Multigrid methods for fractional diffusion equations

Doctoral Dissertation submitted to the
Faculty of Informatics of the Università della Svizzera Italiana
in partial fulfillment of the requirements for the degree of
Doctor of Philosophy

presented by
Ken Trotti

under the supervision of
Prof. Rolf Krause and Prof. Marco Donatelli

12 2021

Dissertation Committee

Cesare Alippi	University of Italian Switzerland, Lugano, Switzerland
Matthias Bolten	University of Wuppertal, Wuppertal, Germany
Michael Multerer	University of Italian Switzerland, Lugano, Switzerland

Dissertation accepted on 15 12 2021

Marco Donatelli

Research Advisor	Co-Advisor
Prof. Rolf Krause	**Prof. Marco Donatelli**

PhD Program Director

The PhD program Director *Walter Binder*

I certify that except where due acknowledgement has been given, the work presented in this thesis is that of the author alone; the work has not been submitted previously, in whole or in part, to qualify for any other academic award; and the content of the thesis is the result of work which has been carried out since the official commencement date of the approved research program.

Ken Trotti

Ken Trotti
Lugano, 15 12 2021

To my family

Abstract

Recent years have seen the rapid growth in interest towards fractional calculus. Fractional calculus plays an important role in modelling anomalous diffusion phenomena, however closed-form analytical solutions of such equations are rarely available, hence numerical estimates are needed.

In this thesis we consider various fractional diffusion equations (FDEs), where different fractional derivative definitions and related discretizations are involved and we focus on multigrid-based approaches for solving the associated linear systems. Precisely, we will leverage the spectral properties of the coefficient matrix, retrieved by exploiting its structure, to design ad-hoc (tailored) multigrid solvers or preconditioners for Krylov methods.

We develop a new approach to compute the Jacobi weight, which is versatile enough to work with various FDEs and allows to build a parameter free multigrid method. Moreover, in the case of uniform meshes, we exploit the knowledge about anisotropic integer-order partial diffusion equations to deal with anisotropic FDEs, by building a robust multigrid-based solver. Furthermore, we study the behavior of multigrid methods as parallel-in-time solvers and, then, we provide a new second-order accurate finite volume approximation and related ad-hoc multigrid solver. Finally, we extend our multigrid strategies to deal with a singular one-dimensional space-FDE discretized over non-uniform meshes.

Acknowledgements

This thesis is the result of a joint supervision between the University of Insubria and the University of Italian Switzerland (USI).

I would like to thank my supervisors Prof. Marco Donatelli (Univ. Insubria) and Prof. Rolf Krause (USI) for all their precious help and support.

Moreover, a special thank to Dr. Mariarosa Mazza (Co-Advisor Univ. Insubria) for her help and remarks, which taught me a lot and improved the quality of my work.

Contents

List of Figures

List of Tables

Introduction

The history of the fractional calculus goes back to seventeenth century, when in 1695 the derivative of order $\alpha = \frac{1}{2}$ was described by Leibniz in his letter to L'Hôpital Leibniz [1849, 1962a,b] [1]. Since then the interest in the theory is grown and applications have been found in biophysics, quantum mechanics, wave theory, polymers, continuum mechanics, Lie theory, field theory, spectroscopy and in group theory (see Tenreiro Machado [2011]; Oldham and Spanier [n.d.]; Podlubny [1998] and references therein).

In this thesis we deal with multigrid methods for different type of fractional diffusion equations (FDEs), which are shown to be useful in modelling anomalous diffusion processes in complex media, e.g., the dispersion of pollutant in underground water Zhang et al. [2021] or the propagation of the electrical potential in heterogeneous cardiac tissue Liu et al. [2015]; Bueno-Orovio et al. [2014]. By anomalous diffusion we mean the mesoscopic process of transport of particles, which differs from the standard diffusion process (Brownian motion).
The fractional diffusion can occur in time and/or space. In time, it generates a non-markovian process with an anomalous diffusion law Michelitsch et al. [2020]; Mainardi et al. [2001], whereas in space it generates a non-local diffusion process with "fat tails" Mainardi et al. [2001]. In this thesis we consider the case in which the fractional diffusion only occurs in space. To visualize the problem let us consider, e.g., the generalization to non-integer derivative order of the two-dimensional standard diffusion equation

$$\begin{aligned}\frac{\partial u(x,y,t)}{\partial t} &= d(x,y,t)\left(\frac{\partial^{\alpha} u(x,y,t)}{\partial_{+} x^{\alpha}} + \frac{\partial^{\alpha} u(x,y,t)}{\partial_{\_} x^{\alpha}}\right) + \\ &+ e(x,y,t)\left(\frac{\partial^{\beta} u(x,y,t)}{\partial_{+} y^{\beta}} + \frac{\partial^{\beta} u(x,y,t)}{\partial_{\_} y^{\beta}}\right) + v(x,y,t),\end{aligned} \tag{1}$$

where $\frac{\partial^{\alpha}}{\partial_{\pm} x^{\alpha}}, \frac{\partial^{\beta}}{\partial_{\pm} y^{\beta}}$ are the left (+) and right (−) fractional derivative operators of

[1] Taken from Katugampola [2014]

order $\alpha, \beta \in (1,2)$ in x- and y-variable, respectively, $d(x,y,t), e(x,y,t)$ are the diffusion coefficients and $v(x,y,t)$ is the forcing term. Concerning the boundary conditions, we mention that, due to the non-locality of the fractional operators, in order to have a well-posed Cauchy problem, the behavior of u must be specified not just at the boundary but also at all points exterior to the domain Defterli et al. [2015].
Of course, when $\alpha = \beta = 1$ equation (1) is an advection equation, while for $\alpha = \beta = 2$ equation (1) is a diffusion equation. Therefore, it is clear that varying $\alpha, \beta \in (1,2)$ equation (1) models different phenomena ranging from advection to diffusion.
The improved physical description of the considered phenomenon, however, translates in a harder numerical treatment of the corresponding discretized problem. Indeed, even when standard local discretization methods like finite differences or finite elements are adopted, the non-locality of the fractional operators $\frac{\partial^\alpha}{\partial_\pm x^\alpha}, \frac{\partial^\beta}{\partial_\pm y^\beta}$ causes absence of sparsity in the discretization matrices Ghorbani and Baleanu [2020]; Breiten et al. [2014]. The good news is that, in presence of uniform grids, and some other special non-uniform meshes, the discretization matrices show a *Toeplitz-like structure* Donatelli et al. [2016, 2018]; Breiten et al. [2014], in the sense that they are expressed as a sum of products between diagonal and (dense) Toeplitz matrices, and this paves the way for the design of iterative solvers specialized for Toeplitz-like linear systems.
Toeplitz matrices Grenander and Szegö [1958] are structured matrices which allow fast matrix-vector products through the *Fast Fourier Transfrom* (FFT) algorithm Brigham [1973] with a computational cost of $O(N \log N)$ operations, where N is the matrix-size. Moreover, in the case of a Toeplitz matrix, the singular value distribution of the matrix can be approximated through a uniform sampling of the function associated to the matrix, called symbol. This spectral result can be extended to the Toeplitz-like matrices through the so-called *Generalized Locally Toeplitz* (GLT) theory, which is briefly introduced in Section 1.3.2 and widely used throughout the thesis. In case of Hermitian matrices, these results can be extended to the eigenvalues.

Throughout the history of fractional calculus, several definitions for the non-integer order derivative operators $\frac{\partial^\alpha}{\partial_\pm x^\alpha}$ have been proposed. Three popular definitions were given by Grünwald-Letnikov (GL), Riemann-Liouville (RL) and Caputo Podlubny [1998]; Samko et al. [1993]. The GL fractional derivative, differently from the RL and Caputo operators, is expressed as a weighted sum of uniformly sampled function. It is usually employed to discretize the RL operator whenever uniform meshes are adopted, under proper smoothness assumptions.

The RL and Caputo fractional derivative operators are each represented as the convolution between a singular kernel and a function. In some cases, the singular kernel causes the solution to exhibit weak singularities near the boundaries and this translates in a harder numerical treatment of the corresponding discretized problems Stynes [2019]; Gracia et al. [2018]; Stynes [2016].
Attempting to avoid this problem, some authors have proposed modifications to the RL and Caputo fractional derivatives, where the singular kernel is replaced by a non-singular one. Even if this new operators are widely used in literature, unfortunately, *these operators with non-singular kernels have serious shortcomings that strongly discourage their use* Diethelm et al. [2020]. Therefore, in this work we will only consider the three aforementioned classical fractional derivative operators.

Traditionally, FDEs were solved through the Gaussian elimination with the prohibitive computational cost of $O(N^3)$ operations Agrawal [2006]; Ashyralyev et al. [2009]. In case of a time-dependent FDE the computational cost further increases, since the Gaussian elimination has to be run at each time step. Then, with the growing interest in FDEs, specialized solvers were developed.
In this thesis we will design ad-hoc (tailored) multigrid methods (MGMs) Ruge and Stüben [1987]. MGMs consist in the combination of two iterative methods known as smoother and Coarse Grid Correction (CGC). The smoother is usually a stationary iterative method like weighted Jacobi or Gauss-Seidel, while the CGC has the purpose to speed up the convergence in the subspace where the smoother is not efficient, by projecting the error equation over a coarser grid, then projecting back the solution over the fine grid and finally applying the correction to the solution approximated by the smoother. Projectors and smoother are set according to the spectral properties of the coefficient matrix, which, in case of structured matrices, are retrieved through a symbol-based analysis. The resulting algorithm is called Two-Grid Method (TGM). An extension which recursively calls TGM, instead of directly solving the error equation on the coarse level, is called V-ccycle.
In order to increase their robustness, MGMs are often used as preconditioners Trottenberg et al. [2000] for Krylov methods, e.g., the well-known *Generalized Minimum RESidual* (GMRES) Liesen and Strakos [2013]. The idea behind preconditioning is to construct a matrix P, whose inverse is cheap to compute and such that the preconditioned linear system $P^{-1}Ax = P^{-1}b$ becomes easier to solve, i.e., $P^{-1}\,A \,\approx\, I$, with I the identity matrix.

Among specialized sequential solvers for time-dependent FDEs similar to the one in equation (1), for the one-dimensional case we mention the circulant preconditioning in Lei and Sun [2013], the MGM in Pang and Sun [2012], and the structure preserving tridiagonal preconditioners in Donatelli et al. [2016]. The latter preconditioners were motivated by the spectral study of the coefficient matrices through the notion of symbol and by the GLT theory.

In the two-dimensional setting, classical preconditioners based on multilevel circulant matrices are not well-suited, while MGMs, possibly used as preconditioners, can be effective and robust solvers. Moreover, a two-dimensional space involves two fractional derivative operators, one for each dimension, with two potentially different fractional derivative orders α, β. In the case where $\alpha \approx \beta$, some multigrid proposals for FDEs discretized with finite differences can be found in Lin et al. [2017b,a]; Moghaderi et al. [2017]. In the first reference, a Toeplitz splitting preconditioner is combined with a MGM having block Jacobi as smoother. In Lin et al. [2017a], a multigrid approach suitable for non-rectangular domains has been designed employing two banded splitting iteration schemes as pre-smoother and post-smoother. In Moghaderi et al. [2017], the spectral approach presented in Donatelli et al. [2016] has been extended to two-dimensional FDEs and has been used to define a multigrid preconditioner built using either rediscretization or a two-dimensional scaled-Laplacian matrix, which is particularly effective when the fractional orders are both close to 2, i.e., when equation (1) is close to a standard integer-order diffusion equation. When finite element or finite volume discretizations are adopted, MGMs are investigated in Jiang and Xu [2015] and Donatelli et al. [2018], respectively.

All the aforementioned multigrid strategies do not take into account the case where α and β are far from each other, i.e., when the FDE problems suffer from some sort of anisotropy. By anisotropy we mean a stronger diffusion along one coordinate axis with respect to the other. In Chapter 2, as a possible source of anisotropy for two-dimensional FDE problems, as in (1), we consider a large difference in the fractional derivative orders. Precisely, when $\alpha \approx 1$ and $\beta \approx 2$, or vice versa, the problem shows an intrinsic anisotropy in the coordinate corresponding to the minimum fractional order. Another source of anisotropy is related to the diffusion coefficients that multiply the fractional operators. The latter kind of anisotropy will not be treated in this thesis. The anisotropy can be emphasized or reduced by the choice of the space grid widths. An ad-hoc grid could potentially avoid the anisotropy, but, as a drawback, in case of a strong anisotropy the ad-hoc grid could be much finer than the actually needed one, leading to an increase in the computational effort required for solving the equation.

Efficient multigrid strategies for anisotropic integer order partial differential equations (PDEs) have been investigated in Fischer and Huckle [2006, 2008]; Oosterlee [1995]; Van Lent and Vandewalle [2002]; Washio and Oosterlee [1998]. In Fischer and Huckle [2006], the authors propose a V-cycle with ad-hoc projectors combined with the weighted Jacobi smoother, and they prove the optimality of the corresponding TGM version. In Fischer and Huckle [2008], the same authors focus on the use of more sophisticated smoothing techniques such as block Jacobi method. An alternative strategy that combines a standard V-cycle and that uses a V-cycle with semi-coarsening as smoother has been proposed in Oosterlee [1995]; Van Lent and Vandewalle [2002]; Washio and Oosterlee [1998]. The resulting algorithm is the non-standard MGM known as “multigrid as smoother" (MG-S).

In Chapter 2, inspired by the ideas in Fischer and Huckle [2006, 2008]; Oosterlee [1995]; Van Lent and Vandewalle [2002]; Washio and Oosterlee [1998] on MGMs for anisotropic PDEs and by the results in Moghaderi et al. [2017] on MGMs for isotropic FDEs, we consider an adaptation to our case of the original MG-S algorithm as GMRES preconditioner, which turned out to be robust with respect to the anisotropy of the problem. Concerning the Jacobi method inside the V-cycle smoother, we extend the spectral analysis in Moghaderi et al. [2017] to the anisotropic case and then we use the retrieved information for estimating the smoothing parameter of weighted Jacobi. We stress that this approach, introduced in Section 2.4, will be used to estimate the Jacobi weight of most of the ad-hoc MGMs we develop in this work, allowing to build a versatile parameter free MGM.

Due to the sequentiality of time integration, with none of the aforementioned approaches we can aspire towards complete independence of time of the overall computational cost. By contrary, an all-at-once rephrasing of the discretized problem over a uniform space-time grid, obtained by considering the time as an additional dimension, yields large (multilevel) Toeplitz linear systems and opens to parallelization.

In this regard, we mention the banded Toeplitz preconditioner proposed in Zhao et al. [2020] for solving non-linear space-FDEs, and the block structured preconditioner given in Bertaccini and Durastante [2019] for dealing with arbitrary dimensional space problems. In Gu et al. [2015]; Ran and Zhang [2020] a Strang-type circulant preconditioner for solving FDEs by boundary value methods has been proposed. In the case of equal left and right diffusion coefficients, we also mention the multigrid reduction in time (MGRIT) discussed in Yue et al. [2019]. Therein, the authors consider finite elements in space and Crank-Nicolson in time, since the MGRIT is specifically tailored for one step methods.

Chapter 3 fits within this framework. Precisely, we build a fast and efficient parallel-in-time structure-based multigrid solver. We fix our attention on the one-dimensional version of the FDE treated in Chapter 2. We stress that this one-dimensional problem turns out to be already a tough one, due to the block structure of the coefficient matrix and to its possibly anisotropic character because of the grid choice and the diffusion coefficients.
For the time discretization, we opt either for Crank-Nicolson (CN) or second-order Backward-Difference Formula (BDF2) schemes. The unconditional stability of CN combined with a Weighted and Shifted Grünwald Difference (WSGD) space discretization has already been proven in Tian et al. [2015]. Concerning BDF2, in Section 3.2 we extend the result of unconditional stability, obtained in Liao et al. [2018] for central finite difference space discretization scheme, to the case of a WSGD space discretization scheme.
In order to build a parallel-in-time multigrid, we consider block Jacobi as smoother, since it is parallelizable and, unlike standard weighted Jacobi, does not ask for a parameter estimation. Moreover, exploiting the Toeplitz structure of the coefficient matrices and related symbols, we define the projectors according to what has been done in the integer-order literature for both isotropic Arico et al. [2004] and anisotropic Toeplitz linear systems Fischer and Huckle [2006].
The performances of the proposed MGMs reveal sensitive to the choice of the time discretization scheme. Indeed, many numerical tests show that Crank-Nicolson prevents the multigrid to yield good convergence results, while BDF2 scheme allows good convergence under certain conditions on the grid and the diffusion coefficients.

Aiming at extending our multigrid approaches to different FDEs and discretization approaches, in Chapter 4 we focus on a two-dimensional conservative steady-state Riesz FDE. As is typical for problems in conservative form Eymard et al. [2000], we adopt a Finite Volume (FV) discretization approach. Precisely, we use both classical FVs and the so-called Finite Volume Elements (FVEs), which consists in a mixed approach between FV and finite elements. While FVEs have already been applied in the context of FDEs Wang and Du [2013]; Liu et al. [2014], classical FVs have only been previously applied in first order discretizations Hejazi et al. [2013, 2014].
The one-dimensional version of such FDE was first treated by FVE in Wang and Du [2013], and a FVE method for a two-sided time-dependent space-FDE was introduced in Liu et al. [2014]. In Feng et al. [2015], the latter scheme was proven to be unconditionally stable and convergent with second-order accuracy. A FV approach to solve an advectiondispersion equation with constant dispersion

coefficient was given in Zhang et al. [2005]. In Hejazi et al. [2014] Hejazi et al. proved its stability and first-order accuracy, while a second-order FV discretization was missing in the literature.
Less work has been done in the treatment by finite volume-based methods of the two-dimensional FDE. In Jia and Wang [2016] Jia and Wang presented a fast FVE method for conservative space-FDEs with variable coefficients on convex domains, while in Yang et al. [2014] Yang et al. extended the FV method to the two-dimensional fractional Laplacian.
Therefore, in Chapter 4 we introduce a new second-order FV discretization over uniform meshes, based on a modification of the WSGD scheme used in Chapter 2. This again leads to Toeplitz-like linear systems, whose structure is exploited to build an ad-hoc multigrid solver, which is then compared to other solvers in literature.
The absence of time dependency of the considered FDE allows to focus on the space discretization scheme avoiding to concern about the stability of the time stepping algorithm. An eventual adaptation of the multigrid-based solver to the sequential time stepping algorithm is almost straightforward and would essentially follow the work done in Chapter 2.

Throughout our studies, we came across models whose solution presents singularities near the boundaries. Since MGMs turned out to be robust solvers and preconditioners when uniform meshes were employed, in Chapter 5 we aim at extending the strategies learned in the previous works to the non-uniform case for solving a conservative Caputo FDE, whose solution is known to exhibit singularities near the boundaries Kopteva and Meng [2020]; Stynes [2016]. We introduce a grid mapped by a smooth function which yields a graded mesh near the singularity and a uniform mesh where the solution is smooth. This gives rise to a partially Toeplitz discretization matrix and allows to speed up the matrix-vector products. Then, through the spectral study of the coefficient matrix by means of the GLT theory, we build an ad-hoc multigrid-based preconditioner. In the numerical results section, we test the behavior of our multigrid proposal over different grids and make comparisons with the composite mesh used in Jia and Wang [2015] and their respective proposal, which consists in a circulant based preconditioner. Our numerical results suggest that MGMs are also suitable preconditioners for FDEs when non-uniform meshes are employed.

We stress that our symbol-based weight estimate for Jacobi, which is used in all our multigrid proposals, except for Chapter 3, has a low computational impact, is versatile and makes multigrid a parameter free solver, avoiding a long and not trivial spectral analysis. Finally, throughout our studies we noticed similar spectral properties of the discretization matrices, i.e., similar ill-conditioning, which suggests that MGMs could be applied in a wider variety of FDEs then the ones considered by us.

The thesis is organized as follows.

- In Chapter 1 we fix the notation and recall definitions and the basic tools needed to understand the reminder of this work. We give the definition of Toeplitz and multilevel Toeplitz matrices and briefly introduce the GLT theory, which is needed to retrieve the spectral properties of the discretization matrices through the notion of symbol. Then we introduce MGMs, by giving the algorithm and recalling some convergence results, and preconditioning, by giving the idea behind it and by briefly introducing Krylov methods. Finally, we recall the construction of the GL, RL and Caputo fractional derivatives.

- In Chapter 2 we deal with an anisotropic time-dependent two-dimensional space FDE. Therein we focus on the efficient solution of the two-level Toeplitz-like linear systems which arise from the discretization of the FDE over an equispaced mesh through the largely used CN-WSGD scheme, consisting in a sequential-in-time second-order accurate finite difference scheme. We provide a spectral study for the anisotropic case and we use the retrieved information for estimating the smoothing parameter of weighted Jacobi. Then we define the non-standard MGM, called MG-S, as preconditioner for GMRES which is applied to a band approximation of the coefficient matrix. In the numerical results section, many tests show MG-S to be robust with respect to the anisotropy of the problem, even in the case of a severe anisotropy.

- In Chapter 3, we consider the one-dimensional case of the FDE treated in Chapter 2 and we develop a parallel-in-time multigrid solver. The parallel-in-time interpretation comes from a reinterpretation of the sequential-in-time discrete equation in which the time dimension is considered as an additional dimension and gives rise to a larger discretization matrix. Such matrix has a block lower triangular structure, due to the sequentiality of time, where the number of blocks is equal to the amount of time steps and each block represents the discretization along the space dimension of the FDE. We consider two different time schemes, such as CN and the second-order accurate BDF2, and we provide the spectral study for both. The retrieved information is used to build a parallel-in-time multigrid solver. The addition of the time dimension in the coefficient matrix introduces a further degree of difficulty, since anisotropy can occur between space and time. In the numerical results section we provide many tests which show that BDF2 is more suitable for parallel-in-time integration through MGMs than CN.

- In Chapter 4, we deal with other discretization approaches, specifically FVs and FVEs, for solving a two-dimensional conservative FDE discretized over an equispaced mesh. Therein we develop a new second-order accurate FV scheme, we provide a spectral study of the discretization matrix and we build an ad-hoc multigrid solver for both discretization approaches. In the numerical results section we compare the two approaches in terms of approximation error and behavior of MGM.
- In Chapter 5, we deal with a conservative Caputo FDE, where the fractional derivative operator is a convex combination between the left and right Caputo fractional derivatives. This kind of equation is known to yield singular solution even in the case of smooth diffusion coefficients and forcing term. Therefore, we adopt non-uniform meshes and we provide the full discretization over an unstructured arbitrary mesh through the FVE approach and, then, we exploit the GLT theory to retrieve the symbol of the coefficient matrix in the case of a mesh mapped by a non-linear function. The spectral information are used to develop a parameter free multigrid preconditioner for GMRES, which in the numerical results section is proven to yield good convergence results even in the case of a strong singularity.

All our main findings are summarized in the conclusion chapter.

Note that Chapters 2 to 5 are an adaptation of papers Donatelli et al. [2020]; Donatelli, Krause, Mazza and Trotti [2021a]; Donatelli, Krause, Mazza, Semplice and Trotti [2021]; Donatelli, Krause, Mazza and Trotti [2021b], respectively, in which abstract and introduction have been removed and part of the preliminaries have been moved to Chapter 1. Moreover, in Chapter 3, differently from paper Donatelli, Krause, Mazza and Trotti [2021a] we added Propositions 3.4.1 and 3.4.2 to provide more details regarding the convergence of the used method. Finally, we mention paper Bogoja et al. [2021] where we used the GLT approach, but that we did not discuss in this work as out-of-scope with respect to our main purpose of providing fast solvers for FDEs.

Chapter 1

Preliminaries

The purpose of this chapter is to fix the notation, to introduce special matrix structures, multigrid methods and different definitions of fractional derivatives, which will be used throughout the thesis. In particular, in Sections 1.1 and 1.2 we respectively provide a list of notations and acronyms. In Section 1.3 we illustrate some known tools necessary to retrieve spectral information of Toeplitz matrices. This tools are then extended to the more general case of Toeplitz-like matrices through the generalized locally Toeplitz theory. In Section 1.4 we introduce multigrid methods and provide convergence results. Then we briefly introduce Krylov methods and we give the idea behind preconditioning. Finally, in Section 1.6 we introduce three famous definitions of fractional derivatives and provide links between them.

1.1 Notation

- $\mathbb{R}^{N\times M}, \mathbb{C}^{N\times M}$, are the linear spaces of the real or complex $N \times M$ matrices, respectively;
- $I_N \in \mathbb{R}^{N\times N}$ is the identity matrix;
- Let $d_1, \dots, d_M \in \mathbb{R}$, then $\mathrm{diag}_{i=1,\dots,M}(d_i)$, defines the $M \times M$ diagonal matrix whose diagonal elements are $d_1, \dots, d_M$;
- Let $D_1, \dots, D_M \in \mathbb{R}^{N\times N}$, then $\mathrm{diag}_{i=1,\dots,M}(D_i)$, defines the $NM \times NM$ block diagonal matrix whose diagonal block are matrices $D_1, \dots, D_M$;
- Let $A \in \mathbb{C}^{N\times N}$, we denote by
  - $\lambda_i(A),\ i = 1, \dots, N$, the eigenvalues of A;
  - $\rho(A) = \max\limits_{i=1,\dots,N} |\lambda_i(A)|$ the spectral radius of A;

  - A^{H} the transpose conjugate of A;
  - $||A||_{\mathrm{tr}}$ the trace norm of A (the sum of its singular values);
  - $||A||_1, ||A||_2, ||A||_\infty$ the induced 1-norm, 2-norm and infinity norm of A, respectively.
- We shorten the notation $\{A_N\}_{N\in\mathbb{N}}$ of a matrix-sequence to $\{A_N\}_N$;
- i is the imaginary unit;
- $\otimes$ denotes the Kronecker product;
- $\mathbb{R}_+, \mathbb{R}_-$ are the sets of positive and negative real numbers, respectively;
- $[a,b]^d$ is the d-dimensional square $[a,b]\times\cdots\times[a,b]\subset\mathbb{R}^d$;
- $\lfloor x\rfloor, \lceil x\rceil$ are the floor and ceiling of x, respectively.

1.2 Acronyms

- **BDF2** second-order Backward-Difference Formula;
- **BJ** Block-Jacobi;
- **BTTB** Block-Toeplitz with Toeplitz-block;
- **CGC** Coarse Grid Correction;
- **CN** Crank-Nicolson;
- **FDE** Fractional Diffusion Equation;
- **FFT** Fast Fourier Transform;
- **FV** Finite Volume;
- **FVE** Finite Volume Element;
- **GL** Grünwald-Letnikov;
- **GLT** Generalized Locally Toeplitz;
- **GMRES** Generalized Minimal RESidual;
- **MGM** MultiGrid Method;
- **MGRIT** MultiGrit Reduction-In-Time;
- **MG-S** MultiGrid as Smoother;
- **PDE** Partial Differential Equation;
- **PFASST** Parallel Full Approximation Scheme in Space and Time;
- **PFCGS** Preconditioned Fast Conjugate Gradient Squared;
- **RL** Riemann-Liouville;
- **SGD** Shifted Grüwald Difference;
- **TGM** Two-Grid Method;
- **WSGD** Weighted and Shifted Grünwald Difference.

1.3 Structured matrices

It is often the case that the discretization of a continuous linear operator yields structured matrices. The knowledge on the structure is of crucial importance when dealing with the solution of the resulting linear systems. In this section we will focus on Toeplitz and Toeplitz-like structures.

Toeplitz matrices are key structured matrices which arise from the discretization of shift invariant one-dimensional operators over uniform meshes, for instance standard integer-order derivatives Serra-Capizzano [2002], integral operators Böttcher et al. [2012] and also fractional differential operators Tian et al. [2015]; Donatelli et al. [2016, 2018]. In the case of a multi-dimensional operator, the resulting discretization matrix shows a multilevel Toeplitz structure. Such a structure is not only interesting from a computational point of view, allowing fast matrix-vector products, but it also provides tools to retrieve spectral information.

In some cases, the discretization yields Toeplitz-like matrices, which consists in apparently unstructured matrices with an hidden structure. Under proper hypothesis, the spectral tools for Toeplitz matrices can be generalized to deal with Toeplitz-like matrices, by means of the *Generalized Locally Toeplitz* (GLT) theory. Finally, we briefly introduce circulant matrices, which are a subset of Toeplitz matrices, and which will be often used as the main ingredient (preconditioner) of a comparison solver for our multigrid proposals.

1.3.1 Toeplitz matrices

In order to explore the properties of such matrices to our aims, we first recall some basic definitions, see, e.g., Bini et al. [1988].

Definition 1.3.1. *Let $f \in \mathrm{L}^1([-\pi,\pi])$ and let $\{f_k\}_{k\in\mathbb{Z}}$ be the sequence of its Fourier coefficients defined as*

$$f_k = \frac{1}{2\pi}\int_{-\pi}^{\pi} f(\theta)e^{-\mathrm{i}k\theta}\mathrm{d}\theta,$$

then the Toeplitz matrix associated with f has the form

$$T_N = \begin{pmatrix} f_0 & f_{-1} & \cdots & f_{-N+2} & f_{-N+1} \\ f_1 & f_0 & f_{-1} & & f_{-N+2} \\ \vdots & f_1 & \ddots & \ddots & \vdots \\ f_{N-2} & & \ddots & \ddots & f_{-1} \\ f_{N-1} & f_{N-2} & \cdots & f_1 & f_0 \end{pmatrix}.$$

The function $f(\theta) = \sum_{k\in\mathbb{Z}} f_k \mathrm{e}^{\mathrm{i}k\theta}$, is called the generating function *of T_N, therefore we denote such a matrix by $T_N(f)$.*

Remark 1.3.1. *If T_n is generated by f, then ${T_n}^{\mathrm{H}}$ is generated by $\overline{f}$. Hence, T_n is Hermitian whenever f is a real function. Furthermore, in the case where coefficients f_k are real, it holds that $\overline{f}(\theta) = f(-\theta)$, hence if f is real then it is also even.*

Toeplitz matrices, defined in Definition 1.3.1, are a particular case of the multilevel Toeplitz matrices obtained below.

Definition 1.3.2. *Let $f \in \mathrm{L}^1([-\pi,\pi]^d)$ and let $\{f_{\boldsymbol{k}}\}_{\boldsymbol{k}\in\mathbb{Z}^d}$ be the sequence of its Fourier coefficients defined as*

$$f_{\boldsymbol{k}} := \frac{1}{(2\pi)^d}\int_{[0,2\pi]^d} f(\boldsymbol{\theta})\mathrm{e}^{-\mathrm{i}\langle \boldsymbol{k},\boldsymbol{\theta}\rangle}\mathrm{d}\boldsymbol{\theta},$$

*where $\langle \boldsymbol{k},\boldsymbol{\theta}\rangle = \sum_{t=1}^{d} k_t\theta_t$. Then the d-*level Toeplitz matrix *of partial orders $\boldsymbol{n} = (n_1,\ldots,n_d)$ associated with f is*

$$T_N^{(d)} := \left[f_{\boldsymbol{i}-\boldsymbol{j}}\right]_{\boldsymbol{i},\boldsymbol{j}=\boldsymbol{1}}^{\boldsymbol{n}} = \left[\cdots\left[\left[f_{i_1-j_1,\ldots,i_d-j_d}\right]_{i_d,j_d=1}^{n_d}\right]_{i_{d-1},j_{d-1}}^{n_{d-1}}\cdots\right]_{i_1,j_1=1}^{n_1},$$

where $N = \prod_{i=1}^{d} n_i$ is the order of the matrix.

To clarify the notation, a 2-level Toeplitz matrix of order N generated by f is given by

$$T_N^{(2)}(f) = \left[\left[f_{[i_1-j_1,i_2-j_2]}\right]_{i_2,j_2=1}^{n_2}\right]_{i_1,j_1=1}^{n_1},$$

or equivalently

$$T_N^{(2)}(f) = \sum_{|j_1|\le n_1}\sum_{|j_2|\le n_2} f_{[j_1,j_2]} J_{n_1}^{[j_1]} \otimes J_{n_2}^{[j_2]},$$

where $J_{n_i}^{[j_i]} \in \mathbb{R}^{n_i\times n_i}$ are matrices whose entry (s,t)-th equals 1 if $s-t = j_i$ and 0 elsewhere. In other words, a 2-level Toeplitz matrix is a block Toeplitz whose blocks are Toeplitz, briefly BTTB. When $d = 1$, we simplify the notation using

$$T_N(f) := T_N^{(1)}(f).$$

A key property of Toeplitz matrices is that approximations or bounds for their spectra can be obtained exploiting the generating function. The following theorem gives an estimated range for the real part of eigenvalues of a Toeplitz matrix through the real part of its generating function. As a corollary, an upper bound for the spectral radius follows.

Theorem 1.3.1 (Bini et al. [1988]). *Let $f \in L^1([0,2\pi])$ be a complex-valued function, $A = T_N(f)$, and $\tilde{f} = \frac{f+\bar{f}}{2} =: \mathrm{Re}(f)$. Then, for each λ eigenvalue of A, we have*

$$\min_{\theta\in[0,2\pi]} \tilde{f}(\theta) < \mathrm{Re}(\lambda) < \max_{\theta\in[0,2\pi]} \tilde{f}(\theta).$$

Theorem 1.3.1 extends to the multilevel case as discussed in Arico et al. [2004].

Corollary 1.3.1. *Let $f \in L^1([0,2\pi]^d)$ be a real-valued function and $A = T_N^{(d)}(f)$. Then,*

$$\rho(A) \leq \max_{\boldsymbol{\theta}\in[0,2\pi]^d} |f(\boldsymbol{\theta})| =: \|f\|_\infty .$$

Remark 1.3.2. *We shall notice that if $f_{min} > 0$ or $f_{max} < 0$ then T_N is positive or negative definite, respectively.*

1.3.2 GLT theory

The GLT class is a matrix-sequence algebra obtained as a closure under some algebraic operations between Toeplitz and diagonal matrix-sequences generated by functions. The GLT class includes matrix-sequences coming from the discretization of differential operators with various techniques, such as finite differences, finite volumes, finite elements, Isogeometric Analysis, etc. The formal definition is difficult and involves a heavy notation, therefore we just give a basic definition and report few properties, in the one-dimensional case, that we will need for our studies (for a more detailed discussion see Garoni and Serra-Capizzano [2017] for the unilevel case and Garoni and Serra-Capizzano [2018] for the extension to the multilevel case).

Definition 1.3.3. *A matrix-sequence whose N-th element is a diagonal matrix $D_N = [d_{i,j}]_{i,j=1}^N \in \mathbb{R}^{N\times N}$ such that $d_{i,i} = d\left(\frac{i}{N}\right)$, $i = 1,\dots,N$ with $d : [0,1] \to \mathbb{C}$ a Riemann-integrable function is called* diagonal sampling sequence.

The functions f in definition 1.3.1 and d in definition 1.3.3 allow to estimate the spectrum of the matrix-sequences $\{T_N(f)\}_N$ and $\{D_N\}$, respectively, in the following sense.

Definition 1.3.4. *Let $f : G \to \mathbb{C}$ be a measurable function, defined on a measurable set $G \subset \mathbb{R}^k$ with $k \geq 1$, $0 < m_k(G) < \infty$, where $m_k(G)$ is the Lebesgue measure of the set G. Let $C_0(\mathbb{K})$ be the set of continuous functions with compact support over $\mathbb{K} \in \{\mathbb{R}_0^+, \mathbb{C}\}$ and let $\{A_N\}_N$, be a sequence of matrices of size N with eigenvalues $\lambda_j(A_N), j = 1,\dots,N$ and singular values $\sigma_j(A_N), j = 1,\dots,N$.*

- $\{A_N\}_N$ *is distributed as the pair* (f,G) *in the sense of the eigenvalues, in formulae* $\{A_N\}_N \sim_\lambda (f,G)$*, if the following limit relation holds for all* $F \in C_0(\mathbb{C})$
$$\lim_{N\to\infty} \frac{1}{N}\sum_{j=1}^{N} F(\lambda_j(A_N)) = \frac{1}{m_k(G)}\int_G F(f(t))\mathrm{d}t. \tag{1.1}$$
In this case, we refer to the function f *as (spectral) symbol.*

- $\{A_N\}_N$ *is distributed as the pair* (f,G) *in the sense of the singular values, in formulae* $\{A_N\}_N \sim_\sigma (f,G)$*, if the following limit relation holds for all* $F \in C_0(\mathbb{R}_0^+)$
$$\lim_{N\to\infty} \frac{1}{N}\sum_{j=1}^{N} F(\sigma_j(A_N)) = \frac{1}{m_k(G)}\int_G F(|f(t)|)\mathrm{d}t. \tag{1.2}$$
In this case, we refer to the function f *as singular values symbol.*

Remark 1.3.3. *An informal interpretation of the limit relation* (1.1) *(resp.* (1.2)*) is that when* N *is sufficiently large, the eigenvalues (resp. singular values) of* A_N *can be approximated by a sampling of* f *(resp.* $|f|$*) on a uniform mesh over the set* G*, up to a relatively small number of potential outliers and where "relatively small" means* $o(N)$*.*

For completeness we mention a classic result of distribution in the case of multilevel Toeplitz matrices.

Theorem 1.3.2 (Grenander and Szegö [1958]; Tilli [1998])**.** *Let* $f \in \mathrm{L}^1([0,2\pi]^d)$ *be a complex function, then*
$$\{T_N^{(d)}(f)\}_{N\in\mathbb{N}} \sim_\sigma \left(f,[0,2\pi]^d\right).$$
Furthermore, if f *is a real valued function, it holds*
$$\{T_N^{(d)}(f)\}_{N\in\mathbb{N}} \sim_\lambda \left(f,[0,2\pi]^d\right).$$

Throughout, we use the following notation
$$\{A_N\}_N \sim_{GLT} \psi(x,\theta), \qquad (x,\theta)\in[0,1]\times[0,2\pi],$$
to say that the sequence $\{A_N\}_N$ is a GLT sequence with symbol $\psi(x,\theta)$.

Here we report five main features of the GLT class.

GLT1 Let $\{A_N\}_{N\in\mathbb{N}} \sim_{\mathrm{GLT}} \psi(x,\theta)$ with $\psi:\Omega\to\mathbb{C}$, $\Omega=[0,1]\times[0,2\pi]$, then $\{A_N\}_{N\in\mathbb{N}} \sim_\sigma (\psi,\Omega)$. If the matrices A_N are Hermitian, then $\{A_N\}_{N\in\mathbb{N}} \sim_\lambda (\psi,\Omega)$.

GLT2 The set of GLT sequences form a $*$-algebra, i.e., it is closed under linear combinations, products, inversion (whenever the symbol vanishes, at most, in a set of zero Lebesgue measure), transposed conjugation: hence, the sequence obtained via algebraic operations on a finite set of input GLT sequences is still a GLT sequence and its symbol is obtained by following the same algebraic manipulations on the corresponding symbols of the input GLT sequences.

GLT3 Every Toeplitz sequence $\{T_N(f)\}_{N\in\mathbb{N}}$ generated by a $L^1([0,2\pi])$ function $f(\theta)$ is such that $\{T_N(f)\}_{N\in\mathbb{N}} \sim_{\mathrm{GLT}} f(\theta)$, with the specifications reported in item **GLT1**. Every diagonal sampling sequence $\{D_N(a)\}_{N\in\mathbb{N}}$, where $a(x)$ is a Riemann integrable function, is such that $\{D_N(a)\}_{N\in\mathbb{N}} \sim_{\mathrm{GLT}} a(x)$.

GLT4 Every sequence which is distributed as the constant zero in the singular value sense is a GLT sequence with symbol zero, and viceversa. In formulae, $\{A_N\}_{N\in\mathbb{N}} \sim_\sigma (0,\Omega)$, $\Omega = [0,1]\times[0,2\pi]$, if and only if $\{A_N\}_{N\in\mathbb{N}} \sim_{\mathrm{GLT}} 0$.

GLT5 Let $\{A_N\}_N \sim_{\mathrm{GLT}} \psi(x,\theta)$, $\Omega = [0,1]\times[0,2\pi]$. If we assume that

$$\lim_{N\to\infty} \frac{\left\|A_N - A_N^{\mathrm{H}}\right\|_{\mathrm{tr}}}{N} = 0,$$

then $\psi(x,\theta)$ is necessarily a real-valued function and $\{A_N\}_N \sim_\lambda \psi(x,\theta)$.

An important GLT result that will be needed in Chapter 5 is reported in Proposition 1.3.1 and concerns the symbol of a diagonal-times-Toeplitz matrix-sequence.

Proposition 1.3.1. *Let $\{D_N\}_N$ be a sequence of diagonal sampling matrices with symbol $d : [0,1] \to \mathbb{R}_{>0}$, and $\{T_N(f)\}_N$ be a sequence of Hermitian Toeplitz matrices with symbol $f : [0,2\pi] \to \mathbb{R}$, then*

$$\{D_N T_N(f)\}_N \sim_\lambda \big(d(x)f(\theta), [0,1]\times[0,2\pi]\big).$$

Notice that proposition 1.3.1 is a consequence of the similitude transformation $D_N^{-\frac{1}{2}} D_N T_N(f) D_N^{\frac{1}{2}} = D_N^{\frac{1}{2}} T_N(f) D_N^{\frac{1}{2}}$ and of the hermitianity of $D_N^{\frac{1}{2}} T_N(f) D_N^{\frac{1}{2}}$ which, for **GLT1-3**, ensure

$$\{D_N^{\frac{1}{2}} T_N(f) D_N^{\frac{1}{2}}\}_N \sim_\lambda \big(\sqrt{d(x)} f(\theta) \sqrt{d(x)} = d(x)f(\theta), [0,1]\times[0,2\pi]\big).$$

In case the diagonal-times-Toeplitz structure is hidden, one can resort to the notion of approximating class of sequences and to the GLT result reported in theorem 1.3.3 which allows to find the symbol of a ‘difficult’ matrix-sequence by means of ‘simpler’ matrix-sequences.

Definition 1.3.5. *Let $\{A_N\}_N$ be a matrix-sequence and let $\{\{B_{N,M}\}_N\}_M$ be a sequence of matrix-sequences. We say that $\{\{B_{N,M}\}_N\}_M$ is an* approximating class of sequences *(a.c.s.) for $\{A_N\}_N$ if the following condition is met: $\forall M\ \exists\ N_M$ such that for $N \geq N_M$,*

$$A_N = B_{N,M} + R_{N,M} + N_{N,M}, \quad \operatorname{rank}(R_{N,M}) \leq c(M)N, \quad \left\|N_{N,M}\right\|_2 \leq \omega(M),$$

where $N_M, c(M), \omega(M)$ depend only on M with

$$\lim_{M\to\infty} c(M) = \lim_{M\to\infty} \omega(M) = 0.$$

Theorem 1.3.3 (Garoni and Serra-Capizzano [2018])**.** *Let $\{A_N\}_N$ be a matrix-sequence. If there exists an a.c.s. $\{\{B_{N,M}\}_N\}_M$ for $\{A_N\}_N$ such that $\{\{B_{N,M}\}_N\}_M \sim_\sigma (f_{N,M}, G)$, with $f_{N,M}$ that converges in measure to f, then*

$$\{A_N\}_N \sim_\sigma (f, G).$$

1.3.3 Circulant matrices

A special case of the Toeplitz matrices are the circulant matrices, whose structure allows a fast matrix-vector product.

Definition 1.3.6. *A circulant matrix $C_N \in \mathbb{C}^{N\times N}$ has the form*

$$C_N = \begin{bmatrix} c_0 & c_{N-1} & \cdots & c_2 & c_1 \\ c_1 & c_0 & c_{N-1} & & c_2 \\ \vdots & c_1 & c_0 & \ddots & \vdots \\ c_{N-2} & & \ddots & \ddots & c_{N-1} \\ c_{N-1} & c_{N-2} & \cdots & c_1 & c_0 \end{bmatrix}.$$

Let $C_N \in \mathbb{C}^{N\times N}$ and $x \in \mathbb{C}^N$, then the matrix-vector product $y = C_N x$ can be performed through the "Fast Fourier Transform" (FFT) algorithm, see, e.g., Brigham [1973], with a computational cost of $\mathrm{O}(N \log(N))$, in the following way:

$$y = IFFT\left(FFT(C_N^{(1)}) \circ FFT(x)\right), \tag{1.3}$$

where $C_N^{(1)}$ is the first column of C_N, $IFFT(\cdot)$ is the Inverse FFT algorithm and $\circ$ is the Hadamard product (point-wise product). Note that the IFFT algorithm is essentially the same algorithm as the FFT, up to a proper permutation and scaling. This means that the computational cost is still $\mathrm{O}(N \log(N))$.

Remark 1.3.4. *It is important to observe that any Toeplitz matrix can be embedded into a circulant matrix, which will be at least twice as big. More precisely, if we choose a proper dimension N of the Toeplitz matrix, the product Toeplitz-vector is performed in $\frac{3}{2}2N\log_2(2N)$ operations.*

1.4 Multigrid Methods

In this section we introduce multigrid methods (MGMs), which are efficient algorithms for solving large linear systems deriving, for instance, from the discretization of PDEs. In literature there are many different MGMs (see, e.g., the V-cycle, W-Cycle and FMG in Briggs et al. [2000]) and, in general, a MGM is an iterative methods defined by extending recursively the two-grid algorithm, to be described below.

1.4.1 Two-Grid method

Two-grid methods (TGM) combine two iterative methods known as smoother and Coarse Grid Correction (CGC). Given the linear system $A_N x = b$, $A_N \in \mathbb{R}^{N\times N}$, the former is typically a stationary iterative method of the form

$$x^{(j+1)} = \left(I_N - W_N^{-1}A_N\right)x^{(j)} + W_N^{-1}b =: \mathscr{S}_N(x^{(j)}, W_N, A_N, b), \tag{1.4}$$

where $W_N \in \mathbb{C}^{N\times N}$ is the preconditioner and the iteration matrix is $S_N = I_N - W_N^{-1}A_N$. Given a full-rank matrix $P_N \in \mathbb{C}^{N\times k}$, with $k < N$, a step of TGM is defined by Algorithm 1.

Algorithm 1 Two-grid method

```
$x^{(j+1)} = \mathbf{TGM}\left(\mathscr{S}_N, P_N, W_N, A_N, x^{(j)}, b\right)$
1) $r_N = b - A_N x^{(j)}$
2) $r_k = P_N^{\mathrm{T}} r_N$
3) $A_k = P_N^{\mathrm{T}} A_N P_N$        (Galerkin approach)
4) Solve $A_k y = r_k$
5) $\hat{x}^{(j)} = x^{(j)} + P_N y$
6) $x^{(j+1)} = \mathscr{S}_N^{\nu}\left(\hat{x}^{(j)}, W_N, A_N, b\right)$
```

Steps 1) to 5) define the CGC, which depends on the projection operator P_N and step 6) consists in applying ν times the "post-smoothing iteration" and hence has iteration matrix S_N^{ν}. To strengthen the algorithm, a second smoother, which is called pre-smoother, could be added before the CGC. Step 4) is often computationally expensive, therefore the direct solution can be approximated by calling recursively the TGM. This process can be iterated until a small dimension linear system is reached in order to compute the direct solution with a low computational cost. The resulting algorithm is known as V-cycle.

In the case of a V-cycle, the Galerkin approach in step 3), i.e., the computation of A_k through products with projectors P_N, P_N^{T} and A_N, could modify the structure of the matrix A_N at lower levels and therefore step 3) could be replaced by a geometric approach, which consists in the rediscretization of the equation over a coarser grid. This leads to a less robust algorithm but allows to maintain the same structure at each level. Moreover, if the matrix A_N shows a Toeplitz-like structure, by choosing the geometric approach the matrix-vector products can be performed by means of the FFT algorithm with a computational cost of $O(N \log N)$ (see Remark 1.3.4).

1.4.2 TGM convergence results

The iteration matrix of the TGM in Algorithm 1 is given by

$$Z_N = S_N^{\nu} \left[I - P_N \left(P_N^{\mathrm{T}} A_N P_N \right)^{-1} P_N^{\mathrm{T}} A_N \right].$$

The basis for the TGM convergence analysis is Theorem 1.4.1 below, given in Ruge and Stüben [1987]. For the sake of simplicity, in the following we assume that the smoothing part is only one post-smoother iteration.

Theorem 1.4.1 (Ruge-Stüben Ruge and Stüben [1987]). *Let A_N be a positive definite matrix of size N and let S_N be the post smoothing iteration matrix. Suppose that $\exists \delta > 0$ independent of N such that*

$$\|S_N x\|_{A_N}^2 \le \|x\|_{A_N}^2 - \delta \|x\|_{A_N D_N^{-1} A_N}^2, \qquad \forall x \in \mathbb{C}^N, \tag{1.5}$$

where D_N is the diagonal matrix having the same diagonal of A_N. Assume that $\exists \epsilon > 0$ independent of N such that

$$\min_{y \in \mathbb{C}^k} \|x - P_N y\|_{D_N}^2 \le \epsilon \|x\|_{A_N}^2, \qquad \forall x \in \mathbb{C}^N. \tag{1.6}$$

Then $\epsilon \ge \delta$ and

$$\|Z_N\|_{A_N} \le \sqrt{1 - \frac{\delta}{\epsilon}}.$$

Equations (1.5) and (1.6) are known as *smoothing property* and *approximation property*, respectively. The smoothing property can usually be satisfied by classical iterative methods like Gauss–Seidel and weighted Jacobi, with a proper choice of the weight, while the approximation property typically requires a nontrivial analysis.

Smoothing analysis. In the following we only consider weighted Jacobi or block Jacobi methods as smoother. Two main advantages of such methods are that they can be parallelized and that the Toeplitz-like structure of the coefficient matrix can be exploited to prove related convergence results.

Consider the linear system $A_N x = b$, we recall that the weighted Jacobi and block Jacobi methods are obtained from equation (1.4) choosing $W_N = \frac{1}{\omega} D_N$ or $W_N = \frac{1}{\omega}\tilde{D}_N$, where $D_N, \tilde{D}_N$ are the main diagonal and block diagonal of A_N, respectively, and $\omega > 0$ is the weight or relaxation parameter.

Remark 1.4.1. *If $W_N = \frac{1}{\omega_{\mathrm{R}}} I_N$, then equation* (1.4) *defines the weighted Richardson iteration which is well-known to satisfy the smoothing property* (1.5)*, for positive definite matrices A_N, whenever it is convergent Donatelli et al. [2015].*

Moreover, we recall the convergence theorem of weighted Jacobi.

Theorem 1.4.2. *Let A_N be a positive definite matrix, then weighted Jacobi converges if it holds $0 < \omega < \frac{2}{\rho\left(D_N^{-1} A_N\right)}$.*

In practical application the convergence rate of multigrid strongly depends on the choice of the weight of Jacobi, hence numerical estimates are needed.

Approximation property. The approximation property (1.6) relies on the study of the grid transfer operators. In order to show how such operators are defined let $M = (N - (N \bmod 2))/2$, and define the one-dimensional down-sampling matrix $K_N^M \in \mathbb{R}^{N \times M}$ as

$$[K_N^M]_{i,j} := \begin{cases} 1 & \text{if } i = 2j - (N+1) \bmod 2, \\ 0 & \text{otherwise}, \end{cases} \qquad j = 1, \dots, M.$$

The 1-dimensional projector $P_N \in \mathbb{R}^{N \times M}$ is then defined as

$$P_N := T_N(p) K_N^M, \tag{1.7}$$

where p is usually a low order trigonometric polynomial, which yields a sparse projector and allows to keep a low computational cost.

The approximation property (1.6) has been proven for matrix trigonometric algebras under the following condition on the polynomial p and on the symbol f of the coefficient matrix A_N (see Serra-Capizzano [2002])

$$\limsup_{\theta \to \theta_0} \frac{p(\hat{\theta})^2}{f(\theta)} = \mathfrak{c} < +\infty, \qquad \forall \hat{\theta} \in \mathscr{M}(\theta), \tag{1.8}$$

with $f \geq 0$ that vanishes only at θ_0, and $\mathscr{M}(\theta) = \{\pi - \theta\}$ the set of the “mirror points" of θ in the 1-dimensional case.

Concerning the 2-dimensional case, the projector $P_N \in \mathbb{R}^{N \times M}$, with $N = n_1 n_2$, $M = m_1 m_2$, is then defined as

$$P_N := T_N^{(2)}(p) U_N^M, \tag{1.9}$$

where $U_N^k = K_{n_1}^{m_1} \otimes K_{n_2}^{m_2}$ is the 2-dimensional down-sampling operator.

Let f be the symbol of the coefficient matrix A_N, then the 2-dimensional version of condition (1.8) is the following (see Serra-Capizzano [2002])

$$\limsup_{\boldsymbol{\theta} \to \boldsymbol{\theta}_0} \frac{p(\hat{\boldsymbol{\theta}})^2}{f(\boldsymbol{\theta})} = \mathfrak{c} < +\infty, \qquad \forall \hat{\boldsymbol{\theta}} \in \mathscr{M}(\boldsymbol{\theta}), \tag{1.10}$$

with $f \geq 0$ that vanishes only at $\boldsymbol{\theta}_0$, and

$$\mathscr{M}(\boldsymbol{\theta}) = \{(\theta_1, \pi - \theta_2), (\pi - \theta_1, \theta_2), (\pi - \theta_1, \pi - \theta_2)\}$$

the set of the “mirror points" of $\boldsymbol{\theta}$ in the 2-dimensional case.

Remark 1.4.2. *For*

$$p(\theta_1, \theta_2) = (1 + \cos\theta_1)(1 + \cos\theta_2), \tag{1.11}$$

i.e., fixed P_N as the standard linear interpolation, relation (1.10) *holds true whenever f has a zero of order smaller or equal to 4 at $\boldsymbol{\theta}_0 = \mathbf{0}$.*

In the case where f vanishes on a whole line, i.e., when the equation suffers from anisotropy, the theory does not apply anymore. We refer the reader to Chapter 2 Section 2.3 for a detailed discussion about how to deal with such problems in the FDE context.

1.4.3 V-cycle convergence results

The convergence analysis of the V-cycle is much more involved and a linear convergence rate has been proven, under a condition stricter than (1.10), only when the Galerkin approach is considered and for matrices in some trigonometric algebras, see Arico et al. [2004]. In details, the symbol p of the grid transfer operator has to satisfy

$$\limsup_{\theta\to\theta_0} \frac{p(\hat{\boldsymbol{\theta}})}{f(\boldsymbol{\theta})} = \mathfrak{c} < +\infty, \qquad \forall \hat{\boldsymbol{\theta}} \in \mathscr{M}(\boldsymbol{\theta}), \tag{1.12}$$

which has to hold at each level of the V-cycle hierarchy.

Note that, compared to (1.10), relation (1.12) does not have the power two in the numerator and hence p has to vanish at the mirror points with double order.

