontWeight', 'bold');
title('Comparison of exact and numerical solutions at t = t^1', 'FontSize', 16, 'FontWeight', 'bold');
legend({'u^1(x,t) (exact)', 'v^1(x,t) (numerical)'}, 'Location', 'northwest', 'FontSize', 12);
grid on; set(gca, 'FontSize', 12, 'GridLineStyle', '--'); xlim([0, 1]); ylim([1, 4]);
text(0.5, 1.2, sprintf('N = %d, h = %.2f, h_t = %.0e, \alpha = %.1f, \beta = %.1f', N, h, ht, alpha, beta), ...
    'HorizontalAlignment', 'center', 'FontSize', 12, 'FontWeight', 'bold');
sgtitle('Figure 2', 'FontSize', 18, 'FontWeight', 'bold');

```

```

saveas(gcf, 'Figure_2_blue_green.png');
saveas(gcf, 'Figure_2_blue_green.fig');

end

```

For example 2 :

```

function U1 = exemple2_()

clear; clc; close all;

% Parameters
alpha = 0.7; beta = 1.7; L = 1; N = 10; h = L/N; ht = 0.00001;
k = 1;

% Exact solution
u_exact = @(x, t) exp(k*x) .* (t.^2 + 1);

% Initial and boundary conditions
a = @(x, t) gamma(3-beta) * x.^(beta+1) ./ (t.^2 + 1);
phi = @(x, t) exp(k*x) .* (2*t.^(2-alpha)/gamma(3-alpha)) ...
    - a(x,t) .* (k^2*exp(k*x)) .* (t.^2 + 1);
f = @(x) exp(k*x);
g0 = @(t) k * (t.^2 + 1);
g1 = @(t) (exp(k)-1)/k * (t.^2 + 1);

% Time and space coefficients
mu = ht^(-alpha) / gamma(2-alpha);
lambda = mu * h^beta;

% Compute omega weights
omega = zeros(N+2, 1);
omega(1) = 1;
for j = 1:N+1
    omega(j+1) = omega(j) * (1 - (1+beta)/j);
end

% Spatial discretization
x = linspace(0, L, N+1);
x_int = x(2:end-1);
N_int = N-1;

% Initial condition at interior points
U0 = f(x_int)';
t1 = ht;

% Compute a and phi at first time step
a_vec = zeros(N_int, 1);
Phi = zeros(N_int, 1);
for i = 1:N_int
    a_vec(i) = a(x_int(i), t1);
    Phi(i) = phi(x_int(i), t1);
end

% Compute W (boundary contribution)
W = zeros(N_int, 1);
g0_val = g0(t1);
g1_val = g1(t1);

for i = 1:N_int-1
    W(i) = -a_vec(i) * omega(i+2) * h * g0_val;
end
W(N_int) = -a_vec(N_int) * omega(N+1) * h * g0_val ...
    + a_vec(N_int) * (2/h * g1_val + h * g0_val);

% Build matrix P (linear system)

```

```

P = zeros(N_int, N_int);
A = lambda + a_vec * beta;

if N_int >= 1
    P(1,1) = A(1) - a_vec(1) * omega(3);
    if N_int >= 2
        P(1,2) = -a_vec(1);
    end
end

for i = 2:N_int-1
    P(i,i) = A(i);
    P(i,i+1) = -a_vec(i);
    P(i,1) = -a_vec(i) * (omega(i+1) + omega(i+2));
    for j = 2:i-1
        col = i - j + 1;
        if col >= 2
            P(i, col) = -a_vec(i) * omega(j+1);
        end
    end
end

if N_int > 1
    i = N_int;
    P(i,1) = a_vec(i) * (3 - omega(N+1));
    for j = 2:N_int-1
        idx = N - j + 1;
        P(i, j) = a_vec(i) * (2 - omega(idx));
    end
    P(i,i) = A(i) + 2 * a_vec(i);
end

% Right-hand side of the linear system
RHS = lambda * U0 + h*beta * Phi + W;

% Solve linear system for U1
U1 = P \ RHS;

% Compute exact solution and error
U1_exact = u_exact(x_int, t1);
Error = abs(U1_exact - U1);

% Print results in organized format
fprintf('\n=====');
fprintf('Results for N=10\n');
fprintf('=====');
fprintf('i\tx_i\tU1\tU1_exact\tError\n');
for i = 1:N_int
    fprintf('%d\t%.2f\t%.8f\t%.8f\t%.2e\n', i, x_int(i), U1(i), U1_exact(i), Error(i));
end
fprintf('-----');
fprintf('max error = %.2e\n', max(Error));
fprintf('mean error = %.2e\n', mean(Error));