1.5 Krylov methods and preconditioning

Krylov methods Liesen and Strakos [2013] are a family of algorithms for solving $A_N x = b$ that search for an approximate solution on a Krylov subspace. The r-th Krylov subspace, denoted by $\mathscr{K}_r(A_N, b)$ is the vector space spanned by the vectors $b, A_N b, ..., {A_N}^{r-1} b$, i.e.,

$$\mathscr{K}_r(A_N, b) = \text{Span}\left(b, A_N b, ..., {A_N}^{r-1} b\right).$$

In particular, the Generalized Minimum RESidual method (GMRES) is a Krylov method that computes at the r-th step the best least-squares solution $x^{(r)}$ from the Krylov subspace $\mathscr{K}_r(A_N, b)$. Specifically, the GMRES method successively solves the following least squares problems:

$$\min_{x^{(r)} \in \mathscr{K}_r(A_N, b)} \left\| b - A_N x^{(r)} \right\|_2, \qquad r \geq 1.$$

Krylov methods are non-stationary iterative methods due to the iteration matrix being dependent on the iteration r. They are known to converge faster than most basic iterative methods, e.g., the class of methods defined in equation (1.4), but when the matrix is ill-conditioned the iterations to convergence still increase with N. In such cases a preconditioning strategy is required.

The philosophy behind the preconditioning is to reduce the condition number of the coefficient matrix, by clustering as much as possible the spectrum of the

coefficient matrix in order to speed up the convergence of the method. Indeed, it is well-known that the more the spectrum is clustered, the higher is the convergence rate of a Krylov method.
In detail, consider the preconditioned linear system $P^{-1}A_N x = P^{-1}b$, where P is the preconditioner. P should be such that P^{-1} is cheap to compute and $P^{-1}A_N$ is as close as possible to an identity matrix.
When considering multigrid as preconditioner, the inversion of P is performed through few iterations of multigrid. In the case where $P = A_N$, the more iterations of multigrid are performed the more P^{-1} is close to ${A_N}^{-1}$. Usually, in order to keep the computational cost as low as possible, only one iteration is performed.
When matrix A_N shows a dense Toeplitz or block Toeplitz structure, to further reduce the computational cost of the preconditioning iteration, a band or block band approximation of matrix A_N is usually considered (see, e.g., the preconditioner defined in Section 2.3).

1.6 Fractional Operators

The aim of this section is to define both fractional integration and differentiation operators with a single formula. This formula will then define integro-differential operators of arbitrary order Samko et al. [1993].
Throughout the history of fractional calculus, several non-equivalent definitions for the non-integer order derivative operators have been proposed. Here we will briefly introduce the three most popular definitions given by GL, RL and Caputo Podlubny [1998]; Samko et al. [1993]. The RL and Caputo fractional derivative operators are obtained from the generalization of the iterated integral, while GL derivative is obtained from the generalization of the integer-order derivative. However, under proper conditions the three definitions coincide (more details in the subsequent sections).

1.6.1 Gamma Function

The Gamma function plays an important role in the theory of fractional derivatives, therefore we collect here two equivalent definitions and some properties.

Definition 1.6.1. *The Gamma function is defined through the Euler limit as*

$$\Gamma(x) = \lim_{N\to\infty}\left(\frac{N!N^x}{x(x+1)(x+2)...(x+N)}\right), \quad x > 0. \tag{1.13}$$

Definition 1.6.2. *The Gamma function can also be defined through an integral transform as*

$$\Gamma(x) = \int_0^\infty y^{x-1} e^{-y} \mathrm{d}y, \quad x > 0. \tag{1.14}$$

Property 1.6.1. $\Gamma(1) = 1$.

Proof. Using integration by parts, it holds that

$$\Gamma(1) = \int_0^\infty y\mathrm{e}^{-y}\mathrm{d}y = -y\mathrm{e}^{-y}\Big|_0^\infty + \int_0^\infty \mathrm{e}^{-y}\mathrm{d}y = -\mathrm{e}^{-y}\Big|_0^\infty = 1.$$

□

Property 1.6.2. $\Gamma(x+1) = x\Gamma(x)$.

Proof. As before, using integration by parts, it holds that

$$\Gamma(x+1) = \int_0^\infty y^x\mathrm{e}^{-y}\mathrm{d}y = -y^x\mathrm{e}^{-y}\Big|_0^\infty + x\int_0^\infty y^{x-1}\mathrm{e}^{-y}\mathrm{d}y = x\Gamma(x).$$

□

As a corollary of Property 1.6.2, the Gamma function can be seen as a generalization of the factorial function.

Property 1.6.3. $\forall n \in \mathbb{N}: \quad \Gamma(n+1) = n!$

Proof. Applying $n+1$ times Property 1.6.2, holds that

$$\Gamma(n+1) = n\Gamma(n) = n(n-1)\Gamma(n-1) = ... = n(n-1)...2\cdot 1\cdot\Gamma(1) = n!.$$

□

Replacing x with $x-1$ in Property 1.6.2 we obtain the recurrence formula $\Gamma(x-1) = \frac{\Gamma(x)}{x-1}$ which can be used to extend Definition 1.6.2 to negative arguments. It is easy to see that $\Gamma(0)$ and $\Gamma(-n)$ are not defined for any $n \in \mathbb{N}$.

In the next chapter we will encounter expressions of the form

$$\binom{j-n-1}{j} \qquad (-1)^j\binom{n}{j} \qquad j,n \in \mathbb{N} \tag{1.15}$$

which turns out to be equivalent. A proof of the following property can be found in Ortigueira [2011].

Property 1.6.4. For any $j,n \in \mathbb{N}$ holds $\binom{j-n-1}{j} = (-1)^j\binom{n}{j}$

1.6.2 Notation

Since differentiation and integration are inverse operations, when applied to a function which is regular enough and under appropriate hypotheses, we can extend the usual notation for the nth derivative of a function $f(x)$ with respect to x

$$\frac{\mathrm{d}^n f}{\mathrm{d}x^n}, \quad n \in \mathbb{N},$$

to multiple integration. Assuming zero as lower bound, we have that

$$\frac{\mathrm{d}^{-1} f}{\mathrm{d}x^{-1}} = \int_0^x f(y)\mathrm{d}y. \tag{1.16}$$

Applying recursively the equation (1.16), we obtain

$$\begin{aligned}
\frac{\mathrm{d}^{-2} f}{\mathrm{d}x^{-2}} &= \int_0^x \int_0^{x_1} f(x_0)\mathrm{d}x_0\mathrm{d}x_1, \\
&\vdots \\
\frac{\mathrm{d}^{-n} f}{\mathrm{d}x^{-n}} &= \int_0^x \dots \int_0^{x_2} \int_0^{x_1} f(x_0)\mathrm{d}x_0\mathrm{d}x_1 \dots \mathrm{d}x_{n-1}.
\end{aligned}$$

To generalize the previous extended definition of integral to lower bounds different from zero, we define

$$\begin{aligned}
\frac{\mathrm{d}^{-1} f}{\mathrm{d}(x-a)^{-1}} &= \int_a^x f(y)\mathrm{d}y, \\
&\vdots \\
\frac{\mathrm{d}^{-n} f}{\mathrm{d}(x-a)^{-n}} &= \int_a^x \dots \int_a^{x_2} \int_a^{x_1} f(x_0)\mathrm{d}x_0\mathrm{d}x_1 \dots \mathrm{d}x_{n-1}.
\end{aligned}$$

This definition follows naturally from the identity

$$\int_a^x f(y)\mathrm{d}y = \int_0^{x-a} f(y+a)\mathrm{d}y \tag{1.17}$$

and we note that

$$\frac{\mathrm{d}^{-n} f}{\mathrm{d}(x-a)^{-n}} \neq \frac{\mathrm{d}^{-n} f}{\mathrm{d}x^{-n}}. \tag{1.18}$$

1.6.3 Riemann-Liouville and Caputo definitions of fractional derivatives

In order to construct Riemann-Liouville and Caputo formulas, we need to express an iterated integral $\frac{\mathrm{d}^{-n}}{\mathrm{d}(x-a)^{-n}}$ as a weighted single integral Loverro [2004].

Let us recall Leibniz's theorem for differentiating an integral which will be useful to prove the next proposition.

Theorem 1.6.1 (Leibniz). *Let* $h \in \mathrm{C}^1(\mathbb{R}^2)$ *and* $a, b \in \mathrm{C}^1(\mathbb{R})$ *then it holds*

$$\frac{\mathrm{d}}{\mathrm{d}x}\left(\int_{a(x)}^{b(x)} h(x,t)\mathrm{d}t\right) = h(x,b(x))\frac{\mathrm{d}}{\mathrm{d}x}b(x) - h(x,a(x))\frac{\mathrm{d}}{\mathrm{d}x}a(x) + \int_{a(x)}^{b(x)} \frac{\partial}{\partial x}h(x,t)\mathrm{d}t.$$

Proposition 1.6.1. *Let f be a locally integrable function, then for $n = 0, 1, 2, \ldots$ it holds*

$$\int_a^x f(\xi)\mathrm{d}\xi = \frac{1}{n!}\frac{\mathrm{d}^n}{\mathrm{d}x^n}\int_a^x (x-\xi)^n f(\xi)\mathrm{d}\xi. \tag{1.19}$$

Proof.

$$\begin{aligned}\frac{1}{n!}\frac{\mathrm{d}^n}{\mathrm{d}x^n}\int_a^x (x-\xi)^n f(\xi)\mathrm{d}\xi &= \frac{1}{n!}\frac{\mathrm{d}^{n-1}}{\mathrm{d}x^{n-1}}\frac{d}{\mathrm{d}x}\int_a^x (x-\xi)^n f(\xi)\mathrm{d}\xi \\ &= \frac{1}{n!}\frac{\mathrm{d}^{n-1}}{\mathrm{d}x^{n-1}}\Bigg((x-x)^n f(x) - (x-a)^n f(a)\cdot 0 + \\ &\qquad + n\int_a^x (x-\xi)^{n-1} f(\xi)\mathrm{d}\xi\Bigg) \\ &= \frac{1}{(n-1)!}\frac{\mathrm{d}^{n-1}}{\mathrm{d}x^{n-1}}\int_a^x (x-\xi)^{n-1} f(\xi)\mathrm{d}\xi.\end{aligned} \tag{1.20}$$

Applying recursively equation (1.20) we obtain

$$\frac{1}{n!}\frac{\mathrm{d}^n}{\mathrm{d}x^n}\int_a^x (x-\xi)^n f(\xi)\mathrm{d}\xi = \cdots = \int_a^x f(\xi)\mathrm{d}\xi.$$

□

The Riemann-Liouville fractional integral follows from integrating the equation (1.19) n times.

$$\frac{\mathrm{d}^{-2}f}{\mathrm{d}(x-a)^{-2}} = \int_a^x \int_a^{x_1} f(x_0)\mathrm{d}x_0\mathrm{d}x_1 = \frac{1}{1!}\int_a^x (x-\xi)f(\xi)\mathrm{d}\xi \qquad \text{if } n = 1$$

$$
\begin{aligned}
\frac{\mathrm{d}^{-3}f}{\mathrm{d}(x-a)^{-3}} &= \frac{1}{2!}\int_a^x (x-\xi)^2 f(\xi)\mathrm{d}\xi \qquad\qquad \text{if } n=2 \\
&\vdots \\
\frac{\mathrm{d}^{-n}f}{\mathrm{d}(x-a)^{-n}} &= \frac{1}{(n-1)!}\int_a^x (x-\xi)^{n-1} f(\xi)\mathrm{d}\xi. \qquad (1.21)
\end{aligned}
$$

Recalling the Property 1.6.3 of Gamma function, formula (1.21) can be generalized as

$$
\begin{aligned}
{}_aJ_x^\alpha f(x) &:= \frac{1}{\Gamma(\alpha)}\int_a^x (x-\xi)^{\alpha-1} f(\xi)\mathrm{d}\xi, \qquad \alpha>0, \\
{}_xJ_b^\alpha f(x) &:= \frac{1}{\Gamma(\alpha)}\int_x^b (\xi-x)^{\alpha-1} f(\xi)\mathrm{d}\xi, \qquad \alpha>0.
\end{aligned} \qquad (1.22)
$$

which are called the left and right Riemann-Liouville (RL) fractional integrals, respectively, and are used to retrieve the left and right RL and Caputo fractional derivatives.

To extend definitions in equation (1.22) to $\alpha \in \mathbb{R}$ we proceed through an example. Consider as differentiation of order, e.g., $\alpha = 2.3$ and select $m \in \mathbb{N}$ such that $m-1 < \alpha < m$, i.e., $m = 3$ At this point we have two possible ways to define the fractional derivative.

Riemann-Liouville fractional derivative

Having found the integer $m = \lceil \alpha \rceil$, the first step of the process is to integrate our function $f(x)$ by order $|\alpha - m| = m - \alpha = 0.7$ (arrow (a) in Figure 1.1). Second, we differentiate the resulting function by order $m = 3$ (arrow (b) in Figure 1.1), thereby achieving a resultant differentiation of order α.

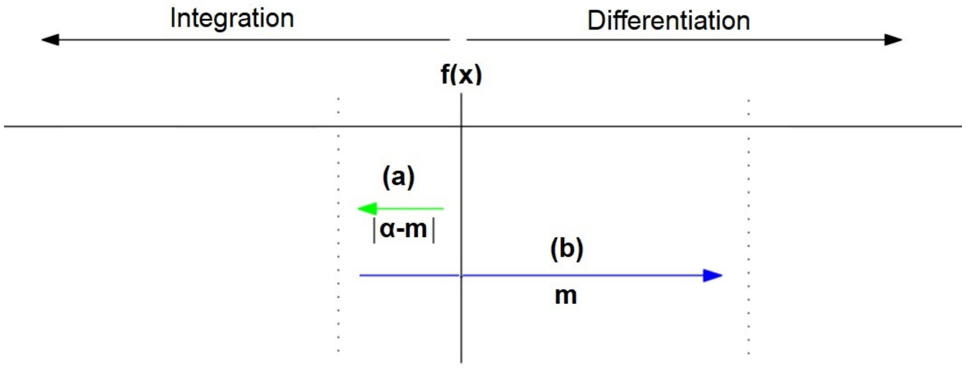


Figure 1.1. Graphical representation of the Riemann-Liouville definition method.

Summing up, let $m = \lceil \alpha \rceil$ and define the left and right RL fractional derivatives as

$$\begin{aligned}\frac{\mathrm{d}^\alpha f(x)}{\mathrm{d}^R(x-a)^\alpha} &:= \frac{\mathrm{d}^m}{\mathrm{d}x^m}\,{}_aJ_x^{m-\alpha} f(x) = \frac{1}{\Gamma(m-\alpha)}\frac{\mathrm{d}^m}{\mathrm{d}x^m}\int_a^x f(\xi)(x-\xi)^{m-\alpha-1}\mathrm{d}\xi,\\ \frac{\mathrm{d}^\alpha f(x)}{\mathrm{d}^R(b-x)^\alpha} &:= \frac{\mathrm{d}^m}{\mathrm{d}x^m}\,{}_xJ_b^{m-\alpha} f(x) = \frac{(-1)^m}{\Gamma(m-\alpha)}\frac{\mathrm{d}^m}{\mathrm{d}x^m}\int_x^b f(\xi)(\xi-x)^{m-\alpha-1}\mathrm{d}\xi,\end{aligned} \tag{1.23}$$

which is the most frequently encountered definition of fractional derivative.

Remark 1.6.1. *If $f(x) \in L^1(\mathbb{R})$ equation (1.23) can be extended to $-\infty, +\infty$ as lower and upper limit and we write*

$$\begin{aligned}\frac{\mathrm{d}^\alpha f(x)}{\mathrm{d}_+^R x^\alpha} &:= \frac{1}{\Gamma(m-\alpha)}\frac{\mathrm{d}^m}{\mathrm{d}x^m}\int_{-\infty}^x f(\xi)(x-\xi)^{m-\alpha-1}\mathrm{d}\xi,\\ \frac{\mathrm{d}^\alpha f(x)}{\mathrm{d}_-^R x^\alpha} &:= \frac{(-1)^m}{\Gamma(m-\alpha)}\frac{\mathrm{d}^m}{\mathrm{d}x^m}\int_x^{+\infty} f(\xi)(\xi-x)^{m-\alpha-1}\mathrm{d}\xi,\end{aligned} \tag{1.24}$$

Caputo fractional derivative

The Caputo fractional derivative is defined essentially in the same way as the RL fractional derivative, but with the inverse order of the operations. We first differentiate our function $f(x)$ by order m (arrow (a) in Figure 1.2) and then we integrate the resulting function by order $|\alpha - m| = 0.7$ (arrow (b) in Figure 1.2). This still lead to a resultant differentiation of order α.

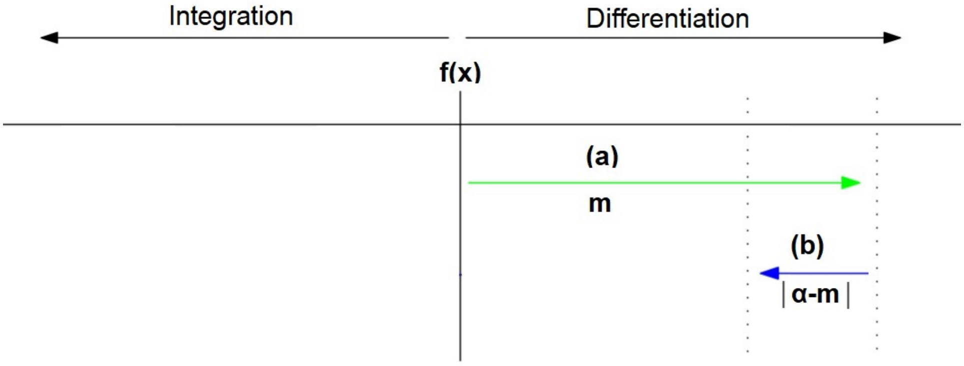


Figure 1.2. Graphical representation of the Caputo definition method.

Summing up, we define the left and right Caputo fractional derivatives as follows

$$\begin{aligned}\frac{d^\alpha f(x)}{d^C(x-a)^\alpha} &:= {}_aJ_x^{m-\alpha}\frac{d^m f(x)}{dx^m} = \frac{1}{\Gamma(m-\alpha)}\int_a^x \frac{d^m f(\xi)}{d\xi^m}(x-\xi)^{m-\alpha-1}d\xi,\\ \frac{d^\alpha f(x)}{d^C(b-x)^\alpha} &:= {}_xJ_b^{m-\alpha}\frac{d^m f(x)}{dx^m} = \frac{(-1)^m}{\Gamma(m-\alpha)}\int_x^b \frac{d^m f(\xi)}{d\xi^m}(\xi-x)^{m-\alpha-1}d\xi,\end{aligned} \tag{1.25}$$

where $m=\lceil\alpha\rceil$.
In fractional models the Caputo derivative often replaces the standard time derivative, and due to the sequentiality of time the most frequently encountered Caputo fractional derivative is the left one.

From a modellistic point of view, applied problems require fractional derivatives which allow physically interpretable initial conditions. Indeed, the Caputo fractional derivative allows the formulation of initial conditions for initial-value problems for FDEs involving only the boundary values of integer-order derivatives, while the RL fractional derivative, unfortunately, leads to initial conditions in terms of RL fractional derivatives of lower orders Podlubny [1998]. However, under opportune hypothesis, the two approaches coincide.

Remark 1.6.2. *From Equations (2.4.8)-(2.4.9) on page 91 of Kilbas et al. [2006], the following relations hold*

$$\begin{aligned}\frac{d^\alpha f(x)}{d^R(x-a)^\alpha} &= \frac{d^\alpha f(x)}{d^C(x-a)^\alpha} + \sum_{k=0}^{\lfloor\alpha\rfloor}\frac{(x-a)^{k-\alpha}}{\Gamma(k-\alpha+1)}\frac{d^k f(a)}{dx^k}\\ \frac{d^\alpha f(x)}{d^R(b-x)^\alpha} &= \frac{d^\alpha f(x)}{d^C(b-x)^\alpha} + \sum_{k=0}^{\lfloor\alpha\rfloor}(-1)^k\frac{(b-x)^{k-\alpha}}{\Gamma(k-\alpha+1)}\frac{d^k f(b)}{dx^k}\end{aligned}$$

Then, in the case where $\frac{\mathrm{d}^k f(a)}{\mathrm{d}x^k} = \frac{\mathrm{d}^k f(b)}{\mathrm{d}x^k} = 0 \ \forall k = 1, ..., \lfloor \alpha \rfloor$*, the Caputo and RL fractional derivatives coincide.*

1.6.4 Grünwald definition of fractional derivatives

Because the RL approach to the fractional derivative began with an expression for the repeated integration of a function, one's first instinct may be to imitate a similar approach for the derivative.

Applying the definition of derivative in terms of a backward difference one or more times to a regular enough function f, we obtain

$$\begin{aligned}
\frac{\mathrm{d}^1 f}{\mathrm{d}x^1} = \frac{\mathrm{d}}{\mathrm{d}x} f(x) &= \lim_{h\to 0} \frac{f(x) - f(x-h)}{h} \\
\frac{\mathrm{d}^2 f}{\mathrm{d}x^2} = \frac{\mathrm{d}^2}{\mathrm{d}x^2} f(x) &= \lim_{h\to 0} \frac{\left[\lim_{h\to 0} \frac{f(x)-f(x-h)}{h} - \lim_{h\to 0} \frac{f(x-h)-f(x-2h)}{h}\right]}{h} \\
&= \lim_{h\to 0} \frac{f(x) - 2f(x-h) + f(x-2h)}{h^2} \\
\frac{\mathrm{d}^3 f}{\mathrm{d}x^3} = \frac{\mathrm{d}^3}{\mathrm{d}x^3} f(x) &= \lim_{h\to 0} \frac{f(x) - 3f(x-h) + 3f(x-2h) - f(x-3h)}{h^3} \\
&\vdots
\end{aligned} \tag{1.26}$$

Let us note that the coefficients that multiply the function in the nth derivative refer to the $(n+1)$th row in Pascal's triangle. Recalling that Pascal's triangle is a triangular array of the binomial coefficients, this suggests the following formula

$$\frac{\mathrm{d}^n f}{\mathrm{d}x^n} = \lim_{h\to 0} \frac{\sum\limits_{j=0}^{n} (-1)^j \binom{n}{j} f(x - jh)}{h^n}, \qquad n \in \mathbb{N}.$$

We now replace h with a sequence $\{h_N\}$ such that $\lim\limits_{N\to\infty} h_N = 0$. Given $a \in \mathbb{R}$ such that $a < x$ we consider the sequence $\{h_N\}$ where $h_N = \frac{x-a}{N}$. Then, since $\binom{n}{j} = 0$ if $j > n$, the nth derivative may be defined as

$$\frac{\mathrm{d}^n f}{\mathrm{d}x^n} = \lim_{N\to\infty} \frac{\sum\limits_{j=0}^{N-1} (-1)^j \binom{n}{j} f\left(x - j\frac{x-a}{N}\right)}{\left(\frac{x-a}{N}\right)^n}. \tag{1.27}$$

Let us now consider the integrals. Applying the definition of integral as Riemann sum we obtain that

$$\begin{aligned}\frac{\mathrm{d}^{-1}f}{\mathrm{d}(x-a)^{-1}} &= \int_a^x f(\xi)\mathrm{d}\xi \\ &= \lim_{h_N\to 0} h_N[f(x)+f(x-h_N)+\ldots+f(x-(N-1)h_N)] \qquad (1.28) \\ &= \lim_{h_N\to 0} h_N \sum_{j=0}^{N-1} f(x-jh_N).\end{aligned}$$

Using two times the equation (1.28) it holds

$$\begin{aligned}\frac{\mathrm{d}^{-2}f}{\mathrm{d}(x-a)^{-2}} &= \int_a^x \int_a^{x_1} f(x_0)\mathrm{d}x_0\mathrm{d}x_1 \\ &= \lim_{h_N\to 0} {h_N}^2[f(x)+2f(x-h_N)+\ldots+Nf(x-(N-1)h_N)] \\ &= \lim_{h_N\to 0} {h_N}^2 \sum_{j=0}^{N-1}(j+1)f(x-jh_N).\end{aligned}$$

Two times is not enough to find a general rule, hence applying again equation (1.28) we have

$$\begin{aligned}\frac{\mathrm{d}^{-3}f}{\mathrm{d}(x-a)^{-3}} &= \int_a^x \int_a^{x_2} \int_a^{x_1} f(x_0)\mathrm{d}x_0\mathrm{d}x_1\mathrm{d}x_2 \\ &= \lim_{h_N\to 0} {h_N}^3 \sum_{j=0}^{N-1} \frac{(j+1)(j+2)}{2} f(x-jh_N),\end{aligned}$$

where $h_N = \frac{x-a}{N}$. This time we notice that all the signs are positive and the sequence $\{1,\ j+1,\ \frac{(j+1)(j+2)}{2},\ \ldots\}$ can be described by $\binom{j+n-1}{j}$ where n is the order of the integral. This yields

$$\frac{\mathrm{d}^{-n}f}{\mathrm{d}(x-a)^{-n}} = \lim_{N\to\infty}\left(\frac{x-a}{N}\right)^n \sum_{j=0}^{N-1}\binom{j+n-1}{j} f\left(x-j\frac{x-a}{N}\right). \qquad (1.29)$$

Comparing equations (1.27) and (1.29) and recalling the Property 1.6.4

$$(-1)^j\binom{n}{j} = \binom{j-n-1}{j} = \frac{\Gamma(j-n)}{\Gamma(-n)\Gamma(j+1)},$$

we see that equations (1.27) and (1.29) are identical and, by extending the integer order n to the fractional order $\alpha \in \mathbb{R}$, we finally obtain the left-sided GL derivative formula for fractional derivatives

$$\frac{\mathrm{d}^\alpha f(x)}{\mathrm{d}^G(x-a)^\alpha} := \lim_{N\to\infty}\left(\frac{x-a}{N}\right)^{-\alpha} \sum_{j=0}^{N-1}(-1)^j\binom{\alpha}{j} f\left(x-j\frac{x-a}{N}\right). \qquad (1.30)$$

Let us notice that using the definition of derivative in terms of a forward difference in (1.26), defining integrals with switched limits $\frac{\mathrm{d}^{-1}f}{\mathrm{d}(b-x)^{-1}} = \int_x^b f(\xi)\mathrm{d}\xi$ and adjusting the respective discretization, we would have built up the right-sided GL which depends on "what happens after x instead of before". Explicitly, given $b > x$, the right-sided GL is

$$\frac{\mathrm{d}^{\alpha} f(x)}{\mathrm{d}^{G}(b-x)^{\alpha}} := \lim_{N\to\infty} \left(\frac{b-x}{N}\right)^{-\alpha} \sum_{j=0}^{N-1} (-1)^j \binom{\alpha}{j} f\left(x + j\frac{b-x}{N}\right). \qquad (1.31)$$

Remark 1.6.3. *If $f \in L^1(\mathbb{R})$, equations* (1.30) *and* (1.31) *can be extended to $-\infty, +\infty$ as lower and upper limit, respectively. For the sake of readability we write $\frac{\mathrm{d}^{\alpha}f(x)}{\mathrm{d}_{+}^{G}x^{\alpha}}, \frac{\mathrm{d}^{\alpha}f(x)}{\mathrm{d}_{-}^{G}x^{\alpha}}$ instead of $\frac{\mathrm{d}^{\alpha}f(x)}{\mathrm{d}^{G}(x-(-\infty))^{\alpha}}, \frac{\mathrm{d}^{\alpha}f(x)}{\mathrm{d}^{G}((+\infty)-x)^{\alpha}}$, respectively, therefore*

$$\begin{aligned} \frac{\mathrm{d}^{\alpha} f(x)}{\mathrm{d}_{+}^{G} x^{\alpha}} &= \lim_{h\to 0} \frac{1}{h^{\alpha}} \sum_{j=0}^{\infty} (-1)^j \binom{\alpha}{j} f(x - jh); \\ \frac{\mathrm{d}^{\alpha} f(x)}{\mathrm{d}_{-}^{G} x^{\alpha}} &= \lim_{h\to 0} \frac{1}{h^{\alpha}} \sum_{j=0}^{\infty} (-1)^j \binom{\alpha}{j} f(x + jh). \end{aligned} \qquad (1.32)$$

When uniform meshes are employed, the RL fractional derivative is usually discretized through the GL formula. Aiming at introducing a second-order approximation of the RL fractional derivative, which will be widely used in the reminder of this work, we first define a shift of the GL operator.

Definition 1.6.3. *For any small fixed $h > 0$, let us define the Shifted Grünwald Difference (SGD) by shifting the k index in* (1.32) *as follows*

$$\begin{aligned} {}_{L}G_{h,p}^{\alpha} f(x) &:= \frac{1}{h^{\alpha}} \sum_{k=0}^{\infty} g_k^{(\alpha)} f(x - (k-p)h), \\ {}_{R}G_{h,q}^{\alpha} f(x) &:= \frac{1}{h^{\alpha}} \sum_{k=0}^{\infty} g_k^{(\alpha)} f(x + (k-q)h), \end{aligned} \qquad (1.33)$$

where $p, q \in \mathbb{Z}$ and $g_k^{(\alpha)} = (-1)^k \binom{\alpha}{k}$, with $\binom{\alpha}{0} := 1$.

As shown in Theorem 1.6.2, left and right SGD operator, which coincide with the GL operator when the shifting parameters $p = q = 0$, are first order approximations of the left and right RL operators, respectively.

Theorem 1.6.2 (Hejazi et al. [2014]). *Let $\alpha, p > 0$, $n = \lceil \alpha \rceil$ and suppose that $f \in C^{2n}(\mathbb{R})$ and all derivatives of f up to order $2n$ belong to $L^1(\mathbb{R})$. Then there* *exist constants c_l independent of h, f, x such that*

$$
\begin{aligned}
{}_L G^{\alpha}_{h,p} f(x) &= \frac{\mathrm{d}^{\alpha} f(x)}{\mathrm{d}^{R}_{+} x^{\alpha}} + \sum_{l=1}^{n-1} c_l \frac{\mathrm{d}^{\alpha+l} f(x)}{\mathrm{d}^{R}_{+} x^{\alpha+l}} h^l + O(h^n) = \frac{\mathrm{d}^{\alpha} f(x)}{\mathrm{d}^{R}_{+} x^{\alpha}} + O(h),\\
{}_R G^{\alpha}_{h,p} f(x) &= \frac{\mathrm{d}^{\alpha} f(x)}{\mathrm{d}^{R}_{-} x^{\alpha}} + \sum_{l=1}^{n-1} c_l \frac{\mathrm{d}^{\alpha+l} f(x)}{\mathrm{d}^{R}_{-} x^{\alpha+l}} h^l + O(h^n) = \frac{\mathrm{d}^{\alpha} f(x)}{\mathrm{d}^{R}_{-} x^{\alpha}} + O(h),
\end{aligned}
$$

uniformly in $x \in \mathbb{R}$.

Remark 1.6.4. *Let $f(x)$ with $\frac{\mathrm{d}^k f(a)}{\mathrm{d}x^k} = \frac{\mathrm{d}^k f(b)}{\mathrm{d}x^k} = 0 \ \forall k = 1, ..., \lfloor \alpha \rfloor$, and that satisfies the hypothesis in Theorem 1.6.2. Then, according to Remark 1.6.2 and Theorem 1.6.2, the Caputo, RL and GL fractional derivatives coincide.*

An appropriate weighting of the SGD operators in (1.33) results in a second-order accurate approximation of the fractional derivatives, denoted by WSGD, as specified by the following theorem.

Theorem 1.6.3 (Tian et al. [2015]). *Assume that f, $\frac{\mathrm{d}^{\alpha+2} f}{\mathrm{d}^{G}_{\pm} x^{\alpha+2}}$, and its Fourier transform belong to $L^1(\mathbb{R})$, and define the left and right Weighted SGD operators as*

$$
\begin{aligned}
{}_L \mathrm{D}^{\alpha}_{h,\boldsymbol{p}} f(x) &:= \frac{\alpha - 2p_2}{2(p_1 - p_2)}\, {}_L G^{\alpha}_{h,p_1} f(x) + \frac{2p_1 - \alpha}{2(p_1 - p_2)}\, {}_L G^{\alpha}_{h,p_2} f(x),\\
{}_R \mathrm{D}^{\alpha}_{h,\boldsymbol{q}} f(x) &:= \frac{\alpha - 2p_2}{2(p_1 - p_2)}\, {}_R G^{\alpha}_{h,p_1} f(x) + \frac{2p_1 - \alpha}{2(p_1 - p_2)}\, {}_R G^{\alpha}_{h,p_2} f(x).
\end{aligned}
\tag{1.34}
$$

Then,

$$
\begin{aligned}
{}_L \mathrm{D}^{\alpha}_{h,\boldsymbol{p}} f(x) &= \frac{\mathrm{d}^{\alpha} f(x)}{\mathrm{d}^{R}_{+} x^{\alpha}} + \mathrm{O}(h^2),\\
{}_R \mathrm{D}^{\alpha}_{h,\boldsymbol{q}} f(x) &= \frac{\mathrm{d}^{\alpha} f(x)}{\mathrm{d}^{R}_{-} x^{\alpha}} + \mathrm{O}(h^2),
\end{aligned}
$$

uniformly for $x \in \mathbb{R}$, where $\boldsymbol{p}, \boldsymbol{q} \in \mathbb{Z}^2$, with $\boldsymbol{p} = (p_1, p_2), \boldsymbol{q} = (q_1, q_2)$ such that $p_1, q_1 \neq p_2, q_2$, respectively.

Remark 1.6.5. *In Chapter 4 a non-integer shift will be needed, therefore here we note that the validity of equation* (1.34) *extends also to the case where $p_1, p_2, q_1, q_2 \in \mathbb{R}$ with $p_1 \neq p_2$ and $q_1 \neq q_2$. The reader can easily verify this assuming that p_1, p_2, q_1, q_2 are real and following the same argument of the proof of Theorem 1 in Tian et al. [2015].*

Remark 1.6.6. *In case of absorbing boundary conditions (see Chapter 2), i.e., the solution is null outside the definition domain, say $[a,b]$, it holds*

$$\frac{\mathrm{d}^{\alpha}u(x)}{\mathrm{d}^{S}(x-a)^{\alpha}} \equiv \frac{\mathrm{d}^{\alpha}u(x)}{\mathrm{d}^{S}x_{+}^{\alpha}},$$

$$\frac{\mathrm{d}^{\alpha}u(x)}{\mathrm{d}^{S}(b-x)^{\alpha}} \equiv \frac{\mathrm{d}^{\alpha}u(x)}{\mathrm{d}^{S}x_{-}^{\alpha}},$$

for $S \in \{\mathrm{GL}, \mathrm{RL}, \mathrm{C}\}$.

Chapter 2

Time stepping scheme

In this chapter we focus on a two-dimensional time-dependent space-FDE. The use of Crank-Nicolson (CN) in time and WSGD in space (see equation (1.34)) leads to the so-called CN-WSGD scheme Tian et al. [2015], which consists in a sequential-in-time second-order accurate finite difference scheme. In Section 2.1, we briefly introduce the CN-WSGD scheme, providing the coefficient matrix at each time step, which shows a 2-level Toeplitz structure. In Section 2.2, by exploiting the structure, we discuss certain spectral features of the coefficient matrices. Section 2.3 is devoted to the presentation of our anisotropic multigrid preconditioning strategy.

Multigrid strategies that exploit such structure are particularly effective when the fractional orders are both close to 2. We seek to investigate how structure-based multigrid approaches can be efficiently extended to the case where only one of the two fractional orders is close to 2, i.e., when the fractional equation shows an intrinsic anisotropy. Precisely, we design a multigrid (block-bandedbanded-block) preconditioner whose grid transfer operator is obtained with a semi-coarsening technique and that has weighted Jacobi as smoother.

The Jacobi relaxation parameter is estimated in Section 2.4 through an automatic symbol-based procedure. A further improvement in the robustness of the proposed MGM is attained using the V-cycle with semi-coarsening as smoother inside an outer full-coarsening.

Finally, in Section 2.5, several numerical results confirm that the resulting multigrid preconditioner1 is computationally effective and outperforms current state of the art techniques. Finally, in Section 2.6 we draw conclusions.

2.1 Problem setting and discretization

In this section we introduce our problem, then we provide its discretization over uniform meshes through the second-order accurate CN-WSGD scheme Tian et al. [2015].

2.1.1 Two-dimensional time-dependent space-FDE

We focus on the following two-dimensional initial-boundary value space-FDE problem

$$\begin{cases} \dfrac{\partial u(x,y,t)}{\partial t} = d_+(x,y,t)\dfrac{\partial^\alpha u(x,y,t)}{\partial_+^R x^\alpha} + d_-(x,y,t)\dfrac{\partial^\alpha u(x,y,t)}{\partial_-^R x^\alpha} + \\ \qquad + e_+(x,y,t)\dfrac{\partial^\beta u(x,y,t)}{\partial_+^R y^\beta} + e_-(x,y,t)\dfrac{\partial^\beta u(x,y,t)}{\partial_-^R y^\beta} + v(x,y,t), \\ \qquad\qquad (x,y,t) \in \Omega \times [0,T], \\ u(x,y,t) = 0, \qquad (x,y,t) \in \left(\mathbb{R}^2 \setminus \Omega\right) \times [0,T], \\ u(x,y,0) = u_0(x,y), \qquad (x,y) \in \overline{\Omega}, \end{cases} \tag{2.1}$$

where $\alpha, \beta \in (1,2)$ are the fractional derivative orders in x- and y-variable, respectively, $\Omega = (a_1, b_1) \times (a_2, b_2)$ is the space domain, $d_\pm(x,y,t), e_\pm(x,y,t)$ are the diffusion coefficients and are nonnegative bounded functions, $v(x,y,t)$ is the forcing term.

Remark 2.1.1. *Due to the non-locality of the fractional derivative operators, the behavior of u must be specified not just at the boundary but also at all points exterior to the domain, see Defterli et al. [2015]. In* (2.1) *we impose the so-called absorbing boundary conditions, that is we assume that the solution is zero on* $\mathbb{R}^2 \setminus \Omega$ *at each time.*

Here, we are interested in the case where the fractional orders α, β with $\alpha \neq \beta$ are in some sense far from each other, that is when the problem shows an intrinsic anisotropy. The left-sided and the right-sided fractional derivatives in (2.1) are given in the RL form, according to definition (1.24), and discretized through the GL formulas.

2.1.2 CN-WSGD scheme for 2D space-FDEs

For the discretization of problem (2.1) we combine the WSGD approximation (1.34) of the fractional operators in space with the CN method in time. The resulting scheme is second-order accurate and is known as CN-WSGD (see Tian et al. [2015]). In this section we review the construction of the CN-WSGD scheme and we recall some related results.

Let us fix $N_x, N_y, M_t \in \mathbb{N}$ and discretize the domain $\Omega \times [0,T]$ with

$$\begin{array}{lll} x_i = a_1 + ih_x, & h_x = \frac{b_1-a_1}{N_x+1}, & i = 0,...,N_x+1, \\ y_j = a_2 + jh_y, & h_y = \frac{b_2-a_2}{N_y+1}, & j = 0,...,N_y+1, \\ t^m = m\Delta t, & \Delta t = \frac{T}{M_t}, & m = 0,...,M_t, \end{array}$$

By choosing $\boldsymbol{p} = \boldsymbol{q} = (1,0)$, from (1.34) we obtain the following discretized second-order accurate WSGD operators

$$\begin{aligned} {}_L\mathrm{D}^{\alpha}_{h_x,\boldsymbol{p}} u(x_i, y_j, t^m) &= \frac{1}{h_x^{\alpha}} \sum_{k=0}^{i} \omega_k^{(\alpha)} u\big(x_{i-k+1}, y_j, t^m\big), \\ {}_R\mathrm{D}^{\alpha}_{h_x,\boldsymbol{q}} u(x_i, y_j, t^m) &= \frac{1}{h_x^{\alpha}} \sum_{k=0}^{N_x-i} \omega_k^{(\alpha)} u\big(x_{i+k-1}, y_j, t^m\big), \\ {}_L\mathrm{D}^{\beta}_{h_y,\boldsymbol{p}} u(x_i, y_j, t^m) &= \frac{1}{h_y^{\beta}} \sum_{k=0}^{j} \omega_k^{(\beta)} u\big(x_i, y_{j-k+1}, t^m\big), \\ {}_R\mathrm{D}^{\beta}_{h_y,\boldsymbol{q}} u(x_i, y_j, t^m) &= \frac{1}{h_y^{\beta}} \sum_{k=0}^{N_y-j} \omega_k^{(\beta)} u\big(x_i, y_{j+k-1}, t^m\big), \end{aligned} \tag{2.2}$$

where

$$\omega_0^{(\gamma)} = \frac{\gamma}{2} g_0^{(\gamma)}, \quad \omega_k^{(\gamma)} = \frac{\gamma}{2} g_k^{(\gamma)} + \frac{2-\gamma}{2} g_{k-1}^{(\gamma)}, \quad k \geq 1. \tag{2.3}$$

with $g_k^{(\alpha)} = (-1)^k \binom{\alpha}{k}$ and $\gamma = \alpha, \beta$. Notice that the sums in (2.2) are finite sums because of the absorbing boundary conditions imposed in (2.1).

For the coefficients in (2.3) the following lemma holds.

Lemma 2.1.1 (Tian et al. [2015]). *Coefficients $\omega_k^{(\gamma)}$ in equation (2.3) satisfy the following properties for $\gamma \in (1,2]$,*

$$\begin{cases} \omega_0^{(\gamma)} = \frac{\gamma}{2}, \ \omega_1^{(\gamma)} = \frac{2-\gamma-\gamma^2}{2} < 0, \ \omega_2^{(\gamma)} = \frac{\gamma(\gamma^2+\gamma-4)}{4}, \\ 1 \geq \omega_0^{(\gamma)} \geq \omega_3^{(\gamma)} \geq \omega_4^{(\gamma)} \geq \cdots \geq 0, \\ \sum\limits_{k=0}^{\infty} \omega_k^{(\gamma)} = 0, \ \sum\limits_{k=0}^{m} \omega_k^{(\gamma)} < 0, \ m \geq 2, \\ \omega_k^{(\gamma)} = O\left(\frac{1}{k^{\gamma+1}}\right). \end{cases}$$

Let us now represent the discretized solution at time t^m by the N-dimensional vector

$$u^m = [u^m_{1,1}, \dots, u^m_{N_x,1}, u^m_{1,2}, \dots, u^m_{N_x,2}, \dots, u^m_{1,N_y}, \dots, u^m_{N_x,N_y}]^{\mathrm{T}},$$

where $N = N_x{\cdot}N_y$ and $u^m_{i,j} \approx u(x_i, y_j, t^m)$. Accordingly, we define the N-dimensional vectors

$$\begin{aligned} d^m_\pm &= [d^{\pm,m}_{1,1}, \dots, d^{\pm,m}_{N_x,1}, d^{\pm,m}_{1,2}, \dots, d^{\pm,m}_{N_x,2}, \dots, d^{\pm,m}_{1,N_y}, \dots, d^{\pm,m}_{N_x,N_y}]^{\mathrm{T}}, \\ e^m_\pm &= [e^{\pm,m}_{1,1}, \dots, e^{\pm,m}_{N_x,1}, e^{\pm,m}_{1,2}, \dots, e^{\pm,m}_{N_x,2}, \dots, e^{\pm,m}_{1,N_y}, \dots, e^{\pm,m}_{N_x,N_y}]^{\mathrm{T}}, \\ v^{m-\frac{1}{2}} &= [v^{m-\frac{1}{2}}_{1,1}, \dots, v^{m-\frac{1}{2}}_{N_x,1}, v^{m-\frac{1}{2}}_{1,2}, \dots, v^{m-\frac{1}{2}}_{N_x,2}, \dots, v^{m-\frac{1}{2}}_{1,N_y}, \dots, v^{m-\frac{1}{2}}_{N_x,N_y}]^{\mathrm{T}}, \end{aligned}$$

with $d^{\pm,m}_{i,j} = d_\pm(x_i, y_j, t^m)$, $e^{\pm,m}_{i,j} = e_\pm(x_i, y_j, t^m)$, $v^{m-\frac{1}{2}}_{i,j} = v(x_i, y_j, t^{m-\frac{1}{2}})$ and the $N \times N$ matrices

$$D^m_\pm = \operatorname{diag}\left(d^m_\pm\right), \quad E^m_\pm = \operatorname{diag}\left(e^m_\pm\right), \tag{2.4}$$

where $t^{m-\frac{1}{2}} = \frac{t^m + t^{m-1}}{2}$, $m = 1, \dots, M_t$.

Thanks to (2.2), the discretization of the fractional derivatives in both spatial dimensions at time t^m can be written in matrix-form as

$$\begin{aligned} \frac{\partial^\alpha u(x,y,t^m)}{\partial^R_+ x^\alpha} &\approx \frac{1}{h^\alpha_x}\left(I_{N_y} \otimes A^\alpha_{N_x}\right) u^m, \\ \frac{\partial^\alpha u(x,y,t^m)}{\partial^R_- x^\alpha} &\approx \frac{1}{h^\alpha_x}\left(I_{N_y} \otimes \left(A^\alpha_{N_x}\right)^{\mathrm{T}}\right) u^m, \\ \frac{\partial^\beta u(x,y,t^m)}{\partial^R_+ y^\beta} &\approx \frac{1}{h^\beta_y}\left(A^\beta_{N_y} \otimes I_{N_x}\right) u^m, \\ \frac{\partial^\beta u(x,y,t^m)}{\partial^R_- y^\beta} &\approx \frac{1}{h^\beta_y}\left(\left(A^\beta_{N_y}\right)^{\mathrm{T}} \otimes I_{N_x}\right) u^m, \end{aligned}$$

where A^γ_N is the following Toeplitz matrix,

$$A^\gamma_N = \begin{pmatrix} \omega^{(\gamma)}_1 & \omega^{(\gamma)}_0 & & & \\ \omega^{(\gamma)}_2 & \omega^{(\gamma)}_1 & \omega^{(\gamma)}_0 & & \\ \vdots & \ddots & \ddots & \ddots & \\ \omega^{(\gamma)}_{N-1} & & \ddots & \ddots & \omega^{(\gamma)}_0 \\ \omega^{(\gamma)}_N & \omega^{(\gamma)}_{N-1} & \cdots & \omega^{(\gamma)}_2 & \omega^{(\gamma)}_1 \end{pmatrix}_{N \times N}. \tag{2.5}$$

Remark 2.1.2. *If $\gamma = 2$, then $\omega^{(2)}_0 = 1$, $\omega^{(2)}_1 = -2$, $\omega^{(2)}_2 = 1$ and $\omega^{(2)}_k = 0$ for $k > 2$ and therefore A^2_N is the Hermitian negative definite matrix representing the discretized one-dimensional Laplacian operator discretized by second-order central finite differences.*

Omitting the scaling factors and collecting all the terms together, the discretization of (2.1) in both spatial dimensions yields the matrices

$$\begin{aligned} A^m_{x,N} &= D^m_+\left(I_{N_y} \otimes A^\alpha_{N_x}\right) + D^m_-\left(I_{N_y} \otimes \left(A^\alpha_{N_x}\right)^{\mathrm{T}}\right), \\ A^m_{y,N} &= E^m_+\left(A^\beta_{N_y} \otimes I_{N_x}\right) + E^m_-\left(\left(A^\beta_{N_y}\right)^{\mathrm{T}} \otimes I_{N_x}\right). \end{aligned} \tag{2.6}$$

On the other hand, the discretization in time of (2.1) follows by the application of the CN scheme, that is

$$u^m = u^{m-1} + \frac{\Delta t}{2}\left(F^m + F^{m-1}\right),$$

where

$$F^m = \left(\frac{1}{h^\alpha_x}A^m_{x,N} + \frac{1}{h^\beta_y}A^m_{y,N}\right)u^m + v^m.$$

Then,

$$\begin{aligned} u^m = u^{m-1} + \frac{\Delta t}{2}\Bigg(&\left(\frac{1}{h^\alpha_x}A^m_{x,N} + \frac{1}{h^\beta_y}A^m_{y,N}\right)u^m + v^m + \\ &\left(\frac{1}{h^\alpha_x}A^{m-1}_{x,N} + \frac{1}{h^\beta_y}A^{m-1}_{y,N}\right)u^{m-1} + v^{m-1}\Bigg), \end{aligned}$$

which by replacing $v^m + v^{m-1}$ with $v^{m-\frac{1}{2}}$, gives the CN-WSGD scheme

$$\left(I_N - rA^m_{x,N} - sA^m_{y,N}\right)u^m = \left(I_N + rA^{m-1}_{x,N} + sA^{m-1}_{y,N}\right)u^{m-1} + \Delta t v^{m-\frac{1}{2}}, \tag{2.7}$$

where $\mathrm{r} = \frac{\Delta t}{2h^\alpha_x}$, $\mathrm{s} = \frac{\Delta t}{2h^\beta_y}$. By multiplying both sides by $\frac{1}{\mathrm{r}}$ or $\frac{1}{\mathrm{s}}$ we obtain these two alternative forms of (2.7)

$$\begin{aligned} \left(\frac{1}{\mathrm{r}}I_N - A^m_{x,N} - \frac{\mathrm{s}}{\mathrm{r}}A^m_{y,N}\right)u^m &= \left(\frac{1}{\mathrm{r}}I_N + A^{m-1}_{x,N} + \frac{\mathrm{s}}{\mathrm{r}}A^{m-1}_{y,N}\right)u^{m-1} + 2h^\alpha_x v^{m-\frac{1}{2}}, \\ \left(\frac{1}{\mathrm{s}}I_N - \frac{\mathrm{r}}{\mathrm{s}}A^m_{x,N} - A^m_{y,N}\right)u^m &= \left(\frac{1}{\mathrm{s}}I_N + \frac{\mathrm{r}}{\mathrm{s}}A^{m-1}_{x,N} + A^{m-1}_{y,N}\right)u^{m-1} + 2h^\beta_y v^{m-\frac{1}{2}}. \end{aligned}$$

Summarizing, at each time step t^m the CN-WSGD scheme requires the solution of the following linear system

$$\mathscr{A}^m_{(\alpha,\beta),N}u^m = b^{m-1}, \tag{2.8}$$

where

$$\begin{aligned}\mathscr{A}^m_{(\alpha,\beta),N} &= \frac{1}{\mathrm{r}} I_N - A^m_{x,N} - \frac{\mathrm{s}}{\mathrm{r}} A^m_{y,N},\\ b^{m-1} &= \left(\frac{1}{\mathrm{r}} I_N + A^{m-1}_{x,N} + \frac{\mathrm{s}}{\mathrm{r}} A^{m-1}_{y,N}\right) u^{m-1} + 2h_x^\alpha v^{m-\frac{1}{2}},\end{aligned} \tag{2.9}$$

or, equivalently,

$$\begin{aligned}\mathscr{A}^m_{(\alpha,\beta),N} &= \frac{1}{\mathrm{s}} I_N - \frac{\mathrm{r}}{\mathrm{s}} A^m_{x,N} - A^m_{y,N},\\ b^{m-1} &= \left(\frac{1}{\mathrm{s}} I_N + \frac{\mathrm{r}}{\mathrm{s}} A^{m-1}_{x,N} + A^{m-1}_{y,N}\right) u^{m-1} + 2h_y^\beta v^{m-\frac{1}{2}}.\end{aligned} \tag{2.10}$$

2.2 Spectral properties of the coefficient matrices

It has already been observed in Lei and Sun [2013] that when $d_\pm(x,y,t) = d$ and $e_\pm(x,y,t) = e$, with $d, e \in \mathbb{R}_+$ the coefficient matrix $\mathscr{A}^m_{(\alpha,\beta),N}$ in equation (2.8) is a BTTB or 2-level Toeplitz matrix. In this section we will exploit this information on the structure to show the unconditional stability of the CN-WSGD scheme in equation (2.7).

The tools introduced in Section 1.3.1 have already been used in Moghaderi et al. [2017] to study the spectral properties of the coefficient matrix $\mathscr{A}^m_{(\alpha,\beta),N}$. Here we review some of these properties with an extra eye to the anisotropic case, and use the tools in Section 1.3.1 to study certain spectral features of the matrices $A^m_{x,N}$ and $A^m_{y,N}$ defined in equation (2.6) that will in turn be used to prove the stability of the CN-WSGD scheme in equation (2.7).

To get to the point, we first discuss the spectral properties of A^γ_N. As shown in Tian et al. [2015], the symbol of the Hermitian part of A^γ_N, defined as $\frac{A^\gamma_N + (A^\gamma_N)^{\mathrm{H}}}{2}$, is the 2π-periodic real-valued even function

$$f_\gamma(\theta) = \frac{1}{2}\left(\sum_{k=0}^{\infty} \omega_k^{(\gamma)} \mathrm{e}^{\mathrm{i}(k-1)\theta} + \sum_{k=0}^{\infty} \omega_k^{(\gamma)} \mathrm{e}^{-\mathrm{i}(k-1)\theta}\right),$$

whose simplified form defined over $[0,\pi]$ is

$$f_\gamma(\theta) = \left(2\sin\left(\frac{\theta}{2}\right)\right)^\gamma \left(\frac{\gamma}{2}\cos\left(\frac{\gamma}{2}(\theta-\pi)-\theta\right) + \frac{2-\gamma}{2}\cos\left(\frac{\gamma}{2}(\theta-\pi)\right)\right). \tag{2.11}$$

The following theorem deals with the properties of function f_γ.

Theorem 2.2.1 (Tian et al. [2015]). *Let $f_\gamma(\theta)$ be the function defined in equation* (2.11)*, then it holds that*

1. *$f_\gamma(\theta)$ is a monotonically decreasing non-positive function;*
2. *it has a zero of order γ at $\theta = 0$ and $\forall \gamma \in (1,2]$.*

Remark 2.2.1. *Notice that from item 1. in Theorem 2.2.1 yields*

$$-4 \leq 2^\gamma(1-\gamma) \leq f_\gamma(\theta) \leq 0.$$

Corollary 2.2.1 is a direct consequence of Theorem 1.3.1 and Theorem 2.2.1.

Corollary 2.2.1 (Tian et al. [2015]). *Let λ be an eigenvalue of A_N^γ in equation* (2.5)*, then* $\mathrm{Re}(\lambda) < 0$ $\forall \gamma \in (1,2]$ *and* $\mathrm{Re}(\lambda) = 0$ *if* $\gamma = 1$.

The following proposition states that the real part of the eigenvalues of both $A_{x,N}^m$ and $A_{y,N}^m$ is negative.

Proposition 2.2.1. *Assume that the diffusion coefficients are constant, i.e., $d_\pm(x,y,t) = d_\pm$, $e_\pm(x,y,t) = e_\pm$ and let λ be an eigenvalue of $A_{x,N}^m$ or $A_{y,N}^m$ in equation* (2.6)*, then* $\mathrm{Re}(\lambda) < 0$.