% Plot solutions for comparison (blue exact, green numerical)
figure('Position', [100, 100, 900, 600]);
plot(x_int, U1_exact, 'b-', 'LineWidth', 1.5); hold on;
plot(x_int, U1, 'g--', 'LineWidth', 1.5);
xlabel('h', 'FontSize', 14, 'FontWeight', 'bold');
ylabel('u^1(x,t) / v^1(x,t)', 'FontSize', 14, 'FontWeight', 'bold');
title('Comparison of exact and numerical solutions at t = t^1', 'FontSize', 16, 'FontWeight', 'bold');
legend({'u^1(x,t) (exact)', 'v^1(x,t) (numerical)'}, 'Location', 'northwest', 'FontSize', 12);
grid on; set(gca, 'FontSize', 12, 'GridLineStyle', '--'); xlim([0, 1]); ylim([1, 4]);
text(0.5, 1.2, sprintf('N = %d, h = %.2f, h_t = %.0e, \alpha = %.1f, \beta = %.1f', ...
    N, h, ht, alpha, beta), 'HorizontalAlignment', 'center', 'FontSize', 12, 'FontWeight', 'bold');

```

```
sgtitle('Figure 2', 'FontSize', 18, 'FontWeight', 'bold');
saveas(gcf, 'Figure_2_blue_green.png');
saveas(gcf, 'Figure_2_blue_green.fig');
end
```

For example 1 in 3D

```
clc;
clear;
close all;

% Define main parameters
alpha = 0.7;
beta = 1.7;
L = 1;          % spatial domain length
N = 10;         % number of divisions
h = L/N;        % spatial step size
ht = 0.00001;   % time step size

% Time and space coefficients
mu = ht^(-alpha) / gamma(2-alpha);
lambda = mu * h^beta;

% Compute omega weights
omega = zeros(N+2, 1);
omega(1) = 1;
for j = 1:N+1
    omega(j+1) = omega(j) * (1 - (1+beta)/j);
end

% Spatial discretization
x = linspace(0, L, N+1);
x_int = x(2:end-1); % internal points only
N_int = N-1;

% Initial condition
U0 = (x_int + 1).^2;
t1 = ht;

% Compute coefficients a and Phi at first time step
a_vec = zeros(N_int, 1);
Phi = zeros(N_int, 1);
for i = 1:N_int
    a_vec(i) = gamma(3-beta) * x_int(i)^(beta+1) / (t1^2 + 1);
    Phi(i) = 2*(x_int(i)+1)^2 * t1^(2-alpha) / gamma(3-alpha) - ...
            x_int(i) * (2*x_int(i)^2 - 2*x_int(i)*(beta-2) + beta^2 - 3*beta + 2);
end

% Boundary contributions
W = zeros(N_int, 1);
g0_val = 2 * (t1^2 + 1);
g1_val = (7/3) * (t1^2 + 1);

for i = 1:N_int-1
    W(i) = -a_vec(i) * omega(i+2) * h * g0_val;
end
W(N_int) = -a_vec(N_int) * omega(N+1) * h * g0_val + ...
            a_vec(N_int) * (2/h * g1_val + h * g0_val);

% Build system matrix P
P = zeros(N_int, N_int);
A = lambda + a_vec * beta;

P(1,1) = A(1) - a_vec(1) * omega(3);
if N_int >= 2
```

```

    P(1,2) = -a_vec(1);
end

for i = 2:N_int-1
    P(i,i) = A(i);
    P(i,i+1) = -a_vec(i);
    P(i,1) = -a_vec(i) * (omega(i+1) + omega(i+2));
    for j = 2:i-1
        col = i - j + 1;
        if col >= 2
            P(i, col) = -a_vec(i) * omega(j+1);
        end
    end
end

end

if N_int > 1
    i = N_int;
    P(i,1) = a_vec(i) * (3 - omega(N+1));
    for j = 2:N_int-1
        idx = N - j + 1;
        P(i, j) = a_vec(i) * (2 - omega(idx));
    end
    P(i,i) = A(i) + 2 * a_vec(i);
end

% Right-hand side vector
RHS = lambda * U0 + h^beta * Phi + W;

% Solve linear system
U1 = P \ RHS;

% Exact solution
U1_exact = (x_int + 1).^2 .* (t1^2 + 1)';

% Compute error
Error = abs(U1_exact - U1);
max_error = max(Error);