Proof. The thesis follows from the 2-level extension of Theorem 1.3.1 by noticing that $A_{x,N}^m$ is a 2-level Toeplitz matrix, whose symbol of the Hermitian part is

$$d_+ f_\alpha(\theta) + d_- f_\alpha(\theta),$$

which by Theorem 2.2.1 is a non-positive function. A similar reasoning shows that the real part of the eigenvalues of $A_{y,N}^m$ is negative. □

The one-dimensional CN-WSGD scheme was already shown to be unconditionally stable when $\alpha \in (1,2]$ and the diffusion coefficients are constant in Tian et al. [2015]. Following the same line of proof used in there and using Proposition 2.2.1, here we extend the unconditional stability of the CN-WSGD scheme also to the two-dimensional case.

Proposition 2.2.2. *Let $\alpha, \beta \in (1,2]$ and consider constant diffusion coefficients, then CN-WSGD scheme* (2.7) *is unconditionally stable.*

Proof. Let us define $B_N = \mathrm{r}A_{x,N}^m + \mathrm{s}A_{y,N}^m$ and rewrite the scheme (2.7) as $(I_N - B_N)u^m = (I_N + B_N)u^{m-1} + \Delta t v^{m-\frac{1}{2}}$. The corresponding iteration matrix is $M_N =$

$(I_N - B_N)^{-1}(I_N + B_N)$, which does not depend on time since we are assuming that the diffusion coefficients are constant.

Showing the thesis is equivalent to prove that $\rho(M_N) < 1$. Let λ be an eigenvalue of B_N, then $\frac{1+\lambda}{1-\lambda}$ is an eigenvalue of M_N. By linearity of the 2-level Toeplitz operator and by Proposition 2.2.1, it holds that $\mathrm{Re}(\lambda) < 0$ and therefore $\left|\frac{1+\lambda}{1-\lambda}\right| < 1$, which completes the proof. □

We end this section with a consequence of Theorem 2.2.1 that gives a precise account of what is the behavior of the symbol of $\{\mathscr{A}^m_{(\alpha,\beta),N}\}_N$ when assuming constant coefficients and depending on the parameters r, s. For more details on the derivation of item (i), we refer the reader to Moghaderi et al. [2017].

Proposition 2.2.3. *Let* $\mathrm{r}, \mathrm{s} \to \infty$ *and consider* $d_\pm(x, y, t) = d$, $e_\pm(x, y, t) = e$ *with* $e, d \in \mathbb{R}_+$. *Fixed formulation* (2.9)

(i) if $\frac{\mathrm{s}}{\mathrm{r}} \to c > 0$, *then* $\{\mathscr{A}^m_{(\alpha,\beta),N}\}_N$ *has symbol* $F_{\alpha,\beta}(\theta_1, \theta_2) = -2df_\alpha(\theta_1) - 2cef_\beta(\theta_2)$ *which has a zero of order* $\min\{\alpha, \beta\} \leq 2$ *at* $(\theta_1, \theta_2) = (0, 0)$*;*

(ii) if $\frac{\mathrm{s}}{\mathrm{r}} \to 0$, *then* $\{\mathscr{A}^m_{(\alpha,\beta),N}\}_N$ *has symbol* $F_\alpha(\theta_1, \theta_2) = -2df_\alpha(\theta_1)$ *which has a zero of order* α *at* $\theta_1 = 0, \forall \theta_2$.

On the other hand, fixed formulation (2.10)

(iii) if $\frac{\mathrm{s}}{\mathrm{r}} \to \infty$, *then* $\{\mathscr{A}^m_{(\alpha,\beta),N}\}_N$ *has symbol* $F_{\alpha,\beta}(\theta_1, \theta_2) = -2ef_\beta(\theta_2)$ *which has a zero of order* β *at* $\theta_2 = 0, \forall \theta_1$.

Notice that item *(i)* can give rise to anisotropic situations when assuming that $\alpha \neq \beta$ and that they are in some sense far from each other. An example that confirms this scenario is reported in Figure 2.1 where we plot the contour lines of $F_{\alpha,\beta}(\theta_1, \theta_2)$ when $d_\pm = e_\pm = 1$, $\frac{\mathrm{s}}{\mathrm{r}} \to 1$, $\beta = 1.1$ and $\alpha = 1.5, 1.9$. For both $\alpha = 1.9$ and $\alpha = 1.5$, the contour lines show an elliptic shape and have foci along the y-axis. This happens because $F_{\alpha,\beta}(\theta_1, \theta_2)$ has a 'stronger dependence' on θ_1 rather than on θ_2 and means that the problem is anisotropic along the y-coordinate. The choice $\frac{\mathrm{s}}{\mathrm{r}} \to 1$ highlights here a kind of anisotropy purely caused by the fractional derivative orders.

On the other hand, the ratio $\frac{\mathrm{s}}{\mathrm{r}}$ depends on the space-time grid widths whose choice can emphasize or reduce the anisotropy of the problem. If, for instance, we consider a cubic mesh such that $\Delta t = h_x = h_y$, then for $\alpha > \beta$ we get $\frac{\mathrm{s}}{\mathrm{r}} \to 0$, that is item *(ii)* in Proposition 2.2.3. This strengthens the dependence on θ_1 of $F_{\alpha,\beta}(\theta_1, \theta_2)$ and hence increases the anisotropy along the y-axis. Of course a similar reasoning applies when $\alpha < \beta$, that in the case of item *(iii)* in Proposition

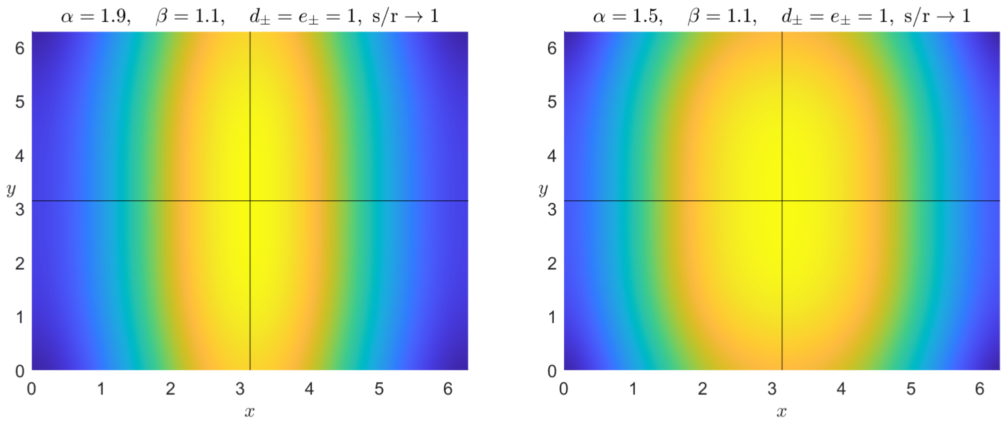


Figure 2.1. Plot of $F_{\alpha,\beta}(\theta_1, \theta_2)$ over $[0, 2\pi] \times [0, 2\pi]$

2.2.3. In other words, all cases *(i)–(iii)* can indicate that the problem suffers from anisotropy along the coordinates and they will be all subject of study in the next sections.

2.3 Multigrid methods for anisotropic FDEs

MGMs have already shown to be valid stand alone solvers as well as efficient preconditioners for Krylov methods for FDE problems Lin et al. [2017a]; Moghaderi et al. [2017]; Pang and Sun [2012]. In this section, we propose a preconditioning multigrid strategy for the FDE problem in (2.1) tailored for all the cases where the problem suffer from anisotropy, in the sense of the discussion at the end of previous section.

Thanks to the structure and the spectral properties of the coefficient matrix $\mathscr{A}^m_{(\alpha,\beta),N}$, in Moghaderi et al. [2017] the authors proposed a geometric MGM based on the classical linear interpolation as grid transfer operator and weighted Jacobi as smoother, and they proved the linear convergence rate of the corresponding TGM for the isotropic case. In the remaining of this section we review the convergence of the TGM method focusing especially on how to adapt both TGM and V-cycle to the anisotropic case.

2.3.1 TGM convergence results

In this section we will discuss the convergence of the two-grid algorithm, which, from Section 1.4.1, we recall that relies on the so-called smoothing property and approximation property.

Smoothing analysis. First, we focus on the smoothing property (1.5) for the weighted Jacobi method. Two main advantages of such method are that it can be parallelized and that the Toeplitz-like structure of the coefficient matrix can be exploited to prove related convergence results.

In the case of constant diffusion coefficients, the matrix $\mathscr{A}^{m}_{(\alpha,\beta),N}$ in equation (2.8) is a positive definite BTTB matrix and its diagonal is $D_N = \hat{c}I_N$ with $\hat{c} > 0$. As a consequence, the Jacobi iteration applied to $\mathscr{A}^{m}_{(\alpha,\beta),N}$ can be considered as a weighted Richardson iteration with weight $\omega_{\mathrm{R}} = \frac{\omega}{\hat{c}}$, and thanks to Remark 1.4.1 the smoothing property (1.5) is satisfied whenever it is convergent.

Theorem 2.3.1 provides a sufficient condition for the convergence of the weighted Jacobi method when applied to $\mathscr{A}^{m}_{(\alpha,\beta),N}$ with $D_{\pm} = dI_N$ and $E_{\pm} = eI_N$. An alternative proof of this theorem in the case where $\frac{\mathrm{s}}{\mathrm{r}} \to c$ can be found in Moghaderi et al. [2017]. Therein, the thesis has been obtained starting directly from the inequality in the smoothing property (1.5). We stress that our proof extends to the cases where $\frac{\mathrm{s}}{\mathrm{r}} \to 0$ or $\frac{\mathrm{r}}{\mathrm{s}} \to 0$.

Theorem 2.3.1. *Let $d_{\pm}(x,y,t) = d$, $e_{\pm}(x,y,t) = e$, then the weighted Jacobi method applied to* (2.8) *is convergent for any $\omega \in (0,1)$.*

Proof. Assuming that $\mathrm{r},\mathrm{s} \to \infty$, in the following we distinguish between these three cases: 1) $\frac{\mathrm{s}}{\mathrm{r}} \to c$, 2) $\frac{\mathrm{s}}{\mathrm{r}} \to 0$, 3) $\frac{\mathrm{r}}{\mathrm{s}} \to 0$.

Let us first consider case 1). From Proposition 2.2.3, we know that the symbol of the BTTB matrix-sequence $\{\mathscr{A}^{m}_{(\alpha,\beta),N}\}_N$ is

$$F_{\alpha,\beta}(\theta_1,\theta_2) = -2d\left(f_\alpha(\theta_1) + \frac{ce}{d}f_\beta(\theta_2)\right) \geq 0 \quad \theta_1,\theta_2 \in [0,\pi],$$

while its diagonal is generated by $D_{\alpha,\beta} = -2d\left(\omega_1^{(\alpha)} + \frac{ce}{d}\omega_1^{(\beta)}\right) \geq 0$. Thanks to Remark 2.2.1, and by denoting $\tilde{c} = \frac{ce}{d}$, it holds

$$F_{\alpha,\beta}(\theta_1,\theta_2) \leq F_{\alpha,\beta}(\pi,\pi) = 2d\left(2^\alpha(\alpha-1) + \tilde{c}2^\beta(\beta-1)\right), \tag{2.12}$$

and by equation (2.3) it follows

$$D_{\alpha,\beta} = d\left(\alpha^2+\alpha-2+\tilde{c}(\beta^2+\beta-2)\right) = d\big((\alpha-1)(\alpha+2)+\tilde{c}(\beta-1)(\beta+2)\big). \tag{2.13}$$

The ratio between (2.12) and (2.13) gives the symbol of $\{D_N^{-1}\mathscr{A}^m_{(\alpha,\beta),N}\}_N$,

$$\frac{F_{\alpha,\beta}(\theta_1,\theta_2)}{D_{\alpha,\beta}} \le \frac{2^{\alpha+1}(\alpha-1)+\tilde{c}2^{\beta+1}(\beta-1)}{(\alpha-1)(\alpha+2)+\tilde{c}(\beta-1)(\beta+2)} =: H(\alpha,\beta)$$

Since the study of the gradient of $H(\alpha,\beta)$ does not reveal any stationary point inside the domain $(1,2)\times(1,2)$, we can restrict our analysis to the edges. We notice that $H(\alpha,\beta)$ is not defined for $\alpha=\beta=1$, but switching to polar coordinates $\alpha = 1+\rho\cos\theta$, $\beta = 1+\rho\sin\theta$ with $\theta\in[0,\frac{\pi}{2}]$ yields

$$\begin{aligned}
\lim_{(\alpha,\beta)\to(1,1)} H(\alpha,\beta) &= \lim_{\rho\to 0}\frac{2^{2+\rho\cos\theta}\rho\cos\theta+\tilde{c}2^{2+\rho\sin\theta}\rho\sin\theta}{(3\rho\cos\theta+\rho^2\cos^2\theta)+\tilde{c}(3\rho\sin\theta+\rho^2\sin^2\theta)}\\
&= \lim_{\rho\to 0}\frac{2^{2+\rho\cos\theta}\cos\theta+\tilde{c}2^{2+\rho\sin\theta}\sin\theta}{(3\cos\theta+\rho\cos^2\theta)+\tilde{c}(3\sin\theta+\rho\sin^2\theta)}\\
&= \lim_{\rho\to 0}\frac{2^{2+\rho\cos\theta}\cos\theta+\tilde{c}2^{2+\rho\sin\theta}\sin\theta}{3(\cos\theta+\tilde{c}\sin\theta)+\rho\left(\cos^2\theta+\tilde{c}\sin^2\theta\right)}\\
&= \frac{4\cos\theta+4\tilde{c}\sin\theta}{3\cos\theta+3\tilde{c}\sin\theta} = \frac{4}{3} < 2.
\end{aligned}$$

Let us fix $\alpha=1$ and $\alpha=2$. If $\alpha=1$, then $H(1,\beta)=\frac{2^{\beta+1}}{\beta+2}$ is a monotonically increasing function and therefore $H(1,\beta)\le H(1,2)=2$. On the other hand, if $\alpha=2$, the maximum of $H(2,\beta)=\frac{8+\tilde{c}2^{\beta+1}(\beta-1)}{4+\tilde{c}(\beta-1)(\beta+2)}$ is still 2 and is obtained for $\beta=1$ and $\beta=2$. The cases where we fix $\beta=1$ or $\beta=2$ provide analogous results.

Let us now move to case 2). The symbol of $\{\mathscr{A}^m_{(\alpha,\beta),N}\}_N$, with $\mathscr{A}^m_{(\alpha,\beta),N}$ as in equation (2.9) and the symbol of $\{D_N\}_N$, with D_N diagonal of $\mathscr{A}^m_{(\alpha,\beta),N}$ are $F_{\alpha,\beta}(\theta_1,\theta_2) = -2df_\alpha(\theta_1)$ and $D_{\alpha,\beta} = -2d\omega_1^{(\alpha)}$, respectively. Therefore the symbol of $\{D_N^{-1}\mathscr{A}^m_{(\alpha,\beta),N}\}_N$ is

$$\frac{F_{\alpha,\beta}(\theta_1,\theta_2)}{D_{\alpha,\beta}} \le \frac{2^{\alpha+1}}{\alpha+2} = H(\alpha,1),$$

and the study of the first derivative leads to $\frac{2^{\alpha+1}}{\alpha+2}\le 2$.

The same reasoning that proves the thesis in case 2) applies to case 3) when formulation (2.9) of $\mathscr{A}^m_{(\alpha,\beta),N}$ is replaced by formulation (2.10).

In conclusion, from Corollary 1.3.1 we have that

$$\rho(D_N^{-1}\mathscr{A}^m_{(\alpha,\beta),N}) \leq \|H(\alpha,\beta)\|_\infty \leq 2$$

independently of α and β and the thesis follows from Theorem 1.4.2. □

Remark 2.3.1. *Theorem 2.3.1 gives a sufficient condition on ω such that the Jacobi iteration matrix, say B_N^ω, satisfies $\rho\left(B_N^\omega\right) < 1, \forall \alpha,\beta \in (1,2]$. For some combinations of α,β, more suitable values of ω could be greater than* 1.

Approximation property. As shown in Section 1.4.2, the approximation property (1.6) relies on the study of the grid transfer operators. In [Moghaderi et al., 2017, Lemma 4.4] the authors proved that when $F_{\alpha,\beta}(\theta_1,\theta_2)$ is the symbol of $\{\mathscr{A}^m_{(\alpha,\beta),N}\}_N$ with $D_\pm = dI$, $E_\pm = eI$ and $\frac{\mathrm{s}}{\mathrm{r}} \to c > 0$, then relation (1.10) is satisfied by p as in equation (1.11) with $\mathfrak{c} = 0$. This is in line with the fact that, thanks to Proposition 2.2.3, $F_{\alpha,\beta}(\theta_1,\theta_2)$ has a unique zero at $(0,0)$ of order $\min\{\alpha,\beta\} \leq 2$.

In the anisotropic cases $\frac{\mathrm{s}}{\mathrm{r}} \to 0$ or $\frac{\mathrm{r}}{\mathrm{s}} \to 0$, from Proposition 2.2.3 it holds that $F_{\alpha,\beta}(\theta_1,\theta_2)$ vanishes on a whole line and hence the theory does not apply anymore. On the other hand, $F_{\alpha,\beta}(\theta_1,\theta_2)$ describes only asymptotically the spectrum of $\mathscr{A}^m_{(\alpha,\beta),N}$, then in practice the standard TGM still works but converges so slowly that it becomes totally impractical.

More efficient alternatives for integer order partial differential equations have been proposed in Fischer and Huckle [2006, 2008]. Therein, the authors investigate the convergence of a TGM method tailored for anisotropic 2-level Toeplitz linear systems. More precisely, they propose a TGM with semi-coarsening along the perpendicular direction to the anisotropy combined with weighted Jacobi as smoother. The results in Fischer and Huckle [2006] apply also to the fractional case when the diffusion coefficients are constant as we show in the following. Assume, for instance, that the anisotropy occurs along the y-axis, i.e., $\frac{\mathrm{s}}{\mathrm{r}} \to 0$ and hence $F_{\alpha,\beta}(\theta_1,\theta_2) = -2df_\alpha(\theta_1)$ according to Proposition 2.2.3 item (ii). The coarsening is then performed only in x-direction defining the projector P_N as

$$P_N = I_{N_y} \otimes P_{N_x},$$

where

$$P_{N_x} = T_{N_x}(1+\cos\theta_1)K_{N_x}^{k_2}. \tag{2.14}$$

The symbol of P_N is then $p(\theta_1, \theta_2) = 1 + \cos\theta_1$ and it verifies

$$\limsup_{\theta_1 \to 0} \frac{1 + \cos(\theta_1 + \pi)}{F_{\alpha,\beta}(\theta_1, \theta_2)} = \mathfrak{c} < +\infty, \qquad \forall \theta_2 \in [0, 2\pi), \tag{2.15}$$

with $\mathfrak{c} = 0$, which is equivalent to condition (32) in Fischer and Huckle [2006].

We refer the reader to Sections 1.4.3 and 2.3.2 for a discussion on both full- and semi-coarsening V-cycle methods, respectively.

Convergence of geometric TGM. A multigrid with geometric approach, i.e., obtained rediscretizing the same FDE on a coarser grid, is usually less robust than a multigrid with Galerkin approach. For instance, in order to guarantee the convergence in the geometric case, a careful scaling of the grid transfer operators is required.

Proposition 2.2.3 shows that, in term of frequencies, FDEs and PDEs have a similar spectral behavior. This is because the symbol analysis can be considered as a generalization of the standard local Fourier analysis (LFA). Precisely, a LFA of the smoother or of the grid transfer operator can be performed scaling the symbol according to the mesh-size and using the order of its zero (see Donatelli [2010] for more details). As a consequence, we expect that classical smoothers defined for PDEs will work for FDEs as well.

In order to discuss the choice of the grid transfer operator in the geometric case, we first recall that isotropic FDEs have order $\min\{\alpha, \beta\} \leq 2$. This is due to the order of the zero at $(0, 0)$ of their symbol $F_{\alpha,\beta}(\theta_1, \theta_2)$ (see again Proposition 2.2.3). Moreover, based on the results in Donatelli [2010], condition (1.10) is equivalent to require that

$$2HF(p) \geq \min\{\alpha, \beta\}, \tag{2.16}$$

where $HF(p)$ is the high frequency order of p introduced in Hemker [1990], and is equal to 2 for p defined as in (1.11). By performing the same LFA discussed in Hemker [1990] for PDEs, it can be shown that condition (2.16) is sufficient to have the convergence of the geometric TGM also when applied to isotropic FDEs.

A similar reasoning applies to anisotropic variants of p as well. For instance, in case of semi-coarsening in the x-direction, condition (2.14) implies that the unidimensional version of (2.16), i.e., $2HF(p) \geq \alpha$, holds true.

2.3.2 V-cycle considerations

According to Section 1.4.3, for p defined as in equation (1.11) and $f(\theta_1, \theta_2) = F_{\alpha,\beta}(\theta_1, \theta_2)$ with $\frac{\mathrm{s}}{\mathrm{r}} \to c > 0$ the condition (1.12) holds true with $\mathfrak{c} = 0$. This suggests that the bilinear interpolation is powerful enough to work also under some perturbations. In particular, one could use the geometric multigrid instead of the Galerkin approach as done in Moghaderi et al. [2017]. We emphasize that the resulting solver could still be not enough efficient when $\alpha \neq \beta$ and the problem shows anisotropy along one of the two coordinates.

Concerning the anisotropic cases $\frac{\mathrm{s}}{\mathrm{r}} \to 0$ or $\frac{\mathrm{r}}{\mathrm{s}} \to 0$, as for TGM, also for the V-cycle the theory does not apply anymore. On the other hand, contrary to what happens for TGM, a proof of the V-cycle optimality for a MGM with semi-coarsening is still missing.

An alternative strategy that combines both full- and semi-coarsening approaches and that revealed extremely effective for anisotropic integer order differential problems has been proposed in Oosterlee [1995]; Van Lent and Vandewalle [2002]; Washio and Oosterlee [1998]. Precisely, based on the fact that the multigrid itself is a stationary method, in Oosterlee [1995]; Van Lent and Vandewalle [2002]; Washio and Oosterlee [1998] the authors design a full-coarsening V-cycle that uses another V-cycle with semi-coarsening along one direction as pre-smoother, and a V-cycle with semi-coarsening along the other direction as post-smoother. The resulting method is known as “multigrid as smoother" (MG-S) and in Van Lent and Vandewalle [2002] was introduced as a tool for dealing with anisotropic problems in the case of two-dimensional rotated diffusion equations. The use of both pre- and post-smoothing with alternate semi-coarsening simulates an alternating line smoother, which is useful for anisotropies between the axes, but computationally expensive.

When applying the MG-S method to anisotropic FDE problems of form (2.1), the computational cost can be reduced removing one of the two semi-coarsening V-cycles. The reason is that the kind of anisotropy we are considering here is only along the coordinates. In the next section we discuss a combination of this ‘light’ version of the MG-S with a banded approximation of the coefficient matrix.

2.3.3 Preconditioning

Aiming at increasing their robustness, MGMs are often applied as preconditioners for Krylov methods. One possibility is to use as preconditioner the approximation provided by few iterations of the chosen multigrid applied directly to the coefficient matrix. Another possibility, that typically results in a lower computational cost, is to replace the coefficient matrix with an approximation whose related linear systems are easier to be solved.

In the context of FDE problems, we mention the proposal in Moghaderi et al. [2017] where the coefficient matrix was approximated by a band preconditioning defined as the scaled Laplacian and combined with a MGM based on the Galerkin approach. The resulting method was shown to be efficient when both α and β were close to 2.

Here we propose a band approximation $\mathscr{P}_N^m$ of $\mathscr{A}_{(\alpha,\beta),N}^m$ in equation (2.8) that is tailored for the case where only one of the two fractional derivative orders, say α, is close to 2. Precisely, $\mathscr{P}_N^m$ is obtained from the coefficient matrix replacing

- $A_{N_x}^{\alpha}$ with the Laplacian matrix $A_{N_x}^2 = T_{N_x}(2-2\cos\theta)$. This choice is in line with the results in Moghaderi et al. [2017];
- $A_{N_y}^{\beta}$ with its penta-diagonal band truncation. This choice is justified by the decay of coefficients $\omega_k^{(\beta)}$ with respect to index k given in Lemma 2.1.1.

The computational cost for solving a linear system associated to $\mathscr{P}_N^m$ is linear in N. Based on this fact and on the discussions in Sections 2.3.1 and 2.3.2, we define a MG-S method having

- only one semi-coarsening V-cycle with weighted Jacobi as smoother;
- a hierarchy of matrices at the coarser levels obtained by the geometric approach (see Remark 2.3.2);

and we expect that applying few iterations of it to $\mathscr{P}_N^m$ gives rise to an effective preconditioner for anisotropic FDEs. Numerical confirmations of this are given in Section 2.5.

Remark 2.3.2. *Although the Galerkin approach would ensure a more robust method, estimating suitable weights for Jacobi in this case could result in a much harder task. In Section 2.4, we present a spectral-based algorithmic procedure for estimating the weights that naturally applies to the geometric strategy.*

We end this section, providing the complete algorithm of our numerical proposal. Without loss of generality, we assume that the anisotropy occurs along the y-axis. As a consequence, we opt for a MG-S preconditioning with no pre-smoother and one iteration of V-cycle with semi-coarsening in x as post-smoother. The latter V-cycle is denoted by **Vjx** and has no pre-smoother and one iteration of Jacobi as post-smoother. The overall algorithm is denoted by **MG-Sx**. In the following, we describe the steps of both **Vjx** and **MG-Sx**.

For the sake of simplicity, we assume that $N_x = N_y = 2^n$, $n \in \mathbb{N}$ and let $\boldsymbol{\mathscr{A}}^{\boldsymbol{m}}_{\boldsymbol{N}} = \{\mathscr{A}^m_{N^{(i,k)}}\}^n_{i,k=0}$, $N^{(i,k)} = N^{(i)}_x N^{(k)}_y$ with $i \geq k$ and $N^{(i)}_x = 2^{n-i}, N^{(k)}_y = 2^{n-k}$, with $N^{(0)}_x = N_x$, $N^{(0)}_y = N_y$ be the geometric hierarchy of matrices needed at each level of the full-coarsening V-cycle that defines **MG-Sx**. Moreover, let $\boldsymbol{\omega} = \{\omega_{(i,k)}\}^{n-1}_{i,k=0}$ be a sequence of proper Jacobi weights corresponding to $\boldsymbol{\mathscr{A}}^{\boldsymbol{m}}_{\boldsymbol{N}}$ (see Section 2.4 for more details on how to define $\boldsymbol{\omega}$), and let $\boldsymbol{P}^{\boldsymbol{xy}}_{\boldsymbol{N}} = \{P^{xy}_{N^{(i)}}\}^{n-1}_{i=0}$ and $\boldsymbol{P}^{\boldsymbol{x}}_{\boldsymbol{N}} = \{P^{x}_{N^{(i,k)}}\}^{n-1}_{i,k=0}$ be the following sequences of projectors

$$P^{xy}_{N^{(i)}} = P_{N^{(i)}_y} \otimes P_{N^{(i)}_x}, \quad P^{x}_{N^{(i,k)}} = I_{N^{(k)}_y} \otimes P_{N^{(i)}_x}, \quad i,k = 0,\dots,n-1, \quad i \geq k,$$

with $P_{N^{(i)}_x}, P_{N^{(i)}_y}$ defined as in (2.14). A step of **MG-Sx** is described by Algorithm 2 with $k=0$, while one iteration of **Vjx** is given in Algorithm 3.

Algorithm 2 MG-Sx method

```
$x^{(j+1)} = \mathbf{MGSx}\left(\boldsymbol{\mathscr{A}}^{\boldsymbol{m}}_{\boldsymbol{N}}, \boldsymbol{P}^{\boldsymbol{xy}}_{\boldsymbol{N}}, \boldsymbol{P}^{\boldsymbol{x}}_{\boldsymbol{N}}, \boldsymbol{\omega}, x^{(j)}, b, k\right)$
1) $r = b - \mathscr{A}^m_{N^{(k,k)}} x^{(j)}$
2) $\tilde{r} = (P^{xy}_{N^{(k)}})^{\mathrm{T}} r$
3) if $k<n$ then   $y = \mathbf{MGSx}\left(\boldsymbol{\mathscr{A}}^{\boldsymbol{m}}_{\boldsymbol{N}}, \boldsymbol{P}^{\boldsymbol{xy}}_{\boldsymbol{N}}, \boldsymbol{P}^{\boldsymbol{x}}_{\boldsymbol{N}}, \boldsymbol{\omega}, \mathbf{0}, \tilde{r}, k+1\right)$ ▷ $\mathbf{0}$ is the null vector
   else            $y = (\mathscr{A}^m_{N^{(n,n)}})^{-1} \tilde{r}$
4) $\hat{x} = x^{(j)} + P^{xy}_{N^{(k)}} y$
5) $x^{(j+1)} = \mathbf{Vjx}\left(\boldsymbol{\mathscr{A}}^{\boldsymbol{m}}_{\boldsymbol{N}}, \boldsymbol{P}^{\boldsymbol{x}}_{\boldsymbol{N}}, \boldsymbol{\omega}, \hat{x}, b, k, k\right)$ ▷ 1 iteration of Vjx given in Algorithm 3
```

Algorithm 3 Vjx method

```
$x^{(j+1)} = \mathbf{Vjx}\left(\boldsymbol{\mathscr{A}}^{\boldsymbol{m}}_{\boldsymbol{N}}, \boldsymbol{P}^{\boldsymbol{x}}_{\boldsymbol{N}}, \boldsymbol{\omega}, x^{(j)}, b, i, k\right)$
1) $r = b - \mathscr{A}^m_{N^{(i,k)}} x^{(j)}$
2) $\tilde{r} = (P^{x}_{N^{(i,k)}})^{\mathrm{T}} r$
3) if $i<n$ then   $y = \mathbf{Vjx}\left(\boldsymbol{\mathscr{A}}^{\boldsymbol{m}}_{\boldsymbol{N}}, \boldsymbol{P}^{\boldsymbol{x}}_{\boldsymbol{N}}, \boldsymbol{\omega}, \mathbf{0}, \tilde{r}, i+1, k\right)$   ▷ $\mathbf{0}$ is the null vector
   else            $y = (\mathscr{A}^m_{N^{(n,k)}})^{-1} \tilde{r}$
4) $\hat{x} = x + P^{x}_{N^{(i,k)}} y$
5) $x^{(j+1)} = \mathbf{J}\left(\mathscr{A}^m_{N^{(i,k)}}, \hat{x}, b, \omega_{(i,k)}\right)$   ▷ $\mathbf{J} = 1$ iteration of $\omega_{(i,k)}$-weighted Jacobi
```

The graph in Figure 2.2 summarizes how our **MG-Sx** looks like in the case where $n=2$. The node (i,k) represents the level of the **MG-Sx** hierarchy where the coefficient matrix is $\mathscr{A}^m_{N(i,k)}$. The arrows stand for the projection to the next level, where full line is full-coarsening and dotted line is semi-coarsening in x. The red nodes are the levels in **MG-Sx** hierarchy that require a smoothing iteration. Since the smoother is a V-cycle, to each red node corresponds a V-structure as highlighted in the blue boxes.

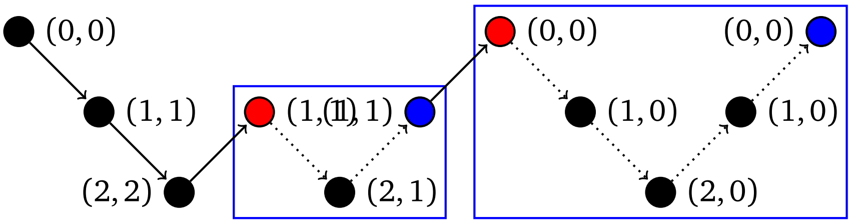


Figure 2.2. Sketch of the MG-Sx algorithm in the case where $n = 2$

2.4 Weight estimate for Jacobi

In the previous section we proved that when $\omega \in (0,1)$, the weighted Jacobi converges. This section is devoted on how to computationally estimate ω. We start showing two results on the spectrum of the weighted Jacobi iteration matrix that will be used later on. Precisely, Proposition 2.4.1 shows that the real part of the eigenvalues of the weighted Jacobi iteration matrix is smaller than 1, while Theorem 2.4.1 proves that its spectral radius is bounded by a constant independent of the chosen discretization grid.

Proposition 2.4.1. *Let $d_\pm = d$, $e_\pm = e$ and let $B^\omega_N = I_N - \omega D_N^{-1}\mathscr{A}^m_{(\alpha,\beta),N}$ be the iteration matrix of weighted Jacobi, then*

$$\mathrm{Re}\left(\lambda\left(B^\omega_N\right)\right) < 1,\ \forall \omega > 0.$$

Proof. From Proposition 2.2.1 it follows that $\mathrm{Re}(\lambda(\mathscr{A}^m_{(\alpha,\beta),N})) > 1$. Due to Lemma 2.1.1 and the fact that the diffusion coefficients are constant, the diagonal of $\mathscr{A}^m_{(\alpha,\beta),N}$ is $D_N = \hat{c}I_N$ with $\hat{c} > 0$. Then, $\mathrm{Re}(\lambda(D_N^{-1}\mathscr{A}^m_{(\alpha,\beta),N})) > 0$ and

$$\mathrm{Re}(\lambda(B^\omega_N)) = 1 - \omega\mathrm{Re}(\lambda(D_N^{-1}\mathscr{A}^m_{(\alpha,\beta),N})) < 1,\ \forall \omega > 0,$$

which concludes the proof. □

Theorem 2.4.1. *Let $\omega > 0$ be fixed and let $\hat{d}, d, \hat{e}, e \in \mathbb{R}_+$ s.t. $\hat{d} > d_\pm(x,y,t) > d$, $\hat{e} > e_\pm(x,y,t) > e$, then $\exists c_{\alpha,\beta} > 0$ s.t. $\rho(B^\omega_N) < c_{\alpha,\beta}$ independently of the chosen grid.*

Proof. We recall that $B_N^\omega = I - \omega D_N^{-1}\mathscr{A}^m_{(\alpha,\beta),N}$, where, from equations (2.9) and (2.4),

$$\begin{aligned}\mathscr{A}^m_{(\alpha,\beta),N} &= \frac{1}{\mathrm{r}}I_N - A^m_{x,N} - \frac{\mathrm{s}}{\mathrm{r}}A^m_{y,N},\\ D_N &= \frac{1}{\mathrm{r}}I_N - \left(D^m_+ + D^m_-\right)\omega_1^{(\alpha)} - \frac{\mathrm{s}}{\mathrm{r}}\left(E^m_+ + E^m_-\right)\omega_1^{(\beta)}.\end{aligned} \tag{2.17}$$

If λ is an eigenvalue of $D_N^{-1}\mathscr{A}^m_{(\alpha,\beta),N}$, then $g(\lambda) = 1-\omega\lambda$ is an eigenvalue of B_N^ω. Therefore, if $\rho(D_N^{-1}\mathscr{A}^m_{(\alpha,\beta),N}) \leq \tilde{c}_{\alpha,\beta}$ for any $\tilde{c}_{\alpha,\beta}$ independent of the grid, then there exists $c_{\alpha,\beta}$ such that $\|g\|_\infty \leq c_{\alpha,\beta}$, since g is a continuous function defined over the compact set $\left[-\tilde{c}_{\alpha,\beta}, \tilde{c}_{\alpha,\beta}\right]$.
By Gershgorin's theorem $\exists k$ such that

$$\left|\lambda - a_{k,k}\right| \leq \sum_{\substack{j=1\\ j\neq k}}^{N} a_{k,j},$$

where $a_{i,j}$ is the (i,j)-element of the matrix $D_N^{-1}\mathscr{A}^m_{(\alpha,\beta),N}$. Assume first that $k \neq 1+jN_x$ and $k\neq(j+1)N_x-1$ for $j=0,\dots,N_y-1$, then from equation (2.17),

$$\sum_{\substack{j=1\\ j\neq k}}^{N} a_{k,j} = \frac{P_1+P_2}{P_3},$$

where, from Lemma 2.1.1,

$$\begin{aligned}P_1 &= |d_k^+\omega_2^{(\alpha)} + d_k^-\omega_0^{(\alpha)}| + |d_k^-\omega_2^{(\alpha)} + d_k^+\omega_0^{(\alpha)}| + d_k^+\sum_{i=3}^{K_1}|\omega_i^{(\alpha)}| + d_k^-\sum_{i=3}^{K_2}|\omega_i^{(\alpha)}|\\ &\leq d_k^+\Big(|\omega_2^{(\alpha)}| + |\omega_0^{(\alpha)}| + \sum_{i=3}^{\infty}|\omega_i^{(\alpha)}|\Big) + d_k^-\Big(|\omega_2^{(\alpha)}| + |\omega_0^{(\alpha)}| + \sum_{i=3}^{\infty}|\omega_i^{(\alpha)}|\Big)\\ &\leq 2\hat{d}\Big(|\omega_2^{(\alpha)}| + |\omega_0^{(\alpha)}| + \sum_{i=3}^{\infty}|\omega_i^{(\alpha)}|\Big) \leq 2\hat{d}\Big(|\omega_1^{(\alpha)}| + 2|\omega_0^{(\alpha)}| + 2|\omega_2^{(\alpha)}|\Big),\\ P_2 &= \frac{\mathrm{s}}{\mathrm{r}}\Big(|e_k^+\omega_2^{(\beta)} + e_k^-\omega_0^{(\beta)}| + |e_k^-\omega_2^{(\beta)} + e_k^+\omega_0^{(\beta)}| + e_k^+\sum_{i=3}^{K_3}|\omega_i^{(\beta)}| + e_k^-\sum_{i=3}^{K_4}|\omega_i^{(\beta)}|\Big)\\ &\leq 2\frac{\mathrm{s}}{\mathrm{r}}\hat{e}\left(|\omega_1^{(\beta)}| + 2|\omega_0^{(\beta)}| + 2|\omega_2^{(\beta)}|\right),\\ P_3 &= \frac{1}{\mathrm{r}} - d_k^+\omega_1^{(\alpha)} - d_k^-\omega_1^{(\alpha)} - \frac{\mathrm{s}}{\mathrm{r}}e_k^+\omega_1^{(\beta)} - \frac{\mathrm{s}}{\mathrm{r}}e_k^-\omega_1^{(\beta)} \geq -2d\omega_1^{(\alpha)} - 2\frac{\mathrm{s}}{\mathrm{r}}e\omega_1^{(\beta)}\\ &= 2d|\omega_1^{(\alpha)}| + 2\frac{\mathrm{s}}{\mathrm{r}}e|\omega_1^{(\beta)}|,\end{aligned}$$

where $d_k^{\pm}, e_k^{\pm}$ are the k-th diagonal elements of $D_{\pm}^m, E_{\pm}^m$, respectively, and $K_i, i = 1, \dots, 4$ are four appropriate indexes. Moreover,

$$\begin{aligned}\frac{P_1+P_2}{P_3} &\leq \frac{2\hat{d}\left(|\omega_1^{(\alpha)}|+2|\omega_0^{(\alpha)}|+2|\omega_2^{(\alpha)}|\right)+2\frac{\mathrm{s}}{\mathrm{r}}\hat{e}\left(|\omega_1^{(\beta)}|+2|\omega_0^{(\beta)}|+2|\omega_2^{(\beta)}|\right)}{2d|\omega_1^{(\alpha)}|+2\frac{\mathrm{s}}{\mathrm{r}}e|\omega_1^{(\beta)}|}\\ &\leq \frac{\hat{d}}{d}\frac{|\omega_1^{(\alpha)}|+2|\omega_0^{(\alpha)}|+2|\omega_2^{(\alpha)}|}{|\omega_1^{(\alpha)}|}+\frac{\hat{e}}{e}\frac{|\omega_1^{(\beta)}|+2|\omega_0^{(\beta)}|+2|\omega_2^{(\beta)}|}{|\omega_1^{(\beta)}|} =: \hat{c}_{\alpha,\beta},\end{aligned}$$

where $\hat{c}_{\alpha,\beta}$ is a positive constant independent of r, s and of the row k. Applying small changes to the above calculations an analogous bound holds also in the cases where $k = 1 + jN_x$ or $k = (j+1)N_x - 1$ for $j = 0, \dots, N_y - 1$.
Since D_N is the diagonal matrix containing the diagonal elements of $\mathscr{A}_{(\alpha,\beta),N}^m$ it holds $a_{k,k} = 1, \ \forall k$ and therefore

$$|\lambda - 1| \leq \hat{c}_{\alpha,\beta} \iff |\lambda| \leq 1 + \hat{c}_{\alpha,\beta} =: \tilde{c}_{\alpha,\beta}$$

for any eigenvalue λ of $D_N^{-1}\mathscr{A}_{(\alpha,\beta),N}^m$, which concludes the proof. □

Remark 2.4.1. *Theorem 2.4.1 still holds if we consider* r, s *to be fixed and untied from the grid points.*

We are now ready to explain our idea about how to estimate ω. Let $N_x, N_y, M_t \in \mathbb{N}$ and consider the matrix $\mathscr{A}_{(\alpha,\beta),N}^m$ in equation (2.7) at time t^m. The crucial point of our reasoning consists in providing an estimate of the weight when Jacobi is applied to a smaller matrix whose eigenvalues approximate sufficiently well the ones of $\mathscr{A}_{(\alpha,\beta),N}^m$. More precisely, let

$$\tilde{\mathscr{A}}_{(\alpha,\beta),\tilde{N}}^m = I_{\tilde{N}} - \mathrm{r}A_{x,\tilde{N}}^m - \mathrm{s}A_{y,\tilde{N}}^m$$

with $\tilde{N} < N$ and s, r the scaling factors of $\mathscr{A}_{(\alpha,\beta),N}^m$. From Remark 2.4.1 and Theorem 2.4.1, we expect that the spectral radius of both

$$B_N^{\omega} = I_N - \omega D_N^{-1}\mathscr{A}_{(\alpha,\beta),N}^m, \quad \tilde{B}_{\tilde{N}}^{\omega} = I_{\tilde{N}} - \omega D_{\tilde{N}}^{-1}\tilde{A}_{(\alpha,\beta),\tilde{N}}^m$$

are bounded by a constant independent of the chosen grid. Furthermore, several numerical tests indicate that

$$\rho(\tilde{B}_{\tilde{N}}^{\omega}) \approx \rho(B_N^{\omega}), \tag{2.18}$$

even for variable diffusion coefficients problems. As a confirmation, we refer the reader to Figure 2.3 where, starting from Example 2 in Section 2.5, we compare the spectral distributions of both $\tilde{B}^1_{\tilde{N}}$ and B^1_N for two different levels in the MG-S hierarchy. As we can see, on both levels the eigenvalues of $\tilde{B}^1_{\tilde{N}}$ (Figures 2.3a, 2.3c) seem to properly mimic the ones of B^1_N (Figures 2.3b, 2.3d). Moreover, the accuracy of the approximation is higher at the finer level where $|\rho(B^1_N)-\rho(\tilde{B}^1_{\tilde{N}})| \approx 0.023$, with $N = 2^{14}, \tilde{N} = 2^8$ rather than at the coarser one where $|\rho(B^1_N)-\rho(\tilde{B}^1_{\tilde{N}})| \approx 0.14$, with $N = 2^{10}, \tilde{N} = 2^8$. Accordingly, we expect that the quality of the resulting estimated weights ω will be higher on the first levels of the hierarchy in MG-S, which are the most relevant ones.

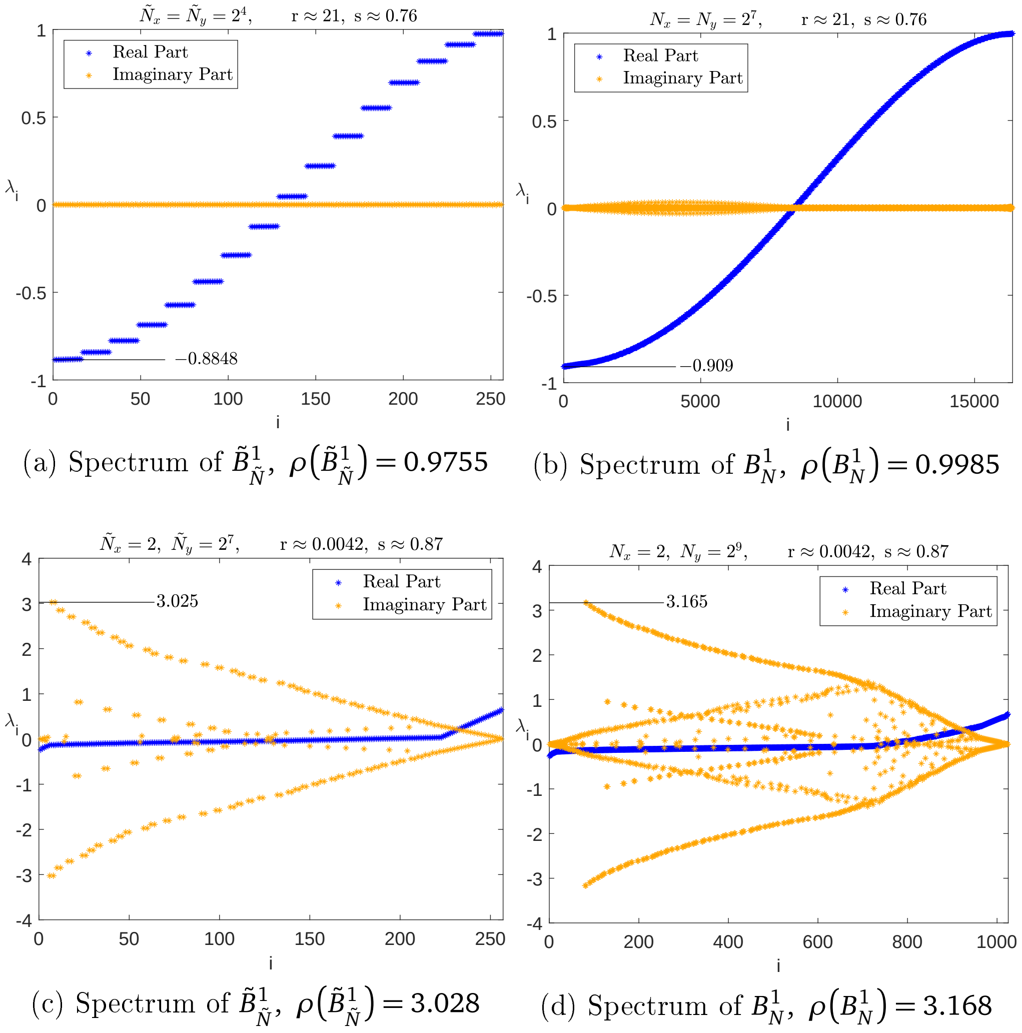
In the light of Proposition 2.4.1, Theorem 2.4.1 and relation (2.18), here we describe the main steps of the proposed spectral-based procedure for estimating the Jacobi weight (see Algorithm 4). For a fixed $\omega > 0$, let $\tilde{\lambda}^{\omega}_i$ be the i-th eigenvalue of $\tilde{B}^{\omega}_{\tilde{N}}$ and let ω_j be few equispaced weights in $(0, 1.5]$. Notice that the range $(0, 1.5]$ is motivated by Remark 2.3.1. In order to estimate a proper relaxation parameter for Jacobi applied to $\tilde{\mathscr{A}}^m_{(\alpha,\beta),\tilde{N}}$, we build a subset $\tilde{O}$ of the complex unit ball. Precisely, we consider

$$\tilde{O} = \{(x, y) \mid x \in I \subset \mathbb{R}, -\tilde{o}(x) < y < \tilde{o}(x)\},$$

where $\tilde{o}(x)$ is defined as $\tilde{o} : I \subset \mathbb{R} \to \mathbb{R}_+$, then we choose ω^* as the biggest ω_j such that $\tilde{\lambda}^{\omega_j}_i \in \tilde{O},\ \forall i$.

In order to define $I = [a, b]$, we first observe that, when assuming constant coefficients, Proposition 2.4.1 implies $\mathrm{Re}(\tilde{\lambda}^{\omega}_i) < 1,\ \forall i,\ \forall \omega > 0$. Although a formal proof that $\mathrm{Re}(\tilde{\lambda}^{\omega}_i) < 1$ holds even in the non-constant case is missing, all the numerical tests that we have carried out indicate so (refer again to Figure 2.3). On the other hand, when $\omega \to 0$, then $\mathrm{Re}(\tilde{\lambda}^{\omega}_i) \to 1$, which means that the only possible choice for b is 1. Concerning the choice of the left extreme, we require that a is far from -1 which prevents $\rho(B^{\omega^*}_N) \geq 1$. Our numerical tests show that choosing $\tilde{o}(x)$ as sum between the semi-unit-circle and the line $y = 0.4x - 0.4$, i.e., $\tilde{o}(x) = \sqrt{1-x^2} + 0.4x - 0.4$, with $I = (-\frac{21}{29}, 1) \approx (-0.7, 1)$ is good enough for our purpose.

It is well-known that the eigenvectors of Toeplitz matrices are distributed like frequencies. Therefore, assuming constant diffusion coefficients, we have $\mathrm{Re}(\lambda(B^{\omega}_N)) \approx 1$, independently on ω, for $\lambda(B^{\omega}_N)$ associated to low frequencies; on the other hand, ω can be properly chosen in order to reduce $|\lambda(B^{\omega}_N)|$ in the high frequencies. Again considering Example 2, in Figure 2.4 we show the number of oscillations of both real and imaginary parts of the eigenvectors v_i corresponding



(a) Spectrum of $\tilde{B}^1_{\tilde{N}}$, $\rho\left(\tilde{B}^1_{\tilde{N}}\right) = 0.9755$

(b) Spectrum of B^1_N, $\rho\left(B^1_N\right) = 0.9985$

(c) Spectrum of $\tilde{B}^1_{\tilde{N}}$, $\rho\left(\tilde{B}^1_{\tilde{N}}\right) = 3.028$

(d) Spectrum of B^1_N, $\rho\left(B^1_N\right) = 3.168$

Figure 2.3. Comparison of the eigenvalues of $\tilde{B}^1_{\tilde{N}}$ and B^1_N considering Example 2 with $\alpha = 1.9$, $\beta = 1.1$

Algorithm 4 Weight estimate

```
$\omega = \textbf{WeightEstimate}(\tilde{\mathscr{A}}^m_{(\alpha,\beta),\tilde{N}})$
$\tilde{D}_{\tilde{N}} = \mathrm{diag}\big(\tilde{\mathscr{A}}^m_{(\alpha,\beta),\tilde{N}}\big)$    ▷ store the diagonal elements of $\tilde{\mathscr{A}}_{(\alpha,\beta),\tilde{N}}$
$\Lambda = \mathrm{eig}\big(\tilde{D}^{-1}_{\tilde{N}}\tilde{A}^m_{(\alpha,\beta),\tilde{N}}\big)$    ▷ compute the spectrum of $\tilde{D}^{-1}_{\tilde{N}}\tilde{\mathscr{A}}_{(\alpha,\beta),\tilde{N}}$
$\tilde{o}(x) = \sqrt{1-x^2} - 0.4x + 0.4$
for $\mathrm{w} = 1.5 : -0.1 : 0.1$ do
    $\Lambda_{\mathrm{w}} = 1 - \mathrm{w}\Lambda$    ▷ compute the spectrum of $\tilde{B}^{\mathrm{w}}_{\tilde{N}}$
    if $\tilde{o}\,(\mathrm{Re}\,(\lambda)) > |\mathrm{Im}\,(\lambda)| \;\; \forall\lambda \in \Lambda_{\mathrm{w}}$ then
        $\omega = \mathrm{w}$
        break
    end if
end for
```

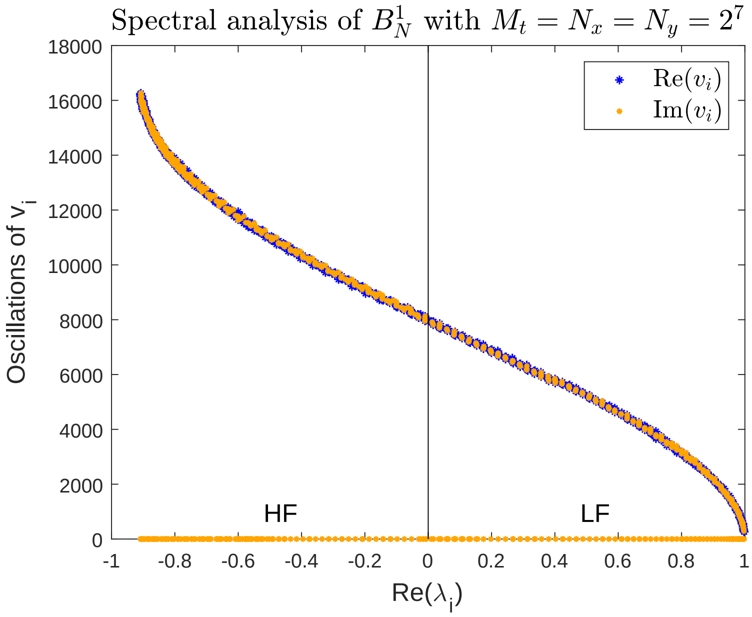


Figure 2.4. Number of oscillations of the real part and of the imaginary part of the eigenvectors of B^1_N.

to the eigenvalues λ_i. We observe that, as for the constant diffusion coefficients case, also for this variable coefficients example the oscillations of both real and imaginary parts of v_i decrease while increasing $\mathrm{Re}(\lambda_i)$ and the subdivision in low and high frequencies seems to hold.

2.5 Numerical results

In this section we test the effectiveness of the banded preconditioner $\mathscr{P}_N^m$ solved by the **MG-Sx** method described in Algorithm 2, and we compare its performances with several state of the art techniques (to be described below). The Jacobi weights required by **MG-Sx** are estimated by means of Algorithm 4 whose input matrix is the opportunely scaled preconditioner $\mathscr{P}_N^m$.