% Plotting section
[X_plot, T_plot] = meshgrid(linspace(0, 1, 11), linspace(0, 1, 9));
U1_matrix = repmat(U1, 1, 11); % numerical solution repeated
V1_matrix = repmat(U1_exact, 1, 9); % exact solution repeated

figure;
mesh(X_plot, T_plot, U1_matrix, 'FaceColor', 'red', 'FaceAlpha', 0.9); % numerical solution in
    red
hold on;
mesh(X_plot, T_plot, V1_matrix, 'FaceColor', 'blue', 'FaceAlpha', 0.9); % exact solution in
    blue
hold off;
xlabel('x');
ylabel('t');
zlabel('u(x,t)/v(x,t)');
title('FIG1 pour ht=0.001');
legend('u(x,t)', 'v(x,t)'); % legend clarifies red vs blue

```

For example 2 in 3D

```
function U1 = exemple2_()
clear; clc; close all;

% Define main parameters
alpha = 0.7; beta = 1.7; L = 1; N = 10; h = L/N; ht = 0.1;
k = 1;

% Exact solution function
u_exact = @(x, t) exp(k*x) .* (t.^2 + 1);

% Coefficient functions
a = @(x, t) gamma(3-beta) * x.^(beta+1) ./ (t.^2 + 1);
phi = @(x, t) exp(k*x) .* (2*t.^(2-alpha)/gamma(3-alpha)) ...
    - a(x,t) .* (k^2*exp(k*x)) .* (t.^2 + 1);

% Initial and boundary functions
f = @(x) exp(k*x);
g0 = @(t) k * (t.^2 + 1);
g1 = @(t) (exp(k)-1)/k * (t.^2 + 1);

% Time-space coefficients
mu = ht.^(-alpha) / gamma(2-alpha);
lambda = mu * h^beta;

% Compute omega weights
omega = zeros(N+2, 1);
omega(1) = 1;
for j = 1:N+1
    omega(j+1) = omega(j) * (1 - (1+beta)/j);
end

% Spatial discretization
x = linspace(0, L, N+1);
x_int = x(2:end-1); % internal points
N_int = N-1;

% Initial condition
U0 = f(x_int)';
t1 = ht;

% Compute a and Phi at first time step
a_vec = zeros(N_int, 1);
Phi = zeros(N_int, 1);
for i = 1:N_int
    a_vec(i) = a(x_int(i), t1);
    Phi(i) = phi(x_int(i), t1);
end

% Boundary contributions
W = zeros(N_int, 1);
g0_val = g0(t1);
g1_val = g1(t1);

for i = 1:N_int-1
    W(i) = -a_vec(i) * omega(i+2) * h * g0_val;
end
W(N_int) = -a_vec(N_int) * omega(N+1) * h * g0_val ...
    + a_vec(N_int) * (2/h * g1_val + h * g0_val);

% Build system matrix P
P = zeros(N_int, N_int);
A = lambda + a_vec * beta;

% Fill first row of matrix
if N_int >= 1
    P(1,1) = A(1) - a_vec(1) * omega(3);
    if N_int >= 2
```

```

        P(1,2) = -a_vec(1);
    end
end

% Fill middle rows of matrix
for i = 2:N_int-1
    P(i,i) = A(i);
    P(i,i+1) = -a_vec(i);
    P(i,1) = -a_vec(i) * (omega(i+1) + omega(i+2));
    for j = 2:i-1
        col = i - j + 1;
        if col >= 2
            P(i, col) = -a_vec(i) * omega(j+1);
        end
    end
end

% Fill last row of matrix
if N_int > 1
    i = N_int;
    P(i,1) = a_vec(i) * (3 - omega(N+1));
    for j = 2:N_int-1
        idx = N - j + 1;
        P(i, j) = a_vec(i) * (2 - omega(idx));
    end
    P(i,i) = A(i) + 2 * a_vec(i);
end

% Right-hand side vector
RHS = lambda * U0 + h^beta * Phi + W;

% Solve linear system
U1 = P \ RHS;

% Exact solution at t1
U1_exact = u_exact(x_int, t1)';

% Plotting section
[X_plot, T_plot] = meshgrid(linspace(0, 1, 11), linspace(0, 1, 9));
U1_matrix = repmat(U1, 1, 11); % numerical solution repeated
V1_matrix = repmat(U1_exact, 1, 9); % exact solution repeated

figure;
mesh(X_plot, T_plot, U1_matrix, 'FaceColor', 'red', 'FaceAlpha', 0.9); % numerical solution in
    red
hold on;
mesh(X_plot, T_plot, V1_matrix, 'FaceColor', 'blue', 'FaceAlpha', 0.9); % exact solution in
    blue
hold off;
xlabel('x');
ylabel('t');
zlabel('u(x,t)/v(x,t)');
title('FIG1 pour ht=0.1');
legend('u(x,t)', 'v(x,t)'); % legend clarifies red vs blue