Remark 2.5.1. *Since the diffusion coefficients in the considered examples do not change too much in time, in order to reduce the CPU-time of the setup phase, Algorithm 4 performs only two estimates of ω at times $t = 0$ and $t = T$. The weight ω_m, which will be used for solving the linear system at time t^m, is computed by linear interpolation as*

$$\omega_m = \omega_0 + m\frac{\omega_{M_t} - \omega_0}{M_t}.$$

As a comparison we consider the following multigrid/circulant preconditioners:

1) **V(Gal)**, which is a V-cycle with Galerkin approach with one iteration of pre- and post- Jacobi smoother whose weights are fixed to $\omega = 1$, applied to the two-dimensional scaled Laplacian, namely $\mathscr{A}_{(2,2),N}^m$ (refer to Moghaderi et al. [2017]);

2) **V(Geo)**, which is the geometric version of the algorithm in point 1) directly applied to the original matrix $\mathscr{A}_{(\alpha,\beta),N}^m$ (refer again to Moghaderi et al. [2017]);

3) **Vx(Gal)** and **Vx(Geo)**, which are the same algorithms as in points 1) and 2), respectively, but with semi-coarsening in x;

4) **Vjx**, described in Algorithm 3 and applied to preconditioner $\mathscr{P}_N^m$ with geometric approach and with Jacobi weights estimated by means of Algorithm 4;

5) **SP**, which is the circulant version of the splitting preconditioner introduced in Lin et al. [2017b]. Such preconditioner was designed for the case where

$d_+(x,y,t) = d_-(x,y,t)$ and $e_+(x,y,t) = e_-(x,y,t)$, but we extend it to the general case by averaging the left and right diffusion coefficients. **SP** is then defined as $W_N \mathscr{C}(T_N)$, where

$$
\begin{aligned}
&W_N = I_N + D_N^m + E_N^m,\\
&D_N^m = \frac{D_+^m + D_-^m}{2}, \quad E_N^m = \frac{E_+^m + E_-^m}{2},\\
&T_N = \overline{\theta}_N I_N + \mathrm{r}\,\overline{d}_N I_{N_y} \otimes \left(A_{N_x}^{\alpha} + A_{N_x}^{\alpha}{}^{\mathrm{T}}\right) + \mathrm{s}\,\overline{e}_N \left(A_{N_y}^{\beta} + A_{N_y}^{\beta}{}^{\mathrm{T}}\right) \otimes I_{N_x},\\
&\overline{\theta}_N = \operatorname{mean}\left(W_N^{-1}\right), \quad \overline{d}_N = \operatorname{mean}\left(D_N^m W_N^{-1}\right), \quad \overline{e}_N = \operatorname{mean}\left(E_N^m W_N^{-1}\right),
\end{aligned}
$$

and $\mathscr{C}(T_N)$ is the the T. Chan optimal circulant approximation of T_N.

The reason why we went for this circulant version of the preconditioner in Lin et al. [2017b] is twofold:

- it allows the use of the FFTs instead of the multigrid with block-Jacobi employed in Lin et al. [2017b] for solving the Toeplitz splitting preconditioner;
- in all our experiments, the number of iteration provided by the circulant splitting grows slowly.

As a consequence, we obtain an overall algorithm which is simpler and more robust than if we would have used the Toeplitz splitting.

Remark 2.5.2. *The direct solution in any of the V-cycles at points 1)-2) occurs at the first level where the mesh size is smaller or equal to* 2^4. *In the V-cycles with semi-coarsening in* x *at points 3)-4) instead, the direct solution is computed at the lowest possible level of the hierarchy, i.e., when the size in the* x*-direction is equal to 1. For the full- and semi-coarsenings occurring inside* ***MG-Sx*** *we combine both choices. All methods use the Matlab function* `backslash` *as solver at the coarsest level.*

In the following, we choose as main solver for solving the linear system (2.8) the GMRES method, and we perform it computationally by the built-in `gmres` Matlab function. The initial guess at time t^m is taken as the solution at time t^{m-1} and for each multigrid preconditioner listed above we perform only 1 iteration. Notice that, the GMRES stopping criterion is based on the preconditioned residual which is usually lower than the residual, therefore to get a tolerance on the residual of 10^{-7}, we set a smaller input tolerance of 10^{-8}.
All the numerical tests have been performed using Matlab 9.3 software on Win-

dows 10 machine with AMD FX-8350 8-core (4.00 GHz) processor and 16 GB (1333 MHz) ram.

In the following tables, for all tested methods, we report the CPU-times in seconds and, in brackets, the average number of iterations

$$\mathrm{It} = \frac{\sum_{k=1}^{M_t} \mathrm{It}_k}{M},$$

where It_k are the iterations required for solving (2.8) at time t^k. The CPU-time of the setup-phase is not reported in the tables since if properly implemented, all methods show a comparable behavior.

When the execution times exceed 1 hour, we display the average iterations till the maximum available time step. Moreover, a dash is displayed if either the input tolerance cannot be reached or the maximum number of iterations per time step, which is set to 100, is exceeded.

For both Example 1 and Example 2, we fix $\beta = 1.1$ and $N_x = N_y = M_t - 1 = n$.

Example 1. Let us consider the CN-WSGD discretization of the FDE problem in (2.1) with $\Omega = [0,1] \times [0,1]$, $T = 1$ and

- $d_+(x,y,t) = 4(1+t)x^\alpha(1+y), \quad d_-(x,y,t) = 4(1+t)(1-x)^\alpha(1+y)$,
- $e_+(x,y,t) = 4(1+t)(1+x)y^\beta, \quad e_-(x,y,t) = 4(1+t)(1+x)(1-y)^\beta$,
- $u_0(x,y,0) = x^3y^3(1-x)^3(1-y)^3$,
- $u(x,y,t) = 0$ for $(x,y) \in \mathbb{R}^2 \setminus \Omega$, $t \in [0,T]$,
- $v(x,y,t)$ is such that the solution to the FDE is given by

$$u(x,y,t) = \mathrm{e}^{-t}x^3y^3(1-x)^3(1-y)^3.$$

For this example both time and space intervals have the same length, then $\Delta t = h_x = h_y$. Table 2.1 shows that, independently of α and n, the number of iterations provided by **MG-Sx** is always smaller than the one provided by the other methods. Moreover, for large-sized problems ($n+1 \geq 2^8$) the **MG-Sx** method outperforms the other solvers also in terms of CPU-times. This is particularly evident for $\alpha = 1.9$, and still occurs for $\alpha = 1.5$, although in the latter case the **MG-Sx** gets worse and the performances of **SP** improve.

α	$n+1$	**MG-Sx**	**Vjx**	**SP**	**V(Gal)**	**Vx(Gal)**	**V(Geo)**	**Vx(Geo)**
		T(s) (It)	T(s) (It)	T(s) (It)	T(s) (It)	T(s) (It)	T(s) (It)	T(s) (It)
1.9	2^4	1.1 (8.6)	0.5 (9.3)	**0.23** (27)	0.52 (26)	0.38 (17)	0.97 (22)	0.77 (12)
	2^5	2.9 (8)	1.4 (11)	1.1 (36)	2 (39)	**0.99** (18)	4.3 (33)	2.4 (13)
	2^6	8.82 (7.6)	**4.79** (13)	6.66 (44)	10.9 (48)	4.81 (21)	20.6 (41)	16 (27)
	2^7	29 (6.9)	**26** (14)	59 (47)	82 (49)	32 (20)	122 (42)	261 (66)
	2^8	**173** (7)	208 (14)	536 (46)	891 (60)	249 (16)	1484 (49)	-
	2^9	**1509** (7.2)	1986 (14)	>1h (41)	>1h (65)	2907 (19)	>1h (48)	-
1.5	2^4	1.6 (15)	0.76 (16)	**0.18** (23)	0.63 (31)	0.43 (20)	0.78 (17)	0.7 (11)
	2^5	4.4 (14)	2.1 (17)	**0.77** (28)	1.7 (34)	1.2 (22)	3.1 (23)	3 (16)
	2^6	13.1 (13)	6.42 (18)	**4.33** (31)	9.02 (41)	5.01 (22)	13.7 (28)	22.7 (38)
	2^7	42 (11)	34 (19)	35 (32)	69 (43)	**32** (20)	80 (29)	-
	2^8	**224** (10)	263 (19)	316 (29)	599 (43)	258 (17)	684 (23)	-
	2^9	**2047** (11)	2495 (19)	2622 (26)	>1h (47)	3272 (22)	>1h (23)	-

Table 2.1. Example 1 - $\beta = 1.1$, $\alpha = 1.5, 1.9$

Notice that, switching from $\alpha = 1.9$ to $\alpha = 1.5$, the anisotropy decreases, while the average iterations of both **MG-Sx** and **Vjx** increase. The reason is that the quality of the approximation provided by the band preconditioner $\mathscr{P}_N^m$ slows down as the distance between α and 2 increases.

We recall that, due to the structure of $\mathscr{P}_N^m$, the computational cost of **MG-Sx** is linear in the matrix-size, and this combined with the low number of iterations justifies its robustness as the grid becomes finer. As the number of iterations stays almost constant with respect to the problem size, we expect a similar trend also for $n+1 > 2^9$.

A comparison of the considered multigrid strategies shows that the use of the semi-coarsening is very effective when combined with the Galerkin approach (refer to **V(Gal)** and **Vx(Gal)** in Table 2.1), while it requires a fair estimation of the Jacobi weight when the Galerkin approach is replaced by the geometric one (compare **Vx(Geo)** with **Vjx**). For this example **Vjx** is indeed already very robust even without being combined with **MG-Sx**.

We note that even if the diffusion coefficients do not satisfy the hypothesis in Theorem 2.4.1, the estimated weights are still good enough to guarantee a faster convergence of **MG-Sx** with respect to the other solvers.

T	$n{+}1$	**MG-Sx**		**Vjx**		**SP**		**V(Gal)**		**Vx(Gal)**		**V(Geo)**		**Vx(Geo)**	
		T(s)	(It)	T(s)	(It)	T(s)	(It)	T(s)	(It)	T(s)	(It)	T(s)	(It)	T(s)	(It)
1	2^4	1.4	(8)	0.59	(9)	**0.17**	(17)	0.61	(23)	0.39	(14)	1.1	(20)	0.8	(11)
	2^5	3.2	(7.4)	1.4	(9)	**0.66**	(22)	1.9	(33)	0.77	(12)	4.2	(29)	2.4	(11)
	2^6	8.91	(7)	3.51	(8)	3.83	(28)	8.56	(37)	**3.26**	(13)	18.5	(35)	7.58	(11)
	2^7	28	(6)	**18**	(9)	34	(30)	61	(38)	22	(13)	103	(35)	50	(12)
	2^8	**151**	(6)	155	(10)	346	(31)	527	(37)	181	(11)	1153	(38)	962	(25)
	2^9	**1232**	(5.8)	1322	(9)	2943	(30)	>1h	(32)	2203	(13)	>1h	(30)	>1h	(47)
2	2^4	1.1	(9)	0.58	(10)	**0.18**	(18)	0.58	(26)	0.43	(16)	1.1	(22)	0.8	(11)
	2^5	3.2	(8)	1.4	(9.4)	**0.76**	(27)	2.1	(37)	0.87	(14)	4.9	(34)	2.5	(12)
	2^6	8.83	(7)	4.15	(10)	6.28	(36)	12.4	(47)	**3.74**	(15)	22.9	(43)	8.33	(12)
	2^7	31	(7)	**21**	(11)	54	(43)	96	(54)	28	(17)	149	(49)	98	(26)
	2^8	**159**	(6)	194	(13)	560	(48)	1013	(65)	251	(16)	$> \frac{1}{2}$h	(59)	$> \frac{1}{2}$h	(62)
	2^9	**1323**	(6.4)	2010	(15)	>1h	(49)	-		3224	(20)	>1h	(61)	-	
4	2^4	1.3	(9.9)	0.6	(10)	**0.19**	(20)	0.66	(29)	0.47	(20)	1.2	(25)	0.89	(12)
	2^5	3.4	(9)	1.5	(10)	0.94	(30)	2.3	(42)	**0.94**	(16)	5.4	(38)	2.7	(13)
	2^6	9.75	(8)	4.41	(11)	6.41	(43)	13.1	(54)	**4.19**	(17)	26.6	(50)	8.95	(13)
	2^7	34	(8)	**26**	(14)	77	(56)	143	(72)	33	(20)	202	(63)	164	(43)
	2^8	**196**	(8)	248	(17)	852	(66)	-		389	(25)	$> \frac{1}{2}$h	(81)	-	
	2^9	**1557**	(8)	2761	(21)	>1h	(71)	-		>1h	(28)	-		-	

Table 2.2. Example 2 - $\alpha = 1.9$, $\beta = 1.1$, and $T \in \{1,2,4\}$

Example 2. For this example we consider the CN-WSGD discretization of the FDE problem in (2.1) with $\Omega = [0,2] \times [0,2]$, $T = 2$ and

- $d_+(x,y,t) = \Gamma(3-\alpha)(1+x)^{\alpha}(1+y)^2$, $d_-(x,y,t) = \Gamma(3-\alpha)(3-x)^{\alpha}(3-y)^2$,
- $e_+(x,y,t) = \Gamma(3-\beta)(1+x)^2(1+y)^{\beta}$, $e_-(x,y,t) = \Gamma(3-\beta)(3-x)^2(3-y)^{\beta}$,
- $u_0(x,y,0) = 16x^2(2-x)^2y^2(2-y)^2$,
- $u(x,y,t) = 0$ for $(x,y) \in \mathbb{R}^2 \setminus \Omega$, $t \in [0,T]$,
- $v(x,y,t)$ is such that the solution to the FDE is given by

$$u(x,t) = 16\mathrm{e}^{-t}x^2(2-x)^2y^2(2-y)^2.$$

Tables 2.2 and 2.3 refer to $\alpha = 1.9$ and $\alpha = 1.5$, respectively, and show the performances of all the considered methods for $T \in \{1,2,4\}$ and then $\Delta t \in \{\frac{h_x}{2}, h_x, 2h_x\}$. As a consequence of this choice, the ratio $\frac{\mathrm{r}}{\mathrm{s}}$ does not change, while the scaling parameters r, s increase. All the methods, except **MG-Sx**, suffer from the subsequent worsening of the ill-conditioning of the coefficient matrix $\mathscr{A}^m_{(\alpha,\beta),N}$. By contrary, **MG-Sx** still shows linear convergence for all T, and outperforms the other solvers both in terms of CPU-times and iterations as the matrix-size increases.

T	$n+1$	**MG-Sx**		**Vjx**		**SP**		**V(Gal)**		**Vx(Gal)**		**V(Geo)**		**Vx(Geo)**	
		T(s)	(It)	T(s)	(It)	T(s)	(It)	T(s)	(It)	T(s)	(It)	T(s)	(It)	T(s)	(It)
1	2^4	1.6	(14)	0.8	(15)	**0.16**	(18)	0.62	(27)	0.43	(18)	0.85	(17)	0.75	(10)
	2^5	4.6	(13)	2	(15)	**0.63**	(21)	1.7	(30)	1	(17)	3.1	(21)	2.3	(11)
	2^6	12.7	(11)	5.63	(15)	**3.2**	(23)	8.28	(36)	3.96	(16)	12.8	(24)	8.91	(13)
	2^7	42	(10)	27	(15)	25	(24)	61	(38)	**25**	(15)	70	(24)	212	(54)
	2^8	**220**	(9.3)	221	(15)	248	(23)	514	(37)	243	(15)	622	(20)	-	
	2^9	**1815**	(9.8)	2004	(15)	2053	(21)	>1h	(42)	3147	(20)	>1h	(22)	-	
2	2^4	1.7	(15)	0.85	(16)	**0.18**	(20)	0.77	(34)	0.5	(22)	0.93	(19)	0.82	(12)
	2^5	5.1	(15)	2.2	(17)	**0.73**	(25)	2.1	(37)	1.2	(21)	3.8	(26)	2.7	(13)
	2^6	14.4	(13)	6.77	(18)	**4.31**	(31)	11	(46)	4.83	(20)	16.4	(31)	12	(18)
	2^7	51	(13)	37	(20)	38	(33)	91	(52)	**34**	(20)	93	(32)	365	(86)
	2^8	**265**	(12)	332	(23)	380	(34)	884	(58)	351	(22)	859	(28)	-	
	2^9	**2160**	(12)	3154	(24)	3176	(32)	>1h	(67)	>1h	(39)	>1h	(31)	-	
4	2^4	1.9	(17)	0.93	(18)	**0.19**	(22)	0.83	(37)	0.61	(27)	1	(20)	0.92	(13)
	2^5	5.7	(17)	2.5	(19)	**0.83**	(28)	2.4	(43)	1.4	(25)	4.2	(29)	3.2	(16)
	2^6	16.4	(16)	7.87	(21)	**5.36**	(37)	13.9	(56)	5.83	(24)	19.1	(36)	16.8	(27)
	2^7	59	(15)	48	(26)	53	(43)	126	(66)	**43**	(25)	119	(40)	-	
	2^8	**324**	(15)	466	(32)	520	(45)	1317	(80)	489	(31)	1126	(37)	-	
	2^9	**2480**	(14)	>1h	(37)	>1h	(44)	-		>1h	(59)	>1h	(39)	-	

Table 2.3. Example 2 - $\alpha = 1.5$, $\beta = 1.1$, and $T \in \{1, 2, 4\}$

2.6 Conclusions

In this chapter, we have investigated multigrid preconditioners for two-dimensional anisotropic FDEs where the anisotropy is generated by fractional derivative orders that are largely different in the two directions x and y. For severe anisotropic cases, the classical semi-coarsening strategy is not enough for obtaining a robust preconditioner and hence we propose two additional improvements. The first is an automatic estimation of the Jacobi weight driven by the symbol analysis previously introduced in Moghaderi et al. [2017] and here extended also to the anisotropic case. The second is the use of the multigrid as smoother according to a simplified version of the proposal in Oosterlee [1995]. The resulting method is applied as preconditioner and the computational cost is further reduced by approximating the fractional operators with matrices that have only few nonzero diagonals. Finally, we stress that the proposed MG-S preconditioner is highly parallelizable thanks to the two-level band structure of the coefficients matrices, the V-cycle and the Jacobi smoother.

Chapter 3

Parallel in time scheme

Despite the encouraging results obtained for the time stepping scheme in Chapter 2, with the sequential approach we cannot aspire towards a complete independency of time of its computational cost. This is because of the sequentiality of the time integration. By contrary, an all-at-once rephrasing of the discretized problem over a uniform space-time grid, obtained by considering the time as an additional dimension, yields large (multilevel) Toeplitz linear systems and opens to parallelization.

In Section 3.1 we fix our attention on the WGSD discretization, as in Chapter 2, of a one-dimensional time-dependent space-FDE with constant diffusion coefficients, by giving the formal expression and the structure of the resulting matrices. We stress that this one-dimensional problem turns out to be already a tough one, due to the block structure of the coefficient matrix and to its possibly anisotropic character because of the grid choice and the diffusion coefficients.
As for the time discretization, we opt either for CN or BDF2 schemes. The unconditional stability of CN-WSGD has already been proven in Tian et al. [2015]. Concerning BDF2, in Liao et al. [2018] it was combined with a central finite difference scheme for solving space-FDEs with diffusion coefficients equal to 1. In that same paper, a proof of unconditional stability of the resulting method was given. In Section 3.2 we extend this result to the case where the space scheme is WSGD and the diffusion coefficients are not necessarily equal to each other.
In Section 3.3, we perform an all-at-once rephrasing of the original matrices and give some results on their spectra, which are leveraged in Section 3.4 for the design of proper multigrid strategies. Finally, several numerical experiments, also in the case of variable diffusion coefficients, are reported in Section 3.5 for

testing the performances of our proposals. Finally, in Section 3.6 we draw conclusions.

3.1 Problem setting and discretization

In this section we introduce the FDE problem we are interested in, which coincides with the one-dimensional version of problem (2.1), and we briefly review the combination of the chosen finite difference space-discretization with two different time discretization schemes (Section 3.1.2).

3.1.1 One-dimensional space-FDE

We focus on the following one-dimensional initial-boundary value space-FDE problem

$$\begin{cases} \dfrac{\partial u(x,t)}{\partial t} = d_+ \dfrac{\partial^\alpha u(x,t)}{\partial_+^R x^\alpha} + d_- \dfrac{\partial^\alpha u(x,t)}{\partial_-^R x^\alpha} + v(x,t), & \\ & (x,t) \in \Omega \times [0,T], \\ u(x,t) = 0, & (x,t) \in (\mathbb{R} \setminus \Omega) \times [0,T], \\ u(x,0) = u_0(x), & x \in \overline{\Omega}, \end{cases} \tag{3.1}$$

where $\Omega = (a,b)$ is the space domain, $d_\pm > 0$ are the diffusion coefficients, $v(x,t)$ is the forcing term.

3.1.2 Space-time discretizations: CN-WSGD and BDF2-WSGD

In the following, we briefly review the finite difference space-discretization of problem (3.1) obtained using the WSGD scheme, already introduced in Section 1.6.4, combined with either CN or BDF2 schemes in time.

Let $N, M \in \mathbb{N}$ and consider the following uniform space-time grid

$$x_i = a + i\Delta x, \quad \Delta x = \frac{b-a}{N+1}, \quad t^m = t_0 + m\Delta t, \quad \Delta t = \frac{T}{M}.$$

As reported in Chapter 2, the discretization of operator $\frac{\partial^\alpha}{\partial_+^R x^\alpha}$ yields the lower Hessenberg Toeplitz matrix $A_N^\alpha = T_N(f_\alpha)$ in equation (2.5), where

$$f_\alpha(x) = \sum_{k=0}^{\infty} \omega_k^{(\alpha)} \mathrm{e}^{\mathrm{i}(k-1)x},$$

with coefficients $\omega_k^{(\alpha)}$ defined in equation (2.3). Similarly, the discretization of $\frac{\partial^\alpha}{\partial_-^R x^\alpha}$ yields an upper Hessenberg Toeplitz matrix, which coincides with A_N^T.

The discretization of the forcing term returns vector $v^m = [v(x_i, t^m)]_{i=1}^N$ and the application of CN and BDF2 schemes in time gives the following linear systems

$$\left(I_N - \mathrm{r}A_{x,N}\right)u^m = \left(I_N + \mathrm{r}A_{x,N}\right)u^{m-1} + \frac{\Delta t}{2}(v^m + v^{m-1}), \tag{3.2}$$

$$\left(I_N - \frac{4}{3}\mathrm{r}A_{x,N}\right)u^m = \frac{4}{3}u^{m-1} - \frac{1}{3}u^{m-2} + \frac{2}{3}\Delta t v^m, \tag{3.3}$$

respectively, where $\mathrm{r}=\frac{\Delta t}{2\Delta x^\alpha}$ and

$$A_{x,N} = d_+ A_N^\alpha + d_- A_N^{\alpha\,\mathrm{T}}.$$

Remark 3.1.1. *In the case of BDF2, the solution u^1 at time t^1 is computed with CN. Note that any other one step method could be used to compute u^1. For example, although Implicit Euler is only first order accurate, if we only use it once it will not compromise the global second-order accuracy. Such a statement can be found in Thomée [1984].*

The following proposition, which plays an important role in the definition of the projectors for our multigrid strategy (see Section 3.4), defines the symbol of the spatial discretization.

Proposition 3.1.1. *Let $d_\pm = d$, then $A_{x,N} = d \cdot T_N(g_\alpha)$, where $g_\alpha(x) = f_\alpha(x) + \overline{f}_\alpha(x)$ is non-positive and has a zero of order α at $x = 0$.*

3.2 Stability of the BDF2-WSGD scheme

In Tian et al. [2015] the authors proved the unconditional stability of the CN-WSGD scheme (3.2), in the constant diffusion coefficients case, as a consequence of the following theorem.

Theorem 3.2.1 (Tian et al. [2015]). *Let λ be an eigenvalue of $A_{x,N}$, then* $\mathrm{Re}(\lambda) < 0$, $\forall \alpha \in (1, 2)$.

Indeed, in case of a one step scheme like CN, the stability relies on the spectral radius of the iterations matrix of the time stepping algorithm, which is required to be lower than 1. We now aim to prove the stability of the BDF2-WSGD scheme.

After the discretization in space we obtain an equation of the form

$$\frac{\mathrm{d}}{\mathrm{d}t}\mathbf{u}(t) = A_{x,N}\mathbf{u}(t) + \mathbf{v}(t), \tag{3.4}$$

where $\mathbf{u}(t), \mathbf{v}(t)$ are, respectively, the semi-discrete in space unknown and forcing term.
The region of absolute stability of a linear multistep method is defined as follows.

Definition 3.2.1 (Absolute stability region). *The region of absolute stability for the linear multistep method*

$$\sum_{j=0}^{s} \alpha_j \mathbf{u}^{n+j} = \lambda \Delta t \sum_{j=0}^{s} \beta_j \mathbf{u}^{n+j},$$

is the set of points $z \in \mathbb{C}$ for which the roots $\{\zeta_j\}_{j=1}^{s}$ of the polynomial

$$\pi(\zeta, z) = \sum_{j=0}^{s} \left(\alpha_j - z\beta_j\right)\zeta^j, \quad z = \lambda \Delta t, \tag{3.5}$$

satisfies the following root conditions:

a) $\left|\zeta_j\right| \leq 1$, *for* $j = 1, \ldots, s$,

b) if ζ_j *is a repeated root, then* $\left|\zeta_j\right| < 1$.

Therefore, again as a consequence of Theorem 3.2.1, the following theorem holds.

Theorem 3.2.2. *Let* $\alpha \in (1,2)$ *and consider* $A_{x,N}$ *to be diagonalizable, then BDF2-WSGD scheme* (3.3) *is unconditionally stable.*

Proof. Since $A_{x,N}$ is diagonalizable there exists an invertible matrix V such that $A_{x,N} = V^{-1}\Lambda V$, where Λ is the diagonal matrix containing the eigenvalues of $A_{x,N}$. Therefore, by introducing $\tilde{u} = Vu$, equation (3.3) can be written as N uncoupled equations with respect to $\tilde{u}$:

$$(I_N - \frac{4}{3}\mathrm{r}\Lambda)\tilde{u}^m = \frac{4}{3}\tilde{u}^{m-1} - \frac{1}{3}\tilde{u}^{m-2} + \frac{2}{3}\Delta t V v^m.$$

Let us fix a row index i, then, by definition of r,

$$(1 - \frac{2}{3}\frac{\Delta t}{\Delta x^{\alpha}}\lambda_i)\tilde{u}_i^m - \frac{4}{3}\tilde{u}_i^{m-1} + \frac{1}{3}\tilde{u}_i^{m-2} = \frac{2}{3}\Delta t (V v^m)_i,$$

which, in the polynomial form of equation (3.5), becomes

$$\pi(\zeta,z)=(1-\frac{2z}{3\Delta x^{\alpha}})\zeta^{2}-\frac{4}{3}\zeta+\frac{1}{3}.$$

By defining $\tilde{z}:=1-\frac{2z}{3\Delta x^{\alpha}}$, which is a complex number and can be written as $\tilde{z}=a+ib$, it follows that the roots are

$$\left|\zeta_{1,2}\right|=\left|\frac{\frac{4}{3}\pm\sqrt{\frac{16}{9}-\frac{4}{3}\tilde{z}}}{2\tilde{z}}\right|\leq\frac{\left|2+2\sqrt{1-\frac{3}{4}\tilde{z}}\right|}{3\left|\tilde{z}\right|}=\frac{\left|2+2\sqrt{1-\frac{3}{4}(a+ib)}\right|}{3\sqrt{a^{2}+b^{2}}}=:g(a,b).$$

From Theorem 3.2.1, we have that $\mathrm{Re}(z)=a>1$ for $\alpha\in(1,2)$, and the study of the maximum of function $g(a,b)$ shows that $\sup\limits_{a>1} g(a,b)<1$. □

The following corollary exploits the density of diagonalizable matrices into the space of square matrices to remove the diagonalizability hypothesis in Theorem 3.2.2.

Corollary 3.2.1. *Let $\alpha\in(1,2)$, then BDF2-WSGD scheme* (3.3) *is unconditionally stable.*

Proof. Let us suppose that $A_{x,N}$ is not diagonalizable, otherwise the thesis follows from Theorem 3.2.2. Let us consider the Schur decomposition QTQ^{H} of $A_{x,N}$, where Q and T are unitary and upper triangular matrices, respectively. Note that due to the structure of T, its diagonal elements are the eigenvalues of T and that, by similarity, they coincide with the eigenvalues of $A_{x,N}$.
Since we are assuming that $A_{x,N}$ is not diagonalizable, T has at least two diagonal elements that are equal. Let us then consider matrix $B_N=Q\tilde{T}Q^{\mathrm{H}}$, where $\tilde{T}$ is obtained from T by properly shifting its diagonal entries such that $\tilde{T}$ becomes diagonalizable. More precisely, for $\epsilon>0$ and for $i=1,\ldots,N$ the i-th diagonal element of $\tilde{T}$ is $\tilde{t}_{ii}=t_{ii}-\delta_i$ with $0\leq\delta_i<\epsilon$ such that $\tilde{t}_{ii}\neq\tilde{t}_{jj}$, $\forall i,j=1,\ldots,N$ and $i\neq j$.
Since B_N is diagonalizable and its eigenvalues have negative real part, Theorem 3.2.2 applies to B_N, and since

$$\left\|B_N-A_{x,N}\right\|_2=\left\|\tilde{T}-T\right\|_2=\max_{i=1,\ldots,N}\delta_i<\epsilon,$$

by letting $\epsilon\to 0$ the thesis is proven. □

Remark 3.2.1. *The unconditional stability of BDF2 combined with a central finite difference scheme for discretizing the fractional derivative operator, was given in*

Liao et al. [2018]. Therein, the diffusion coefficients were both equal to 1. Under the diagonalizability hypothesis, Theorem 3.2.2 extends the unconditional stability of BDF2 to the case where the space scheme is WSGD and the diffusion coefficients are not necessarily equal to each other. Corollary 3.2.1 generalizes Theorem 3.2.2 to the case of a non diagonalizable matrix $A_{x,N}$.

3.3 All-at-once rephrasing of our problem and related spectral study

An all-at-once approach consists in considering the time like an additional dimension. Starting from equations (3.2) and (3.3), and chaining the unknown u^m at each time step into a unique vector as

$$u_{\mathrm{CN}} = [u^1, ..., u^M]^{\mathrm{T}} \in \mathbb{R}^{NM},\ u_{\mathrm{BDF2}} = [u^2, ..., u^M]^{\mathrm{T}} \in \mathbb{R}^{N(M-1)}$$

in the case of CN and BDF2, respectively, we can rephrase the original discretized problem as the following large block linear systems

$$A_{\mathrm{S}} u_{\mathrm{S}} = b_{\mathrm{S}},\ \mathrm{S} \in \{\mathrm{CN,BDF2}\}, \tag{3.6}$$

where

$$A_{\mathrm{CN}} = \begin{pmatrix} I - \mathrm{r}A_{x,N} & 0 & 0 & 0 \\ -I - \mathrm{r}A_{x,N} & I - \mathrm{r}A_{x,N} & 0 & 0 \\ 0 & \ddots & \ddots & 0 \\ 0 & 0 & -I - \mathrm{r}A_{x,N} & I - \mathrm{r}A_{x,N} \end{pmatrix}, \tag{3.7}$$

$$A_{\mathrm{BDF2}} = \begin{pmatrix} I - \frac{4}{3}\mathrm{r}A_{x,N} & 0 & 0 & 0 & 0 \\ -\frac{4}{3}I & I - \frac{4}{3}\mathrm{r}A_{x,N} & 0 & 0 & 0 \\ \frac{1}{3}I & -\frac{4}{3}I & I - \frac{4}{3}\mathrm{r}A_{x,N} & 0 & 0 \\ 0 & \ddots & \ddots & \ddots & 0 \\ 0 & 0 & \frac{1}{3}I & -\frac{4}{3}I & I - \frac{4}{3}\mathrm{r}A_{x,N} \end{pmatrix}, \tag{3.8}$$

and

$$b_{\mathrm{CN}} = \begin{pmatrix} (I + \mathrm{r}A_{x,N})u^0 + \Delta t v^{1-\frac{1}{2}} \\ \Delta t v^{2-\frac{1}{2}} \\ \vdots \\ \Delta t v^{M-\frac{1}{2}} \end{pmatrix}, \qquad b_{\mathrm{BDF2}} = \begin{pmatrix} \frac{4}{3}u^1 - \frac{1}{3}u^0 + \frac{2}{3}\Delta t v^2 \\ -\frac{1}{3}u^1 + \frac{2}{3}\Delta t v^3 \\ \frac{2}{3}\Delta t v^4 \\ \vdots \\ \frac{2}{3}\Delta t v^M \end{pmatrix}.$$

Note that both coefficient matrices $A_{\mathrm{CN}}, A_{\mathrm{BDF2}}$ are two-level Toeplitz according to Definition 1.3.2, and hence we can compute their symbol.

From Proposition 3.1.1 we recall that $A_{x,N} = d \cdot T_N(g_\alpha)$, where $g_\alpha(x)$ has a zero of order α at $x=0$. Then, by assuming $d_\pm = d$ and $N, M \to \infty$, we have

$$A_{\mathrm{S}} = \mathrm{T}^{(2)}_{NM}(h_{\mathrm{S}}),\ \mathrm{S} \in \{\mathrm{CN},\mathrm{BDF2}\},$$

with h_{S} defined as follows:

(i) if r is constant, then

$$\begin{aligned} &\bullet\ \ h_{\mathrm{CN}}(x,t) = 1 - \mathrm{e}^{it} - d \cdot \mathrm{r} g_\alpha(x)\big(1+\mathrm{e}^{it}\big),\\ &\bullet\ \ h_{\mathrm{BDF2}}(x,t) = 1 - \frac{4}{3}\mathrm{e}^{it} + \frac{1}{3}\mathrm{e}^{2it} - \frac{4}{3} d \cdot \mathrm{r} g_\alpha(x), \end{aligned} \tag{3.9}$$

and both functions have a unique zero at $(x,t) = (0,0)$ of order 1 and α in t and x, respectively;

(ii) if $\mathrm{r} \to 0$, then from (3.9) we have

$$\begin{aligned} &\bullet\ \ h_{\mathrm{CN}}(x,t) = 1 - \mathrm{e}^{it},\\ &\bullet\ \ h_{\mathrm{BDF2}}(x,t) = 1 - \frac{4}{3}\mathrm{e}^{it} + \frac{1}{3}\mathrm{e}^{2it}, \end{aligned}$$

and both functions have a zero of order 1 at $t=0,\ \forall x$;

(iii) if $\mathrm{r} \to \infty$, by grouping up r in (3.9), we have

$$\begin{aligned} &\bullet\ \ h_{\mathrm{CN}}(x,t) = -d(1+\mathrm{e}^{it})g_\alpha(x),\\ &\bullet\ \ h_{\mathrm{BDF2}}(x,t) = -\frac{4}{3} d \cdot g_\alpha(x), \end{aligned}$$

where h_{CN} has a zero of order 1 at $t = \pi,\ \forall x$ and a zero of order α at $x=0,\ \forall t$, while h_{BDF2} vanishes only at $x=0,\ \forall t$ with order α.

The presence of at least a line of zeros in the symbol is called *anisotropy*. The latter becomes stronger as the number of such lines increases. We expect then case (iii) for CN to be much harder to be numerically treated than all other cases.

Remark 3.3.1. *If we suppose* r *to be constant and let* $d \to \infty$ *or* $d \to 0$ *then the same results as in case (ii) and (iii), with* r *in place of* d*, hold. In practice, since* r, d *are fixed coefficients, the anisotropy arises when* $d \cdot \mathrm{r}$ *is very large or very small.*

Remark 3.3.2. *The study of the symbol* h_S *can easily be extended to the case where* $d_+ \neq d_-$ *using the results in Donatelli et al. [2016].*

3.4 Multigrid methods for all-at-once systems

This section is devoted to the design of multigrid strategies based on the spectral study performed in Section 3.3 for the linear systems in (3.6).

The convergence of the V-cycle relies in the hypothesis in equation (1.12), when the symbol of the coefficient matrix has a unique zero. In the case where f has a whole line of zeros along the axes, relation (1.12) does not hold anymore. In such a case, an efficient alternative to standard V-cycle is given by the semi-coarsening.

As done in Section 2.3.1, we define the projector according to condition (1.12) and the properties of the symbol h_{S} defined in Section 3.3. Let us first introduce the polynomials

$$p_1(x)=2+2\cos(x), \quad p_2^+(t)=1+\mathrm{e}^{\mathrm{i}t}, \quad p_2^-(t)=1-\mathrm{e}^{\mathrm{i}t}.$$

Note that:

- $p_1(x)$ has a zero of order 2 at $x=\pi$;
- $p_2^+(t)$ has a zero of order 1 at $t=\pi$;
- $p_2^-(t)$ has a zero of order 1 at $t=0$.

In the case of CN, according to the analysis in Section 3.3, we distinguish the following three cases:

1) If r is constant, then h_{CN} has a unique zero of minimum order 1 at $(x,t)=(0,0)$. The mirror points of $(x,t)=(0,0)$ are $(0,\pi),(\pi,0),(\pi,\pi)$. Since $p_1(x)$ and $p_2^+(t)$ vanish at $x=\pi$ and $t=\pi$, respectively, $p(x,t)=p_1(x)p_2^+(t)$ vanishes at the mirror points with a minimum order of 1 and hence satisfies relation (1.12).

2) In the anisotropic case where r → 0, h_{CN} is zero on the whole x-axis, then we opt for semi-coarsening in time. Precisely, according to the discussion related to the approximation property in Section 2.3.1, by considering variable x as a parameter, h_{CN} has a unique zero of order 1 at $t=0$. Then, we generate $P_{t,M}$ through $p_2^+(t)$ that has a zero of order 1 at the mirror point $t=\pi$. The projector $P_{x,N}$ is given by the identity matrix I_N.

3) In the anisotropic case, where r → ∞, h_{CN} is zero on both axes. The theory does not apply to this scenario. Nevertheless, as a first attempt to dominate (at least partially) this other kind of anisotropy, we use again standard semi-coarsening in both time and space. Precisely,

- When x is considered as a parameter, h_{CN} has a zero of order 1 at $t=\pi$. Therefore, we perform semi-coarsening in time replacing $P_{x,N}$ with I_N and generating $P_{t,M}$ through $p_2^-(t)$, whose first order zero at $t=0$ satisfies relation (1.12) for x fixed, i.e., considering only the one-dimensional problem in the variable t.
- When t is considered as a parameter, h_{CN} has a zero of order α at $x=0$. Therefore, the semi-coarsening in space is defined by replacing $P_{t,M}$ with the identity matrix I_M and generating $P_{x,N}$ through $p_1(x)$, which has a zero of order 2 at the mirror point $x=\pi$, again according to condition (1.12) for t fixed.

In the case of BDF2, items 1) and 2) are identical. Regarding item 3), i.e., when $\mathrm{r}\to\infty$, symbol h_{BDF2} vanishes only over the line $x=0$, $\forall t$ with order α. Hence we have a standard anisotropy, like in item 2), but this time along the t-axis. Therefore, we consider a semi-coarsening in space by setting $P_{t,M}=I_M$ and by generating $P_{x,N}$ through $p_1(x)$, whose position and order of the zero satisfies relation (1.12) for t fixed.

Concerning the smoother, we consider ω-weighted block Jacobi method (ω-BJ), where the diagonal blocks are of the form $I-\xi\mathrm{r}A_{x,N}$, with $\xi=1$ for CN and $\xi=\frac{4}{3}$ for BDF2. The reason for such a choice is that it is parallelizable and it allows to exploit the structure of the coefficient matrix. Moreover, ω-BJ converges for any $\omega\in(0,1]$ whenever the blocks are of size $N\times N$ (see Propositions 3.4.1 and 3.4.2), hence the study of its relaxation parameter is only related to the smoothing property along the time axis (see Section 3.5.1 for numerical discussion about this issue).

We stress that, choosing standard Jacobi would decrease the computational cost and would also exploit the structure of the coefficient matrix, but its use would ask for a tougher study of the relaxation parameter in order to ensure the convergence.

Proposition 3.4.1. *Let $A_{\mathrm{S}}\in\mathbb{R}^{NM\times NM}$ be the BTTB in equation* (3.6)*, where M is the number of diagonal blocks and N is the size of the blocks. Then,* 1*-BJ converges in M iterations for* $\mathrm{S}\in\{\text{CN},\text{BDF2}\}$ *and for any choice of the initial guess $x^{(0)}\in\mathbb{R}^{NM}$.*

Proof. We recall that 1-BJ is the iterative method in equation (1.4) with preconditioner $W_N=\tilde{D}$, where $\tilde{D}=\operatorname{diag}_{i=1,\dots,M}(\tilde{D}_i)$ is the block diagonal of A_{S}, and we rewrite it as a sequence of linear systems:

$$\tilde{D}x^{(k+1)}=\left(\tilde{D}-A_{\mathrm{S}}\right)x^{(k)}+b.$$

We now prove by induction the unconditional convergence of 1-BJ when applied to the CN-WSGD scheme. The same reasoning of course applies also to the BDF2-WSGD scheme.

Let us fix an initial guess $x^{(0)} \in \mathbb{R}^{NM}$, then split $x^{(0)}$ into M blocks

$$x^{(0)} = \begin{pmatrix} x_1^{(0)} \\ \vdots \\ x_M^{(0)} \end{pmatrix}, \quad x_i^{(0)} \in \mathbb{R}^N,\ i = 1, \dots, M$$

and proceed by induction.

As a base case, we prove that the first iteration of block Jacobi solves the first linear system in the CN-WSGD scheme. The first iteration yields $x^{(1)}$ as solution of the block linear system

$$\tilde{D} x^{(1)} = \left(\tilde{D} - A_{\mathrm{CN}}\right) x^{(0)} + b. \tag{3.10}$$

Due to the block structure of matrix $\tilde{D}$, the linear system (3.10) can be split into M independent linear systems. From the definition of matrix A_{CN}, in equation (3.7), and its diagonal $\tilde{D}$, it holds

$$\left(I_N - \mathrm{r}A_{x,N}\right) x_1^{(1)} = \left(I_N + \mathrm{r}A_{x,N}\right) u^0 + \Delta t \tilde{v}^{1-\frac{1}{2}}, \tag{3.11}$$

which coincides with the first step of the CN-WSGD scheme in equation (3.2) and gives the solution $x_1^{(1)}$ at time t^1.

Let us now suppose that at the k-th iteration, vector $x^{(k)}$ is such that $x_1^{(k)}, \dots, x_k^{(k)}$ are the solutions at the times $t^1, \dots, t^k$ respectively. Then the $(k+1)$-th block of equation (3.10) is

$$\left(I_N - \mathrm{r}A_{x,N}\right) x_{k+1}^{(k+1)} = \left(I_N + \mathrm{r}A_{x,N}\right) x_k^{(k)} + \Delta t \tilde{v}^{k-\frac{1}{2}}, \tag{3.12}$$

which coincides with the $(k+1)$-step of the CN-WSGD scheme in equation (3.2) and gives the solution $x_1^{(k+1)}$ at time t^{k+1} if $x_k^{(k)}$ is the numerical solution at the previous step, which is true by induction hypothesis and therefore $x_{k+1}^{(k+1)}$ is the numerical solution at time t^{k+1}.
It is easy to note that at the $(k+1)$-th iteration, it holds

$$x_1^{(k+1)} = x_1^{(k)} \ , \ \dots \ , \ x_k^{(k+1)} = x_k^{(k)},$$

up to a rounding error due to the machine precision, and therefore $x^{(k+1)}$ is such that $x_1^{(k+1)}, \dots, x_{k+1}^{(k+1)}$ are the solution at times $t^1, \dots, t^{k+1}$ and the thesis follows by induction. □

More in general, when $\omega \in (0,1)$ the following proposition holds.

Proposition 3.4.2. *Let $A_{\mathrm{S}} \in \mathbb{R}^{NM\times NM}$ be the BTTB in equation* (3.6)*, where M is the number of diagonal blocks and N is the size of the blocks. Then, ω-BJ converges $\forall \omega \in (0,1]$ for $\mathrm{S} \in \{\mathrm{CN}, \mathrm{BDF2}\}$ and for any choice of the initial guess $x^{(0)} \in \mathbb{R}^{NM}$.*

Proof. Let $W_N = \frac{1}{\omega}\tilde{D}$ be the preconditioner in equation (1.4) which defines the ω-BJ method, where $\tilde{D} = \mathrm{diag}_{i=1,\dots,M}(\tilde{D}_i)$ is the block diagonal of A_{S} in equation (3.6). Then, denoting by $*$ the subdiagonal blocks, the iteration matrix of ω-BJ is

$$J_{NM} = I_{NM} - \omega\tilde{D}^{-1}A_{\mathrm{S}} = \begin{pmatrix} (1-\omega)I_N & 0 & \cdots & 0 \\ * & \ddots & \ddots & \vdots \\ * & \ddots & \ddots & 0 \\ 0 & * & * & (1-\omega)I_N \end{pmatrix}.$$

Due to the block lower triangular structure of matrix J_{NM}, it holds that $\rho(J_{NM}) = 1-\omega < 1, \forall\omega \in (0,1]$ and therefore ω-BJ converges for any $\omega \in (0,1]$. □

Remark 3.4.1. *Note that in case of BDF2, when $d \to \infty$ or $\mathrm{r} \to \infty$, the coefficient matrix in equation* (3.6) *tends to a block diagonal Toeplitz matrix. This means that, when $d \cdot \mathrm{r}$ becomes large, using the multigrid is pointless since its smoother is already computing the solution accurately enough.*

3.5 Numerical results

In this section we investigate the performances of ω-BJ, Two-grid method (TGM) and V-cycle (V), mainly used as standalone solvers for solving the linear system in (3.6). Few numerical results concerning the use of both TGM and V as preconditioners (only one iteration) for GMRES are also given.

Both TGM and V-cycle will have one iteration of ω-BJ as post-smoother and no pre-smoothing iteration (the reason is given in Section 3.5.1). In our tests, the inversion of the blocks in block-Jacobi is performed through the Matlab function `backslash`. In V-cycle we halt the coarsening at the 5-th level, when the coefficient matrix has a minimum size of $N_5 \times M_5$ with $N_5 \geq \frac{N}{2^5}$ and $M_5 \geq \frac{N}{2^5}$, depending on the coarsening technique, and the solution on the coarsest level is performed through the `backslash` Matlab function, which is a direct solver.

As already clarified in Remark 3.1.1, the solution u^1 at the first time step is computed outside the coefficient matrix. One could of course include the computation of u^1 in the coefficient matrix as done in Gu et al. [2020]. In our case, due to the computationally expensive smoother we use, the difference between the two approaches is negligible. A comparison between the two approaches in terms of iterations can be found in Section 3.5.6.

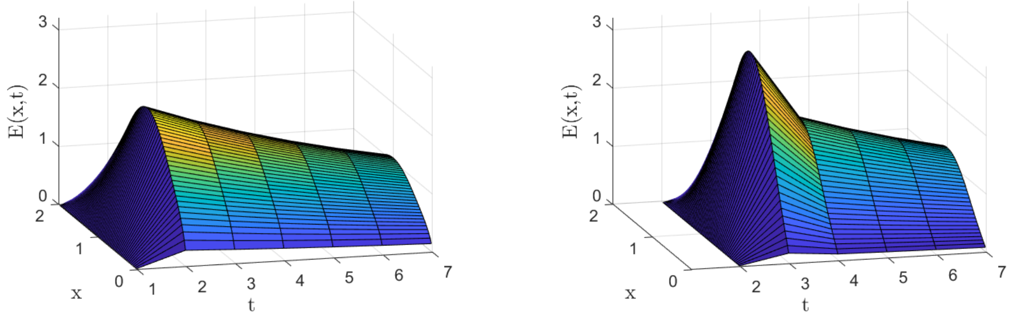
The section is organized as follows. In the first part we aim at explaining how we fix the fractional derivative order α, and the relaxation parameter ω in our numerical examples. Precisely, in Section 3.5.1 we test the performances of ω-BJ for two different values of ω and we show that it generates jumps along the time axis, independently of ω. In Section 3.5.2, we check how much TGM is sensitive to α, and numerical results show that its behavior is only slightly α-dependent.

Aside from α and ω, we also need to clarify how we choose between the two projector generators p_2^+ and p_2^- discussed in Section 3.4 when performing semi-coarsening in time for CN. This is the subject of Section 3.5.3. In Sections 3.5.4–3.5.5 we perform few tests with large N, M to numerically check the robustness of TGM and V as N, M increase in both constant and variable diffusion coefficients cases. Finally, in Section 3.5.6 we provide a two-dimensional example.

All our tests have been run on a server with Intel(R) Xeon(R) Silver 4114 at 2.20GHz with Matlab 2019b. For all methods we fix the tolerance to 10^{-7} and the initial guess as the null vector. We use the built-in `gmres` function, whose preconditioner is left-sided. For this reason we force GMRES to reach the required tolerance on the actual residual through a 'by hand' restart. A right preconditioned GMRES could be of course employed and it would basically give the same amount of iterations.

Notation. In the following, we denote with TGMp (resp. Vp), p$\in$\{x,t,xt\} the TGM (resp. V) that uses ω-BJ as post-smoother and performs semi-coarsening in space (p=x), time (p=t), or both space and time (p=xt). Precisely:

- 'x' denotes the space semi-coarsening, whose projector is generated by $p_1(x)$;
- 't$_\pm$' denotes the time semi-coarsening, whose projector is generated by $p_2(t)^\pm$;
- 'xt$_\pm$' denotes the full-coarsening, whose projector is generated by $p_1(x)p_2(t)^\pm$.



(a) CN scheme

(b) BDF2 scheme

Figure 3.1. Example 1 - Error $E(x,t)$ after one iteration of 1-BJ

The addition of '(G)' after the solver name stands for 'Galerkin approach'. In case nothing is specified, geometric approach is adopted. Finally, the presence of '($\mathscr{P}$)' in the name of the solver means that the considered MGM is set as GMRES preconditioner.

We point out that due to space limitations, in the key of each figure we omit the name of the method and specify only the projector. For instance, we write simply xt$_+$ in place of TGMxt$_+$. The name of the method will be clear from the caption of the figure.

All the results contained in the Sections 3.5.1–3.5.4 refer to the following example.

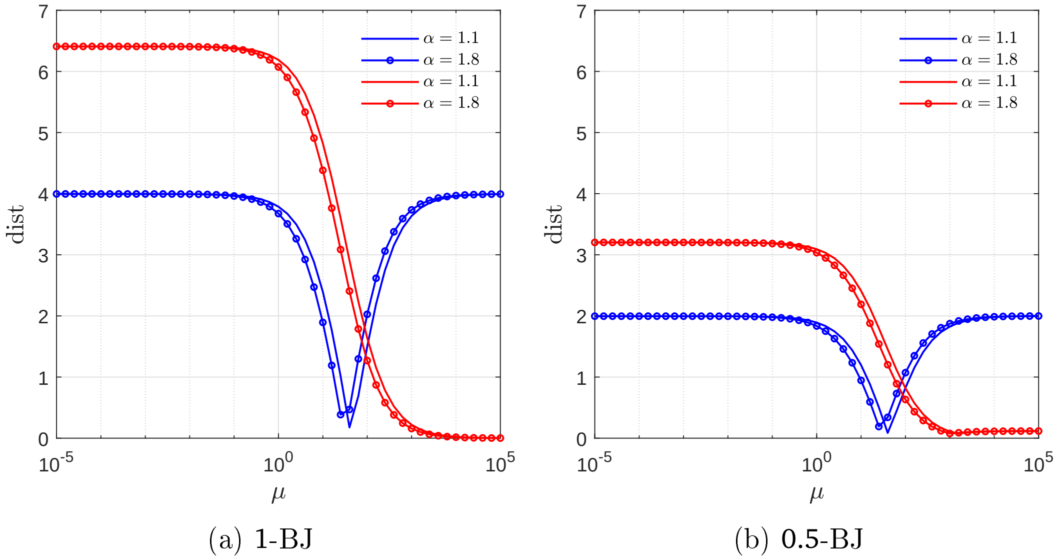
Example 1. *In this example we assume the diffusion coefficients to be constant and equal, that is $d_\pm = d$. The space and time domains in problem* (3.1) *are fixed as $\Omega = (0,2)$, and $[0,1]$ respectively, while the true solution and the solution at $t = 0$ are given by*

$$u_{ex}(x,t) = 4\mathrm{e}^{-t}x^2(2-x)^2, \qquad u_0(x) = 4x^2(2-x)^2.$$

The numerical approximation of v is computed starting from the discretized exact solution.

3.5.1 Behavior of ω-BJ smoother

Here we test the "smoothing properties" of ω-BJ. Let us consider $\alpha = 1.5$, $d = 1$, $N = 63$ and, to better point out the behavior of ω-BJ along the time axis, we fix



(a) 1-BJ

(b) 0.5-BJ

Figure 3.2. Example 1 - Vertical displacement dist(E), in equation (3.13), after 1 iteration of ω-BJ when using both CN (blue) and BDF2 (red)

$M=7\ll N$.

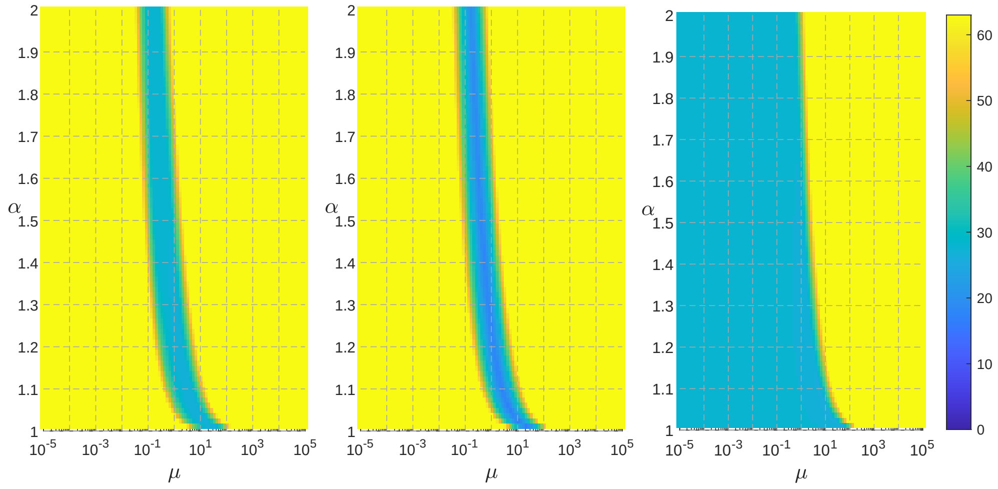
Fig. 3.1 shows the error, reshaped as a space-time surface, after one iteration of ω-BJ for both linear systems in equation (3.6). We note that, in case of CN, one iteration of 1-BJ generates a jump along the time axis from t^1 to t^2. In Figure 3.1b, where the 2-step method BDF2 is considered, such a jump involves also t^3 due to the longer stencil of BDF2 with respect to CN. On the other hand, both surfaces in Figure 3.1 do not show any jump along the x-axis.

We now analyze the jump in time varying the magnitude of $\mu := d\cdot$r, where r is the grid dependent scale parameter. We introduce the function

$$\text{dist}(E)=|E(1,t^2)-E(1,t^1)|+|E(1,t^3)-E(1,t^2)|, \qquad (3.13)$$

which measures the vertical displacement of the discrete error $E(x,t)$ in the first three time steps at the midpoint $x=1$. We note that dist(E)$=0$ if and only if E is constant in the first three time steps. In other words, as far as dist(E) ≈ 0, E is smooth, and this indicates that ω-BJ is a good smoother.

Figures 3.2a and 3.2b show how dist(E) behaves for both CN and BDF2, fixed



(a) TGMxt (b) TGMx (c) TGMt

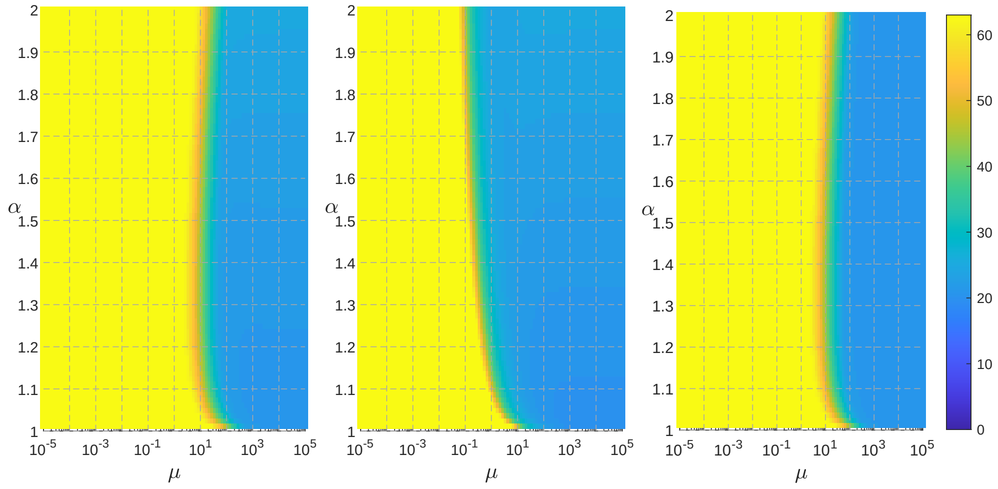
Figure 3.3. Example 1 - Iterations to tolerance varying α and μ, fixed $\omega = 0.5$, and using CN scheme

$N = M = 63$, $\omega = 1, 0.5$, $\alpha = 1.1, 1.8$, and varying $\mu \in [10^{-5}, 10^{5}]$. As we can see, both discretizations are characterized by a region where the jump is negligible. In detail, the jump generated in CN is negligible only when $\mu \approx 10$. In BDF2, instead, the jump becomes negligible as μ increases. Moreover, for both CN and BDF2, the jump slightly moves while varying α, and it halves its magnitude when switching from $\omega = 1$ to $\omega = 0.5$.

In summary, in all the considered cases ω-BJ generates jumps along the time axis which means that the projection along such axis could be inaccurate. In order to face this drawback, in the following we only consider ω-BJ as post-smoother avoiding pre-smoothing iteration at the first iteration. This choice is supported by the idea that applying the CGC before the smoother could reduce the jump, preventing then the projection of a non-smooth error.

3.5.2 Behavior of TGM varying α

In Section 3.5.1, we observed that dist(E), in equation (3.13), slightly varies with α. This could lead to a difference in the behavior of the multigrid depending on α, when solving the two linear systems in (3.6). Here we perform few tests



(a) TGMxt (b) TGMx (c) TGMt

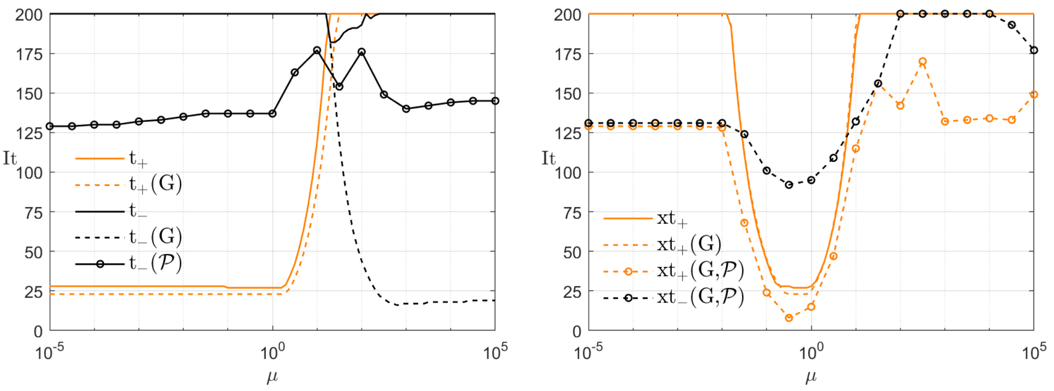
Figure 3.4. Example 1 - Iterations to tolerance varying α and μ, fixed $\omega = 0.5$, and using BDF2 scheme

which show that the behavior of the proposed TGM is almost independent of α and hence that justify the choice of a fixed value for α in the reminder of the numerical tests.

In Fig. 3.3 and Fig. 3.4, we check the number of iterations of TGMx, TGMt, and TGMxt, with fixed $N = M = 2^6 - 1$, and varying $\mu \in [10^{-5}, 10^5]$, and $\alpha \in (1,2]$. Concerning the choice of ω in the ω-BJ smoother, several tests (not reported here because of space limitations) show that, in the case of CN, the choice of $\omega = 1$ causes bad convergence results for both TGMxt and TGMt. On the other hand, $\omega = 0.5$ provides a good convergence, according also to the analysis in Section 3.5.1, for both CN and BDF2. Therefore, in the rest of this section we fix $\omega = 0.5$. We stress that such discussion on the relaxation parameter is not intended as a substitute of a rigorous study, and that a theoretical approach to the subject will be investigated in a future work.

Fig. 3.3, where we use CN scheme, shows that by increasing α the optimal region of convergence (blue) shifts to the left for any of the considered algorithms. Regarding BDF2, instead, Fig. 3.4 shows that the blue region shifts to the left as α increases only in the case of TGMx. In the other two cases, their number of iterations stays almost independent of α.

Summarizing, the width of the blue regions does not seem to significantly change while varying α. Therefore, in the following we restrict our analysis to the case



(a) TGM semi-coarsening in time

(b) TGM full-coarsening

Figure 3.5. Example 1 - Time projectors performances for the CN scheme

where $\alpha = 1.5$.

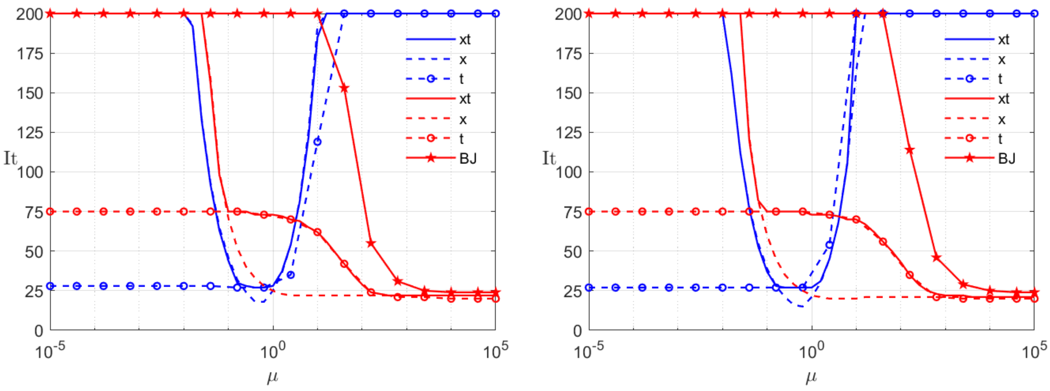
3.5.3 Time projection performances for the CN scheme

In Section 3.3 we have shown that the symbol h_{CN} has a zero in t that moves from 0 to π depending on how r or d behave asymptotically, and then on the magnitude of $\mu = d \cdot \mathrm{r}$. As discussed in Section 3.4, this means that the projector in time must change as well from t_+ to t_- according to μ. Here, we show that the latter does not work satisfactorily in practical applications when μ is large.

Let us fix $\alpha = 1.5$ and $N = M = 2^7–1$. Fig. 3.5a shows the iterations to tolerance of $\mathrm{TGMt}_\pm$, $\mathrm{TGMt}_\pm(\mathrm{G})$, $\mathrm{TGMt}_-(\mathscr{P})$ while varying the magnitude of $\mu \in [10^{-5}, 10^5]$. We note, in line with the discussion in Section 3.4, that the Galerkin approach allows $\mathrm{TGMt}_+(\mathrm{G})$ to converge in a low amount of iterations when $\mu \in [10^{-5}, 1]$, that is for small values of μ. When considering the less robust geometric approach, TGMt_+ still yields good convergence results in the same range of μ, even if the iteration number slightly increases.

In the case where $\mu \gg 1$, the only working method is $\mathrm{TMGt}_-(\mathrm{G})$. Unfortunately, the Galerkin approach is not of practical use since it is too computationally expensive. Regarding the geometric approach, TGMt_- results unpractical also when used as GMRES preconditioner (refer to $\mathrm{TGMt}_-(\mathscr{P})$ in Fig. 3.5a).

The reason why geometric and Galerkin methods behave differently is due to the large difference between the matrices at the coarser level obtained with the two approaches. Indeed, the convergence condition given in (1.12) requires the Galerkin approach, which leads to a coarser matrix having a symbol that van-



(a) $N\!=\!M\!=\!2^7\!-\!1$

(b) $N\!=\!M\!=\!2^8\!-\!1$

Figure 3.6. Example 1 - TGM performances either using CN scheme (blue) or BDF2 scheme (red), and fixed $\alpha = 1.5$

ishes at the origin (see Arico et al. [2004] for details). Differently, the geometric approach, which consists in discretizing the same problem over a coarser grid, builds a coarser matrix that vanishes again at $t\!=\!\pi$ and that shows then opposite spectral behavior with respect to Galerkin.

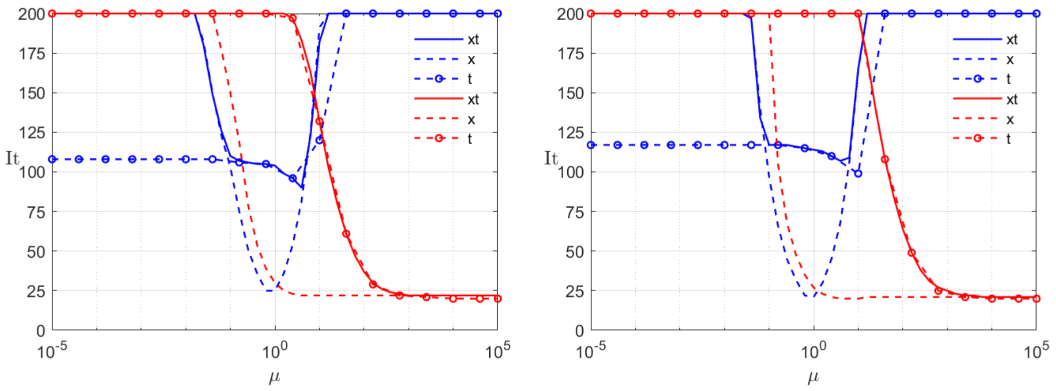
In Figure 3.5b, TGM with full-coarsening is shown not to work in the anisotropic cases $\mu\!<\!10^{-1}$ and $\mu\!>\!10$, independently of the time projectors and the approach for computing the matrix at the coarser level.

In conclusion, in the following we only consider the time projector given by t_+ and we denote it simply with ‘t’, since it is the only projector that allows TGM with both semi-coarsening in time and full-coarsening to yield good convergence results for the geometric approach.

3.5.4 Comparison between CN and BDF2: TGM and V-cycle performances

Now we discuss how the performances of TGM and V-cycle with both semi- and full-coarsening vary depending on the adopted discretization scheme, i.e., CN or BDF2.

Let us discuss first the behavior of TGM. In Fig. 3.6, we compare the iterations to tolerance of TGMxt, TGMx, TGMt for both CN and BDF2 varying the magnitude of $\mu \in [10^{-5}, 10^{5}]$ and fixed $N = M \in \{2^7 - 1, 2^8 - 1\}$. In case of BDF2, the



(a) $N\!=\!M\!=\!2^7\!-\!1$

(b) $N\!=\!M\!=\!2^8\!-\!1$

Figure 3.7. Example 1 - V-cycle performances either using CN scheme (blue) or BDF2 scheme (red), and fixed $\alpha = 1.5$

iteration number of 1-BJ used as a standalone solver are displayed as well.

We note that, in the case of CN, the iterations to tolerance of TGMt look stable for $\mu < 1$ as N, M increase. The same holds for TGMxt and TGMx when $\mu \approx 1$. Nothing seems to work when $\mu > 10$ again independently of N, M, which is what we are expecting due to the strong anisotropy of this specific case discussed in Section 3.4.

In the case of BDF2, again according to our theoretical analysis, TGMx and TGMt yield good convergence results when $\mu > 1$ and $\mu < 1$, respectively, and both are stable as N, M increase. In line with what we observed in Section 3.5.1, the high number of iterations of TGMt, even if constant, could be due to the bad smoothing effects along the time axis of 0.5-BJ. Concerning TGMxt, it yields the same iterations to tolerance as TGMt when $\mu > 10^{-1}$. Note that for this example both TGMxt and TGMt work where they are not supposed to, i.e., in the anisotropic case $\mu \gg 1$. This is due to the smoother that, according to Remark 3.4.1, is already a robust enough solver.

Due to the high computational cost of TGM, in Fig. 3.7 we switch from TGM to V-cycle and we check its behavior depending on the chosen time discretization scheme.

In the case of CN, conversely to the results obtained for TGM, the only projector which seems to allow V-cycle to converge in a reasonable amount of iterations, is the semi-coarsening in space. However, it works in a really small region, i.e., when $\mu \approx 1$, which is close to the region where one iteration of 0.5-BJ yields a smooth solution (go back to Fig. 3.2).

Regarding BDF2, Vx converges in almost the same amount of iterations as TGMx when $\mu > 1$. In particular, for $\mu \in \left[1, 10^3\right]$ the block diagonal part of A_{BDF2} is not dominant and hence 0.5-BJ has a slow convergence, but when it is used as smoother in Vx we obtain a robust and fast convergent method. Moreover, we note that the region where 0.5-BJ used as standalone solver is already enough robust becomes smaller as the mesh-size increases. This is not the case for Vx, which is then faster than 0.5-BJ in a wider range of μ as N, M become large. Concerning Vt and Vxt, their plots are basically superposed independently of μ, and they perform well only for large values of μ again because of the ω-BJ smoother.

In the case of a semi-coarsening in space only, since time interpolation is not involved, larger values of ω could be used. Tests which are not reported here show a reduction in the iteration number of the multigrid with ω-BJ, with $\omega \approx 1$, when applied to both CN and BDF2 for almost the same values of μ where it performs well with $\omega = 0.5$.

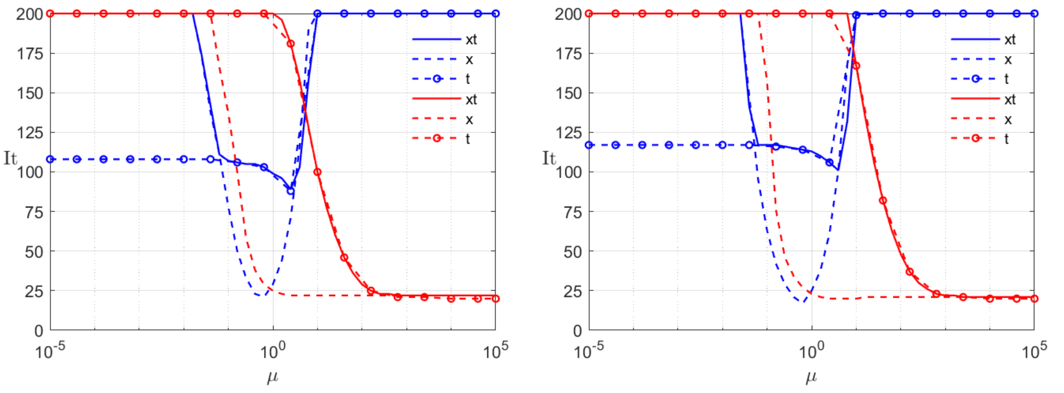
We note that, for both CN and BDF2, the iterations to tolerance of all the tested V-cycles stay almost stable as N, M increase. Moreover our results are in line with the results reported in Figure 2, Section 4.4 of Horton and Vandewalle [1995], where multigrid with colored pointwise Gauss-Seidel as smoother is used to solve a space-time linear system obtained from the discretization of a standard time-dependent diffusion equation.

3.5.5 A variable diffusion coefficients example

We now consider the example taken from Lei and Sun [2013] in which the diffusion coefficients are not constant.

Example 2. *We assume the space and time domains in problem* (3.1) *as* $\Omega = (0, 2)$*, and* $[0, 1]$ *respectively, and define the diffusion coefficients, the true solution and the solution at* $t = 0$ *as follows*

$$\begin{aligned} d_-(x,t) &= d \cdot \Gamma(3-\alpha)x^\alpha, & d_+(x,t) &= d \cdot \Gamma(3-\alpha)(2-x)^\alpha, \\ u_{ex}(x,t) &= 4\mathrm{e}^{-t}x^2(2-x)^2, & u_0(x) &= 4x^2(2-x)^2, \end{aligned}$$



(a) $N\!=\!M\!=\!2^7\!-\!1$

(b) $N\!=\!M\!=\!2^8\!-\!1$

Figure 3.8. Example 2 - V-cycle performances either using CN scheme (blue) or BDF2 scheme (red), and fixed $\alpha = 1.5$

where $d > 0$. When $d\!=\!1$, the forcing term is given by

$$v(x,t) = -32e^{-t}(x^2+\frac{1}{8}((2-x)^2)(8+x^2) - \frac{3(x^3+(2-x)^3)}{3-\alpha} + \frac{3(x^4+(2-x)^4))}{(4-\alpha)(3-\alpha)},$$

while in the remaining cases, the numerical approximation of v is computed starting from the discretized exact solution.

In Figure 3.8, like in Section 3.5.4, we test the behavior of Vx, Vt, Vxt for two fine grids with $N\!=\!M\!=\!2^7\!-\!1$ and $N\!=\!M\!=\!2^8\!-\!1$.

We note that, as in the case of constant and equal diffusion coefficients, the results are not significantly sensitive to N, M. Moreover, like in Example 1, Vx is the only V-cycle, between the three tested, that yields good convergence results for both CN and BDF2. Finally, also in this variable coefficients example, the optimal convergence region, given by the magnitude of $\mu\!=\!d\cdot$r, is much bigger in the case of BDF2 than of CN.

We note that, independently of the constant or variable diffusion coefficients character, none of the tested methods is robust enough to deal with the case where $\mu < 1$. Further tests, not reported here, show that this holds unchanged even when using the V-cycle as preconditioner for the GMRES. On the other hand, we stress that, since d is fixed, the choice of an opportune grid could lead to $\mu > 1$, choosing Δt and Δx such that $d\Delta t > 2\Delta x^\alpha$ and making V-cycle a suitable solver again.

3.5.6 Two-dimensional case

We end the numerical section by providing numerical results in the two-dimensional case. We consider the following extension of the one-dimensional FDE in equation (3.1):

$$\begin{cases} \dfrac{\partial u(x,y,t)}{\partial t} = d_+ \dfrac{\partial^\alpha u(x,y,t)}{\partial_+ x^\alpha} + d_- \dfrac{\partial^\alpha u(x,y,t)}{\partial_- x^\alpha} \\ \qquad\qquad e_+ \dfrac{\partial^\beta u(x,y,t)}{\partial_+ y^\beta} + e_- \dfrac{\partial^\beta u(x,y,t)}{\partial_- y^\beta} + v(x,y,t), \\ \qquad\qquad\qquad\qquad\qquad\qquad (x,y,t) \in \Omega \times [0,T], \\ u(x,y,t) = 0, \qquad\qquad\qquad\qquad (x,y,t) \in \left(\mathbb{R}^2 \setminus \Omega\right) \times [0,T], \\ u(x,y,0) = u_0(x,y), \qquad\qquad\quad (x,y) \in \overline{\Omega}, \end{cases} \tag{3.14}$$

where $\Omega = (a_1, b_1) \times (a_2, b_2)$ is the space domain and $d_\pm, e_\pm > 0$ are the diffusion coefficients.

The discretization follows from the one-dimensional case and yields the same coefficient matrices as in equation (3.6), but with the extension through Kronecker product in two dimensions of each block. In the case where $d_+ = d_- = e_+ = e_- = d$, $\alpha = \beta$ and both spatial steps $\Delta x, \Delta y$ are equal, it holds that the grid dependent scale factor which multiplies the matrix representing the discretization in space is the same as in the one-dimensional case, i.e., $\mu = d\frac{\Delta t}{2\Delta x^\alpha}$.

Example 3. *For our test we extend* Example *1 to the two-dimensional case by assuming $d_\pm = e_\pm = d$, $\Omega = (0,2) \times (0,2)$, and taking as final time step $T = 1$. The true solution and the solution at the initial time $t = 0$ are, respectively, given by*

$$u_{ex}(x,y,t) = 4\mathrm{e}^{-t} x^2 (2-x)^2 y^2 (2-y)^2, \qquad u_0(x,y) = 4x^2(2-x)^2 y^2 (2-y)^2.$$

As done in Example *1, the numerical approximation of v is computed starting from the discretized exact solution.*

Since the time-coarsening does not seems to be effective in the one-dimensional case, here we only consider the coarsening in both spatial dimensions and we use 0.95-BJ as post-smoother (higher weights seemed more suitable for this case).

Due to hardware limitations we cannot choose too dense grids, therefore we fix $N_x = N_y = M = 2^6 - 1$, where N_x and N_y are the amount of points over the grids in the first and second spatial dimensions and M are the points over the time grid. Moreover, when considering the BDF2 scheme, as in the one-dimensional case we use CN to compute the solution at the first time step. As done in Gu et al. [2020], we consider the case where the computation of the solution u^1 at

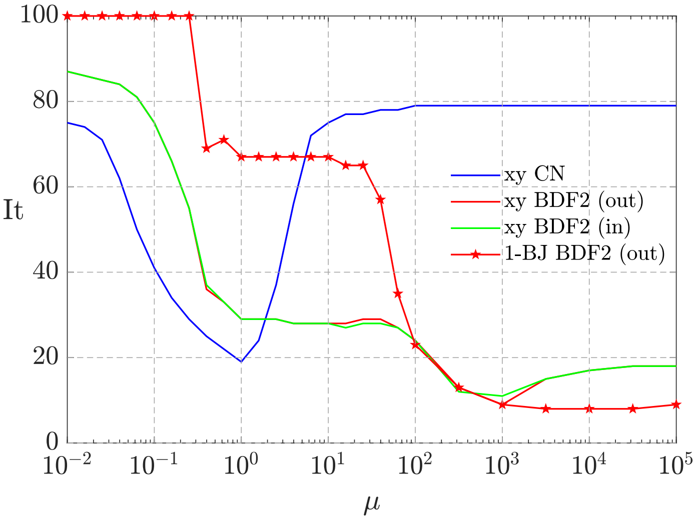


Figure 3.9. Example 3 - Performances of 1-BJ using BDF2 with outer CN (starred line) and of V-cycle using CN scheme (blue), BDF2 scheme with outer CN (red), BDF2 scheme with inner CN (green) and fixed $\alpha=1.5$.

the first the step is included in the coefficient matrix (inner CN) and we compare it with the previously considered case (outer CN), where u^1 is computed outside the coefficient matrix.

In Figure 3.9 we show the iterations to tolerance of the multigrid used as standalone solver (denoted by 'xy'), for solving equation (3.14) discretized with CN, BDF2 with inner CN and BDF2 with outer CN. In the case of BDF2 with outer CN, we compare the results with 1-BJ.
We note that, even in the two-dimensional case, multigrid applied to CN is efficient when $\mu \approx 1$ and applied to BDF2 when $\mu \geq 1$. Moreover, we observe that the plots of BDF2 with inner and outer CN overlap almost everywhere, therefore the addition of CN inside the coefficient matrix does not seem to compromise the convergence of multigrid.
When $\mu \geq 10^2$, as in the one-dimensional case, 1-BJ is an efficient solver, since the coefficient matrix becomes block diagonally dominant.

3.6 Conclusions

In this chapter we focused on an all-at-once rephrasing of a time-dependent one-dimensional space-FDE with constant diffusion coefficients discretized with

WSGD in space and CN or BDF2 in time. The unconditional stability of the BDF2-WSGD scheme has been proven, and the two-level Toeplitz structure of the resulting linear systems has been leveraged to design multigrid strategies that use block Jacobi as smoother and whose projectors definition is driven by the symbol.

We have shown that that V-cycle with semi-coarsening in x-variable is the only multigrid, among all the tested ones, that yields good convergence results for both BDF2 and CN schemes. Moreover, it performs satisfactorily under less restrictive assumptions on the magnitude of $\mu = d \cdot \mathrm{r}$ in the case of BDF2 than in the case of CN, and this let us to conclude that BDF2 is a much better alternative to CN for parallel-in-time integration with multigrid, when μ is large.

Chapter 4

Comparison between FV and FVE

In this chapter, we focus on a two-dimensional conservative steady-state Riesz FDE. As is typical for problems in conservative form, we adopt a FV discretization approach. Precisely, we use both classical FVs and FVEs. While FVEs have already been applied in the context of FDEs, classical FVs had only been applied in first order discretizations. We will see that FVs are more suitable than FVEs when the fractional derivative orders are close to 2, since they allow a lower approximation error and, at the same time, a fast convergence rate of our solver.

This chapter is organized as follows. In Section 4.1, we report the problem and provide a finite difference discretization of the fractional derivative and some preliminary results. In Section 4.2, we introduce the second-order FVE discretization, while in Section 4.3 we build a second-order FV discretization providing numerical evidences on the choice of some parameters. Moreover, by exploiting the Toeplitz-like structure of the resulting coefficient matrices, we perform a qualitative study of their spectrum and conditioning through their symbol. Using the obtained information, in Section 4.4 we design efficient ad-hoc banded preconditioners and MGMs. Finally, in Section 4.5 we compare both FVs and FVEs discretizations in terms of approximation error and iterations to tolerance of the multigrid solver. Our conclusions are drawn in Section 4.6.

4.1 Preliminaries

In this section we define our problem, then we discuss the discretization of the fractional derivative operator through the WSGD formula with a fractional shift over uniform meshes. Finally, we provide the generating function of the resulting

Toeplitz matrices and give some preliminary theoretical results needed to discuss the convergence of MGMs.

4.1.1 Two-dimensional space-FDE

We focus on the following two-dimensional boundary-value steady-state conservative Riesz FDE of order $2-\alpha, 2-\beta$, with $0 < \alpha, \beta < 1$ and with absorbing boundary conditions:

$$\begin{cases} -\dfrac{\partial}{\partial x}\left(K_x(x,y)\dfrac{\partial^{1-\alpha}u(x,y)}{\partial\,|x|^{1-\alpha}}\right)-\dfrac{\partial}{\partial y}\left(K_y(x,y)\dfrac{\partial^{1-\beta}u(x,y)}{\partial\,|y|^{1-\beta}}\right)=v(x,y), \\ \qquad\qquad\qquad\qquad (x,y)\in\Omega, \\ u(x,y)=0, \qquad\qquad (x,y)\in\left(\mathbb{R}^2\setminus\Omega\right), \end{cases} \tag{4.1}$$

where $\frac{\partial^{1-\alpha}u(x,y)}{\partial|x|^{1-\alpha}}, \frac{\partial^{1-\beta}u(x,y)}{\partial|y|^{1-\beta}}$ are the Riesz fractional derivative operators with respect to x- and y-variables, respectively, $\Omega = (a_1, b_1) \times (a_2, b_2)$ is the spatial domain, $K_x(x,y), K_y(x,y)$ are the nonnegative bounded diffusion coefficients, $v(x,y)$ is the forcing term.

The Riesz fractional operator in the x-variable is defined as

$$\frac{\partial^{1-\alpha}u(x,y)}{\partial\,|x|^{1-\alpha}}=\eta(\alpha)\left[\frac{\partial^{1-\alpha}u(x,y)}{\partial_+^R x^{1-\alpha}}+\frac{\partial^{1-\alpha}u(x,y)}{\partial_-^R x^{1-\alpha}}\right], \quad \eta(\alpha)=-\frac{1}{2\cos\left(\frac{(1-\alpha)\pi}{2}\right)},$$

where the left and right derivatives are given in the RL form (see (1.24)). Similarly one can define the Riesz fractional operator in the y-variable.

4.1.2 Fractional derivative discretization and preliminary spectral results

As shown in Section 1.6.4, under proper hypothesis an alternative definition of the left and right fractional derivatives is based on the GL formulas (1.32), whose shifted form SGD is defined in equation (1.33). Similar definitions can be given for the fractional derivatives in the y-variable.

Consider $u : \mathbb{R}^2 \to \mathbb{R}$ with $\mathrm{supp}(u) \in [0,1]^2$ and the equispaced grid

$$\begin{aligned} x_i =& ih_x,\ i=1,\dots,N_x, \qquad h_x=\frac{1}{N_x+1}, \\ y_j =& jh_y,\ j=1,\dots,N_y, \qquad h_y=\frac{1}{N_y+1}, \end{aligned}$$

with $N_x, N_y \in \mathbb{N}$. Then the left fractional derivative operator in equation (1.32), which coincides with (1.33) with $p = 0$, can be written as

$$\frac{\partial^{1-\alpha} u(x_i, y_j)}{\partial_+^G x^{1-\alpha}} = {}_L G_{h_x,0}^{1-\alpha} u(x_i, y_j) = \frac{1}{h_x^{1-\alpha}} \left(G_{+,0} u^{(j)} \right)_i ,$$

where

$$G_{+,0} = \begin{pmatrix} t_0^{(1-\alpha)} & 0 & \cdots & 0 \\ t_1^{(1-\alpha)} & t_0^{(1-\alpha)} & \ddots & \vdots \\ \vdots & \ddots & \ddots & 0 \\ t_{N_x-1}^{(1-\alpha)} & \cdots & t_1^{(1-\alpha)} & t_0^{(1-\alpha)} \end{pmatrix} \in \mathbb{R}^{N_x \times N_x}, \; u^{(j)} = \begin{pmatrix} u(x_1, y_j) \\ u(x_2, y_j) \\ \vdots \\ u(x_{N_x}, y_j) \end{pmatrix} \in \mathbb{R}^{N_x}. \tag{4.2}$$

Matrix $G_{+,0}$ is a Toeplitz matrix and represents the left fractional derivative operator. The choice of $p \neq 0$ yields the same structured matrix, but with the diagonals shifted to the right of p positions, if $p > 0$, and the diagonals shifted to the left of $|p|$ positions, if $p < 0$. We denote such an operator by $G_{+,p}$.

Note that in the case of $p > 0$ we have to compute p new coefficients $t_{N_x}^{(1-\alpha)}, \ldots, t_{N_x+p-1}^{(1-\alpha)}$ to fill the bottom left diagonals. Furthermore, when $q = 0$, the right fractional derivative operator in equation (1.33) is $G_{-,0} = -G_{+,0}^{\mathrm{T}}$. If $q \neq 0$ then we denote such an operator by $G_{-,q}$ and it holds $G_{-,q} = -G_{+,q}^{\mathrm{T}}$.

Remark 4.1.1. *Let $N \in \mathbb{N}$ and consider an arbitrary equispaced grid $\{x_i\}_{i=1}^N$, where h is the step length. Due to the FV discretization, we are required to evaluate the fractional derivative operators between two grid points, e.g., in $(x_i - \frac{h}{2}, y)$, which leads to*

$$\begin{aligned} {}_L G_{h,p}^{1-\alpha} u(x_i - \tfrac{h}{2}, y) &= \frac{1}{h^{1-\alpha}} \sum_{k=0}^{\infty} t_k^{(1-\alpha)} u\big(x_i - (k - p + \tfrac{1}{2})h, y\big), \\ {}_R G_{h,q}^{1-\alpha} u(x_i - \tfrac{h}{2}, y) &= -\frac{1}{h^{1-\alpha}} \sum_{k=0}^{\infty} t_k^{(1-\alpha)} u\big(x_i + (k - q - \tfrac{1}{2})h, y\big). \end{aligned} \tag{4.3}$$

This motivates the need of a non-integer shift.

In Hejazi et al. [2013, 2014] a non-integer shift was used to define a first order FV approximation. A second-order FV scheme is still missing in literature.

In order to fill this gap we recall that, as shown in Theorem 1.6.3, an opportune average of two SGD with two different integer shifting parameters leads to a second-order approximation of the RL formulas in equation (1.24). Moreover, since Remark 4.1.1 highlights the need of a non-integer shift, from Remark 1.6.5 we recall that Theorem 1.6.3 extends also to the case where $p_1, p_2, q_1, q_2 \in \mathbb{R}$ with $p_1 \neq p_2$ and $q_1 \neq q_2$.

As a consequence, for a generic step length $h > 0$ and for $p_1 \neq p_2 \in \mathbb{Z} + \frac{1}{2}$, $q_1 \neq q_2 \in \mathbb{Z} + \frac{1}{2}$, we can write

$$\begin{aligned}
\frac{\partial^{1-\alpha} u(x_i - \frac{h}{2}, y_j)}{\partial_+^R x^{1-\alpha}} =& \frac{1}{h^{1-\alpha}} \Bigg(w_{\boldsymbol{p}}^\alpha \sum_{k=0}^{\infty} t_k^{(1-\alpha)} u\big(x_i - (k - p_1 + \tfrac{1}{2})h, y_j\big) + \\
&+ (1 - w_{\boldsymbol{p}}^\alpha) \sum_{k=0}^{\infty} t_k^{(1-\alpha)} u\big(x_i - (k - p_2 + \tfrac{1}{2})h, y_j\big) \Bigg) + O(h^2); \\
\frac{\partial^{1-\alpha} u(x_i - \frac{h}{2}, y_j)}{\partial_-^R x^{1-\alpha}} =& \frac{1}{h^{1-\alpha}} \Bigg(w_{\boldsymbol{q}}^\alpha \sum_{k=0}^{\infty} t_k^{(1-\alpha)} u\big(x_i + (k - q_1 - \tfrac{1}{2})h, y_j\big) + \\
&+ (1 - w_{\boldsymbol{q}}^\alpha) \sum_{k=0}^{\infty} t_k^{(1-\alpha)} u\big(x_i + (k - q_2 - \tfrac{1}{2})h, y_j\big) \Bigg) + O(h^2),
\end{aligned} \tag{4.4}$$

where $w_{\boldsymbol{p}}^\alpha = \frac{1-\alpha-2p_2}{2(p_1-p_2)}$ and $w_{\boldsymbol{q}}^\alpha = \frac{1-\alpha-2q_2}{2(q_1-q_2)}$ with $\boldsymbol{p} = (p_1, p_2)$ and $\boldsymbol{q} = (q_1, q_2)$. We refer the reader to Theorem 4.1.1 for the matrix form of equation (4.4).

We end this section writing explicitly the symbol of the (properly scaled) Toeplitz matrices representing the discretized operators in equation (4.3) of Remark 4.1.1. Such symbol will be useful in the computation of the symbol for the FV discretization of (4.1) performed in Section 4.3. Having this in mind, we start with a couple of intermediate results.

Proposition 4.1.1. *Let $N \in \mathbb{N}$, then it holds that $G_{+,0}, G_{-,0}$ defined in equation* (4.2) *are such that*

$$G_{+,0} = T_N(g^\alpha(x)), \qquad G_{-,0} = T_N(-\overline{g^\alpha(x)}),$$

with $g^\alpha(x) = (1 - \mathrm{e}^{\mathrm{i}x})^{1-\alpha}$.

Proof. According to the definition of symbol and by means of the generalized Newton binomial, it holds

$$g^\alpha(x) = \sum_{k\in\mathbb{Z}} t_k^{(1-\alpha)} \mathrm{e}^{\mathrm{i}kx} = \sum_{k\in\mathbb{Z}} (-1)^k \binom{1-\alpha}{k} \mathrm{e}^{\mathrm{i}kx} = (1 - \mathrm{e}^{\mathrm{i}x})^{1-\alpha},$$

which completes the proof. □

The following result is needed for later analysis.

Lemma 4.1.1. *For all $x \in [0,\pi]$ it holds that*

$$g^{\alpha}(x)+\overline{g^{\alpha}}(x)=2^{2-\alpha}\sin^{1-\alpha}\left(\tfrac{x}{2}\right)\sin\left(\tfrac{x+\alpha(\pi-x)}{2}\right) \tag{4.5}$$

and

$$g^{\alpha}(x)\mathrm{e}^{\mathrm{i}x}+\overline{g^{\alpha}}(x)\mathrm{e}^{-\mathrm{i}x}=2^{2-\alpha}\sin^{1-\alpha}(\tfrac{x}{2})\left[\sin(x)\cos\left(\tfrac{x+\alpha(\pi-x)}{2}\right)+\cos(x)\sin\left(\tfrac{x+\alpha(\pi-x)}{2}\right)\right]. \tag{4.6}$$

Proof. By means of the Euler formulas

$$\begin{aligned} \mathrm{e}^{\mathrm{i}x}-\mathrm{e}^{\mathrm{i}y} =& 2\mathrm{i}\frac{\mathrm{e}^{\mathrm{i}\frac{x-y}{2}}-\mathrm{e}^{-\mathrm{i}\frac{x-y}{2}}}{2i}\mathrm{e}^{\mathrm{i}\frac{x+y}{2}}=2\mathrm{i}\sin\left(\frac{x-y}{2}\right)\mathrm{e}^{\mathrm{i}\frac{x+y}{2}},\\ \mathrm{e}^{\mathrm{i}x}+\mathrm{e}^{\mathrm{i}y} =& 2\frac{\mathrm{e}^{\mathrm{i}\frac{x-y}{2}}+\mathrm{e}^{-\mathrm{i}\frac{x-y}{2}}}{2}\mathrm{e}^{\mathrm{i}\frac{x+y}{2}}=2\cos\left(\frac{x-y}{2}\right)\mathrm{e}^{\mathrm{i}\frac{x+y}{2}}, \end{aligned} \tag{4.7}$$

we have

$$\begin{aligned} (1-\mathrm{e}^{\mathrm{i}x})^{1-\alpha}+(1-\mathrm{e}^{-\mathrm{i}x})^{1-\alpha} &= (2\mathrm{i}\sin(-\tfrac{x}{2})\mathrm{e}^{\mathrm{i}\frac{x}{2}})^{1-\alpha}+(2\mathrm{i}\sin(\tfrac{x}{2})\mathrm{e}^{-\mathrm{i}\frac{x}{2}})^{1-\alpha}\\ &= (2\sin(\tfrac{x}{2}))^{1-\alpha}\left[(-\mathrm{i}\mathrm{e}^{\mathrm{i}\frac{x}{2}})^{1-\alpha}+(\mathrm{i}\mathrm{e}^{-\mathrm{i}\frac{x}{2}})^{1-\alpha}\right]\\ &= (2\sin(\tfrac{x}{2}))^{1-\alpha}\left[\mathrm{e}^{\mathrm{i}(\frac{x}{2}-\frac{\pi}{2})(1-\alpha)}+\mathrm{e}^{-\mathrm{i}(\frac{x}{2}-\frac{\pi}{2})(1-\alpha)}\right]\\ &= 2^{2-\alpha}\sin^{1-\alpha}(\tfrac{x}{2})\cos\left(\tfrac{\pi}{2}-\tfrac{\alpha\pi+(1-\alpha)x}{2}\right)\\ &= 2^{2-\alpha}\sin^{1-\alpha}(\tfrac{x}{2})\sin\left(\tfrac{\alpha\pi+(1-\alpha)x}{2}\right). \end{aligned}$$

and the proof of equation (4.5) follows by rearranging the argument of the sine. Now, again by means of the Euler formulas (4.7), we have

$$\begin{aligned} g^{\alpha}(x)\mathrm{e}^{\mathrm{i}x}+\overline{g^{\alpha}}(x)\mathrm{e}^{-\mathrm{i}x} &= \left(\mathrm{e}^{-\mathrm{i}\frac{x}{2}}-\mathrm{e}^{\mathrm{i}\frac{x}{2}}\right)^{1-\alpha}\left(\mathrm{e}^{\mathrm{i}(\frac{x}{2}(1-\alpha)+x)}+(-1)^{1-\alpha}\mathrm{e}^{-\mathrm{i}(\frac{x}{2}(1-\alpha)+x)}\right)\\ &= \left(2\sin(\tfrac{x}{2})\right)^{1-\alpha}\left(\mathrm{e}^{\mathrm{i}((1-\alpha)(\frac{x}{2}-\frac{\pi}{2})+x)}+\mathrm{e}^{-\mathrm{i}((1-\alpha)(\frac{x}{2}-\frac{\pi}{2})+x)}\right)\\ &= \left(2\sin(\tfrac{x}{2})\right)^{1-\alpha}2\cos\left((1-\alpha)(\tfrac{x}{2}-\tfrac{\pi}{2})+x\right)\\ &= 2^{2-\alpha}\sin^{1-\alpha}(\tfrac{x}{2})\ \cos\left(\tfrac{\pi}{2}-\tfrac{3x+\alpha(\pi-x)}{2}\right)\\ &= 2^{2-\alpha}\sin^{1-\alpha}(\tfrac{x}{2})\ \sin\left(x+\tfrac{x+\alpha(\pi-x)}{2}\right)\\ &= 2^{2-\alpha}\sin^{1-\alpha}(\tfrac{x}{2})\Big[\sin(x)\cos\left(\tfrac{x+\alpha(\pi-x)}{2}\right)+\\ &\qquad\qquad+\cos(x)\sin\left(\tfrac{x+\alpha(\pi-x)}{2}\right)\Big], \end{aligned}$$

which proves equation (4.6) and concludes the proof. □

As a Corollary to Proposition 4.1.1, we obtain the symbol of the Toeplitz matrices representing the second-order discretization of the fractional derivative operators given in equation (1.34).

Corollary 4.1.1. *Let $N \in \mathbb{N}$, $h = \frac{1}{N+1}$ and $\boldsymbol{p}, \boldsymbol{q} \in \mathbb{Z}^2$, then equation* (1.34) *can be written as follows*

$$\begin{aligned}\frac{\partial^{1-\alpha}u(x_i,y_j)}{\partial_+^R x^{1-\alpha}} &= \frac{1}{h^{1-\alpha}}\left(T_N\left(g^\alpha_{+,\boldsymbol{p}}(x)\right)u^{(j)}\right)_i + O(h^2),\\ \frac{\partial^{1-\alpha}u(x_i,y_j)}{\partial_-^R x^{1-\alpha}} &= \frac{1}{h^{1-\alpha}}\left(T_N\left(g^\alpha_{-,\boldsymbol{q}}(x)\right)u^{(j)}\right)_i + O(h^2),\end{aligned}$$

where $u^{(j)}$ is defined in equation (4.2) *and*

$$\begin{aligned}g^\alpha_{+,\boldsymbol{p}}(x) &= g^\alpha(x)\left(w^\alpha_{\boldsymbol{p}}\mathrm{e}^{-\mathrm{i}p_1x} + (1-w^\alpha_{\boldsymbol{p}})\mathrm{e}^{-\mathrm{i}p_2x}\right),\\ g^\alpha_{-,\boldsymbol{q}}(x) &= -\overline{g^\alpha}(x)\left(w^\alpha_{\boldsymbol{q}}\mathrm{e}^{\mathrm{i}q_1x} + (1-w^\alpha_{\boldsymbol{q}})\mathrm{e}^{\mathrm{i}q_2x}\right).\end{aligned}$$

Proof. According to the definition of symbol, shifting the diagonals by p positions to the right or left consists in multiplying the symbol by $\mathrm{e}^{-\mathrm{i}px}$ or $\mathrm{e}^{\mathrm{i}px}$, respectively. Therefore, the proof follows by the discussion at the begin of Section 4.1.2. □

We are now ready to provide the symbol of the Toeplitz matrices corresponding to the fractional left and right operators evaluated at the midpoint $x_{i-\frac{1}{2}}$ given in equation (4.3).

Theorem 4.1.1. *Let $N \in \mathbb{N}$, $h = \frac{1}{N+1}$ and $\boldsymbol{p}, \boldsymbol{q} \in \mathbb{Z}^2 + \frac{1}{2}$, then equation* (4.4) *can be written as follows*

$$\begin{aligned}\frac{\partial^{1-\alpha}u(x_i-\frac{h}{2},y_j)}{\partial_+^R x^{1-\alpha}} &= \frac{1}{h^{1-\alpha}}\left(H_{+,\boldsymbol{p}}u^{(j)}\right)_i + O(h^2),\\ \frac{\partial^{1-\alpha}u(x_i-\frac{h}{2},y_j)}{\partial_-^R x^{1-\alpha}} &= \frac{1}{h^{1-\alpha}}\left(H_{-,\boldsymbol{q}}u^{(j)}\right)_i + O(h^2),\end{aligned} \tag{4.8}$$

where $u^{(j)}$ is defined in equation (4.2) *and*

$$H_{+,\boldsymbol{p}} = T_N\left(g^\alpha_{+,\boldsymbol{p}}(x)\mathrm{e}^{\mathrm{i}\frac{x}{2}}\right), \quad H_{-,\boldsymbol{q}} = T_N\left(g^\alpha_{-,\boldsymbol{q}}(x)\mathrm{e}^{\mathrm{i}\frac{x}{2}}\right).$$

Proof. We only provide the proof in the case of the left fractional derivative since for the other case the proof follows the same steps. From equation (4.4), which represents the i-th row of the matrix-vector product in equation (4.8), we have that the resulting matrix is a Toeplitz generated by

$$\begin{aligned}
w_p^\alpha \sum_{k=0}^{\infty} t_k^{(1-\alpha)} \mathrm{e}^{\mathrm{i}(k-p_1+\frac{1}{2})x} + (1-w_p^\alpha) \sum_{k=0}^{\infty} t_k^{(1-\alpha)} \mathrm{e}^{\mathrm{i}(k-p_2+\frac{1}{2})x} = \\
&= w_p^\alpha g^\alpha(x) \mathrm{e}^{\mathrm{i}(-p_1+\frac{1}{2})x} + (1-w_p^\alpha) g^\alpha(x) \mathrm{e}^{\mathrm{i}(-p_2+\frac{1}{2})x} \\
&= g^\alpha(x) \left(w_p^\alpha \mathrm{e}^{-\mathrm{i}p_1 x} + (1-w_p^\alpha) \mathrm{e}^{-\mathrm{i}p_2 x} \right) \mathrm{e}^{\mathrm{i}\frac{x}{2}} \\
&= g_{+,p}^\alpha(x) \mathrm{e}^{\mathrm{i}\frac{x}{2}},
\end{aligned}$$

which completes the proof. □

4.2 Finite volume-type discretizations

Here we review the idea of a finite volume discretization applied to problem (4.1). In particular, in Section 4.2.1 we recall the FVE approach. The classical FV approach will be treated in Section 4.3.

In the most general case, applying a finite volume approach to (4.1) consists in covering Ω with a mesh $\cup_{i=1}^{n} Q_i$, where $\mu(Q_i \cap Q_j) = 0$, $i \neq j$, with μ the Lebesgue measure, and integrating over Q_i. In our specific case, given $N_x, N_y \in \mathbb{N}$ and defined the following uniform mesh of $\Omega = [a_1, b_1] \times [a_2, b_2]$

$$\begin{aligned}
h_x &= \tfrac{b_1-a_1}{N_x+1}, \quad x_i = a_1 + ih_x,\ i = 1, \ldots, N_x, \\
h_y &= \tfrac{b_2-a_2}{N_y+1}, \quad y_j = a_2 + jh_x,\ j = 1, \ldots, N_y.
\end{aligned}$$

We decompose the domain Ω with the rectangles $Q_{ij} = [x_{i-\frac{1}{2}}, x_{i+\frac{1}{2}}] \times [y_{j-\frac{1}{2}}, y_{j+\frac{1}{2}}]$, and we integrate equation (4.1) over Q_{ij}. As a consequence, we end up with $S_1 + S_2 = S_3$, where

$$\begin{aligned}
S_1 &= -\int_{y_{j-\frac{1}{2}}}^{y_{j+\frac{1}{2}}} \int_{x_{i-\frac{1}{2}}}^{x_{i+\frac{1}{2}}} \frac{\partial}{\partial x} \left(K_x(x,y) \frac{\partial^{1-\alpha} u(x,y)}{\partial |x|^{1-\alpha}} \right) \mathrm{d}x \mathrm{d}y, \\
S_2 &= -\int_{y_{j-\frac{1}{2}}}^{y_{j+\frac{1}{2}}} \int_{x_{i-\frac{1}{2}}}^{x_{i+\frac{1}{2}}} \frac{\partial}{\partial y} \left(K_y(x,y) \frac{\partial^{1-\beta} u(x,y)}{\partial |y|^{1-\beta}} \right) \mathrm{d}x \mathrm{d}y, \\
S_3 &= \int_{y_{j-\frac{1}{2}}}^{y_{j+\frac{1}{2}}} \int_{x_{i-\frac{1}{2}}}^{x_{i+\frac{1}{2}}} v(x,y) \mathrm{d}x \mathrm{d}y.
\end{aligned}$$

We approximate S_3 by means of the tensor product of Simpson rules, which is an order 3 scheme, so that the approximation of the right hand side will not influence the solution and the comparison of the FV and FVE discretization approaches. Therefore,

$$S_3 = \frac{h_x h_y}{36}\Big(v(x_{i-\frac{1}{2}}, y_{j-\frac{1}{2}}) + 4v(x_{i-\frac{1}{2}}, y_j) + v(x_{i-\frac{1}{2}}, y_{j+\frac{1}{2}}) + 4v(x_i, y_{j-\frac{1}{2}}) + 16v(x_i, y_j) + \\ + 4v(x_i, y_{j+\frac{1}{2}}) + v(x_{i+\frac{1}{2}}, y_{j-\frac{1}{2}}) + 4v(x_{i+\frac{1}{2}}, y_j) + v(x_{i+\frac{1}{2}}, y_{j+\frac{1}{2}})\Big) + O(h_x^3 + h_y^3). \tag{4.9}$$

Moreover, the discretization of S_1 can be simplified as follows

$$S_1 = K_x(x_{i-\frac{1}{2}}, y_j)\int_{y_{j-\frac{1}{2}}}^{y_{j+\frac{1}{2}}} \frac{\partial^{1-\alpha} u(x_{i-\frac{1}{2}}, y)}{\partial\, |x|^{1-\alpha}} \mathrm{d}y - K_x(x_{i+\frac{1}{2}}, y_j)\int_{y_{j-\frac{1}{2}}}^{y_{j+\frac{1}{2}}} \frac{\partial^{1-\alpha} u(x_{i+\frac{1}{2}}, y)}{\partial\, |x|^{1-\alpha}} \mathrm{d}y + O(h_y^2). \tag{4.10}$$

A similar reasoning of course applies to S_2. At this point, we can proceed in two different ways: either approximating again S_1 as

$$S_1 = h_y K_x(x_{i-\frac{1}{2}}, y_j)\frac{\partial^{1-\alpha} u(x_{i-\frac{1}{2}}, y_j)}{\partial\, |x|^{1-\alpha}} - h_y K_x(x_{i+\frac{1}{2}}, y_j)\frac{\partial^{1-\alpha} u(x_{i+\frac{1}{2}}, y_j)}{\partial\, |x|^{1-\alpha}} + O(h_x^2 + h_y^2) \tag{4.11}$$

which brings to the FV approach or restricting the admissible solutions to a certain finite element space, which gives rise to the FVE approach.

4.2.1 FVE discretization matrices and its spectral study

FVE have already been applied to FDE problems in Liu et al. [2014]; Feng et al. [2015]; Jia and Wang [2016]. Here we briefly recall the FVE discretization matrices obtained in case of piecewise linear elements. Let us consider the basis functions $\{\phi_k^x(x)\otimes\phi_l^y(y)\}_{k,l=1}^{N_x,N_y}$, where

$$\phi_k^x(x) = \begin{cases} \frac{x - x_{k-1}}{h_x}, & x \in (x_{k-1}, x_k), \\ \frac{x_{k+1} - x}{h_x}, & x \in (x_k, x_{k+1}), \\ 0, & \text{elsewhere}, \end{cases}$$

for $k = 1, \ldots, N_x$, and define similarly $\phi_l^y(y)$ with y_l in place of x_k and h_y in place of h_x. Then, we replace $u(x, y)$ in equation (4.10) with its piecewise linear approximation $\tilde{u}(x, y) = \sum_{k,l=1}^{N_x,N_y} u_{kl}\phi_k^x(x)\phi_l^y(y)$ leading to

$$S_1 = \sum_{k,l=1}^{N_x,N_y} u_{kl}\Big(\frac{\partial^{1-\alpha}\phi_k^x(x_{i-\frac{1}{2}})}{\partial\, |x|^{1-\alpha}} K_x(x_{i-\frac{1}{2}}, y_j)\int_{y_{j-\frac{1}{2}}}^{y_{j+\frac{1}{2}}} \phi_l^y(y)\mathrm{d}y +$$

$$-\frac{\partial^{1-\alpha}\phi_k^x(x_{i+\frac{1}{2}})}{\partial\,|x|^{1-\alpha}}K_x(x_{i+\frac{1}{2}},y_j)\int_{y_{j-\frac{1}{2}}}^{y_{j+\frac{1}{2}}}\phi_l^y(y)\mathrm{d}y\Bigg).$$

Since the support of $\phi_l^y(y)$ is compact, then

$$\int_{y_{k-\frac{1}{2}}}^{y_{k+\frac{1}{2}}}\phi_l^y(y)\mathrm{d}y\neq 0,\quad \text{only if}\quad l=k-1,k,k+1,$$

which evaluates to $\frac{h_y}{8},\frac{6h_y}{8},\frac{h_y}{8}$, respectively, and leads to the tridiagonal mass matrix

$$B_{N_y}=\ \text{tridiag}\left(\frac{1}{8},\frac{6}{8},\frac{1}{8}\right)\in\mathbb{R}^{N_y\times N_y}.$$

Let $u^l=[u_{1l},u_{2l},\ldots,u_{N_xl}]^{\mathrm{T}}$, then, by performing the same computations done in Wang and Du [2013], it follows that

$$\sum_{k=1}^{N_x}\frac{\partial^{1-\alpha}\tilde{u}(x_{i-\frac{1}{2}},y)}{\partial\,|x|^{1-\alpha}}=\phi_l^y(y)\sum_{k=1}^{N_x}u_{kl}\frac{\partial^{1-\alpha}\phi_k^x(x_{i-\frac{1}{2}})}{\partial\,|x|^{1-\alpha}}=\phi_l^y(y)\frac{\eta(\alpha)}{\Gamma(\alpha+1)h_x^{1-\alpha}}(G_{\alpha,N_x}u^l)_i,$$

where $G_{\alpha,N_x}=T_{N_x}(\hat{g}^\alpha(x))$, with $\hat{g}^\alpha(x)=\sum_{k\in\mathbb{Z}}\hat{t}_k^{(\alpha)}\mathrm{e}^{\mathrm{i}kx}$ and

$$\hat{t}_k^{(\alpha)}=\begin{cases}\left(\frac{3}{2}\right)^\alpha-3\left(\frac{1}{2}\right)^\alpha, & k=1,\\ \left(k+\frac{1}{2}\right)^\alpha+\left(k-\frac{3}{2}\right)^\alpha-2\left(k-\frac{1}{2}\right)^\alpha & k\geq 2,\\ -\hat{t}_{-k+1}^{(\alpha)}, & k\leq 0.\end{cases}$$

Therefore, the FVE discretization of the FDE problem in (4.1) yields the linear system

$$A_{\text{FVE}}u=b \tag{4.12}$$

where the right-hand side b follows from equation (4.9), the solution is $u=\{u_{kl}\}_{k,l=1}^{N_x,N_y}$ and the coefficient matrix is the following $N\times N$, with $N=N_xN_y$, matrix

$$\begin{aligned}A_{\text{FVE}}=&r\left(K_{x,L}(B_{N_y}\otimes G_{\alpha,N_x})+K_{x,R}(B_{N_y}\otimes G_{\alpha,N_x}^{\mathrm{T}})\right)+\\&+s\left(K_{y,L}(G_{\beta,N_y}\otimes B_{N_x})+K_{y,R}(G_{\beta,N_y}^{\mathrm{T}}\otimes B_{N_x})\right),\end{aligned}\tag{4.13}$$

with

$$K_{x,L}=\text{diag}\left(\{K_x(x_{i-\frac{1}{2}},y_j)\}_{i,j=1}^{N_x,N_y}\right),\quad K_{x,R}=\text{diag}\left(\{K_x(x_{i+\frac{1}{2}},y_j)\}_{i,j=1}^{N_x,N_y}\right),$$
$$K_{y,L}=\text{diag}\left(\{K_y(x_i,y_{j-\frac{1}{2}})\}_{i,j=1}^{N_x,N_y}\right),\quad K_{y,R}=\text{diag}\left(\{K_y(x_i,y_{j+\frac{1}{2}})\}_{i,j=1}^{N_x,N_y}\right).$$

The grid dependent scale factors are $r=\frac{\eta(\alpha)h_y}{\Gamma(\alpha+1)h_x^{1-\alpha}}$, $s=\frac{\eta(\beta)h_x}{\Gamma(\beta+1)h_y^{1-\beta}}$.