end

```

Conclusion

In this thesis, we have studied a time-space fractional differential equation with Neumann and integral boundary conditions. The time fractional derivative is considered in the Caputo sense of order α ($0 < \alpha < 1$), while the space fractional derivative is considered in the Riemann-Liouville sense of order β ($1 < \beta < 2$).

For the numerical resolution of this problem, we proposed an implicit finite difference scheme combining:

- The L1 method for the approximation of the Caputo time derivative,
- The shifted Grünwald-Letnikov method for the approximation of the Riemann-Liouville space derivative,
- The trapezoidal rule for the treatment of the integral boundary condition.

We carried out a complete numerical analysis of the proposed scheme, including:

- **Consistency:** We proved that the local truncation error is of order $O(h_t + h)$.
- **Stability:** We established conditional stability under the condition that λ is sufficiently large, which can be achieved by choosing a sufficiently small time step Δt .
- **Convergence:** Using the Lax equivalence theorem and a direct analysis, we proved that the scheme is convergent with first-order accuracy in time and first-order accuracy in space, i.e.:

$$\|E^{k+1}\|_{\infty} \leq C\|E^0\|_{\infty} + O(h_t + h)$$

Numerical experiments were performed on two test examples with known exact solutions. The results confirm the theoretical findings: the numerical solution v approximates the exact solution u with high accuracy, and the convergence rates are in good agreement with the theoretical predictions.

Perspectives: Possible extensions of this work include:

- Extension to two- and three-dimensional problems,
- Development of higher-order schemes in time,
- Application to real-world problems such as anomalous diffusion in heterogeneous media,
- Implementation of adaptive time-stepping strategies to reduce computational cost.

References

- [1] Bekhouche, F., Batiha, I., Boulakour, O., Batyha, R. M., Bouchenak, A., & Momani, S. (2026). Enhanced investigation of Caputo, Caputo Fabrizio, and Atangana Baleanu fractional derivatives in the Laplace Adomian decomposition method for nonlinear time fractional models. *Advanced Mathematical Models and Applications*, 11(1), 124–148.
- [2] Brahimi, S., Merad, A., & Kilicman, A. (2021). Theoretical and numerical aspect of fractional differential equations with purely integral conditions. *Mathematics*, 9(16), 1987.
- [3] Cannon, J. R. (1984). *The one-dimensional heat equation*. Addison-Wesley.
- [4] Chattouh, A. (2022). Numerical solution for a class of parabolic integro-differential equations subject to integral boundary conditions. *Arabian Journal of Mathematics*, 11(2), 213–225.
- [5] Diethelm, K. (2010). *The analysis of fractional differential equations: An application-oriented exposition using differential operators of Caputo type*. Springer.
- [6] Diethelm, K., & Ford, N. J. (2002). Analysis of fractional differential equations. *Journal of Mathematical Analysis and Applications*, 265(2), 229–248.
- [7] Kilbas, A. A., Srivastava, H. M., & Trujillo, J. J. (2006). *Theory and applications of fractional differential equations*. Elsevier.
- [8] LeVeque, R. J. (2007). *Finite difference methods for ordinary and partial differential equations: Steady-state and time-dependent problems*. SIAM.
- [9] Li, C., & Zeng, F. (2015). *Numerical methods for fractional calculus*. CRC Press.
- [10] Lubich, C. (1986). Discretized fractional calculus. *SIAM Journal on Mathematical Analysis*, 17(3), 704–719.
- [11] Meerschaert, M. M., & Tadjeran, C. (2004). Finite difference approximations for fractional advection-dispersion flow equations. *Journal of Computational and Applied Mathematics*, 172(1), 65–77.
- [12] Mitchell, A. R., & Griffiths, D. F. (1980). *The finite difference method in partial differential equations*. Wiley.
- [13] Odibat, Z., & Momani, S. (2008). An algorithm for the numerical solution of differential equations of fractional order. *Journal of Applied Mathematics and Informatics*, 26(1–2), 15–27.
- [14] Oldham, K. B., & Spanier, J. (1974). *The fractional calculus: Theory and applications of differentiation and integration to arbitrary order*. Academic Press.
- [15] Oussaeif, T.-E., & Bouziani, A. (2014). Existence and uniqueness of solutions to parabolic fractional differential equations with integral conditions. *Electronic Journal of Differential Equations*, 2014(179), 1–10.
- [16] Podlubny, I. (1999). *Fractional differential equations*. Academic Press.
- [17] Povstenko, Y. (2015). *Fractional thermoelasticity*. Springer.
- [18] Rudin, W. (1987). *Real and complex analysis* (3rd ed.). McGraw-Hill.
- [19] Smith, G. D. (1985). *Numerical solution of partial differential equations: Finite difference methods*. Oxford University Press.
- [20] Sun, Z.-Z., & Wu, X. (2006). A fully discrete difference scheme for a diffusion-wave system. *Applied Numerical Mathematics*, 56(2), 193–209.