As already observed in Donatelli et al. [2018], in the one-dimensional case with constant diffusion coefficients, the symbol of the coefficient matrix is $\left(\hat{g}^\alpha(x)+\overline{\hat{g}^\alpha}(x)\right)$, which is a nonnegative function with a unique zero of order lower than 2 at $x=0$. In the case of a two-dimensional equation with constant diffusion coefficients the symbol of A_{FVE} is

$$\hat{g}^\alpha_{2\text{D}}(x,y)=rK_x m(y)\left(\hat{g}^\alpha(x)+\overline{\hat{g}^\alpha}(x)\right)+sK_y m(x)\left(\hat{g}^\alpha(y)+\overline{\hat{g}^\alpha}(y)\right),$$

where $m(z)=\frac{6+2\cos(z)}{8}$ is the symbol of the mass matrix B_{N_z}, with $z=x,y$.

Remark 4.2.1. *Note that $\hat{g}^\alpha_{2D}(x,y)$ has a unique zero of order lower than 2 at $(x,y)=(0,0)$. This is because the symbol of the mass matrix is a strictly positive function.*

4.3 FV discretization matrices and related spectral study

First order accurate FV discretizations for FDE problems appeared in Hejazi et al. [2013, 2014]; Zhang et al. [2005]. Here we build a second-order scheme by imposing some reasonable constraints on the shift parameters involved in the approximation of the fractional derivatives. In addition, on the same line of what has been done in Donatelli et al. [2018], we provide a spectral study of the resulting coefficient matrices which allows to build ad-hoc solvers for the associated linear systems in Section 4.4.

Let us go back to (4.10). The choice of approximating S_1 as in (4.11) yields a $N\times N$ linear system, whose structure of the coefficient matrix A_{FV}, except for the mass matrices that are replaced by identities, is the same as A_{FVE} in equation (4.13). In detail we have to solve

$$A_{\text{FV}}u=b, \tag{4.14}$$

where b follows from equation (4.9), $u=\{u_{ij}\}_{i,j=1}^{N_x,N_y}$, with $u_{ij}\approx u(x_i,y_j)$, and $A_{\text{FV}}:=A_x+A_y$ with

$$\begin{aligned} A_x =& r\left(K_{x,L}(I_{N_y}\otimes M_{\alpha,L})-K_{x,R}(I_{N_y}\otimes M_{\alpha,R})\right),\\ A_y =& s\left(K_{y,L}(M_{\beta,L}\otimes I_{N_x})-K_{y,R}(M_{\beta,R}\otimes I_{N_x})\right),\end{aligned} \tag{4.15}$$

where the new scaling factors are $r=\frac{\eta(\alpha)h_y}{h_x^{1-\alpha}}$, $s=\frac{\eta(\beta)h_x}{h_y^{1-\beta}}$ and the Toeplitz matrices $M_{\alpha,L},M_{\alpha,R},M_{\beta,L},M_{\beta,R}$ represent the discretized fractional operators by means of the shifted weighted GL formulas in equation (4.8). Specifically, the matrices $M_{\alpha,L}, M_{\alpha,R}$, are such that

$$\begin{aligned}
\frac{\partial^{1-\alpha} u(x_{i-\frac{1}{2}}, y_j)}{\partial |x|^{1-\alpha}} &= r\left(\left(I_{N_y} \otimes M_{\alpha,L}\right) u\right)_{i+N_x(j-1)} + O(h_x^2),\\
\frac{\partial^{1-\alpha} u(x_{i+\frac{1}{2}}, y_j)}{\partial |x|^{1-\alpha}} &= r\left(\left(I_{N_y} \otimes M_{\alpha,R}\right) u\right)_{i+N_x(j-1)} + O(h_x^2),
\end{aligned}$$

i.e., $M_{\alpha,L}$ coincides with $T_{N_x}\big(f_\alpha^{(\boldsymbol{p},\boldsymbol{q})}(x)\big)$, where

$$f_\alpha^{(\boldsymbol{p},\boldsymbol{q})}(x) = g^\alpha_{+,\boldsymbol{p}}(x) \mathrm{e}^{\mathrm{i}\frac{x}{2}} + g^\alpha_{-,\boldsymbol{q}}(x) \mathrm{e}^{\mathrm{i}\frac{x}{2}},$$

while $M_{\alpha,R}$ is obtained by $M_{\alpha,L}$ shifting its diagonals one position forward, that is, $M_{\alpha,R} = T_{N_x}\left(f_\alpha^{(\boldsymbol{p},\boldsymbol{q})}(x)\mathrm{e}^{-\mathrm{i}x}\right)$. The matrices $M_{\beta,L}, M_{\beta,R}$ are similarly defined.

4.3.1 Properties of the symbol of A_{FV}

In the following we study the properties of A_{FV} and we explain what is a good choice for the shifting parameters $\boldsymbol{p} = (p_1, p_2)$, $\boldsymbol{q} = (q_1, q_2)$. In this view, we note that in case of constant diffusion coefficients $K_x(x,y) = K_x > 0$, from equation (4.15) we have $A_x = rK_x\, I_{N_y} \otimes (M_{\alpha,L} - M_{\alpha,R})$, where

$$M_{\alpha,L} - M_{\alpha,R} = T_{N_x}\left(F_\alpha^{(\boldsymbol{p},\boldsymbol{q})}(x)\right), \tag{4.16}$$

with $F_\alpha^{(\boldsymbol{p},\boldsymbol{q})}(x) = f_\alpha^{(\boldsymbol{p},\boldsymbol{q})}(x) - f_\alpha^{(\boldsymbol{p},\boldsymbol{q})}(x)\mathrm{e}^{-\mathrm{i}x}$.
Having in mind the design of an ad-hoc MGM for the linear systems associated to A_{FV}, we ask that $F_\alpha^{(\boldsymbol{p},\boldsymbol{q})}(x)$ is a nonnegative function with a unique zero (see Section 4.4.1 for more details). Let us first require that $F_\alpha^{(\boldsymbol{p},\boldsymbol{q})}(x)$ is a real-valued function. Since there are many free parameters we fix $\boldsymbol{q} = \boldsymbol{p}$. Under this constraint function $F_\alpha^{(\boldsymbol{p},\boldsymbol{p})}(x)$ reads as

$$F_\alpha^{(\boldsymbol{p},\boldsymbol{p})}(x) = g^\alpha_{+,\boldsymbol{p}} \mathrm{e}^{\mathrm{i}\frac{x}{2}} - \overline{g^\alpha_{+,\boldsymbol{p}}} \mathrm{e}^{\mathrm{i}\frac{x}{2}} - \left(g^\alpha_{+,\boldsymbol{p}} \mathrm{e}^{\mathrm{i}\frac{x}{2}} - \overline{g^\alpha_{+,\boldsymbol{p}}} \mathrm{e}^{\mathrm{i}\frac{x}{2}}\right) \mathrm{e}^{-\mathrm{i}x} = \left(g^\alpha_{+,\boldsymbol{p}} - \overline{g^\alpha_{+,\boldsymbol{p}}}\right)\left(1 - \mathrm{e}^{-\mathrm{i}x}\right)\mathrm{e}^{\mathrm{i}\frac{x}{2}}.$$

and

$$F_\alpha^{(\boldsymbol{p},\boldsymbol{p})}(x) - \overline{F_\alpha^{(\boldsymbol{p},\boldsymbol{p})}}(x) = \left(g^\alpha_{+,\boldsymbol{p}} - \overline{g^\alpha_{+,\boldsymbol{p}}}\right)\left(\mathrm{e}^{\mathrm{i}\frac{x}{2}} - \mathrm{e}^{-\mathrm{i}\frac{x}{2}}\right) - \left(\overline{g^\alpha_{+,\boldsymbol{p}}} - g^\alpha_{+,\boldsymbol{p}}\right)\left(\mathrm{e}^{-\mathrm{i}\frac{x}{2}} - \mathrm{e}^{\mathrm{i}\frac{x}{2}}\right),$$

which is zero $\forall x \in (-\pi, \pi]$ and $\forall p_1, p_2$, and this implies that $F_\alpha^{(\boldsymbol{p},\boldsymbol{p})}(x)$ is a real-valued function independently of $\boldsymbol{p}$.

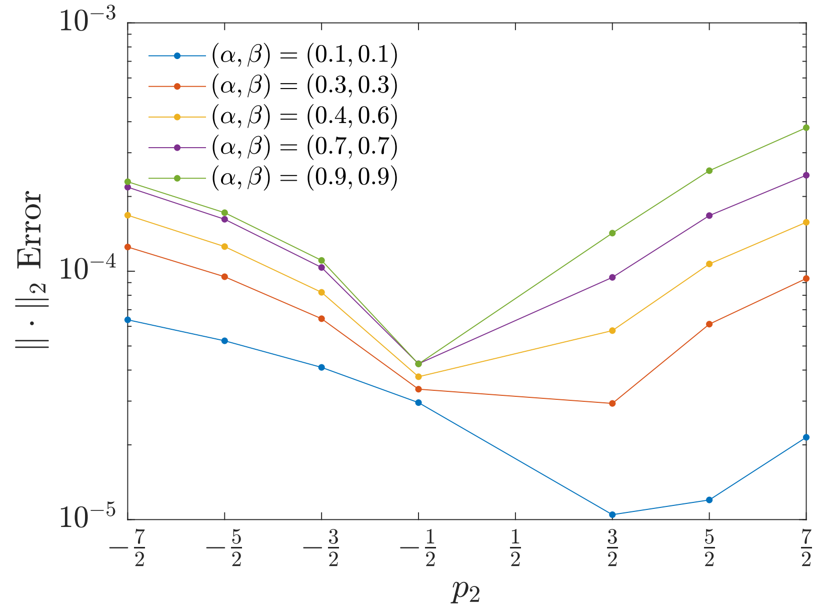


Figure 4.1. Relative error varying (α,β) and p_2, with fixed $p_1=\frac{1}{2}$.

In order to make a reasonable choice of $\boldsymbol{p}$, we numerically check how the relative 2-norm approximation error varies with $\boldsymbol{p}$ while solving (4.14) in the case where $K_x=K_y=1$ and solution $u(x,y)$ with related forcing term $f(x,y)$ are the ones reported in Section 4.5. Many tests show that choosing p_1,p_2 too far from 0 leads to an increase in the error. Hence, we fix $p_1=\frac{1}{2},-\frac{1}{2}$. Figure 4.1 shows the relative 2-norm error for $p_1=\frac{1}{2}$, $p_2\in[-\frac{7}{2},\frac{7}{2}]$ and varying α,β. We note that the optimal $\boldsymbol{p}$ seems to be $\boldsymbol{p}=(\frac{1}{2},-\frac{1}{2})$, since it gives the lowest error for a wider range of fractional derivative orders if compared to other combinations.

We do not show the results for $p_1=-\frac{1}{2}$ since every tested combination with $p_2\in[-\frac{7}{2},\frac{7}{2}]$ leads to high ill-conditioned linear systems with a large increase in approximation error except for $\boldsymbol{p}=(-\frac{1}{2},\frac{1}{2})$, which yields the same results as the shift $\boldsymbol{p}=(\frac{1}{2},-\frac{1}{2})$ due to the symmetry of formulas in equation (4.8) with respect to the shifting parameters p_1,p_2 and q_1,q_2, respectively. Therefore, from now onwards, we will fix $\boldsymbol{p}=(\frac{1}{2},-\frac{1}{2})$.
The numerical results in Section 4.5 show that such a choice of $\boldsymbol{p}$ and $\boldsymbol{q}$ leads to a second-order accurate numerical scheme for equation (4.1) (see Figure 4.2).

Remark 4.3.1. *Interestingly enough, when $\alpha,\beta\approx 0$, $\boldsymbol{p}=(\frac{1}{2},\frac{3}{2})$ has almost one third of the approximation error than $\boldsymbol{p}=(\frac{1}{2},-\frac{1}{2})$. Moreover, some preliminary* *numerical checks, which are not reported here, seem to indicate that the resulting coefficient matrix is positive definite and therefore it could be another interesting combination to investigate.*

We now check whether for $\boldsymbol{p} = (\frac{1}{2}, -\frac{1}{2})$, the symbol $F_\alpha^{(\boldsymbol{p},\boldsymbol{p})}(x)$ is nonnegative with a unique zero. For the sake of readability, we omit the superscript $(\boldsymbol{p},\boldsymbol{p})$ in the symbol and rewrite it as

$$
\begin{aligned}
F_\alpha(x) &= \mathrm{e}^{\mathrm{i}\frac{x}{2}}(1-\mathrm{e}^{-\mathrm{i}x})\Big(g^\alpha(x)(w_p^\alpha \mathrm{e}^{-\mathrm{i}\frac{x}{2}} + (1-w_p^\alpha)\mathrm{e}^{\mathrm{i}\frac{x}{2}}) - \overline{g^\alpha}(x)(w_p^\alpha \mathrm{e}^{\mathrm{i}\frac{x}{2}} + (1-w_p^\alpha)\mathrm{e}^{-\mathrm{i}\frac{x}{2}})\Big)\\
&= g^\alpha(x)\Big(w_p^\alpha(1-\mathrm{e}^{-\mathrm{i}x}) + (1-w_p^\alpha)(\mathrm{e}^{\mathrm{i}x}-1)\Big) - \overline{g^\alpha}(x)\Big(w_p^\alpha(\mathrm{e}^{\mathrm{i}x}-1) + (1-w_p^\alpha)(1-\mathrm{e}^{-\mathrm{i}x})\Big)\\
&= (2w_p^\alpha - 1)(g^\alpha(x) + \overline{g^\alpha}(x)) + \mathrm{e}^{\mathrm{i}x}\Big(g^\alpha(x)(1-w_p^\alpha) - \overline{g^\alpha}(x)w_p^\alpha\Big) +\\
&\qquad + \mathrm{e}^{-\mathrm{i}x}\Big(\overline{g^\alpha}(x)(1-w_p^\alpha) - g^\alpha(x)w_p^\alpha\Big)\\
&= (2w_p^\alpha - 1)(g^\alpha(x) + \overline{g^\alpha}(x)) - w_p^\alpha(\mathrm{e}^{\mathrm{i}x} + \mathrm{e}^{-\mathrm{i}x})(g^\alpha(x) + \overline{g^\alpha}(x)) + \mathrm{e}^{\mathrm{i}x}g^\alpha(x) + \mathrm{e}^{-\mathrm{i}x}\overline{g^\alpha}(x)\\
&= (g^\alpha(x) + \overline{g^\alpha}(x))(2w_p^\alpha - 1 - w_p^\alpha(\mathrm{e}^{\mathrm{i}x} + \mathrm{e}^{-\mathrm{i}x})) + \mathrm{e}^{\mathrm{i}x}g^\alpha(x) + \mathrm{e}^{-\mathrm{i}x}\overline{g^\alpha}(x).
\end{aligned}
$$

Then, from equation (1.34), we have $w_p^\alpha = \frac{2-\alpha}{2}$ and from Lemma 4.1.1 and the Euler formulas, we have

$$
\begin{aligned}
F_\alpha(x) =& 2^{2-\alpha}\sin^{1-\alpha}(\tfrac{x}{2})\Big(\sin\Big(\tfrac{x+\alpha(\pi-x)}{2}\Big)(1-\alpha-(2-\alpha)\cos(x)) + \sin(x)\cos\Big(\tfrac{x+\alpha(\pi-x)}{2}\Big) +\\
&+ \cos(x)\sin\Big(\tfrac{x+\alpha(\pi-x)}{2}\Big)\Big)\\
=& 2^{2-\alpha}\sin^{1-\alpha}(\tfrac{x}{2})\Big(\sin\Big(\tfrac{x+\alpha(\pi-x)}{2}\Big)(1-\alpha)(1-\cos(x)) + \sin(x)\cos\Big(\tfrac{x+\alpha(\pi-x)}{2}\Big)\Big).
\end{aligned}
$$

The following theorem answers positively to our request of having a symbol $M_{\alpha,L} - M_{\alpha,R}$ which is nonnegative with a single zero. The proof follows from the study of the two multiplicative factors of the symbol.

Theorem 4.3.1. *Function $F_\alpha(x)$ has a unique zero at $x = 0$ of order $2-\alpha$ for $0 < \alpha < 1$ and $x \in [0,\pi]$.*

Proof. Let us first show that $F_\alpha(x)$ is nonnegative, rewriting $F_\alpha(x) = t_1(x)t_2(x)$, with

$$
\begin{aligned}
t_1(x) &= 2^{2-\alpha}\sin^{1-\alpha}(\tfrac{x}{2}),\\
t_2(x) &= \sin\Big(\tfrac{x+\alpha(\pi-x)}{2}\Big)(1-\alpha)(1-\cos(x)) + \sin(x)\cos\Big(\tfrac{x+\alpha(\pi-x)}{2}\Big).
\end{aligned}
$$

For $(x,\alpha) \in Q = [0,\pi] \times (0,1)$, we have that $\frac{x+\alpha(\pi-x)}{2} \in [0, \frac{\pi}{2}]$ and therefore $F_\alpha(x) \geq 0$, being sums and products of nonnegative functions. In order to prove that $F_\alpha(x)$ has a unique zero at 0, let us consider $F_\alpha'(x) = t_1'(x)t_2(x) + t_1(x)t_2'(x)$, where

$$
t_1'(x) = 2^{1-\alpha}(1-\alpha)\sin^{-\alpha}(\tfrac{x}{2})\cos(\tfrac{x}{2}),
$$

$$\begin{aligned}
t_2'(x) &= \cos\left(\tfrac{x+\alpha(\pi-x)}{2}\right)\tfrac{1-\alpha}{2}(1-\alpha)(1-\cos(x))+\\
&\quad+\sin\left(\tfrac{x+\alpha(\pi-x)}{2}\right)\tfrac{1-\alpha}{2}\sin(x)+\cos(x)\cos\left(\tfrac{x+\alpha(\pi-x)}{2}\right)\\
&= \cos\left(\tfrac{x+\alpha(\pi-x)}{2}\right)\Big(-\tfrac{(1-\alpha)^2}{2}(\cos(x)-1)+\cos(x)-1+1\Big)+\\
&\quad+\sin\left(\tfrac{x+\alpha(\pi-x)}{2}\right)\tfrac{1-\alpha}{2}\sin(x)\\
&= \cos\left(\tfrac{x+\alpha(\pi-x)}{2}\right)\Big(1-(1-\cos(x))\Big(1-\tfrac{(1-\alpha)^2}{2}\Big)\Big)+\\
&\quad+\sin\left(\tfrac{x+\alpha(\pi-x)}{2}\right)\tfrac{1-\alpha}{2}\sin(x).
\end{aligned}$$

It is easy to see that $t_1'(x)t_2(x) \geq 0$ and that $t_1'(x)t_2(x) = 0$ only if $x = 0$ or $x = \pi$. Moreover, since

$$0 \leq (1-\cos(x))\Big(1-\tfrac{(1-\alpha)^2}{2}\Big) < 1,$$

we have that $t_2'(x) \geq 0$ and $t_2'(x) = 0$ only for $x = \pi$. Hence, $t_1(x)t_2'(x) = 0$ for $x = 0$ or $x = \pi$. As a consequence, $F_\alpha'(x) \geq 0$ in Q and $F_\alpha'(x) = 0$ for $x = 0$ or $x = \pi$, which means that $F_\alpha'(x)$ is monotonically increasing for $x \in (0,\pi)$ and $\alpha \in (0,1)$. On the other hand, $F_\alpha(0) = 0$, therefore $F_\alpha(x)$ has a unique zero at 0. Moreover, for $x \to 0$, it holds

$$F_\alpha(x) \sim 2^{2-\alpha}x^{1-\alpha}\big[\sin(\tfrac{\alpha\pi}{2})(1-\alpha)\tfrac{1}{2}x^2 + x\cos(\tfrac{\alpha\pi}{2})\big] = O(x^{2-\alpha}),$$

which proves that the order of the zero at 0 is $2-\alpha$. □

Remark 4.3.2. *It is well-known that in case of a one-dimensional second-order diffusion equation, the symbol of the coefficient matrix has a zero of order* 2 *at* $x = 0$*, which is in accordance with the limit case* $\alpha = 0$ *where we have* $F_0(x) = 2(2—\cos x)$*, i.e., a multiple of the Laplacian symbol.*

It is easy to see that the properties of $F_\alpha(x)$ transfer to the symbol of A_{FV}. First recall that $F_\alpha(x)$ is the symbol of $M_{\alpha,L} - M_{\alpha,R}$ in equation (4.16). Therefore, multiplying by the scaling parameters and diffusion coefficients we have

$$A_x = rK_xT_N(F_\alpha(x)).$$

Similarly, along the second spatial dimension,

$$A_y = sK_yT_N(F_\beta(y)).$$

Therefore,

$$A_{FV} = T_N(\mathscr{F}_{\alpha,\beta}(x,y)), \text{ where } \mathscr{F}_{\alpha,\beta}(x,y) = r\,K_xF_\alpha(x) + sK_yF_\beta(y).$$

If we suppose $\frac{r}{s} \to c$, with $c \in \mathbb{R}^+$, when $N_x, N_y \to \infty$, then from Theorem 4.3.1 the following corollary immediately follows.

Corollary 4.3.1. *Let $\alpha, \beta \in (0,1)$, $\frac{r}{s} \to c$ as $N_x, N_y \to \infty$ and take constant diffusion coefficients, then the symbol $\mathscr{F}_{\alpha,\beta}(x)$ is a nonnegative function that has a unique zero at $(x,y) = (0,0)$ of order* $\min\{2-\alpha, 2-\beta\}$.

4.4 Symbol-based fast solvers

Based on the analysis performed in Section 4.3, in this section we propose two iterative strategies for solving (4.12) and (4.14). Precisely, we present a MGM with damped Jacobi as smoother and a band preconditioner whose inverse is approximated through one iteration of the aforementioned multigrid.

4.4.1 Multigrid methods

From Section 1.4 we recall that the convergence of the V-cycle relies on the smoothing property and approximation property. In order to discuss the convergence analysis of V-cycle applied either to (4.12) or (4.14), we consider constant diffusion coefficients and weighted Jacobi as smoother.
Under these assumptions and because of the Toeplitz structure of the considered matrices, from Remark 1.4.1, the weighted Jacobi is well-known to satisfy the smoothing property for positive definite matrices, whenever it is convergent. Moreover, thanks to Remarks 4.2.1, 4.3.2 and Theorem 4.3.1, the approximation property holds with the same projectors as in the case of the Laplacian (see Moghaderi et al. [2017]) for both FV and FVE approaches.

We stress that it is hard to preserve the block Toeplitz-like structure of A_{FV} and A_{FVE} at the coarser levels using a Galerkin approach, while implementing the geometric approach allows to perform the matrix-vector products by fast Fourier transforms at each coarser grid. Therefore, our multigrid hierarchy is built through the geometric approach and the amount of levels is given by $\text{lvl} = \lfloor \log_2(N_x) \rfloor$, i.e., the coarsest level has size 1×1. Note that in order to make the V-cycle properly working, the linear systems must be scaled such that the right-hand side does not contain any grid dependent scaling factor. Therefore, we scale both $A_{FV}x = b$ and $A_{FVE}x = b$ by $h_x h_y$.

At each iteration of V-cycle one iteration of weighted Jacobi as pre- and post-smoother is performed. The weight ω is estimated through Algorithm 4, introduced in Chapter 2 when dealing with a different FDE. Numerical results in Section 4.5 show that Algorithm 4 fits also here.

Remark 4.4.1. *Since matrices A_{FV} and A_{FVE} are both sums and product between* *diagonal matrices and Toeplitz matrices, thanks to Remark 1.3.4 the matrix-vector product can be performed in $O(N \log N)$ operations, without assembling the coefficient matrix, and the storage only requires $O(N)$ floating points. Therefore, one iteration of V-cycle has a computational cost of $O(N \log N)$ operations.*

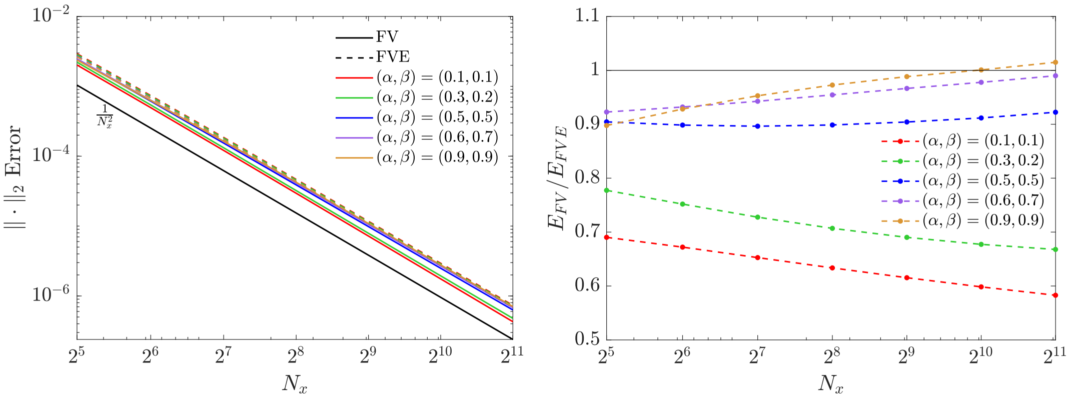
4.4.2 Banded preconditioner

In Meerschaert and Tadjeran [2004] and Donatelli et al. [2018] it was respectively proven that coefficients $t_k^{(1-\alpha)}, \hat{t}_k^{(\alpha)} \to 0$ as $k \to \infty$. This motivates the choice, of a band truncation of the discretized fractional operators. Here we consider a band truncation of matrices $G_{\gamma,N_x}, G_{\gamma,N_y}$ and $M_{\gamma,L}, M_{\gamma,R}$, $\gamma \in \{\alpha, \beta\}$ for FVE and FV, respectively. The resulting block-banded banded-block matrix $\tilde{A}$ is used as GMRES preconditioner. Instead of inverting $\tilde{A}$, we apply one iteration of V-cycle before each iteration of GMRES. The resulting GMRES preconditioner is denoted by $\mathscr{P}_{\mathbf{VB}}$, where $\mathbf{B}$ is an odd integer number which denotes the block bandwidth and the bandwidth of each block.
The hierarchy of $\mathscr{P}_{\mathbf{VB}}$ is built through the geometric approach. Due to the sparsity of $\tilde{A}$, a more robust approach could be obtained building the hierarchy of $\mathscr{P}_{\mathbf{VB}}$ by means of the Galerkin approach, but it would be harder to estimate the relaxation parameter of Jacobi and a different smoother should be adopted. Note that, due to the band structure, the preconditioning iteration has a linear cost.

4.5 Numerical Results

In this section we check the second-order convergence of the FV scheme proposed in Section 4.3 and we test the performances of the methods presented in Section 4.4 when applied to both (4.12) and (4.14). Precisely, we compare the V-cycle algorithm given in Section 4.4.1 as both main solver (denoted by $\mathbf{V}$) and GMRES preconditioner (denoted by $\mathscr{P}_{\mathbf{V}}$), with the banded preconditioner $\mathscr{P}_{\mathbf{VB}}$ given in Section 4.4.2.

Our numerical test have been run on a server with Intel(R) Xeon(R) Silver 4114 at 2.20GHz, 64 GB of RAM and Matlab 2019b. In all considered examples $N_x = N_y \in \{2^4-1, ..., 2^{11}-1\}$, and the initial guess $x^{(0)}$ is the null vector. The stopping criterion for the V-cycle is $\frac{\|Ax^{(k)}-b\|_2}{\|b\|_2} <$ tol, where the tolerance is tol$= 10^{-7}$ and $x^{(k)}$ is the unknown at the k-th iteration, while for the built-in GMRES Matlab function it is $\frac{\|P^{-1}Ax^{(k)}-P^{-1}b\|_2}{\|P^{-1}b\|_2} <$ tol, where P is the preconditioner.



(a) (b)

Figure 4.2. (a) Behaviour of the relative 2-norm errors E_{FV} (continuous lines) and E_{FVE} (dashed lines) as N_x increases and (α,β) vary, (b) Behaviour of the ratio $\frac{E_{FV}}{E_{FVE}}$ as N_x increases and (α,β) vary.

Let us consider function $\tilde{u}(x) = x^2(1-x)^2,\ x \in \Omega = [0,1]$. From Pan et al. [2017], the exact Riesz fractional derivatives of order $1-\alpha$ and $2-\alpha$ of $\tilde{u}$, are

$$\begin{aligned}\frac{\mathrm{d}^{1-\alpha}\tilde{u}(x)}{\mathrm{d}\,|x|^{1-\alpha}} &= \eta(\alpha)\sum_{k=1}^{3} a_k \frac{(x^{\alpha+k}-(1-x)^{\alpha+k})}{\Gamma(\alpha+k+1)}\\ \frac{\mathrm{d}^{2-\alpha}\tilde{u}(x)}{\mathrm{d}\,|x|^{2-\alpha}} &= \eta(\alpha)\sum_{k=1}^{3} a_k \frac{(x^{\alpha+k-1}+(1-x)^{\alpha+k-1})}{\Gamma(\alpha+k)}.\end{aligned} \tag{4.17}$$

where $(a_1,a_2,a_3)=(2,-12,24)$. In the following examples we consider $u(x,y)=\tilde{u}(x)\tilde{u}(y)$ and build the exact forcing term $v(x,y)$ through the formulas in equation (4.17), for these two choices of the diffusion coefficients:

- *Choice 1*: $K_x(x,y)=K_y(x,y)=1$;
- *Choice 2* Pan et al. [2017]: $K_x(x,y)=K_y(x,y)=\mathrm{e}^{4x+4y}$.

Example 1 First we test the accuracy provided by both FVE and FV approaches while considering *Choice 1*. Figure 4.2b reports the relative 2-norm error in FV (E_{FV}) and in FVE (E_{FVE}), while Figure 4.2a reports the ratio between the two as N_x increases and α,β vary.

In Figure 4.2a, a comparison with the black line representing the square of the step length $h_x (= h_y)$ confirms the convergence of order 2 for both FV and FVE.

When the ratio between the errors in Figure 4.2b is smaller than 1, then the FV approach allows better approximation of the solution than the FVE approach. We note that FV has a lower approximation error than FVE in the cases where $\alpha, \beta \geq 0.5$. Especially, when $\alpha, \beta \approx 1$ the error in FVE is decreasing faster than in FV, therefore we expect FVE to yield better results than FV when $N_x > 2^{11}-1$. On the contrary, when $\alpha, \beta \approx 0$ the error in FV decreases faster and reaches almost half the error of FVE for $N_x = 2^{11} - 1$. Therefore, it is reasonable to expect further improvements in approximation error for FV with respect to FVE when $N_x > 2^{11}-1$. Further tests, which are not reported here, show that similar results are achieved also for *Choice 2*.

Example 2 We now test the behavior of our proposals for solving the two linear systems obtained from FVE and FV when considering *Choice 2*. Table 4.1 and 4.2 respectively show iterations to tolerance (IT) and CPU times of algorithms $\mathbf{V}$, $\mathscr{P}_{\mathrm{V}}$ and $\mathscr{P}_{\mathrm{V5}}$ described in Section 4.4 compared with:

- $\mathscr{P}_{\mathrm{VL}(geo)}$, which is the 2D Laplacian preconditioner introduced in Donatelli et al. [2018] inverted through one iteration of V-cycle with the geometric approach and Jacobi weight $\omega = 0.75$ (as in Donatelli et al. [2018]);
- $\mathscr{P}_{\mathrm{VL}(gal)}$, which is the same as preconditioner $\mathscr{P}_{\mathrm{VL}(geo)}$, but implemented through the Galerkin approach;
- $\mathbf{V}(\tilde{\omega})$ and $\mathscr{P}_{\mathrm{V}(\tilde{\omega})}$, which are the same as $\mathbf{V}$ and $\mathscr{P}_{\mathrm{V}}$ but with Jacobi weight fixed as $\tilde{\omega} = 0.75 + \frac{\sqrt{\min(\alpha,\beta)}}{4}$ (see Donatelli et al. [2018]).

We do not consider any circulant preconditioner for two different reasons: first, in Donatelli et al. [2018] it has been shown that circulant matrices are slower than MGMs; second, it is well-known that if used as preconditioners for multilevel Toeplitz matrices, multilevel circulant matrices cannot ensure a superlinear convergence character (see Serra-Capizzano and Tyrtyshnikov [n.d.]).

In Table 4.1 and 4.2, the numbers in bold highlight, in each row, the combination with the fastest computational time. We note that, as expected, when $\alpha = \beta$, the convergence of $\mathbf{V}$ and $\mathscr{P}_{\mathrm{V}}$ is almost independent of the grid size and the amount of iterations is low. When $\alpha, \beta \approx 0$, the block-banded-banded-block preconditioner $\mathscr{P}_{\mathrm{V5}}$ yields almost the same iterations as the full matrix $\mathscr{P}_{\mathrm{V}}$, but with lower

$\begin{pmatrix}\alpha\\ \beta\end{pmatrix}$	N_x+1	V-cycle				Preconditioned GMRES									
		V		$\mathbf{V}(\tilde{\omega})$		$\mathscr{P}_{\mathbf{V5}}$		$\mathscr{P}_{\mathbf{VL}(geo)}$		$\mathscr{P}_{\mathbf{VL}(gal)}$		$\mathscr{P}_{\mathbf{V}}$		$\mathscr{P}_{\mathbf{V}(\tilde{\omega})}$	
		FVE	FV	FVE	FV	FVE	FV	FVE	FV	FVE	FV	FVE	FV	FVE	FV
$\begin{pmatrix}0.1\\ 0.1\end{pmatrix}$	2^6	10	16	11	15	**7**	10	9	12	9	12	7	10	7	8
	2^7	10	15	11	16	8	**10**	11	12	11	12	7	10	8	9
	2^8	10	15	12	16	9	**10**	12	14	11	14	7	9	8	9
	2^9	11	15	12	17	10	**10**	14	**13**	12	16	8	9	8	9
	2^{10}	11	16	13	17	10	**10**	13	17	14	17	8	10	8	12
	2^{11}	**11**	16	13	18	10	13	16	18	15	18	8	10	8	11
$\begin{pmatrix}0.3\\ 0.2\end{pmatrix}$	2^6	24	22	18	25	**11**	11	20	22	17	22	11	11	10	12
	2^7	20	24	20	27	12	**13**	22	23	22	24	10	11	12	12
	2^8	23	26	22	29	12	**13**	26	29	23	29	13	13	12	14
	2^9	25	28	24	32	14	**14**	33	36	28	34	15	13	12	14
	2^{10}	25	31	26	34	14	**15**	36	37	34	37	14	15	13	16
	2^{11}	27	33	28	37	15	**16**	37	40	35	44	15	16	14	16
$\begin{pmatrix}0.5\\ 0.5\end{pmatrix}$	2^6	8	11	9	11	9	**9**	19	21	20	23	6	8	6	7
	2^7	**9**	**11**	10	12	**11**	**12**	26	29	26	29	6	7	6	8
	2^8	9	11	10	13	14	**12**	30	31	30	34	6	8	7	8
	2^9	**10**	12	11	13	14	15	40	39	40	42	7	8	7	8
	2^{10}	**10**	12	11	14	18	16	43	46	44	56	7	9	7	8
	2^{11}	**11**	13	12	14	20	18	54	57	54	63	7	9	8	9
$\begin{pmatrix}0.6\\ 0.7\end{pmatrix}$	2^6	13	16	13	18	12	**12**	31	34	31	35	8	9	9	10
	2^7	14	18	14	19	14	**14**	36	45	36	49	9	9	8	10
	2^8	16	19	16	21	16	**16**	50	52	51	54	10	11	9	10
	2^9	17	21	17	22	19	**18**	60	74	61	67	10	11	9	11
	2^{10}	18	22	18	24	23	**21**	92	88	94	90	12	11	10	12
	2^{11}	19	24	20	26	30	**24**	93	103	106	115	12	12	11	14
$\begin{pmatrix}0.9\\ 0.9\end{pmatrix}$	2^6	7	9	7	9	12	**12**	32	33	33	36	5	6	5	6
	2^7	**7**	9	7	9	16	15	40	50	41	55	5	6	5	6
	2^8	**7**	10	8	10	21	18	69	59	63	77	5	6	5	6
	2^9	**8**	10	**8**	10	27	23	84	87	86	92	5	7	5	7
	2^{10}	**8**	10	**8**	10	36	30	103	109	102	110	5	7	6	7
	2^{11}	**8**	11	9	11	48	38	147	154	145	156	6	7	6	7

Table 4.1. Iterations to tolerance of the V-cycles **V**, $\mathbf{V}(\tilde{\omega})$, and the preconditioned GMRES with preconditioners $\mathscr{P}_{\mathbf{V5}}$, $\mathscr{P}_{\mathbf{VL}(geo)}$, $\mathscr{P}_{\mathbf{VL}(gal)}$, $\mathscr{P}_{\mathbf{V}}$, $\mathscr{P}_{\mathbf{V}(\tilde{\omega})}$.

CPU times due to the lower computational cost per iteration. Moreover, preconditioners $\mathscr{P}_{\mathbf{VL}(geo)}$ and $\mathscr{P}_{\mathbf{VL}(gal)}$ are less robust than $\mathscr{P}_{\mathbf{V5}}$ and comparing $\mathbf{V}(\tilde{\omega})$ with **V** we note that the adaptive choice of the Jacobi weight explained in Section 4.4 allows slightly faster convergence with respect to the fixed weight $\tilde{\omega}$.

When $\alpha, \beta \approx 1$, instead, the block-banded-banded-block preconditioner seems not to be suitable anymore. This is due to the decay of the coefficient of the matrix. As shown in Meerschaert and Tadjeran [2004], the coefficients of the GL formulas tend to zero as the index increases with an order that depends on the fractional derivative order $2-\alpha$, i.e., the larger is α, the slower the coefficients tends to zero. Therefore, a good band approximation of the coefficient matrix

$\begin{pmatrix}\alpha\\ \beta\end{pmatrix}$	N_x + 1	V-cycle				Preconditioned GMRES									
		V		**V**($\tilde{\omega}$)		$\mathscr{P}_{\mathbf{V5}}$		$\mathscr{P}_{\mathbf{VL}(geo)}$		$\mathscr{P}_{\mathbf{VL}(gal)}$		$\mathscr{P}_{\mathbf{V}}$		$\mathscr{P}_{\mathbf{V}(\tilde{\omega})}$	
		FVE	FV	FVE	FV	FVE	FV	FVE	FV	FVE	FV	FVE	FV	FVE	FV
$\begin{pmatrix}0.1\\ 0.1\end{pmatrix}$	2^6	5.9e-2	9.3e-2	6.6e-2	8.7e-2	**2.5e-2**	3.7e-2	3.0e-2	4.0e-2	3.0e-2	4.1e-2	9.2e-2	1.1e-1	7.0e-2	7.6e-2
	2^7	1.3e-1	1.8e-1	1.7e-1	2.4e-1	1.3e-1	**1.2e-1**	1.4e-1	1.3e-1	1.3e-1	1.3e-1	2.2e-1	3.1e-1	2.5e-1	2.6e-1
	2^8	4.3e-1	6.2e-1	5.1e-1	6.6e-1	6.2e-1	**3.9e-1**	4.3e-1	4.7e-1	4.2e-1	4.9e-1	6.9e-1	7.7e-1	8.5e-1	8.7e-1
	2^9	1.6e+0	2.0e+0	1.7e+0	2.3e+0	2.4e+0	**1.4e+0**	1.9e+0	**1.4e+0**	1.6e+0	2.0e+0	3.0e+0	2.6e+0	2.8e+0	2.9e+0
	2^{10}	6.4e+0	8.9e+0	7.4e+0	9.3e+0	9.9e+0	**5.4e+0**	6.4e+0	8.0e+0	8.4e+0	8.3e+0	1.3e+1	1.4e+1	1.2e+1	1.7e+1
	2^{11}	**3.6e+1**	4.9e+1	4.3e+1	5.5e+1	4.6e+1	4.0e+1	4.6e+1	4.2e+1	4.4e+1	4.6e+1	7.0e+1	7.4e+1	6.4e+1	9.0e+1
$\begin{pmatrix}0.3\\ 0.2\end{pmatrix}$	2^6	1.4e-1	1.2e-1	1.1e-1	1.2e-1	**3.2e-2**	3.4e-2	5.8e-2	6.1e-2	4.6e-2	6.1e-2	1.2e-1	1.2e-1	1.2e-1	1.3e-1
	2^7	2.6e-1	3.2e-1	2.9e-1	4.1e-1	1.7e-1	**1.5e-1**	2.2e-1	2.0e-1	2.1e-1	2.1e-1	2.8e-1	2.8e-1	4.2e-1	3.5e-1
	2^8	9.8e-1	1.1e+0	9.4e-1	1.2e+0	6.3e-1	**4.7e-1**	8.6e-1	8.2e-1	7.3e-1	8.5e-1	1.4e+0	1.2e+0	1.1e+0	1.2e+0
	2^9	3.5e+0	3.8e+0	3.3e+0	4.3e+0	2.9e+0	**1.7e+0**	3.7e+0	3.4e+0	3.0e+0	3.3e+0	5.5e+0	3.8e+0	3.8e+0	4.0e+0
	2^{10}	1.4e+1	1.7e+1	1.5e+1	1.9e+1	1.2e+1	**7.8e+0**	1.5e+1	1.4e+1	1.5e+1	1.4e+1	1.9e+1	2.0e+1	1.7e+1	2.1e+1
	2^{11}	8.6e+1	9.9e+1	8.8e+1	1.1e+2	5.8e+1	**4.2e+1**	7.8e+1	7.3e+1	7.6e+1	8.4e+1	1.1e+2	1.1e+2	1.1e+2	1.1e+2
$\begin{pmatrix}0.5\\ 0.5\end{pmatrix}$	2^6	4.8e-2	6.4e-2	5.4e-2	6.4e-2	3.3e-2	**2.9e-2**	5.6e-2	5.7e-2	5.9e-2	6.3e-2	6.2e-2	9.7e-2	6.2e-2	6.9e-2
	2^7	**1.4e-1**	**1.4e-1**	1.6e-1	1.8e-1	**1.4e-1**	**1.4e-1**	2.6e-1	2.6e-1	2.6e-1	2.7e-1	1.6e-1	2.1e-1	1.7e-1	2.7e-1
	2^8	3.9e-1	4.4e-1	4.2e-1	5.4e-1	7.1e-1	**3.7e-1**	9.5e-1	9.0e-1	9.5e-1	9.4e-1	4.7e-1	8.7e-1	6.9e-1	7.2e-1
	2^9	**1.4e+0**	1.6e+0	1.6e+0	1.7e+0	2.3e+0	1.7e+0	4.2e+0	3.5e+0	4.3e+0	3.9e+0	2.3e+0	2.9e+0	2.3e+0	2.4e+0
	2^{10}	**5.7e+0**	6.7e+0	6.3e+0	7.7e+0	1.3e+1	7.4e+0	1.8e+1	1.6e+1	1.8e+1	2.0e+1	9.4e+0	1.3e+1	9.4e+0	1.1e+1
	2^{11}	**3.4e+1**	4.0e+1	3.7e+1	4.4e+1	6.4e+1	4.0e+1	1.1e+2	1.0e+2	1.1e+2	1.1e+2	5.1e+1	6.9e+1	5.6e+1	7.1e+1
$\begin{pmatrix}0.6\\ 0.7\end{pmatrix}$	2^6	7.7e-2	9.2e-2	7.7e-2	1.0e-1	4.0e-2	**3.5e-2**	8.3e-2	8.4e-2	8.2e-2	8.8e-2	9.9e-2	1.0e-1	1.1e-1	1.1e-1
	2^7	2.2e-1	2.3e-1	1.9e-1	2.9e-1	1.6e-1	**1.4e-1**	3.3e-1	3.8e-1	3.2e-1	4.1e-1	3.6e-1	2.5e-1	2.7e-1	3.0e-1
	2^8	6.8e-1	7.8e-1	6.8e-1	8.7e-1	7.0e-1	**4.7e-1**	1.5e+0	1.4e+0	1.5e+0	1.5e+0	1.0e+0	1.1e+0	8.0e-1	8.3e-1
	2^9	2.4e+0	2.8e+0	2.4e+0	2.9e+0	2.9e+0	**1.8e+0**	5.9e+0	6.2e+0	6.0e+0	5.8e+0	3.4e+0	3.5e+0	2.7e+0	3.4e+0
	2^{10}	1.0e+1	1.2e+1	1.0e+1	1.3e+1	1.4e+1	**8.2e+0**	3.5e+1	2.8e+1	3.5e+1	3.0e+1	1.8e+1	1.4e+1	1.4e+1	1.5e+1
	2^{11}	6.0e+1	7.3e+1	6.6e+1	7.9e+1	8.7e+1	**4.6e+1**	1.8e+2	1.7e+2	2.0e+2	1.9e+2	1.1e+2	8.4e+1	8.5e+1	1.0e+2
$\begin{pmatrix}0.9\\ 0.9\end{pmatrix}$	2^6	4.2e-2	5.2e-2	4.2e-2	5.3e-2	3.3e-2	**3.0e-2**	8.5e-2	8.2e-2	8.7e-2	8.9e-2	5.4e-2	6.1e-2	5.5e-2	6.1e-2
	2^7	**9.4e-2**	1.3e-1	1.1e-1	1.4e-1	1.5e-1	1.2e-1	3.4e-1	4.1e-1	3.5e-1	4.6e-1	1.4e-1	1.6e-1	1.5e-1	1.7e-1
	2^8	**3.1e-1**	4.1e-1	3.4e-1	4.2e-1	7.9e-1	4.9e-1	2.1e+0	1.6e+0	1.8e+0	2.0e+0	4.0e-1	4.4e-1	4.1e-1	4.6e-1
	2^9	**1.1e+0**	1.4e+0	**1.1e+0**	1.4e+0	3.7e+0	2.1e+0	8.0e+0	7.2e+0	8.4e+0	7.8e+0	1.4e+0	2.2e+0	1.4e+0	2.2e+0
	2^{10}	**4.6e+0**	5.6e+0	**4.6e+0**	5.6e+0	2.0e+1	1.1e+1	3.8e+1	3.4e+1	3.7e+1	3.6e+1	5.5e+0	9.3e+0	8.7e+0	9.2e+0
	2^{11}	**2.5e+1**	3.4e+1	2.8e+1	3.4e+1	1.2e+2	7.1e+1	2.7e+2	2.5e+2	2.7e+2	2.6e+2	4.6e+1	5.1e+1	4.8e+1	4.9e+1

Table 4.2. CPU times of the V-cycles **V**, **V**($\tilde{\omega}$), and the preconditioned GMRES with preconditioners $\mathscr{P}_{\mathbf{V5}}$, $\mathscr{P}_{\mathbf{VL}(geo)}$, $\mathscr{P}_{\mathbf{VL}(gal)}$, $\mathscr{P}_{\mathbf{V}}$, $\mathscr{P}_{\mathbf{V}(\tilde{\omega})}$.

requires a wider band than in the case of $\alpha, \beta \approx 0$. This of course affects the CPU times and tests not reported in Tables 4.1–4.2 show $\mathscr{P}_{\mathbf{V11}}$ to be a robust solver, but still slower than **V**.

When $\alpha \neq \beta$, the number of iterations of all methods tends to increase as N_x increases. This is due to the anisotropy of the diffusion along the two coordinate axes. Since hypothesis $\frac{r}{s} \to c$ in Corollary 4.3.1 is not satisfied, neither is the approximation property, therefore the projectors in V-cycle should be built differently and a strategy like that proposed in Chapter 2 should be explored. Nevertheless, using the GMRES with $\mathscr{P}_{\mathbf{V}}$, not only halves the iteration with respect to **V**, but also seems to be much more robust in the anisotropic cases. Consequently, using the lighter preconditioner $\mathscr{P}_{\mathrm{V5}}$ instead of $\mathscr{P}_{\mathrm{V}}$ allows to reach the lowest CPU times without losing in robustness.

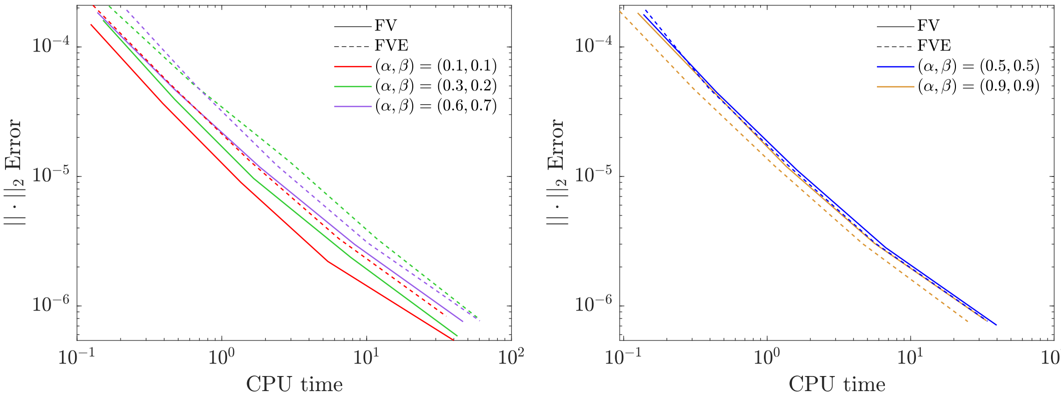
Now, let us fix $N_x = 2^{11}-1$ and consider the solvers with the lowest CPU time in Table 4.2 for FV and FVE and for each combination of (α,β). More precisely we consider solver $\mathscr{P}_{\mathrm{V5}}$ for FV except for $\alpha=\beta=0.5$ and $\alpha=\beta=0.9$, where we use **V**, and solver **V** for FVE except for $(\alpha,\beta)=(0.3,0.2)$, where we use $\mathscr{P}_{\mathrm{V5}}$. Figure 4.3 shows the 2-norm error versus the CPU time of such solvers for FV (solid line) and FVE (dashed line). We note that when $\alpha,\beta\approx 0$, the FV method is more efficient since it allows to compute solutions with smaller error than FVE in the same amount of time, despite the fact that for a given grid FVE is sometimes faster (see Table 4.2). FV seems to be more efficient than FVE in the anisotropic cases too, even for large α,β where FVE has a higher accuracy. Instead, in the isotropic cases with $\alpha,\beta\approx 1$ both approaches allow similar CPU times and, therefore, FVE becomes more suitable than FV.

We stress that due to the presence of the tridiagonal mass matrices, each matrix-vector product is more expensive in FVE than in FV. This goes in favor of FV since allows V-cycle to yield faster results than in the case of FVE, even when a larger number of iterations is required.

Remark 4.5.1. *Note that it is not possible to compare the iterations of* $\mathscr{P}_{\mathrm{VL}(geo)}$ *and **V**($\tilde{\omega}$) in Table 4.1 with preconditioners* P_{2_N} *and* $MGM_{2D}(J)$ *in Donatelli et al. [2018] because therein the 2D discretization is different from the one given equation* (4.13). *Indeed, in Donatelli et al. [2018]; Pan et al. [2017] the authors replaced the tridiagonal mass matrix with an identity matrix resulting in a mixed FV and FVE approach.*

4.6 Conclusions

We have introduced a second-order FV discretization for problem (4.1) and numerical results confirm that it is a good alternative to the FVE approach when $\alpha,\beta\approx 0$. Moreover, we have proposed a block-banded-banded-block preconditioner for GMRES that allows a fast solution of the resulting linear systems in an amount of iterations to tolerance that is stable as the matrix-size increases. When $\alpha,\beta\approx 1$, the FVE approach revealed more accurate than FV. In this case, a MGM used as standalone solver for the discretized problem should be preferred. Same as in Donatelli et al. [2018], we used damped Jacobi as smoother, but here we selected its weight adaptively, as in Chapter 2, which yields better results if



(a) (b)

Figure 4.3. Trend of the 2-norm error versus the CPU time of the fastest solver for various combinations of (α,β).

compared to the fixed weight proposed in Donatelli et al. [2018].

Chapter 5

Non-uniform meshes

In this chapter we focus on a conservative steady-state Caputo FDE. It is known that the solution of such equations can exhibit singularities near the boundaries Kopteva and Meng [2020]; Gracia et al. [2018]; Jia and Wang [2015]. As consequence of this and due to the conservative nature of the problem, we adopt a FVE discretization approach over a generic non-uniform mesh.

In Section 5.1 we explicitly provide all the coefficients of the resulting linear system. Moreover, we introduce composite meshes and meshes mapped functions, which are both partially uniform and partially graded near the singularity, in order to yield a partial Toeplitz structure of the coefficient matrix, but that differ in terms of regularity.

By restricting our studies to meshes mapped by continuous bijective functions, in Section 5.2 we perform a spectral analysis of the coefficient matrix, by retrieving its symbol. Such information will be used in Section 5.3 to design an ad-hoc multigrid preconditioner, which is parameter free since the relaxing parameter of Jacobi smoother is estimated through the approach introduced in Section 2.4. Finally, in Section 5.4, numerical results show the stability of the iterations to tolerance of our multigrid used as preconditioner when increasing the size of the coefficient matrix and while varying some parameters of the equation and the discretization grid. Our conclusions are drawn in Section 5.5.

5.1 Problem setting and discretization

In this section we introduce the FDE problem we are interested in, then we provide the full discretization over arbitrary uniform meshes, and finally we give the formal definition of the composite mesh and the mesh mapped by a non-linear function, which will be numerically tested in Section 5.4.

5.1.1 Two-dimensional space-FDE

We focus on the following conservative steady-state Caputo FDE of order $2-\beta$, $0<\beta<1$, with inhomogeneous Dirichlet boundary-value conditions Wang and Yang [2017]; Gracia et al. [2020], i.e.,

$$\begin{cases} -\dfrac{\mathrm{d}}{\mathrm{d}x}\left(K(x)\left(\gamma\,\dfrac{\mathrm{d}^{1-\beta}u(x)}{\mathrm{d}^C(x-0)^{1-\beta}}+(1-\gamma)\,\dfrac{\mathrm{d}^{1-\beta}u(x)}{\mathrm{d}^C(1-x)^{1-\beta}}\right)\right)=f(x), \qquad x\in[0,1],\\ u(0)=u_l, \quad u(1)=u_r, \end{cases} \tag{5.1}$$

where $K(x)$ is a positive diffusion coefficient, $f(x)$ is the source term, u_l, u_r are the Dirichlet boundary values and $0\leq\gamma\leq 1$ indicates the anisotropy in the diffusion, i.e., $\gamma\approx 0$ and $\gamma\approx 1$ imply a strong forward and backward diffusivity, respectively. The operators $\frac{\mathrm{d}}{\mathrm{d}x}\frac{\mathrm{d}^{1-\beta}}{\mathrm{d}^C(1-x)^{1-\beta}}$ and $\frac{\mathrm{d}}{\mathrm{d}x}\frac{\mathrm{d}^{1-\beta}}{\mathrm{d}^C(x-0)^{1-\beta}}$ are known as RiemannLiouvilleCaputo fractional derivatives Jia et al. [2019] or as PatieSimon fractional derivatives Baeumer et al. [2018]; Kelly et al. [2019]; Patie and Simon [2012].

5.1.2 FVE scheme

From Chapter 4 we recall that FVE approach consists in restricting the admissible solutions $u(x)$ of the FDE (5.1) to a certain finite element space, covering the definition interval $[0,1]$ with a mesh $\cup_{i=1}^n I_i$, where $\mu(I_i\cap I_j)=0,\ i\neq j$, with μ the Lebesgue measure, and finally integrating the FDE (5.1) over I_i. In our specific case we consider the unknown $u(x)$ to belong to the space of the piecewise linear polynomial functions. In the following we provide a full discretization over a generic non-structured mesh (Section 5.1.3), then we consider the special case of a uniform mesh (Section 5.1.4) and make a comparison with the discretization in Donatelli et al. [2018]. Finally, in Section 5.1.5 we focus on two non-uniform structured meshes, obtained as a combination of a non-uniform part and a uniform one.

5.1.3 Generic non-uniform mesh

Let $N \in \mathbb{N}$ and denote by $\{x_i\}_{i=0}^{N+1}$ a generic mesh on $[0,1]$, such that $x_i > x_{i-1}$, $\forall i = 1,...,N+1$ with $x_0 = 0, x_{N+1} = 1$, then a

$$\tilde{u}(x) = \sum_{i=1}^{N} u_i \phi_i(x) + u_l \phi_0(x) + u_r \phi_{N+1}(x),$$

where $\{\phi_i\}_{i=0}^{N+1}$ is the set of hat (linear) functions with

$$\phi_0(x) = \begin{cases} \frac{x_1 - x}{h_1}, & x \in (x_0, x_1) \\ 0, & \text{otherwise} \end{cases}$$

$$\phi_i(x) = \begin{cases} \frac{x - x_{i-1}}{h_i}, & x \in (x_{i-1}, x_i) \\ \frac{x_{i+1} - x}{h_{i+1}}, & x \in (x_i, x_{i+1}) \\ 0, & \text{otherwise} \end{cases} \qquad \text{for } i = 1,...,N,$$

$$\phi_{N+1}(x) = \begin{cases} \frac{x - x_N}{h_{N+1}}, & x \in (x_N, x_{N+1}) \\ 0, & \text{otherwise} \end{cases}$$

with $h_i = x_i - x_{i-1}$, $i = 1,...,N+1$ being the step length. Replacing $u(x)$ with $\tilde{u}(x)$in equation (5.1) and integrating over $\Omega_i = [x_{i-\frac{1}{2}}, x_{i+\frac{1}{2}}]$, where $x_{i-\frac{1}{2}} = \frac{x_i + x_{i-1}}{2}$, equation (5.1) can be written as the linear system

$$A_N u = b, \tag{5.2}$$

where $b \in \mathbb{R}^N$ and $A_N \in \mathbb{R}^{N \times N}$, with

$$\begin{aligned} a_{i,j} &= -K(x) \left(\gamma \, \frac{\mathrm{d}^{1-\beta}}{\mathrm{d}_-^C x^{1-\beta}} + (1-\gamma) \, \frac{\mathrm{d}^{1-\beta}}{\mathrm{d}_+^C x^{1-\beta}} \right) \phi_j(x) \Big|_{x=x_{i-\frac{1}{2}}}^{x=x_{i+\frac{1}{2}}} \\ b_i &= \int_{x_{i-\frac{1}{2}}}^{x_{i+\frac{1}{2}}} f(x)\mathrm{d}x + K(x) \left(\gamma \, \frac{\mathrm{d}^{1-\beta}}{\mathrm{d}_-^C x^{1-\beta}} + (1-\gamma) \, \frac{\mathrm{d}^{1-\beta}}{\mathrm{d}_+^C x^{1-\beta}} \right) (u_l \phi_0(x) + u_r \phi_{N+1}(x)) \Big|_{x=x_{i-\frac{1}{2}}}^{x=x_{i+\frac{1}{2}}}. \end{aligned}$$

for $i, j = 1,...,N$. Explicitly, the entries of b are

$$\begin{aligned} b_1 =& f_1 \frac{h_1 + h_2}{2} + \frac{K_{\frac{3}{2}}}{\Gamma(\beta+1)} \Big[u_L \gamma \frac{1}{h_1} \left((x_{\frac{3}{2}} - x_1)^\beta - x_{\frac{3}{2}}^\beta \right) + u_R (1-\gamma) \frac{1}{h_{N+1}} \left((1 - x_{\frac{3}{2}})^\beta - (x_N - x_{\frac{3}{2}})^\beta \right) \Big] + \\ & - \frac{K_{\frac{1}{2}}}{\Gamma(\beta+1)} \Big[-u_L \gamma \frac{1}{h_1} x_{\frac{1}{2}}^\beta + (1-\gamma) \left(-u_L \frac{1}{h_1} (x_1 - x_{\frac{1}{2}})^\beta + u_R \frac{1}{h_{N+1}} \left((1 - x_{\frac{1}{2}})^\beta - (x_N - x_{\frac{1}{2}})^\beta \right) \right) \Big], \end{aligned}$$

$$\begin{aligned} b_i =& f_i \frac{h_i + h_{i+1}}{2} + \frac{K_{i+\frac{1}{2}}}{\Gamma(\beta+1)} \Big[u_L \gamma \frac{1}{h_1} \left((x_{i+\frac{1}{2}} - x_1)^\beta - x_{i+\frac{1}{2}}^\beta \right) + u_R (1-\gamma) \frac{1}{h_{N+1}} \left((1 - x_{i+\frac{1}{2}})^\beta - (x_N - x_{i+\frac{1}{2}})^\beta \right) \Big] + \\ & - \frac{K_{i-\frac{1}{2}}}{\Gamma(\beta+1)} \Big[u_L \gamma \frac{1}{h_1} \left((x_{i-\frac{1}{2}} - x_1)^\beta - x_{i-\frac{1}{2}}^\beta \right) + u_R (1-\gamma) \frac{1}{h_{N+1}} \left((1 - x_{i-\frac{1}{2}})^\beta - (x_N - x_{i-\frac{1}{2}})^\beta \right) \Big], \end{aligned}$$

for $i = 2, ..., N-1$, and

$$\begin{aligned}
b_N =& f_N \frac{h_N + h_{N+1}}{2} + \frac{K_{N+\frac{1}{2}}}{\Gamma(\beta+1)}\Big[\gamma\Big(\frac{u_L}{h_1}\Big((x_{N+\frac{1}{2}} - x_1)^\beta - x^\beta_{N+\frac{1}{2}}\Big) + \frac{u_R}{h_{N+1}}(x_{N+\frac{1}{2}} - x_N)^\beta\Big) + (1-\gamma)\frac{u_R}{h_{N+1}}(1 - x_{N+\frac{1}{2}})^\beta\Big] + \\
&- \frac{K_{N-\frac{1}{2}}}{\Gamma(\beta+1)}\Big[\gamma\frac{u_L}{h_1}\Big((x_{N-\frac{1}{2}} - x_1)^\beta - x^\beta_{N-\frac{1}{2}}\Big) + (1-\gamma)\frac{u_R}{h_{N+1}}\Big((1 - x_{N-\frac{1}{2}})^\beta - (x_N - x_{N-\frac{1}{2}})^\beta\Big)\Big];
\end{aligned}$$

while, for $i = 1, ..., N$, the entries of A_N are

$$\begin{aligned}
a_{i,i-k} =& \frac{K_{i-\frac{1}{2}}}{\Gamma(\beta+1)}\gamma\left[\frac{(\frac{h_i}{2} + \sum_{j=1}^{k} h_{i-j})^\beta - (\frac{h_i}{2} + \sum_{j=1}^{k-1} h_{i-j})^\beta}{h_{i-k}} + \frac{(\frac{h_i}{2} + \sum_{j=1}^{k-2} h_{i-j})^\beta - (\frac{h_i}{2} + \sum_{j=1}^{k-1} h_{i-j})^\beta}{h_{i-k+1}}\right] + \\
&- \frac{K_{i+\frac{1}{2}}}{\Gamma(\beta+1)}\gamma\left[\frac{(\frac{h_{i+1}}{2} + \sum_{j=0}^{k} h_{i-j})^\beta - (\frac{h_{i+1}}{2} + \sum_{j=0}^{k-1} h_{i-j})^\beta}{h_{i-k}} + \frac{(\frac{h_{i+1}}{2} + \sum_{j=0}^{k-2} h_{i-j})^\beta - (\frac{h_{i+1}}{2} + \sum_{j=0}^{k-1} h_{i-j})^\beta}{h_{i-k+1}}\right],
\end{aligned}$$

for $2 \le k \le i-1$, and

$$\begin{aligned}
a_{i,i-1} =& \frac{K_{i-\frac{1}{2}}}{\Gamma(\beta+1)}\left[\gamma\frac{(h_{i-1} + \frac{h_i}{2})^\beta - (\frac{h_i}{2})^\beta}{h_{i-1}} - \frac{(\frac{h_i}{2})^\beta}{h_i}\right] + \\
&- \frac{K_{i+\frac{1}{2}}}{\Gamma(\beta+1)}\gamma\left[\frac{(h_{i-1} + h_i + \frac{h_{i+1}}{2})^\beta - (h_i + \frac{h_{i+1}}{2})^\beta}{h_{i-1}} + \frac{(\frac{h_{i+1}}{2})^\beta - (h_i + \frac{h_{i+1}}{2})^\beta}{h_i}\right], \\
a_{i,i} =& \frac{K_{i-\frac{1}{2}}}{\Gamma(\beta+1)}\left[\frac{(\frac{h_i}{2})^\beta}{h_i} + (1-\gamma)\frac{(\frac{h_i}{2})^\beta - (h_{i+1} + \frac{h_i}{2})^\beta}{h_{i+1}}\right] - \frac{K_{i+\frac{1}{2}}}{\Gamma(\beta+1)}\left[\gamma\frac{(h_i + \frac{h_{i+1}}{2})^\beta - (\frac{h_{i+1}}{2})^\beta}{h_i} - \frac{(\frac{h_{i+1}}{2})^\beta}{h_{i+1}}\right], \\
a_{i,i+1} =& \frac{K_{i-\frac{1}{2}}}{\Gamma(\beta+1)}(1-\gamma)\left[\frac{(h_{i+1} + \frac{h_i}{2})^\beta - (\frac{h_i}{2})^\beta}{h_{i+1}} + \frac{(h_{i+1} + \frac{h_i}{2})^\beta - (h_{i+2} + h_{i+1} + \frac{h_i}{2})^\beta}{h_{i+2}}\right] + \\
&- \frac{K_{i+\frac{1}{2}}}{\Gamma(\beta+1)}\left[\frac{(\frac{h_{i+1}}{2})^\beta}{h_{i+1}} + (1-\gamma)\frac{(\frac{h_{i+1}}{2})^\beta - (h_{i+2} + \frac{h_{i+1}}{2})^\beta}{h_{i+2}}\right],
\end{aligned}$$

and finally,

$$\begin{aligned}
a_{i,i+k} =& \frac{K_{i-\frac{1}{2}}}{\Gamma(\beta+1)}(1-\gamma)\left[\frac{(\frac{h_i}{2} + \sum_{j=1}^{k} h_{i+j})^\beta - (\frac{h_i}{2} + \sum_{j=1}^{k-1} h_{i+j})^\beta}{h_{i+k}} + \frac{(\frac{h_i}{2} + \sum_{j=1}^{k} h_{i+j})^\beta - (\frac{h_i}{2} + \sum_{j=1}^{k+1} h_{i+j})^\beta}{h_{i+k+1}}\right] + \\
&- \frac{K_{i+\frac{1}{2}}}{\Gamma(\beta+1)}(1-\gamma)\left[\frac{(\frac{h_{i+1}}{2} + \sum_{j=2}^{k} h_{i+j})^\beta - (\frac{h_{i+1}}{2} + \sum_{j=2}^{k-1} h_{i+j})^\beta}{h_{i+k}} + \frac{(\frac{h_{i+1}}{2} + \sum_{j=2}^{k} h_{i+j})^\beta - (\frac{h_{i+1}}{2} + \sum_{j=2}^{k+1} h_{i+j})^\beta}{h_{i+k+1}}\right],
\end{aligned}$$

for $2 \le k \le N-i$.

Remark 5.1.1. *When $\beta = 0$, we have $a_{i,i-k} = a_{i,i+k} = 0, \forall k \ge 2$ and the dense structure of A_N collapses into the tridiagonal matrix representing the 1D discrete Laplacian operator, which does not depend on γ anymore. On the contrary, when $\beta = 1$ we still have $a_{i,i-k} = a_{i,i+k} = 0, \forall k \ge 2$ independently of γ, but if $K(x)$ is constant, then A_N becomes a skew-symmetric matrix.*

5.1.4 Uniform mesh

Under the conditions

$$K(x) = K, \qquad \gamma = \frac{1}{2}, \qquad h_i = h \ \forall i, \tag{5.3}$$

with $c = \frac{Kh^{\beta-1}}{2^\beta \Gamma(\beta+1)}$, it holds

$$
\begin{aligned}
a_{i,i-k} =& \frac{c}{2}\left[3(2k+1)^\beta - 3(2k-1)^\beta + (2k-3)^\beta - (2k+3)^\beta\right] \qquad 2 \le k \le i-1,\\
a_{i,i-1} =& \frac{c}{2}\left[3^{\beta+1} - 4 - 5^\beta\right],\\
a_{i,i} =& \frac{c}{2}\left[6 - 2 \cdot 3^\beta\right],\\
a_{i,i+1} =& \frac{c}{2}\left[3^{\beta+1} - 4 - 5^\beta\right],\\
a_{i,i+k} =& \frac{c}{2}\left[3(2k+1)^\beta - 3(2k-1)^\beta + (2k-3)^\beta - (2k+3)^\beta\right] \qquad 2 \le k \le N-i.
\end{aligned}
\tag{5.4}
$$

Therefore, under the assumptions in equation (5.3), matrix A_N in (5.2) is a symmetric Toeplitz matrix and coincides with the coefficient matrix considered in paper Donatelli et al. [2018], where the authors took a FVE discretization of equation (5.1) with RL fractional derivative operators in place of Caputo's. This is indeed not surprising since, from Remark 1.6.2 and by $\phi_i(0) = \phi_i(1) = 0$, $\forall i = 1, \dots, N$, we have

$$
\begin{aligned}
\frac{\mathrm{d}^{1-\beta}\phi_i(x)}{\mathrm{d}_-^R x^{1-\beta}} &= \frac{\mathrm{d}^{1-\beta}\phi_i(x)}{\mathrm{d}_-^C x^{1-\beta}},\\
\frac{\mathrm{d}^{1-\beta}\phi_i(x)}{\mathrm{d}_+^R x^{1-\beta}} &= \frac{\mathrm{d}^{1-\beta}\phi_i(x)}{\mathrm{d}_+^C x^{1-\beta}},
\end{aligned}
$$

which means that the only difference between the FVE discretization of equation (5.1) on uniform meshes and the discretized equation in Donatelli et al. [2018] lies in the right-hand side.

5.1.5 Graded and composite meshes

The discretization of equation (5.1) over uniform meshes yields matrices with a Toeplitz structure, which allows fast matrix-vector product in $\mathrm{O}(N \log N)$, while in case of a generic non-uniform mesh discretization the Toeplitz structure is lost. On the other hand, the solution of the FDE (5.1) may exhibit singularities near the boundaries, therefore uniform grids should be avoided and non-uniform meshes should be preferred.

In order not to completely lose the structure of the coefficient matrices, in the following we consider two mixed approaches of graded mesh near the singularity and uniform mesh where the solution is smooth. This yields matrices with a partial Toeplitz structure that can be exploited to allow a fast matrix-vector product. In the following, we only consider singularities at $x = 0$, however, the approach can be straightforwardly extended to the case of singularities at $x = 1$ or at both boundaries.

Graded meshes We consider the non-uniform grid generated by projection of the uniform mesh through the endomorphism

$$g_{q,\epsilon}(x) = \begin{cases} x^q, & 0 \le x \le \epsilon_1, \\ ax^2 + bx + c, & \epsilon_1 \le x \le \epsilon_1 + \epsilon_2, \\ mx + p, & \epsilon_1 + \epsilon_2 \le x \le 1, \end{cases} \tag{5.5}$$

with $\epsilon = (\epsilon_1, \epsilon_2)$, $0 < \epsilon_1 + \epsilon_2 \le 1$, $\epsilon_2 > 0$ and a, b, c, m, p such that $g_{q,\epsilon} \in \mathrm{C}^1([0,1])$. Proposition 5.1.1 shows that $g_{q,\epsilon}(x)$ is well-defined.

Proposition 5.1.1. *Let* $g_{q,\epsilon}(x)$ *be as in* (5.5)*, with* $\epsilon = (\epsilon_1, \epsilon_2)$*. Then, for* $0 < \epsilon_1 + \epsilon_2 \le 1$ *with* $\epsilon_2 > 0$*, function* $g_{q,\epsilon}(x)$ *is well-defined.*

Proof. The explicit form of coefficients a, b, c, m, q is obtained by solving the following equation

$$\begin{cases} g_{q,\epsilon}(\epsilon_1^-) & = g_{q,\epsilon}(\epsilon_1^+) \\ g_{q,\epsilon}((\epsilon_1+\epsilon_2)^-) & = g_{q,\epsilon}((\epsilon_1+\epsilon_2)^+) \\ g'_{q,\epsilon}(\epsilon_1^-) & = g'_{q,\epsilon}(\epsilon_1^+) \\ g'_{q,\epsilon}((\epsilon_1+\epsilon_2)^-) & = g'_{q,\epsilon}((\epsilon_1+\epsilon_2)^+) \\ g_{q,\epsilon}(1) & = 1 \end{cases} \tag{5.6}$$

where $g_{q,\epsilon}(\xi^\pm) = \lim_{x \to \xi^\pm} g_{q,\epsilon}(x)$. Equation (5.6) can be seen as a linear system with coefficient matrix

$$G = \begin{pmatrix} \epsilon_1^2 & \epsilon_1 & 1 & 0 & 0 \\ (\epsilon_1+\epsilon_2)^2 & \epsilon_1+\epsilon_2 & 1 & -(\epsilon_1+\epsilon_2) & -1 \\ 2\epsilon_1 & 1 & 0 & 0 & 0 \\ 2(\epsilon_1+\epsilon_2) & 1 & 0 & -1 & 0 \\ 0 & 0 & 0 & 1 & 1 \end{pmatrix},$$

whose determinant is $\det(G) = 2\epsilon_1\epsilon_2 - 2\epsilon_2 + \epsilon_2^2$. Finally, since $\det(G) = 0$ if and only if $\epsilon_1 = \frac{2-\epsilon_2}{2}$, and $0 < \epsilon_1 + \epsilon_2 \le 1$ with $\epsilon_2 > 0$, we conclude that $\det(G) \neq 0$ and therefore $g_{q,\epsilon}(x)$ is well-defined. □

In the case where $\epsilon_2 = 0$ then we define

$$g_{q,\epsilon}(x) = \begin{cases} x^q, & 0 \leq x \leq \epsilon_1, \\ mx + p, & \epsilon_1 \leq x \leq 1, \end{cases}$$

with m, p such that $g_{q,\epsilon} \in \mathrm{C}([0,1])$.

In the interval $[0, \epsilon_1]$ we have the singular part of the function which accumulates grid points near the origin, in $[\epsilon_1 + \epsilon_2, 1]$ we have a line, which represents the uniform part of the mesh and gives the partial Toeplitz structure to the coefficient matrix, and in between we have a quadratic function that acts as a smooth connection of length ϵ_2 between the singular part and the uniform mesh, whose only purpose is to increase the smoothness of $g_{q,\epsilon}$. Therefore it is clear that $\epsilon_1 + \epsilon_2$ represents the length of the non-uniform part of the grid over the interval $[0,1]$.

In equation (5.5), where ϵ_1 and ϵ_2 are fixed, the only free parameter is q which we choose to be

$$q = q_\beta = \frac{1+\beta}{1-\beta}, \tag{5.7}$$

as done in Kopteva and Meng [2020], where a Caputo time-fractional derivative is involved and a $L1$ approximation is considered. When N is large, $g_{q,\epsilon}(x)$ could yield a grid that has too short intervals, i.e $h_i \ll 10^{-16}$ for some i. The shortest interval is the first one, and it is obtained by the projection through x^q. Therefore, in case q_β is too large we replace it with q such that $h_1 = 10^{-16}$.

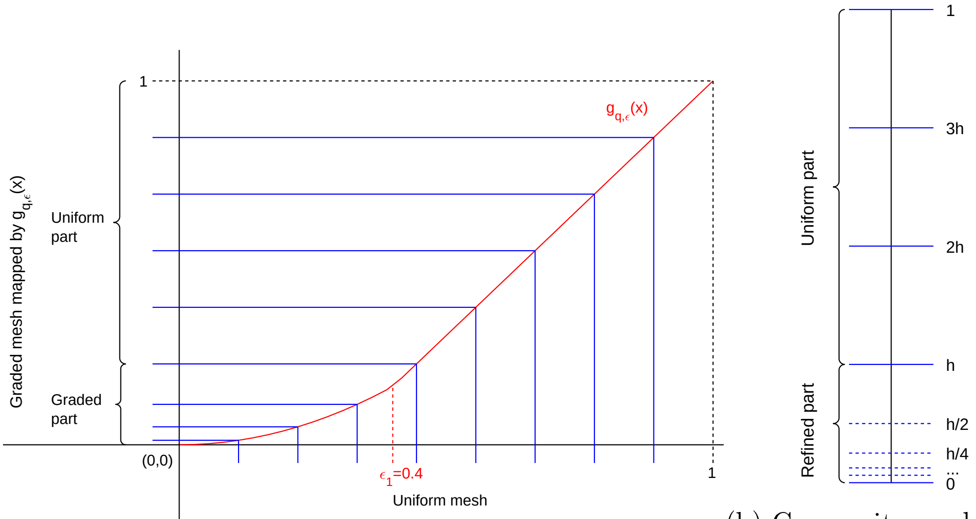
Composite mesh For our numerical comparisons, we will consider also the composite mesh used in Jia and Wang [2015], which proved to be effective in the case where $\beta \approx 1$. Let $N_1, N_2 \in \mathbb{N}$, and consider an uniform mesh with step $h = \frac{1}{N_2+1}$. Then we divide interval $[0,h]$ into $N_1 + 1$ subintervals, whose length from left to right is h_i, $i = 1, \ldots, N_1$ with

$$\begin{cases} h_i = 2^{-N_1}h, & \text{if } i = 1, 2; \\ h_i = 2^{i-2-N_1}h, & \text{if } i = 3, \ldots, N_1. \end{cases}$$

Therefore the grid points are $x_0 = 0$, $x_i = x_{i-1} + h_i$ for $i = 1, \ldots, N_1$, $x_i = x_{i-1} + h$ for $i = N_1 + 1, \ldots, N_1 + N_2$ and $x_{N_1+N_2+1} = 1$.
We choose

$$N_1 = g_G(N), \quad \text{and} \quad N_2 = N - N_1, \tag{5.8}$$

such that the total amount of grid points is N, where $g_G(N) : \mathbb{N} \to \mathbb{N}$, e.g., $g_G(N) = \lfloor \sqrt{N} \rfloor$ or $g_G(N) = \lfloor \log_2 N \rfloor$, with $\lfloor \cdot \rfloor$ being the floor function. This choice



(a) Graded mesh projected by $g_{q,\epsilon}$, with $q = 2$ and $\epsilon = (0.4, 0)$.

(b) Composite mesh with $N_1 = 4$, $N_2 = 3$ and $h = \frac{1}{N_2+1}$.

Figure 5.1. Example of a graded mesh generated by a function and a composite mesh.

slightly differs from the one taken in Jia and Wang [2015], where the authors first fix N_2 and then choose $N_1 \approx \sqrt{N_2}$.

For the sake of clarity, in Figure 5.1 we see two example of meshes. In Figure 5.1a we have the plot of the mapping function $g_{q,\epsilon}(x)$ with $q = 2$ and $\epsilon = (0.4, 0)$, which maps the uniform mesh (on the x axis) into the graded mesh (on the y axis), while in Figure 5.1b we have the plot of the composite mesh with $N_1 = 4$, $N_2 = 3$.

5.2 Spectral properties of the coefficient matrices

In this section we first recall the spectral symbol of the coefficient matrices A_N in presence of uniform meshes already given in Donatelli et al. [2018]. Then, following the idea in Garoni and Serra-Capizzano [2018] (p. 212, Section 10.5.4), we compute the symbol of the coefficient matrices when considering non-uniform grids mapped by functions. In both cases we fix $\gamma = \frac{1}{2}$.

In case of uniform meshes, from equation (5.4), we have that $A_N = cT(p_N^{\beta}(\theta))$

where $c = \frac{Kh^{\beta-1}}{2^{\beta}\Gamma(\beta+1)}$ and

$$p_N^{\beta}(\theta) = \frac{1}{c}a_{1,1} + \frac{2}{c}\sum_{k=1}^{N-1} a_{1,k+1}\cos(k\theta), \tag{5.9}$$

with $a_{1,k}$ in equation (5.4). Moreover, according to Donatelli et al. [2018], the following proposition holds.

Proposition 5.2.1. *For $N \to \infty$, $p_N^{\beta}(\theta)$ in* (5.9) *converges to a positive real-valued even function, say $p^{\beta}(\theta)$, that has a unique zero at $\theta = 0$ of order lower than 2, for every $\beta \in (0,1)$ and such that*

$$\left\{\left(\frac{1}{N+1}\right)^{1-\beta} A_N\right\}_N \sim_{\lambda} \left(\frac{K}{2^{\beta}\Gamma(\beta+1)}p^{\beta}(\theta), [-\pi,\pi]\right).$$

In order to compute the symbol of the coefficient matrix for non-uniform grids mapped by functions, let $N \in \mathbb{N}$ and consider

- a generic grid $\{x_i\}_{i=0}^{N+1} \subset [0,1]$ with $0 = x_0 < x_1 < \cdots < x_{N+1} = 1$;
- the uniform grid $\{\hat{x}_i\}_{i=0}^{N+1}$ with $\hat{x}_i = ih$, $i = 0,\dots,N+1$, and $h = \frac{1}{N+1}$;
- an increasing function $g(x)$ such that $x_i = g(\hat{x}_i)$, $i = 0,\dots,N+1$.

The following theorem, proven by borrowing ideas from Garoni and Serra-Capizzano [2018] (p. 212, Section 10.5.4), holds.

Theorem 5.2.1. *Let $K(x) = K$ and suppose $g : [0,1] \to [0,1]$ is an increasing bijective map in $C^3([0,1])$. Then, if $g'(\hat{x})$ has a finite amount of zeros of limited order, it holds*

$$\left\{\left(\frac{1}{N+1}\right)^{1-\beta} A_N\right\}_N \sim_{\sigma} \left(f_{\beta}(x,\theta), [0,1]\times[-\pi,\pi]\right), \tag{5.10}$$

with

$$f_{\beta}(x,\theta) = \frac{K}{2^{\beta}\Gamma(\beta+1)(g'(x))^{1-\beta}}p^{\beta}(\theta). \tag{5.11}$$

Proof. Let us fix $N \in \mathbb{N}$ and let $\{\hat{x}_i\}_{i=0}^{N+1}$ be the uniform grid with $\hat{x}_i = ih$, $i = 0,\dots,N+1$, and $h = \frac{1}{N+1}$. Then letting $x_i = g(\hat{x}_i)$, $i = 0,\dots,N+1$, from the Taylor expansion of $g(\hat{x}_{i-1})$, $\forall i$, it holds

$$g(\hat{x}_{i-1}) = g(\hat{x}_i) - g'(\hat{x}_i)h + g''(\hat{x}_i)\frac{h^2}{2} + O(h^3), \qquad \forall i = 1,\dots,N,$$

and since $h_i = g(\hat{x}_i) - g(\hat{x}_{i-1})$, we have

$$h_i = g'(\hat{x}_i)h - g''(\hat{x}_i)\frac{h^2}{2} + \mathrm{O}(h^3). \tag{5.12}$$

In the case where $k \in \mathbb{Z}$, from (5.12) we have

$$h_{i+k} = g'(\hat{x}_{i+k})h - g''(\hat{x}_{i+k})\frac{h^2}{2} + \mathrm{O}(h^3), \tag{5.13}$$

and through the Taylor expansions of $g'(\hat{x}_{i+k})$ and $g''(\hat{x}_{i+k})$ we finally obtain

$$h_{i+k} = g'(\hat{x}_i)h + g''(\hat{x}_i)h^2\frac{2k-1}{2} + \mathrm{O}(k^2h^3).$$

Then, with $\gamma = \frac{1}{2}$, $K(x) = K$ and $\tilde{K} = \frac{K}{\Gamma(\beta+1)}$, from equation (5.2) we have

$$\begin{aligned} a_{i,i} &= \tilde{K}\left[\frac{(\frac{h_i}{2})^\beta}{h_i} + \frac{1}{2}\frac{(\frac{h_i}{2})^\beta - (h_{i+1} + \frac{h_i}{2})^\beta}{h_{i+1}} - \frac{1}{2}\frac{(h_i + \frac{h_{i+1}}{2})^\beta - (\frac{h_{i+1}}{2})^\beta}{h_i} + \frac{(\frac{h_{i+1}}{2})^\beta}{h_{i+1}}\right] \\ &= \tilde{K}\left(S_1 + \frac{1}{2}S_2 - \frac{1}{2}S_3 + S_4\right), \end{aligned}$$

where

$$\begin{aligned} S_1 &= \frac{(\frac{h_i}{2})^\beta}{h_i} = \frac{1}{2^\beta}h_i^{\beta-1} = \frac{1}{2^\beta}\left(g_i'h + \mathrm{O}(h^2)\right)^{\beta-1} = \frac{1}{2^\beta(g_i'h)^{1-\beta}}\left(1 + \mathrm{O}(h)\right); \\ S_2 &= \frac{(\frac{h_i}{2})^\beta - (h_{i+1} + \frac{h_i}{2})^\beta}{h_{i+1}} = \frac{(g_i'h)^\beta\left(\left(\frac{1}{2} + \mathrm{O}(h)\right)^\beta - \left(\frac{3}{2} + \mathrm{O}(h)\right)^\beta\right)}{g_i'h(1 + \mathrm{O}(h))} \\ &= \frac{1}{2^\beta(g_i'h)^{1-\beta}}\frac{\left(1 + \mathrm{O}(h)\right)^\beta - 3^\beta\left(1 + \mathrm{O}(h)\right)^\beta}{1 + \mathrm{O}(h)} \\ &= \frac{1}{2^\beta(g_i'h)^{1-\beta}}\left(1 - 3^\beta + \mathrm{O}(h)\right)\left(1 + \mathrm{O}(h)\right) = \frac{1}{2^\beta(g_i'h)^{1-\beta}}\left(1 - 3^\beta + \mathrm{O}(h)\right); \\ S_3 &= \frac{(h_i + \frac{h_{i+1}}{2})^\beta - (\frac{h_{i+1}}{2})^\beta}{h_i} = \frac{(g_i'h)^\beta\left(\left(\frac{3}{2} + \mathrm{O}(h)\right)^\beta - \left(\frac{1}{2} + \mathrm{O}(h)\right)^\beta\right)}{g_i'h(1 + \mathrm{O}(h))} \\ &= \frac{1}{2^\beta(g_i'h)^{1-\beta}}\frac{3^\beta\left(1 + \mathrm{O}(h)\right)^\beta - \left(1 + \mathrm{O}(h)\right)^\beta}{1 + \mathrm{O}(h)} \\ &= \frac{1}{2^\beta(g_i'h)^{1-\beta}}\left(3^\beta - 1 + \mathrm{O}(h)\right)\left(1 + \mathrm{O}(h)\right) = \frac{1}{2^\beta(g_i'h)^{1-\beta}}\left(3^\beta - 1 + \mathrm{O}(h)\right); \\ S_4 &= \frac{(\frac{h_{i+1}}{2})^\beta}{h_{i+1}} = \frac{1}{2^\beta}h_{i+1}^{\beta-1} = \frac{1}{2^\beta}\left(g_i'h + \mathrm{O}(h^2)\right)^{\beta-1} = \frac{1}{2^\beta(g_i'h)^{1-\beta}}\left(1 + \mathrm{O}(h)\right); \end{aligned}$$

with $g_i' = g'(\hat{x}_i)$ and $g_i'' = g''(\hat{x}_i)$.
Assembling $a_{i,i}$ we obtain

$$h^{1-\beta} a_{i,i} = \frac{\tilde{K}}{2^\beta g_i'^{1-\beta}} \left(3 - 3^\beta\right) + \mathrm{O}(h). \tag{5.14}$$

With the same approach we obtain

$$h^{1-\beta} a_{i,i\pm 1} = \frac{\tilde{K}}{2^\beta g_i'^{1-\beta}} \left[3^{\beta+1} - 4 - 5^\beta\right] + \mathrm{O}(h); \tag{5.15}$$

$$h^{1-\beta} a_{i,i\pm k} = \frac{\tilde{K}}{2^\beta g_i'^{1-\beta}} \left[3(2k+1)^\beta - 3(2k-1)^\beta + (2k-3)^\beta - (2k+3)^\beta\right] + \mathrm{O}(kh), \tag{5.16}$$

with $1 < k \leq N^q,\ 0 < q < 1$, such that $kh \to 0$ as $k \to \infty$.

In the case where $N^q < k \leq N$ the approximation yields a large error, therefore we prove that $a_{i,i+k} = \mathrm{o}(h)$. Let $r = \sum_{j=1}^{k} h_{i+j}$, then $0 < r < 1\ \forall k$ and

- if $k = \mathrm{O}(N^q) = \mathrm{O}(h^{-q})$, we have

$$r = \sum_{j=1}^{\mathrm{O}(N^q)} h_{i+j} = \sum_{j=1}^{\mathrm{O}(N^q)} \left(g_i' h + \mathrm{O}(jh^2)\right) = g_i' \mathrm{O}(N^q) h + \mathrm{O}(N^{2q} h^2) = g_i' \mathrm{O}(h^{1-q}), \tag{5.17}$$

- while if $k = \mathrm{O}(N) = \mathrm{O}(h^{-1})$, r is a constant independent of N.

By collecting r, in $a_{i,i+k}$ we have terms of the form $(1+\tilde{h})^\beta$, with $\tilde{h} = \frac{h_i}{2r}, \frac{h_i}{2r} - \frac{h_{i+k}}{r}, \ldots$. In order to use the Taylor expansion of $(1+\tilde{h})^\beta$ we first need to prove that $\tilde{h} \to 0$ as $N \to \infty$. We divide the analysis in two cases:

1) if $g'(\hat{x}) \neq 0$ in $[0,1]$, then from equations (5.12) and (5.17) we have

$$\frac{h_{i+k}}{r} = \frac{g_{i+k}' h + \mathrm{O}(h^2)}{g_i' \mathrm{O}(h^{1-q})} = \mathrm{O}(h^q),$$

which tends to zero as $N \to \infty$ for any $0 < q < 1$.

2) if $g'(\hat{x})$ vanishes in $[0,1]$, the worst possible scenario happens when $h_i \gg h_{i+k}$. Without restrictions to the general case we assume that $g'(\hat{x})$ has a zero of order t at $\hat{x}_0 = 0$, hence $g(\hat{x}) \approx \hat{x}^{t+1}$ when $\hat{x} \to 0$. Then by

considering $k = -N^q$ and $i = N^q + 1$, such that $h_{N^q+1} = h_i \gg h_{i+k} = h_1$, we have

$$\frac{h_{N^q+1}}{\sum_{j=1}^{N^q} h_{N^q+1-j}} = \frac{x_{N^q+1} - x_{N^q}}{h_1 + h_2 + \ldots + h_{N^q}} = \frac{g(\hat{x}_{N^q+1}) - g(\hat{x}_{N^q})}{g(\hat{x}_{N^q}) - g(\hat{x}_0)} = \frac{\left(\frac{N^q+1}{N}\right)^{t+1} - \left(\frac{N^q}{N}\right)^{t+1}}{\left(\frac{N^q}{N}\right)^{t+1}}$$
$$= \left(1 + \frac{1}{N^q}\right)^{t+1} - 1 = \mathrm{O}(h^{qt}),$$

which tends to zero as $N \to \infty$ for any $0 < q < 1$.

Now, let $N^q < k \leq N$ and approximate the coefficient $a_{i,i+k}$ as follows:

$$\begin{aligned} a_{i,i+k} =& \frac{\tilde{K}}{2}\Bigg[\frac{r^\beta(1+\frac{h_i}{2r})^\beta - r^\beta(1+\frac{h_i}{2r}-\frac{h_{i+k}}{r})^\beta}{h_{i+k}} + \frac{r^\beta(1+\frac{h_i}{2r})^\beta - r^\beta(1+\frac{h_i}{2r}+\frac{h_{i+k+1}}{r})^\beta}{h_{i+k+1}} + \\ & - \frac{r^\beta(1-\frac{h_{i+1}}{2r})^\beta - r^\beta(1-\frac{h_{i+1}}{2r}-\frac{h_{i+k}}{r})^\beta}{h_{i+k}} - \frac{r^\beta(1-\frac{h_{i+1}}{2r})^\beta - r^\beta(1-\frac{h_{i+1}}{2r}+\frac{h_{i+k}}{r})^\beta}{h_{i+k+1}} \Bigg] \\ =& \frac{\tilde{K} r^\beta}{2 h_{i+k} h_{i+k+1}} \Bigg[h_{i+k+1} \Big((1+\frac{h_i}{2r})^\beta - (1+\frac{h_i}{2r}-\frac{h_{i+k}}{r})^\beta - (1-\frac{h_{i+1}}{2r})^\beta + (1-\frac{h_{i+1}}{2r}-\frac{h_{i+k}}{r})^\beta \Big) + \\ & + h_{i+k} \Big((1+\frac{h_i}{2r})^\beta - (1+\frac{h_i}{2r}+\frac{h_{i+k+1}}{r})^\beta - (1-\frac{h_{i+1}}{2r})^\beta + (1-\frac{h_{i+1}}{2r}+\frac{h_{i+k}}{r})^\beta \Big) \Bigg]. \end{aligned}$$

By replacing each $(1+\tilde{h})^\beta$ with its Taylor expansion

$$(1+\tilde{h})^\beta = 1 + \beta\tilde{h} + \frac{\beta(\beta-1)}{2}\tilde{h}^2 + \mathrm{O}(\tilde{h}^3),$$

we observe an exact cancellation of the terms of degree 0 and 1 inside the square brackets in $a_{i,i+k}$. The exact cancellation happens even for the term of degree 2 but it is harder to see, therefore we report the computations below.

$$\begin{aligned} a_{i,i+k} =& \frac{\tilde{K} r^\beta \beta(\beta-1)}{4 h_{i+k} h_{i+k+1}} \Bigg[h_{i+k+1} \Big(\frac{h_i^2}{4r^2} + \mathrm{O}(h_i^3) - (\frac{h_i}{2r} - \frac{h_{i+k}}{r})^2 + \mathrm{O}((h_i - h_{i+k})^3) - \frac{h_{i+1}^2}{4r^2} + \mathrm{O}(h_{i+1}^3) + \\ & + (\frac{h_{i+1}}{2r} + \frac{h_{i+k}}{r})^2 + \mathrm{O}((h_{i+1} + h_{i+k})^3)) \Big) + \\ & + h_{i+k} \Big(\frac{h_i^2}{4r^2} + \mathrm{O}(h_i^3) - (\frac{h_i}{2r} + \frac{h_{i+k+1}}{r})^2 + \mathrm{O}((h_i + h_{i+k+1})^3) - \frac{h_{i+1}^2}{4r^2} + \mathrm{O}(h_{i+1}^3) + \\ & + (-\frac{h_{i+1}}{2r} + \frac{h_{i+k+1}}{r})^2 + \mathrm{O}((-h_{i+1} + h_{i+k+1})^3)) \Big) \Bigg] \\ =& \frac{\tilde{K} r^{\beta-2} \beta(\beta-1)}{4 h_{i+k} h_{i+k+1}} \big(h_{i+k+1}(h_i h_{i+k} + h_{i+1} h_{i+k}) - h_{i+k}(h_i h_{i+k+1} + h_{i+1} h_{i+k+1}) + \mathrm{O}(h^4) \big) \\ =& \frac{r^{\beta-2}}{{g'_{i+k}}^2} \mathrm{O}(h^2). \end{aligned}$$

In the case where

- $k = \mathrm{O}(N^q)$, from equation (5.17) we have

$$h^{1-\beta} a_{i,i+k} = \mathrm{O}(h^{1-\beta+2+(1-q)(\beta-2)}) = \mathrm{O}(h^{1-\beta+2+\beta-2+2q-\beta q}) = \mathrm{o}(h^{1+q}), \tag{5.18}$$

- $k = \mathrm{O}(N)$, since r has a constant value, we have

$$h^{1-\beta} a_{i,i+k} = \mathrm{O}(h^{1-\beta+2}) = \mathrm{o}(h^2). \tag{5.19}$$

Let $B_{N,M}$ be a diagonal-times-Toeplitz banded matrix of the form

$$B_{N,M} = D_N(d(x)) T_N(p_M^{\beta}(\theta)),$$

with $d(x) = \frac{\tilde{K}}{2^{\beta}(g'(x))^{1-\beta}}$ and $p_M^{\beta}(\theta)$ being the symbol in equation (5.9). From Proposition 1.3.1 we have

$$\{\{B_{N,M}\}_N\}_M \sim_{\sigma} \left(d(x) p_M^{\beta}(\theta), [0,1] \times [-\pi, \pi]\right). \tag{5.20}$$

We now prove that $\{\{B_{N,M}\}_N\}_M$ is an a.c.s for $\{A_N\}_N$.

Suppose that $g'(\tilde{x}) \neq 0$ in $[0,1]$, then, by choosing $M = N^q$, from equations (5.14), (5.15), (5.16) and (5.18) we have that matrix $h^{1-\beta} A_N - B_{N,N^q}$ is a "symmetric Toeplitz" matrix, whose first row is

$$\mathrm{O}(h) \quad \underbrace{\mathrm{O}(h) \quad \mathrm{O}(2h) \quad \mathrm{O}(3h) \quad \cdots \quad \mathrm{O}(N^q h)}_{N^q \text{ coefficients}} \quad \underbrace{\mathrm{o}(h^{1+q}) \quad \cdots \quad \mathrm{o}(h^{1+q})}_{N - N^q - 1 \text{ coefficients}} . \tag{5.21}$$

Note that the "symmetric Toeplitz" structure holds only while keeping $\mathrm{O}(\cdot)$ and $\mathrm{o}(\cdot)$. When we replace $\mathrm{O}(\cdot)$ and $\mathrm{o}(\cdot)$ with the exact values the structure could be lost.

Thanks to the structure we have

$$\begin{aligned}
\left\|h^{1-\beta} A_N - B_{N,N^q}\right\|_1 &\leq 2\left(\mathrm{O}(h) + \sum_{k=1}^{N^q} \mathrm{O}(kh) + \mathrm{o}(h^{1+q})(N - N^q - 1)\right) \\
&= \mathrm{O}(h) + \mathrm{O}(N^{2q} h) + \mathrm{o}((N - N^q - 1) h^{1+q}) \\
&= \mathrm{O}(h) + \mathrm{O}(h^{1-2q}) + \mathrm{o}(h^q - h - h^{1+q}), \\
\left\|h^{1-\beta} A_N - B_{N,N^q}\right\|_\infty &\leq \mathrm{O}(h) + \mathrm{O}(h^{1-2q}) + \mathrm{o}(h^q - h - h^{1+q}),
\end{aligned}$$

and through the Hölder inequality,

$$\begin{aligned}
\left\|h^{1-\beta} A_N - B_{N,N^q}\right\|_2 &\leq \sqrt{\left\|h^{1-\beta} A_N - B_{N,N^q}\right\|_1 \left\|h^{1-\beta} A_N - B_{N,N^q}\right\|_\infty} \\
&\leq \mathrm{O}(h) + \mathrm{O}(h^{1-2q}) + \mathrm{o}(h^q - h - h^{1+q}),
\end{aligned} \tag{5.22}$$

which tends to zero as $N \to \infty$ if $0 < q < \frac{1}{2}$. From Definition 1.3.5 it follows that $\{\{B_{N,M}\}_N\}_M$ is an a.c.s for $\{h^{1-\beta}A_N\}_N$, and from equation (5.20) and Theorem 1.3.3 we have the thesis, since from Proposition 5.2.1 it holds that $p_M^{\beta}(\theta)$ converges to $p^{\beta}(\theta)$.

Suppose now that there exist $\tilde{x}^{(1)}, \dots, \tilde{x}^{(s)} \in [0,1]$ such that $g'(\tilde{x}^{(k)}) = 0,\ \forall k$ and consider the intervals $B(\tilde{x}^{(k)}, \frac{1}{M}) = \{\tilde{x} \in [0,1]\ :\ \left|\tilde{x} - \tilde{x}^{(k)}\right| < \frac{1}{M}\}$. The function $g'(x)$ is continuous and strictly positive on $[0,1] \setminus I_M\ \forall M$, where

$$I_M = \bigcup_{k=1}^{s} B\Big(\tilde{x}^{(k)}, \frac{1}{M}\Big),$$

therefore we write $h^{1-\beta}A_N - B_{N,M} = N_{N,M} + R_{N,M}$, where matrices $N_{N,M}, R_{N,M}$ have small-norm and low-rank, respectively.
If we define $\tilde{a}_{i,j} = \left(h^{1-\beta}A_N - B_{N,M}\right)_{i,j}$, the matrix $R_{N,M}$ is sparse and its entries are

$$r_{i,j} = \begin{cases} \tilde{a}_{i,j}, & \text{if } \hat{x}_i \in I_M \text{ or } \hat{x}_j \in I_M \\ 0, & \text{otherwise.} \end{cases}$$

Hence, given the equispaced grid $\hat{x}_i = ih,\ i = 1, \dots, N$, then

$$\operatorname{rank}\big(R_{N,M}\big) \le 2|I_M| \le 2s\Big(\frac{2/M}{h} + 1\Big) = 2s\left(\frac{2}{M} + \frac{2}{NM} + \frac{1}{N}\right)N.$$

Moreover, as $R_{N,M}$ contains every coefficient $\tilde{a}_{i,j}$ which has a g_i', with $i \in I_M$ in the denominator, the matrix $N_{N,M}$ can be approximated as in (5.22). In conclusion, for each M there exists N_M such that for $N > N_M$, rank$(R_{N,M}) \le \frac{3sN}{M}$ and $\left\|N_{N,M}\right\|_\infty \le \frac{1}{M}$, which, from Definition 1.3.5, means that $\{\{B_{N,M}\}_N\}_M$ is an a.c.s of $\{A_N\}_N$ and the thesis again follows from equation (5.20) and Theorem 1.3.3. □

By combining Theorem 5.2.1 with **GLT1** and **GLT5**, the following theorem holds.

Theorem 5.2.2. *Let $K(x) = K$ and suppose $g : [0,1] \to [0,1]$ is an increasing bijective map in $C^3([0,1])$. Then, if $g'(\hat{x}) > 0$, it holds*

$$\left\{\left(\frac{1}{N+1}\right)^{1-\beta} A_N\right\}_N \sim_\lambda \big(f_\beta(x,\theta), [0,1] \times [-\pi,\pi]\big), \tag{5.23}$$

with $f_\beta(x,\theta)$ defined as in equation (5.11).

Proof. The thesis is proven by combining Theorem 5.2.1 with **GLT1** and **GLT5**. Therefore, we only need to show that **GLT5** holds for the matrix sequence in equation (5.10). We recall that **GLT5** consists in proving that

$$\lim_{N\to\infty}\frac{\left\|h^{1-\beta}A_N-h^{1-\beta}A_N^{\mathrm{H}}\right\|_{\mathrm{tr}}}{N}=0,$$

with $h=\frac{1}{N+1}$.
Let us now denote $\tilde{a}_{i,j}=h^{1-\beta}\left(A_N-A_N^{\mathrm{H}}\right)_{ij}$. Then, since equations (5.18) and (5.19) hold also for negative values of k, we have

$$\tilde{a}_{i,i+k}=h^{1-\beta}(a_{i,i+k}-a_{i+k,i})=h^{1-\beta}(a_{i,i+k}-a_{j,j-k})=O(h^{1+q}),$$

with $j=i+k$, for $N^q<|k|\leq N$. From equation (5.16) we have,

$$\begin{aligned}
\tilde{a}_{i,i+k}=&h^{1-\beta}(a_{i,i+k}-a_{j,j-k})\\
=&\frac{\tilde{K}}{2^\beta g_i'^{1-\beta}}\left[3(2k+1)^\beta-3(2k-1)^\beta+(2k-3)^\beta-(2k+3)^\beta\right]+\mathrm{O}(kh)+\\
&-\left(\frac{\tilde{K}}{2^\beta g_j'^{1-\beta}}\left[3(2k+1)^\beta-3(2k-1)^\beta+(2k-3)^\beta-(2k+3)^\beta\right]+\mathrm{O}(kh)\right)\\
=&\left(\frac{1}{g_i'^{1-\beta}}-\frac{1}{g_{i+k}'^{1-\beta}}\right)\frac{\tilde{K}}{2^\beta}\left[3(2k+1)^\beta-3(2k-1)^\beta+(2k-3)^\beta-(2k+3)^\beta\right]+\mathrm{O}(kh)\\
=&\,O(kh),
\end{aligned}$$

for $0\leq|k|\leq N^q$. Therefore, a similar reasoning to the one made for $h^{1-\beta}A_N-B_{N,N^q}$ in equation (5.21) can be done also for $h^{1-\beta}\left(A_N-A_N^{\mathrm{H}}\right)$. Finally, from Hölder inequality it follows

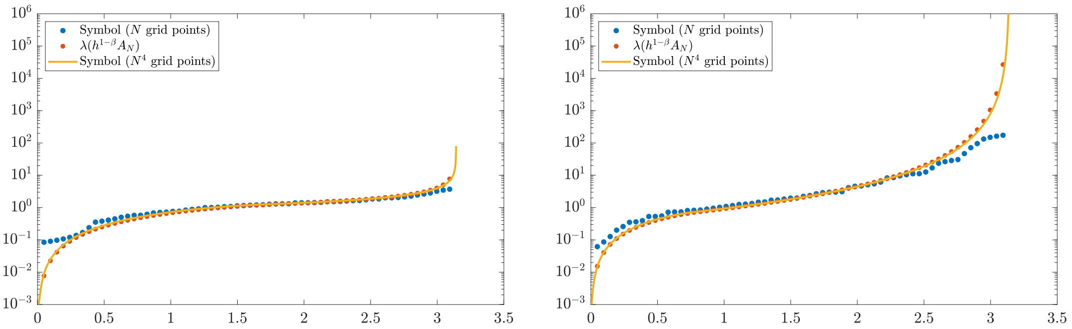
$$\frac{h^{1-\beta}\left\|A_N-A_N^{\mathrm{H}}\right\|_{\mathrm{tr}}}{N}\leq\frac{Nh^{1-\beta}\left\|A_N-A_N^{\mathrm{H}}\right\|_2}{N}\leq\mathrm{O}(h)+\mathrm{O}(h^{1-2q})+\mathrm{o}(h^q-h-h^{1+q}),$$

which tends to 0 as $N\to\infty$ and this concludes the proof. □

Remark 5.2.1. *Combining Theorem 5.2.2 and Proposition 5.2.1, we have that the symbol $f_\beta(x,\theta)$ of A_N has a unique zero at $\theta=0$ of order lower than* 2.

The constraint $g'(x)>0$ is taken to facilitate the proof as, under this hypothesis, we could show that **GLT5** holds, but numerical results seem to indicate that the symbol $f_\beta(x,\theta)$ still approximates the eigenvalue distribution of $h^{1-\beta}A_N$ even without such constraint.

In view of Remark 1.3.3, in Figure 5.2 we give a graphical interpretation of Theorem 5.2.2. From Proposition 5.2.1 it follows that $f_\beta(x,\theta)$ is even with respect



(a) $g(x) = x^2$. (b) $g(x) = x^4$.

Figure 5.2. Plot of the eigenvalues of $h^{1-\beta}A_N$ with $\beta = 0.5$, $N = 2^6$ and the non-uniform mesh mapped by $g(x)$.

to θ, hence we consider $J = [0,1] \times [0,\pi]$ to obtain a more accurate plot. Fixed $N = 2^6$ and $\beta = 0.5$, Figures 5.2a and 5.2b compare the sorted eigenvalues of $h^{1-\beta}A_N$ with the sorted uniform sampling of the symbol $f_\beta(x,\theta)$ over the meshes mapped by $g(x) = x^q$, $q = 2, 4$ over both

(i) $\{\hat{x}_i, \theta_j\}_{i,j=1}^{\sqrt{N}}$ with $\hat{x}_i = i\frac{1}{\sqrt{N}}$, $\theta_j = j\frac{\pi}{\sqrt{N}+1}$,

(ii) $\{\hat{x}_i, \theta_j\}_{i,j=1}^{N^2}$ with $\hat{x}_i = i\frac{1}{N^2}$, $\theta_j = j\frac{\pi}{N^2+1}$.

In both Figures 5.2a and 5.2b we observe a similar shape between the eigenvalues of $h^{1-\beta}A_N$ and the sampling of the symbol over the grid with N points, with a lack of overlapping at initial and final grid points. This discrepancy is immediately overcome by making a comparison between the eigenvalues of $h^{1-\beta}A_N$ and the sampling of the symbol $f_\beta(x,\theta)$ over the much finer mesh (N^4 points).

5.3 Multigrid methods

In order to discuss the convergence analysis of V-cycle applied to (5.2), we consider the mesh to be uniform, the diffusion coefficient K to be constant, $\gamma = \frac{1}{2}$, such that the resulting coefficient matrix is positive definite, and we use weighted Jacobi as smoother. Under these assumptions, and because of the Toeplitz structure of the considered matrices, the weighted Jacobi is well-known to satisfy the smoothing property for positive definite matrices, whenever it is convergent (see Remark 1.4.1). Moreover, according to Proposition 5.2.1, in case of uniform meshes the symbol $f_\beta(x,\theta)$ of A_N vanishes with order lower than 2 at $\theta = 0$ and the approximation property holds true with the same standard projectors as in the case of the Laplacian (see Moghaderi et al. [2017]).

In the case of the considered non-uniform meshes, we propose a V-cycle that still keeps the standard projectors and whose hierarchy is built through the geometric approach with amount of levels given by $\mathrm{lvl} = \log_2(N+1) - 1$ (smallest possible size 3×3). Note that, in order to make the V-cycle properly working, the linear systems must be scaled such that the right-hand side does not contain any grid dependent scaling factor. Therefore, we multiply both members of $A_N x = b$ by the diagonal matrix $H_N = \mathrm{diag}_{i=1,\ldots,N}(\frac{1}{h_i})$.
Concerning the construction of the hierarchy, in the case of a grid mapped by function $g(x) : [0,1] \to [0,1]$, the grid on the ℓ-th level is, as usual, the projection of an uniform mesh with step length $\hat{h}_\ell = \frac{N - N \mathrm{mod}(2)}{2^{\ell-1}}$ through function $g(x)$. When composite meshes are employed, on the contrary, the choice of the grid on the coarser level is non-trivial. We consider a down-sampling of the grid points on the finest level. Let $N \in \mathbb{N}$, then the grid points on the finest ($\ell = 0$) level are $x_i,\ i = 0, \ldots, N+1$ and on the ℓ-th level, with $\ell > 0$, the grid points are

$$\begin{cases} x_j^{(\ell)} = & 0, \qquad j = 0; \\ x_j^{(\ell)} = & x_{a+(j-1)2^\ell}, \qquad 1 < j < N^{(\ell)}; \\ x_j^{(\ell)} = & 1, \qquad j = N^{(\ell)} + 1; \end{cases} \tag{5.24}$$

with $N^{(\ell)} = \lfloor \frac{N}{2^\ell} \rfloor$ and $a = \lfloor 2^{\sqrt{\ell}} \rfloor$. Note that the inner points of the grid at each level are $N^{(\ell)}$ and, therefore, the coefficient matrix at the ℓ-th level has size $N^{(\ell)} \times N^{(\ell)}$. The choice of $x_1^{(\ell)} = x_a$, which slowly moves to the right when increasing ℓ, is taken through trial and error aiming to minimize the iterations to tolerance of our solver when $\beta \approx 1$. This choice seems to yield better results than the natural one, which consists in taking a grid point every two with respect to the grid on the finer level, i.e., with $a = 2^l$. On the other hand, as shown in Section 5.4, when $\beta \ll 1$, the composite mesh does not seem to allow low approximation error, therefore we are not interested in reducing the iterations of our solver in this case.

Regarding the smoother, at each iteration of V-cycle one iteration of weighted Jacobi as pre- and post-smoother is performed. The weight ω is estimated through the approach introduced in Chapter 2 and numerical results in Section 5.4 show the suitability of the choice $\tilde{o}(x) = \sqrt{1-x^2} + 0.475x - 0.475$ and $I = \left[-\frac{1239}{1961}, 1\right] \approx [-0.63, 1]$ such that $\tilde{o}(x) \geq 0$. Note that our choice slightly differs from the one proposed in Chapter 2, where we dealt with a different model.

Remark 5.3.1. *From Remark 5.1.1, in the case where $\beta = 0$, A_N becomes the positive definite one-dimensional discrete Laplacian, therefore we expect multigrid to perform well for $\beta \approx 0$ even in the anisotropic cases where $\gamma \approx 0$ or $\gamma \approx 1$. When $\beta = 1$, A_N becomes skew-symmetric, hence we do not expect multigrid to perform well in the cases where $\beta \approx 1$ and $\gamma \approx 0$ or $\gamma \approx 1$.*

5.4 Numerical results

In this section we first compare few (graded and composite) grids by reporting the reconstruction error and the convergence order (Test 1), then we check the performances of our multigrid applied to equation (5.1) (Tests 2–4) varying the grid, γ and β. Precisely, aiming at increasing its robustness, we use the MGM described in Section 5.3 as preconditioner for GMRES by performing one iteration of V-cycle applied directly to the coefficient matrix. Throughout, we denote our solver by $\mathscr{P}$-GMRES.

Our numerical tests have been run on a server with AMD 3600 6-core (4.20 GHz) processor and 64 GB (3600 MHz) RAM and Matlab 2020b. In all our tests we consider equation (5.1) with $f(x) = \frac{(1-\gamma)(1-\beta)}{\Gamma(\beta)x(1-x)^{1-\beta}}$, $K(x) = 1$ with $\gamma \in [0,1]$ and $\beta \in (0,1)$, whose exact solution, according to Jia and Wang [2015], is $u(x) = x^{1-\beta}$.

For all involved iterative methods the initial guess $x^{(0)}$ is the null vector, the maximum amount of iterations is 100 and the stopping criterion is

$$\frac{\left\|Ax^{(k)} - b\right\|_2}{\|b\|_2} < \text{tol},$$

where the tolerance is tol$= 10^{-7}$ and $x^{(k)}$ is the unknown at the k-th iteration.

Test 1 We first test the quality of q_β, defined in (5.7), the grading parameter of function $g_{q,\epsilon}(x)$ which generates the graded mesh discussed in Section 5.1.5. We set $N = 2^{10}$ and compare q_β, in terms of reconstruction error, with the numerically computed optimal value q_{opt}, which is the one that minimizes the infinity norm of the error and is found through trial and error.

Table 5.1 shows the optimal value q_{opt} and the infinity norm errors e^{opt} and e^{β} yield when discretizing equation (5.1) over the grid $g_{q,\epsilon}(x)$ with $q = q_{\text{opt}}$ and $q = q_\beta$, respectively, for various values of $\epsilon = (\epsilon_1, \epsilon_2)$. Note that q_β is not shown in the

γ	β	$\epsilon^{(1)}=(0.1,0.05)$			$\epsilon^{(2)}=(0.2,0.05)$			$\epsilon^{(3)}=(0.25,0)$			$\epsilon^{(4)}=(0.45,0.05)$			$\epsilon^{(5)}=(0.5,0)$			$\epsilon^{(6)}=(1,0)$		
		q_{opt}	e^{opt}	e^{β}	q_{opt}	e^{opt}	e^{β}	q_{opt}	e^{opt}	e^{β}	q_{opt}	e^{opt}	e^{β}	q_{opt}	e^{opt}	e^{β}	q_{opt}	e^{opt}	e^{β}
	0.2	1.9	1.2e-5	3.3e-5	1.8	1.3e-5	3.3e-5	1.8	1.3e-5	3.3e-5	1.7	1.5e-5	3.3e-5	1.7	1.5e-5	3.3e-5	1.7	1.8e-5	3.3e-5
0.3	0.5	2.7	5.2e-5	5.7e-5	3	2.6e-5	2.6e-5	2.9	3.1e-5	3.2e-5	3.1	2.1e-5	2.2e-5	3	2.2e-5	2.2e-5	2.8	3.5e-5	3.7e-5
	0.8	4.4	1.9e-3	6.3e-3	5.3	7.1e-4	7.1e-4	4.6	1.6e-3	5.7e-3	5.3	7.1e-4	7.1e-4	5.3	7.1e-4	7.1e-4	5.3	7.1e-4	7.1e-4
	0.2	1.7	9.0e-6	2.1e-5	1.7	9.6e-6	2.1e-5	1.7	9.7e-6	2.1e-5	1.7	1.1e-5	2.1e-5	1.7	1.1e-5	2.1e-5	1.6	1.3e-5	2.1e-5
0.5	0.5	2.9	1.3e-5	1.4e-5	3	1.1e-5	1.1e-5	3	1.2e-5	1.2e-5	2.9	1.5e-5	1.5e-5	2.9	1.5e-5	1.5e-5	2.7	2.4e-5	2.6e-5
	0.8	4.3	1.0e-3	4.2e-3	5.3	3.5e-4	3.7e-4	4.6	8.0e-4	2.9e-3	5.3	3.4e-4	3.4e-4	5.3	3.4e-4	3.4e-4	5.3	3.4e-4	3.4e-4
	0.2	1.7	5.6e-6	1.3e-5	1.7	6.0e-6	1.3e-5	1.7	6.1e-6	1.3e-5	1.7	6.9e-6	1.3e-5	1.7	7.0e-6	1.3e-5	1.6	8.1e-6	1.3e-5
0.7	0.5	2.6	1.4e-5	1.8e-5	2.7	8.2e-6	1.0e-5	2.7	9.8e-6	1.5e-5	2.7	8.5e-6	8.7e-6	2.7	8.7e-6	9.0e-6	2.6	1.3e-5	1.5e-5
	0.8	3.8	2.1e-4	1.9e-3	4.1	4.4e-5	4.6e-4	3.7	2.5e-4	5.3e-3	4.2	3.7e-5	5.1e-4	4.2	3.7e-5	5.5e-4	4.2	3.7e-5	5.2e-4

Table 5.1. Comparison between the numerically computed optimal value q_{opt} and q_β, in terms of infinity norm reconstruction error varying γ, β and ϵ.

table since it only depends on β, i.e., $q_{0.2} = 1.5$, $q_{0.5} = 3$. Since $q_{0.8} = \frac{1+0.8}{1-0.8} = 9$ yields a step length much below machine precision for $N = 2^{10}$ ($h_1 \approx 10^{-28}$), we set $q_{0.8} = 5.3$ such that $h_1 = 10^{-16}$, according to the discussion in Section 5.1.5.

When $\beta = 0.2$, in Table 5.1 we observe an increase in the error when choosing q_β with respect to q_{opt}, i.e., $e^\beta \approx 2e^{\text{opt}}$ independently of the choice of γ and of ϵ. Moreover, the interval lengths ϵ_1, ϵ_2 does not seem to affect the error in the case of q_β. This also means that the use of a smooth function to generate the grid does not decrease the error in comparison with a non-smooth function, therefore smaller values of ϵ_1 and ϵ_2 can be chosen, in order to speed up the matrix-vector product. When $\beta = 0.5$, the difference between e^β and e^{opt} is almost negligible and both do not vary that much with γ and ϵ and again the choice of ϵ_1, ϵ_2 is not so crucial. When $\beta = 0.8$, unlike the previous two cases, if the non-linear part of function $g_{q,\epsilon}(x)$ is too short, e.g., when considering $\epsilon^{(1)}$, the error increases greatly. This happens because more grid points are needed near $x = 0$ to deal with the singularity of the solution, and by increasing ϵ_1, $g_{q,\epsilon}(x)$ projects more grid points near $x = 0$.

Moreover, by comparing $\epsilon^{(2)}$ with $\epsilon^{(3)}$ we note that when $g_{q,\epsilon}(x)$ is smooth, i.e., when considering $\epsilon = \epsilon^{(2)}$, the error is lower than in the case of a non-smooth function, especially when $\beta \geq 0.5$, even if the length of the interval where $g_{q,\epsilon}(x)$ is non-linear does not change. Therefore, a smooth function $g_{q,\epsilon}(x)$ is recommended when $\beta \approx 1$.

Note that, in any of the tested case, a full non-uniform mesh does not seem to be necessary, since the lowest error is always reached for mixed meshes.

Test 2 We now fix $\gamma = 0.5$ and provide the numerical results to show the robustness of $\mathscr{P}$-GMRES by reporting the iterations to tolerance (It), needed for solving equation (5.1) discretized over the grids reported in Section 5.1.5, and the infinity norm numerical error e^{∞}. The dash (—) means that the solver exceeded the maximum amount of iterations (It=100).
We recall that, as defined in (5.8), $g_G(N)$ is the function which defines how many points are near $x = 0$ in the composite mesh.

Table 5.2 shows It, e^{∞} and the convergence order computed as the ratio between the infinity-norm error over two grids, with N and $2N$ points, in $\log_2$ scale.

When considering $\epsilon^{(6)}$, we note that the convergence order seems to be $1 + \beta$, which is the theoretical convergence rate obtained in Kopteva and Meng [2020]. Moreover, as expected we note that in almost any of the considered cases, the grid mapped by $g_{q,\epsilon}(x)$ allows to reach the lowest error with the same N.
According to Table 5.2, when $\beta = 0.2$, we note that $\epsilon^{(1)}$ yields the same error as $\epsilon^{(6)}$. When $\beta = 0.5$, $g_{q,\epsilon}(x)$ with $\epsilon = \epsilon^{(1)}$ seems to have a too short non-linear part, since for $N < 2^{10}$ $\epsilon^{(1)}$ yields a larger error in comparison with other choices of ϵ. When $\beta = 0.8$ and N is small, if the non-linear part of $g_{q,\epsilon}(x)$ is too short, e.g., $\epsilon^{(1)}$ or $\epsilon^{(2)}$, the grid projected by $g_{q,\epsilon}(x)$ is not enough smooth due to q being large and this compromises the functioning of the multigrid. When increasing N, instead, the larger amount of points allows a smoother projected grid and the performance of multigrid improves. This seems to indicate that it would be more effective to use a $\epsilon = (\epsilon_1, \epsilon_2)$ that depends on N and β. In particular, ϵ_1, ϵ_2 have to be large when N is small and/or $\beta \approx 1$ and ϵ_1, ϵ_2 can be small when N is large and/or $\beta \approx 0$.

Furthermore, considering a composite mesh, the parameter a in equation (5.24), which is needed to build the multigrid hierachy, could be chosen differently improving the convergence of our solver when $\beta \leq 0.5$. Nevertheless, we do not further investigate into this, since the composite mesh is not suited for that cases yielding larger errors than the mesh mapped by $g_{q,\epsilon}(x)$.

Test 3 We now fix $\gamma = 0.5$, $\beta = 0.9$ and compare $\mathscr{P}$-GMRES with the Preconditioned Fast Conjugate Gradient Squared (PFCGS) introduced in Jia and Wang [2015], which consists in a T.Chan's block circulant preconditioner.

Table 5.3 shows e^{∞} and It of PFCGS and It of $\mathscr{P}$-GMRES in case of the composite

β	$N+1$	$g_G(N)=\lfloor\sqrt{N}\rfloor$			$g_G(N)=\lfloor\log_2 N\rfloor$			$\epsilon^{(1)}=(0.1,0.05)$			$\epsilon^{(2)}=(0.2,0.05)$			$\epsilon^{(4)}=(0.45,0.05)$			$\epsilon^{(6)}=(1,0)$		
		It	e^∞	ord	It	e^∞	ord	It	e^∞	ord	It	e^∞	ord	It	e^∞	ord	It	e^∞	ord
0.2	2^4	9	2.9e-3		9	2.9e-3		8	3.3e-3		8	3.0e-3		10	3.0e-3		10	3.0e-3	
	2^5	11	1.0e-3	1.5	9	1.2e-3	1.2	8	1.4e-3	1.3	8	1.3e-3	1.2	8	1.3e-3	1.1	8	1.3e-3	1.1
	2^6	14	4.7e-4	1.1	11	5.8e-4	1.1	8	5.9e-4	1.2	8	5.9e-4	1.2	8	5.9e-4	1.2	8	5.9e-4	1.2
	2^7	18	2.5e-4	0.9	13	2.9e-4	1	8	2.6e-4	1.2	8	2.6e-4	1.2	8	2.6e-4	1.2	8	2.6e-4	1.2
	2^8	23	1.4e-4	0.8	16	1.5e-4	0.9	8	1.1e-4	1.2	8	1.1e-4	1.2	8	1.1e-4	1.2	8	1.1e-4	1.2
	2^9	29	8.1e-5	0.8	17	8.4e-5	0.9	8	4.9e-5	1.2	8	4.9e-5	1.2	8	4.9e-5	1.2	8	4.9e-5	1.2
	2^{10}	35	4.6e-5	0.8	24	4.7e-5	0.8	8	2.1e-5	1.2	8	2.1e-5	1.2	8	2.1e-5	1.2	8	2.1e-5	1.2
0.5	2^4	8	2.3e-2		8	2.3e-2		9	2.0e-2		10	1.2e-2		9	5.1e-3		11	5.0e-3	
	2^5	8	8.3e-3	1.5	8	1.1e-2	1	11	9.4e-3	1.1	9	4.1e-3	1.5	10	1.8e-3	1.5	8	1.8e-3	1.5
	2^6	10	2.9e-3	1.5	10	5.7e-3	1	9	3.2e-3	1.6	9	1.2e-3	1.8	10	6.4e-4	1.5	9	6.4e-4	1.5
	2^7	13	1.3e-3	1.1	9	2.8e-3	1	9	9.0e-4	1.8	10	3.1e-4	1.9	10	2.3e-4	1.5	12	2.3e-4	1.5
	2^8	14	9.1e-4	0.6	13	1.4e-3	1	9	2.3e-4	2	9	8.0e-5	1.9	12	8.0e-5	1.5	11	1.0e-4	1.1
	2^9	15	6.4e-4	0.5	14	7.8e-4	0.9	11	5.6e-5	2	11	2.8e-5	1.5	12	3.0e-5	1.4	11	5.2e-5	1
	2^{10}	19	4.5e-4	0.5	15	5.1e-4	0.6	9	1.4e-5	2	11	1.1e-5	1.3	12	1.5e-5	1	13	2.6e-5	1
0.8	2^4	7	1.1e-1		7	1.1e-1		12	1.2e-1		-	-		15	9.0e-2		9	2.2e-2	
	2^5	9	7.6e-2	0.6	7	8.7e-2	0.4	-	-	–	-	-		13	5.0e-2	0.8	10	6.5e-3	1.8
	2^6	8	5.0e-2	0.6	8	6.6e-2	0.4	-	-		-	-		18	2.4e-2	1	13	1.9e-3	1.8
	2^7	9	2.5e-2	1	8	5.0e-2	0.4	-	-		-	-		10	4.0e-3	2.6	12	9.2e-4	1.1
	2^8	9	1.3e-2	1	8	3.8e-2	0.4	-	-		24	1.3e-2	-	15	5.8e-4	2.8	12	5.8e-4	0.7
	2^9	9	5.5e-3	1.2	9	2.9e-2	0.4	32	1.7e-2	-	18	2.7e-3	2.3	11	4.5e-4	0.4	11	4.3e-4	0.4
	2^{10}	12	4.8e-3	0.2	10	2.2e-2	0.4	23	4.2e-3	2	15	3.7e-4	2.8	12	3.4e-4	0.4	10	3.4e-4	0.3

Table 5.2. Iterations to tolerance and convergence order for $\gamma = 0.5$.

mesh given in Jia and Wang [2015] and described in Section 5.1.5. We recall that N_1 is the number of points of the graded part of the mesh, while N_2 is the number of points that compose the uniform part of the mesh and therefore the composite mesh has $N = N_1 + N_2$ points. We note that $\mathscr{P}$-GMRES has stable It when increasing N with respect to PFCGS. Furthermore, unlike the PFCGS Jia and Wang [2015], $\mathscr{P}$-GMRES is more versatile since it also works in the case where $\gamma \neq 0.5$. Hence, we only consider $\mathscr{P}$-GMRES and further test it over both composite and graded meshes.

Table 5.4 shows the It, e^∞ and the 2-norm relative numerical error e^{rel} of $\mathscr{P}$-GMRES varying N and the grid. We note that $\epsilon^{(6)}$ yields the lowest e^∞ for $2^4 \leq N \leq 2^6$, then e^∞ stops decreasing probably due to the lower cap imposed on the smallest step size, i.e., $\min_i\{h_i\} = 10^{-16}$. Our choice for the lower cap can surely be refined, but a lower cap is necessary since its absence would lead to a numerically singular coefficient matrix when N is large.

Moreover, when considering $\epsilon^{(6)}$ we observe that with $N = 2^6 - 1$ we obtain an e^∞ smaller than the e^∞ obtained when using the composite mesh with $g_G(N) = \lfloor\sqrt{N}\rfloor$ with $N = 2^{10} - 1$. Therefore, by improving the lower cap on the step size, the mesh mapped by $g_{q,\epsilon}(x)$ could potentially allows to reach lower errors with much smaller sizes compared to the composite mesh.

As already observed in Test 2, we stress once again that since $\beta \approx 1$ the non-linear part of $g_{q,\epsilon}(x)$ should be large when N is small, otherwise multigrid does not converge. Finally, we note that It are stable or stabilize as N increases for any of the tested grids, which makes $\mathscr{P}$-GMRES a suitable solver.

N_1	N_2	e^{∞}	It PFCGS	It $\mathscr{P}$-GMRES
2^3	2^8	7.9306e-2	13	8
2^4	2^9	4.2326e-2	16	9
2^5	2^{10}	1.3025e-2	25	9

Table 5.3. Iterations to tolerance of PFCGS and $\mathscr{P}$-GMRES with $\gamma = 0.5$ and $\beta = 0.9$

Test 4 We have shown that $\mathscr{P}$-GMRES works in the case where $\gamma = 0.5$. Here we show that, under some constraints, $\mathscr{P}$-GMRES is a suitable solver even in the extreme anisotropic cases where $\gamma = 0$ or $\gamma = 1$.

Table 5.5 shows It, the infinity norm error e^{∞} and the relative 2-norm error e^{rel} varying $\beta \in \{0.1, 0.3, 0.7\}$ with $\gamma \in \{0, 1\}$. When $\beta = 0.1$ and $\beta = 0.3$, despite the strong spatial anisotropy, the coefficient matrix is close to Hermitian (see Remark 5.3.1), and therefore $\mathscr{P}$-GMRES is expected to be a suitable preconditioner. In fact, we note that when considering $\epsilon^{(1)}, \epsilon^{(4)}, \epsilon^{(6)}$, It does not increase with N for both choices of γ and both errors seem to decrease with order $1 + \beta$ as observed in Test 2 with $\gamma = 0.5$.

When $\beta = 0.7$, due to the matrix being close to skew-symmetric, multigrid does not seem to be effective anymore. In fact, when $\gamma = 0$ we observe stable iterations only for the composite meshes, but the error does not seem to decrease

$N+1$	$g_G(N) = \lfloor\sqrt{N}\rfloor$			$g_G(N) = \lfloor\log_2 N\rfloor$			$\epsilon_1 = (0.1, 0.05)$			$\epsilon_2 = (0.2, 0.05)$			$\epsilon_4 = (0.45, 0.05)$			$\epsilon_6 = (1, 0)$		
	It	e^{∞}	e^{rel}	It	e^{∞}	e^{rel}	It	e^{∞}	e^{rel}	It	e^{∞}	e^{rel}	It	e^{∞}	e^{rel}	It	e^{∞}	e^{rel}
2^4	7	1.9e-1	8.7e-2	7	1.9e-1	8.7e-2	-	-	-	-	-	-	17	1.8e-1	1.2e-1	8	6.4e-2	1.0e-1
2^5	7	1.5e-1	5.3e-2	7	1.6e-1	5.6e-2	-	-	-	-	-	-	15	1.2e-1	6.4e-2	9	3.2e-2	3.3e-2
2^6	7	1.3e-1	3.1e-2	7	1.4e-1	3.5e-2	-	-	-	20	1.6e-1	4.3e-2	9	4.4e-2	2.4e-2	9	2.9e-2	1.6e-2
2^7	8	8.9e-2	1.6e-2	7	1.2e-1	2.2e-2	-	-	-	21	9.7e-2	2.3e-2	9	2.9e-2	1.2e-2	9	4.1e-2	1.2e-2
2^8	8	6.3e-2	8.5e-3	7	1.1e-1	1.3e-2	29	9.8e-2	1.7e-2	16	3.7e-2	1.2e-2	8	4.1e-2	1.0e-2	8	2.7e-2	6.3e-3
2^9	9	3.6e-2	4.1e-3	9	9.4e-2	8.3e-3	21	5.2e-2	1.2e-2	14	3.3e-2	9.7e-3	8	4.6e-2	1.1e-2	8	4.6e-2	7.9e-3
2^{10}	9	1.8e-2	2.2e-3	8	8.2e-2	5.1e-3	18	4.9e-2	1.1e-2	11	4.5e-2	1.1e-2	7	4.5e-2	9.6e-3	8	4.6e-2	7.7e-3

Table 5.4. Iterations of $\mathscr{P}$-GMRES with $\gamma = 0.5$ and $\beta = 0.9$

as expected (see Test 2). When considering the grid mapped by $g_{q,\epsilon}(x)$ with $\epsilon = \epsilon^{(6)}$, the iterations are not stable while increasing N, but the predicted convergence order seems to be restored. When $\gamma = 1$, $\mathscr{P}$-GMRES does not seem to converge for any of the tested grids mapped by $g_{q,\epsilon}(x)$. When using a composite mesh, instead, $\mathscr{P}$-GMRES converges but It are large, they increase with N and the error decreases too slowly.

5.5 Conclusions

In this chapter we have investigated multigrid preconditioners for conservative steady-state Caputo FDEs. We provided a full FVE discretization over a generic mesh of the FDE, then we computed the symbol of the coefficient matrix in case of non-uniform meshes mapped by a function, and we retrieved the related spectral information needed to build an ad-hoc multigrid preconditioner. In the numerical results section we have shown that not only such a multigrid is a valid alternative to the circulant preconditioner developed in Jia and Wang [2015], but it also works without the restriction $\gamma = 0.5$, imposed to build the circulant preconditioner. Furthermore, the automatic estimation of the relaxation weight for Jacobi, defined according to the strategy introduced in Chapter 2, makes it a parameter free multigrid preconditioner.

Regarding the grid, we have provided numerical comparisons between the composite meshes used in Jia and Wang [2015] and certain graded meshes mapped by non-linear functions. Numerical results show that the optimal grading parameter q_β, taken from Kopteva and Meng [2020], is a suitable parameter to generate the grid and allows to get a low error, close to the numerically optimal one. When $\beta \approx 1$, meshes mapped by not necessarily fully non-linear smooth functions still yield much lower infinity norm errors than composite meshes with the same amount of grid points, allowing fast matrix-vector products due to the Toeplitz structure of the involved coefficient matrix.

γ	β	$N+1$	$g_G(N)=\lfloor\sqrt{N}\rfloor$			$g_G(N)=\lfloor\log_2 N\rfloor$			$\epsilon^{(1)}=(0.1,0.05)$			$\epsilon^{(4)}=(0.45,0.05)$			$\epsilon^{(6)}=(1,0)$		
			It	e^{∞}	e^{rel}	It	e^{∞}	e^{rel}	It	e^{∞}	e^{rel}	It	e^{∞}	e^{rel}	It	e^{∞}	e^{rel}
0	0.1	2^5	10	9.2e-4	1.3e-3	11	1.0e-3	1.4e-3	7	1.5e-3	2.0e-3	7	1.5e-3	2.0e-3	7	1.5e-3	2.1e-3
		2^6	12	4.7e-4	6.2e-4	11	5.1e-4	6.7e-4	7	7.4e-4	9.5e-4	7	7.3e-4	9.9e-4	7	7.4e-4	1.0e-3
		2^7	21	2.6e-4	3.3e-4	11	2.7e-4	3.4e-4	7	3.6e-4	4.6e-4	7	3.6e-4	4.8e-4	7	3.6e-4	4.9e-4
		2^8	23	1.4e-4	1.8e-4	18	1.5e-4	1.8e-4	7	1.8e-4	2.2e-4	7	1.8e-4	2.3e-4	7	1.8e-4	2.4e-4
		2^9	27	7.9e-5	9.5e-5	18	7.9e-5	9.5e-5	7	8.5e-5	1.1e-4	7	8.5e-5	1.1e-4	7	8.5e-5	1.1e-4
		2^{10}	-	-	-	25	4.3e-5	5.1e-5	7	4.1e-5	5.1e-5	7	4.1e-5	5.4e-5	7	4.1e-5	5.5e-5
	0.3	2^5	14	6.1e-3	7.4e-3	14	7.4e-3	8.8e-3	17	6.5e-3	7.8e-3	17	4.5e-3	6.8e-3	17	4.6e-3	7.2e-3
		2^6	12	3.1e-3	3.6e-3	14	3.9e-3	4.5e-3	13	2.7e-3	3.2e-3	13	2.0e-3	3.0e-3	13	2.0e-3	3.2e-3
		2^7	12	1.8e-3	2.0e-3	11	2.2e-3	2.4e-3	9	1.1e-3	1.3e-3	10	9.0e-4	1.3e-3	10	9.0e-4	1.4e-3
		2^8	20	1.1e-3	1.2e-3	12	1.3e-3	1.4e-3	10	4.5e-4	5.4e-4	11	3.9e-4	5.7e-4	11	3.9e-4	6.1e-4
		2^9	16	6.9e-4	7.4e-4	15	7.5e-4	8.0e-4	11	1.9e-4	2.3e-4	10	1.7e-4	2.5e-4	10	1.7e-4	2.6e-4
		2^{10}	30	4.3e-4	4.5e-4	20	4.5e-4	4.8e-4	10	7.9e-5	9.6e-5	10	7.2e-5	1.1e-4	10	7.2e-5	1.1e-4
	0.7	2^5	17	1.1e-1	1.2e-1	17	1.3e-1	1.4e-1	-	-	-	33	3.2e-2	3.7e-2	20	1.8e-2	3.2e-2
		2^6	18	6.3e-2	6.9e-2	18	8.8e-2	9.4e-2	39	9.1e-2	9.5e-2	33	1.1e-2	1.3e-2	22	6.7e-3	1.2e-2
		2^7	18	3.0e-2	3.1e-2	18	6.1e-2	6.3e-2	-	-	-	42	3.6e-3	4.3e-3	19	2.3e-3	4.3e-3
		2^8	19	1.6e-2	1.7e-2	18	4.2e-2	4.3e-2	-	-	-	51	1.1e-3	1.4e-3	23	8.0e-4	1.5e-3
		2^9	20	9.5e-3	9.6e-3	18	2.9e-2	3.0e-2	-	-	-	56	3.5e-4	4.6e-4	26	2.7e-4	5.1e-4
		2^{10}	27	6.9e-3	7.0e-3	19	2.1e-2	2.1e-2	-	-	-	51	1.3e-4	1.9e-4	32	1.2e-4	2.3e-4
1	0.1	2^5	11	8.0e-5	6.1e-5	11	1.4e-4	8.6e-5	8	4.2e-4	3.3e-4	8	4.2e-4	3.6e-4	8	4.2e-4	3.7e-4
		2^6	13	4.5e-5	5.4e-5	11	4.0e-5	3.8e-5	8	2.0e-4	1.3e-4	8	2.0e-4	1.4e-4	8	2.0e-4	1.4e-4
		2^7	21	3.4e-5	3.9e-5	12	2.8e-5	3.2e-5	8	9.3e-5	4.8e-5	8	9.3e-5	5.3e-5	8	9.3e-5	5.4e-5
		2^8	32	2.1e-5	2.4e-5	15	1.9e-5	2.2e-5	8	4.3e-5	1.7e-5	8	4.3e-5	1.9e-5	8	4.3e-5	1.9e-5
		2^9	-	-	-	18	1.2e-5	1.4e-5	8	2.0e-5	5.9e-6	8	2.0e-5	6.3e-6	8	2.0e-5	6.3e-6
		2^{10}	-	-	-	25	7.4e-6	8.0e-6	8	9.5e-6	2.2e-6	8	9.5e-6	2.1e-6	8	9.5e-6	2.1e-6
	0.3	2^5	16	1.3e-3	9.4e-4	19	1.1e-3	8.0e-4	21	1.3e-3	1.0e-3	19	4.6e-4	6.4e-4	21	4.5e-4	6.3e-4
		2^6	15	8.9e-4	5.7e-4	15	8.0e-4	5.1e-4	13	5.6e-4	4.5e-4	15	2.2e-4	3.1e-4	15	2.2e-4	3.1e-4
		2^7	20	5.7e-4	3.2e-4	13	5.2e-4	3.0e-4	12	2.3e-4	2.0e-4	10	1.0e-4	1.4e-4	12	1.0e-4	1.4e-4
		2^8	26	3.5e-4	1.7e-4	15	3.3e-4	1.6e-4	11	9.6e-5	8.3e-5	11	4.7e-5	6.1e-5	11	4.7e-5	6.2e-5
		2^9	-	-	-	16	2.1e-4	8.8e-5	11	3.9e-5	3.5e-5	11	2.1e-5	2.6e-5	11	2.1e-5	2.7e-5
		2^{10}	-	-	-	21	1.3e-4	4.7e-5	11	1.6e-5	1.4e-5	11	8.8e-6	1.1e-5	11	8.8e-6	1.1e-5
	0.7	2^5	17	5.2e-2	1.4e-2	17	6.3e-2	1.6e-2	-	-	-	-	-	-	-	-	-
		2^6	21	2.8e-2	5.2e-3	18	4.1e-2	7.5e-3	-	-	-	-	-	-	-	-	-
		2^7	29	9.7e-3	1.5e-3	19	2.7e-2	3.5e-3	-	-	-	-	-	-	-	-	-
		2^8	36	3.4e-3	6.0e-4	23	1.8e-2	1.6e-3	-	-	-	-	-	-	-	-	-
		2^9	-	-	-	27	1.2e-2	7.7e-4	-	-	-	-	-	-	-	-	-
		2^{10}	-	-	-	28	7.7e-3	3.7e-4	-	-	-	-	-	-	-	-	-

Table 5.5. Iterations of $\mathscr{P}$-GMRES and error when solving the extreme anisotropic equation with $\gamma \in \{0, 1\}$

Chapter 6

Conclusions

In this thesis we dealt with different kind of FDEs, from a theoretical and practical point of view. For each equation we exploited the Toeplitz or GLT structure of the discretization matrix to retrieve related spectral information, which was then used to study the stability and to build ad-hoc multigrid solvers or preconditioners. Several numerical benchmarks are provided to test the robustness of our proposals.

We developed a symbol-based automatic procedure for the estimate of the Jacobi weight, which has been used in almost all of the proposed multigrid solvers. We stress that this approach has a low computational impact, allows to avoid a long and non trivial spectral analysis and makes multigrid a parameter free solver. The only flaw is that this method is intended to work when the multigrid hierarchy is computed through rediscretization, therefore, we aim at extending this algorithm also to the case where the Galerkin approach is adopted.

When dealing with parallel-in-time integration, we discovered that multigrid is sensitive to the time discretization scheme. We have shown that, when considering the BDF2 scheme, multigrid converges much faster than in the case of CN for a wider combination of parameters, such as grid and diffusion coefficients. In this regard, we plan to extend our study to different or higher order time discretization schemes. Furthermore, we aim at making comparisons with other state-of-the-art solvers like MGRIT Yue et al. [2019], parareal Wu and Zhou [2017] and PFASST Emmett and Minion [2012].

Finally, we introduced a new second-order FV discretization, based on a modification of the space discretization scheme WSGD used in Chapters 2 and 3. Such a scheme is shown to be a good alternative to the standard FVE approach. We plan to further improve the FV scheme through the use of different shifts for the WSGD scheme (see Remark 4.3.1), such that it becomes a more accurate scheme in case where the fractional derivative orders are close to 1.

Throughout our studies, we experienced similar spectral properties of the discretization matrices, even when considering different fractional derivative definitions, meshes and discretization methods. This suggests that MGMs could be applied to a much wider set of FDEs. At this point, a natural extension of our work would be to apply MGMs to other FDEs and to provide the symbol of the resulting discretization matrices. Specifically, in the anisotropic case, dealing with more than two-dimensions will ask for a careful treatment of the extra sources of anisotropy and relative extension of the strategies developed in Chapter 2. Moreover, such strategies could also be adapted to the parallel-in-time environment to deal with time-dependent high dimensional space-FDEs. With respect to the FDE treated in Chapter 3, a higher dimensional space would require an ad-hoc solver which is able to deal with anisotropies not only between space and time, but also between different spatial dimensions.

When non-uniform meshes are adopted to discretize two-dimensional FDEs, the extension of the work done in Chapter 5 would be almost straightforward in the case where the function $g(x,y)$, which generates the mesh, can be written as $g(x,y) = g_1(x)g_2(y)$. Unfortunately, this is a very restrictive requirement which, when imposed, could lead to an unnecessary increase of the coefficient matrix-size. However, it is less restrictive to assume, or impose, that $g(x,y)$ is partially linear (linear over a subset of the domain) such that the discretization matrix would have a partial Toeplitz structure.
A widely used MGM to deal with unstructured meshes is the algebraic multigrid (AMG) Ruge and Stüben [1987], which differs from the geometric MGMs treated in this thesis by the computation of the projectors. In the case of a dense coefficient matrix, like discretized FDEs, the computation of the projectors for the AMG is expensive, while the projectors in a geometric MGM are easy to compute but impractical whenever unstructured meshes are adopted. In the case of a partial Toeplitz structure, it would be interesting to study how the two multigrid approaches can be combined together to exploit the advantages of both techniques.

Finally, the employment of neural networks, trained to return ad-hoc projectors when non-uniform meshes are used, could further speed up our multigrid proposals Luz et al. [2020].

Bibliography

Agrawal, O. P. [2006]. A formulation and a numerical scheme for fractional optimal control, *IFAC Proceedings Volumes* **39**(11): 68–72. 2nd IFAC Workshop on Fractional Differentiation and its Applications.

Arico, A., Donatelli, M. and Serra-Capizzano, S. [2004]. V-cycle optimal convergence for certain (multilevel) structured linear systems, *SIAM Journal on Matrix Analysis and Applications* **26**(1): 186–214.

Ashyralyev, A., Dal, F. and Pinar, Z. [2009]. On the numerical solution of fractional hyperbolic partial differential equations, *Mathematical Problems in Engineering* **2009**.

Baeumer, B., Kovács, M., Meerschaert, M. M. and Sankaranarayanan, H. [2018]. Boundary conditions for fractional diffusion, *Journal of Computational and Applied Mathematics* **336**: 408–424.

Bertaccini, D. and Durastante, F. [2019]. Block structured preconditioners in tensor form for the all-at-once solution of a finite volume fractional diffusion equation, *Applied Mathematics Letters* **95**: 92–97.

Bini, D. A., Capovani, M. and Menchi, O. [1988]. *Metodi numerici per l'algebra lineare*, Zanichelli.

Bogoja, M., Serra-Capizzano, S. and Trotti, K. [2021]. Upper Hessenber and Toeplitz Bohemian matrix-sequences: A note on their asymptotical eigenvalues and singular values. Accepted for publication at Electronic Transactions on Numerical Analysis on 01.10.2021.

Böttcher, A., Gohberg, I. and Junghanns, P. [2012]. *Toeplitz matrices and singular integral equations: the Bernd Silbermann anniversary volume*, Vol. 135, Birkhäuser.

Breiten, T., Simoncini, V. and Stoll, M. [2014]. *Fast iterative solvers for fractional differential equations*, Max Planck Institute for Dynamics of Complex Technical Systems.

Briggs, W. L., Henson, V. E. and McCormick, S. F. [2000]. *A multigrid tutorial*, SIAM.

Brigham, E. O. [1973]. *The fast Fourier transform: An introduction to its theory and application*, Prentice Hall.

Bueno-Orovio, A., Kay, D., Grau, V., Rodriguez, B. and Burrage, K. [2014]. Fractional diffusion models of cardiac electrical propagation: role of structural heterogeneity in dispersion of repolarization, *Journal of The Royal Society Interface* **11**(97): 20140352.

Defterli, O., DElia, M., Du, Q., Gunzburger, M., Lehoucq, R. and Meerschaert, M. M. [2015]. Fractional diffusion on bounded domains, *Fractional Calculus and Applied Analysis* **18**(2): 342–360.

Diethelm, K., Garrappa, R., Giusti, A. and Stynes, M. [2020]. Why fractional derivatives with nonsingular kernels should not be used, *Fractional Calculus and Applied Analysis* **23**(3): 610–634.

Donatelli, M. [2010]. An algebraic generalization of local Fourier analysis for grid transfer operators in multigrid based on Toeplitz matrices, *Numerical Linear Algebra with Applications* **17**(2-3): 179–197.

Donatelli, M., Garoni, C., Manni, C., Serra-Capizzano, S. and Speleers, H. [2015]. Two-grid optimality for Galerkin linear systems based on B-splines, *Computing and visualization in science* **17**(3): 119–133.

Donatelli, M., Krause, R., Mazza, M., Semplice, M. and Trotti, K. [2021]. Spectral-based fast solvers for finite volume discretizations of a conservative fractional diffusion problem. Under review at Numerical Linear Algebra and Applications.

Donatelli, M., Krause, R., Mazza, M. and Trotti, K. [2020]. Multigrid preconditioners for anisotropic space-fractional diffusion equations, *Advances in Computational Mathematics* **46**: 1–31.

Donatelli, M., Krause, R., Mazza, M. and Trotti, K. [2021a]. All-at-once multigrid approaches for one-dimensional space-fractional diffusion equations, *Calcolo* **58**(45).

Donatelli, M., Krause, R., Mazza, M. and Trotti, K. [2021b]. Multigrid methods for Caputo space-fractional diffusion equations over non-uniform meshes. Submitted.

Donatelli, M., Mazza, M. and Serra-Capizzano, S. [2016]. Spectral analysis and structure preserving preconditioners for fractional diffusion equations, *Journal of Computational Physics* **307**: 262–279.

Donatelli, M., Mazza, M. and Serra-Capizzano, S. [2018]. Spectral analysis and multigrid methods for finite volume approximations of space-fractional diffusion equations, *SIAM Journal on Scientific Computing* **40**(6): A4007–A4039.

Emmett, M. and Minion, M. [2012]. Toward an efficient parallel in time method for partial differential equations, *Communications in Applied Mathematics and Computational Science* **7**(1): 105–132.

Eymard, R., Gallouët, T. and Herbin, R. [2000]. Finite volume methods, *Handbook of numerical analysis* **7**: 713–1018.

Feng, L., Zhuang, P., Liu, F. and Turner, I. [2015]. Stability and convergence of a new finite volume method for a two-sided space-fractional diffusion equation, *Applied Mathematics and Computation* **257**: 52–65.

Fischer, R. and Huckle, T. [2006]. Multigrid methods for anisotropic BTTB systems, *Linear Algebra and its applications* **417**(2-3): 314–334.

Fischer, R. and Huckle, T. [2008]. Multigrid solution techniques for anisotropic structured linear systems, *Applied numerical mathematics* **58**(4): 407–421.

Garoni, C. and Serra-Capizzano, S. [2017]. *Generalized locally Toeplitz sequences: Theory and applications*, Vol. 1, Springer.

Garoni, C. and Serra-Capizzano, S. [2018]. *Generalized locally Toeplitz sequences: Theory and applications*, Vol. 2, Springer.

Ghorbani, A. and Baleanu, D. [2020]. Fractional spectral differentiation matrices based on Legendre approximation, *Advances in Difference Equations* **2020**(1): 1–9.

Gracia, J. L., ORiordan, E. and Stynes, M. [2018]. A fitted scheme for a Caputo initial-boundary value problem, *Journal of Scientific Computing* **76**(1): 583–609.

Gracia, J. L., ORiordan, E. and Stynes, M. [2020]. Convergence analysis of a finite difference scheme for a two-point boundary value problem with a Riemann–Liouville–Caputo fractional derivative, *BIT Numerical Mathematics* **60**(2): 411–439.

Grenander, U. and Szegö, G. [1958]. *Toeplitz forms and their applications*, Univ of California Press.

Gu, X.-M., Huang, T.-Z., Zhao, X.-L., Li, H.-B. and Li, L. [2015]. Strang-type preconditioners for solving fractional diffusion equations by boundary value methods, *Journal of Computational and Applied Mathematics* **277**: 73–86.

Gu, X.-M., Zhao, Y.-L., Zhao, X.-L., Carpentieri, B. and Huang, Y.-Y. [2020]. A note on parallel preconditioning for the all-at-once solution of Riesz fractional diffusion equations, *arXiv preprint arXiv:2003.07020* .

Hejazi, H., Moroney, T. and Liu, F. [2013]. A finite volume method for solving the two-sided time-space fractional advection-dispersion equation, *Central European Journal of Physics* **11**(10): 1275–1283.

Hejazi, H., Moroney, T. and Liu, F. [2014]. Stability and convergence of a finite volume method for the space fractional advection–dispersion equation, *Journal of Computational and Applied Mathematics* **255**: 684–697.

Hemker, P. [1990]. On the order of prolongations and restrictions in multigrid procedures, *Journal of Computational and Applied Mathematics* **32**(3): 423–429.

Horton, G. and Vandewalle, S. [1995]. A space-time multigrid method for parabolic partial differential equations, *SIAM Journal on Scientific Computing* **16**(4): 848–864.

Jia, J. and Wang, H. [2015]. A preconditioned fast finite volume scheme for a fractional differential equation discretized on a locally refined composite mesh, *Journal of Computational Physics* **299**: 842–862.

Jia, J. and Wang, H. [2016]. A fast finite volume method for conservative space-fractional diffusion equations in convex domains, *Journal of Computational Physics* **310**: 63–84.

Jia, L., Chen, H. and Ervin, V. J. [2019]. Existence and regularity of solutions to 1-D fractional order diffusion equations, *Electronic Journal of Differential Equations* **2019**: 1–21.

Jiang, Y. and Xu, X. [2015]. Multigrid methods for space fractional partial differential equations, *Journal of Computational Physics* **302**: 374–392.

Katugampola, U. [2014]. Existence and uniqueness results for a class of generalized fractional differential equations bulletin of mathematical analysis and applications, *arXiv preprint arXiv:1411.5229* .

Kelly, J. F., Sankaranarayanan, H. and Meerschaert, M. M. [2019]. Boundary conditions for two-sided fractional diffusion, *Journal of Computational Physics* **376**: 1089–1107.

Kilbas, A. A., Srivastava, H. M. and Trujillo, J. J. [2006]. *Theory and applications of fractional differential equations*, Vol. 204, Elsevier.

Kopteva, N. and Meng, X. [2020]. Error analysis for a fractional-derivative parabolic problem on quasi-graded meshes using barrier functions, *SIAM Journal on Numerical Analysis* **58**(2): 1217–1238.

Lei, S.-L. and Sun, H.-W. [2013]. A circulant preconditioner for fractional diffusion equations, *Journal of Computational Physics* **242**: 715–725.

Leibniz, G. [1962a]. Letter from Hanover, Germany to Johann Bernoulli, December 28, 1695, *Leibniz Mathematische Schriften, Olms-Verlag, Hildesheim, Germany* **226**.

Leibniz, G. [1962b]. Letter from Hanover, Germany to John Wallis, May 28, 1697, *Leibniz Mathematische Schriften, Olms-Verlag, Hildesheim, Germany* p. 25.

Leibniz, G. W. [1849]. Letter from Hanover, Germany to GFA LHospital, September 30, 1695, *Mathematische Schriften* **2**: 301–302.

Liao, H.-l., Lyu, P. and Vong, S. [2018]. Second-order BDF time approximation for Riesz space-fractional diffusion equations, *International Journal of Computer Mathematics* **95**(1): 144–158.

Liesen, J. and Strakos, Z. [2013]. *Krylov subspace methods: Principles and analysis*, Oxford University Press.

Lin, X.-l., Ng, M. K. and Sun, H.-W. [2017a]. A multigrid method for linear systems arising from time-dependent two-dimensional space-fractional diffusion equations, *Journal of Computational Physics* **336**: 69–86.

Lin, X.-l., Ng, M. K. and Sun, H.-W. [2017b]. A splitting preconditioner for Toeplitz-like linear systems arising from fractional diffusion equations, *SIAM Journal on Matrix Analysis and Applications* **38**(4): 1580–1614.

Liu, F., Zhuang, P., Turner, I., Anh, V. and Burrage, K. [2015]. A semi-alternating direction method for a 2-D fractional Fitzhugh–Nagumo monodomain model on an approximate irregular domain, *Journal of Computational Physics* **293**: 252–263.

Liu, F., Zhuang, P., Turner, I., Burrage, K. and Anh, V. [2014]. A new fractional finite volume method for solving the fractional diffusion equation, *Applied Mathematical Modelling* **38**(15-16): 3871–3878.

Loverro, A. [2004]. Fractional calculus: History, definitions and applications for the engineer, *Rapport technique, Univeristy of Notre Dame: Department of Aerospace and Mechanical Engineering* pp. 1–28.

Luz, I., Galun, M., Maron, H., Basri, R. and Yavneh, I. [2020]. Learning algebraic multigrid using graph neural networks, *International Conference on Machine Learning*, PMLR, pp. 6489–6499.

Mainardi, F., Luchko, Y. and Pagnini, G. [2001]. The fundamental solution of the space-time fractional diffusion equation, *Fractional Calculus and Applied Analysis* **4**: 153–192.

Meerschaert, M. M. and Tadjeran, C. [2004]. Finite difference approximations for fractional advection–dispersion flow equations, *Journal of computational and applied mathematics* **172**(1): 65–77.

Michelitsch, T. M., Riascos, A. P., Collet, B. A., Nowakowski, A. F. and Nicolleau, F. C. [2020]. Generalized space–time fractional dynamics in networks and lattices, *Nonlinear Wave Dynamics of Materials and Structures*, Springer, pp. 221–249.

Moghaderi, H., Dehghan, M., Donatelli, M. and Mazza, M. [2017]. Spectral analysis and multigrid preconditioners for two-dimensional space-fractional diffusion equations, *Journal of Computational Physics* **350**: 992–1011.

Oldham, K. B. and Spanier, J. [n.d.]. *Theory and applications of differentiation and integration to arbitrary order*, Academic Press: New York, NY, USA; London, UK.

Oosterlee, C. W. [1995]. The convergence of parallel multiblock multigrid methods, *Applied numerical mathematics* **19**(1-2): 115–128.

Ortigueira, M. D. [2011]. *Fractional calculus for scientists and engineers*, Vol. 84, Springer Science & Business Media.

Pan, J., Ng, M. and Wang, H. [2017]. Fast preconditioned iterative methods for finite volume discretization of steady-state space-fractional diffusion equations, *Numerical Algorithms* **74**(1): 153–173.

Pang, H.-K. and Sun, H.-W. [2012]. Multigrid method for fractional diffusion equations, *Journal of Computational Physics* **231**(2): 693–703.

Patie, P. and Simon, T. [2012]. Intertwining certain fractional derivatives, *Potential analysis* **36**(4): 569–587.

Podlubny, I. [1998]. *Fractional differential equations: an introduction to fractional derivatives, fractional differential equations, to methods of their solution and some of their applications*, Elsevier.

Ran, M. and Zhang, C. [2020]. A high-order accuracy method for solving the fractional diffusion equations, *Journal of Computational Mathematics* **38**(2).

Ruge, J. W. and Stüben, K. [1987]. Algebraic multigrid, *Multigrid methods*, SIAM, pp. 73–130.

Samko, S. G., Kilbas, A. A. and Marichev, O. I. [1993]. *Fractional integrals and derivatives: Theory and applications*, Gordan and Breach Science Publishers.

Serra-Capizzano, S. [2002]. Convergence analysis of two-grid methods for elliptic Toeplitz and PDEs matrix-sequences, *Numerische mathematik* **92**(3): 433–465.

Serra-Capizzano, S. and Tyrtyshnikov, E. [n.d.]. Any circulant-like preconditioner for multilevel Toeplitz matrices is not superlinear, *Journal on Matrix Analysis and Applications* **21**(2): 431–439.

Stynes, M. [2016]. Too much regularity may force too much uniqueness, *Fractional Calculus and Applied Analysis* **19**(6): 1554–1562.

Stynes, M. [2019]. Singularities, *Handbook of Fractional Calculus with Applications* **3**: 287–305.

Tenreiro Machado, J. [2011]. *And I say to myself: What a fractional world!*, Vol. 14, Springer.

Thomée, V. [1984]. Galerkin finite element methods for parabolic problems, *Lecture notes in mathematics* **1054**: 241–277.

Tian, W., Zhou, H. and Deng, W. [2015]. A class of second order difference approximations for solving space fractional diffusion equations, *Mathematics of Computation* **84**(294): 1703–1727.

Tilli, P. [1998]. A note on the spectral distribution of Toeplitz matrices, *Linear and Multilinear Algebra* **45**(2-3): 147–159.

Trottenberg, U., Oosterlee, C. W. and Schuller, A. [2000]. *Multigrid*, Elsevier.

Van Lent, J. and Vandewalle, S. [2002]. Multigrid waveform relaxation for anisotropic partial differential equations, *Numerical Algorithms* **31**(1): 361–380.

Wang, H. and Du, N. [2013]. A superfast-preconditioned iterative method for steady-state space-fractional diffusion equations, *Journal of Computational Physics* **240**: 49–57.

Wang, H. and Yang, D. [2017]. Wellposedness of Neumann boundary-value problems of space-fractional differential equations, *Fractional Calculus and Applied Analysis* **20**(6): 1356–1381.

Washio, T. and Oosterlee, C. W. [1998]. Flexible multiple semicoarsening for three-dimensional singularly perturbed problems, *SIAM Journal on Scientific Computing* **19**(5): 1646–1666.

Wu, S.-L. and Zhou, T. [2017]. Fast parareal iterations for fractional diffusion equations, *Journal of Computational Physics* **329**: 210–226.

Yang, Q., Turner, I., Moroney, T. and Liu, F. [2014]. A finite volume scheme with preconditioned Lanczos method for two-dimensional space-fractional reaction–diffusion equations, *Applied Mathematical Modelling* **38**(15-16): 3755–3762.

Yue, X., Shu, S., Xu, X., Bu, W. and Pan, K. [2019]. Parallel-in-time multigrid for space–time finite element approximations of two-dimensional space-fractional diffusion equations, *Computers & Mathematics with Applications* **78**(11): 3471–3484.

Zhang, X., Crawford, J. W., Deeks, L. K., Stutter, M. I., Bengough, A. G. and Young, I. M. [2005]. A mass balance based numerical method for the fractional advection-dispersion equation: Theory and application, *Water resources research* **41**(7).

Zhang, Y., Zhou, D., Wei, W., Frame, J. M., Sun, H., Sun, A. Y. and Chen, X. [2021]. Hierarchical fractional advection-dispersion equation (FADE) to quantify anomalous transport in river corridor over a broad spectrum of scales: Theory and applications, *Mathematics* **9**(7): 790.

Zhao, Y.-L., Zhu, P.-Y., Gu, X.-M., Zhao, X.-L. and Jian, H.-Y. [2020]. A preconditioning technique for all-at-once system from the nonlinear tempered fractional diffusion equation, *Journal of Scientific Computing* **83**(1): 1–27.