NTNU – Trondheim
Norwegian University of
Science and Technology

Nonlinear integro-differential Equations

Numerical Solutions by using Spectral
Methods

Stein-Olav Hagen Davidsen

Master of Science in Physics and Mathematics
Submission date: June 2013
Supervisor: Espen Robstad Jakobsen, MATH

Norwegian University of Science and Technology
Department of Mathematical Sciences

Preface

This thesis completes my master's degree in industrial mathematics at NTNU.

I would like to thank my supervisor, professor Espen Robstad Jakobsen, for his good help and support with the project and many interesting mathematical conversations in the course of my studies in Trondheim. The thesis builds on some of his previous works, and his brilliant understanding of the topic has therefore been of great importance in achieving the goals of this project.

Trondheim, June 11, 2013

Stein-Olav Davidsen

Abstract

This article deals with numerical solutions of nonlinear integro-differential convection-diffusion equations using spectral methods. More specifically, the spectral vanishing viscosity method is introduced and analyzed to show that its family of numerical solutions is compact, and that its solutions converge to the vanishing viscosity solutions. The method is implemented in code, and numerical results including qualitative plots and convergence estimates are given. The article concludes with a discussion of some important implementation concerns and recommendations for further work related to the topic.

Sammendrag

I denne artikkelen behandles numeriske løsninger av ikkelineære integropartielle konveksjons-diffusjonsdifferensialikninger ved bruk av spektralmetoder. Mer spesifikt introduseres og analyseres metoder med spektralt forsvinnende viskositet. Det vises at metodene danner en kompakt familie av numeriske løsninger, og at disse løsningene konvergerer mot løsningene av likningen med forsvinnende viskositet. Metoden implementeres i kode, og det gis numeriske resultater som viser kvalitative plott av løsninger samt konvergensestimater. Artikkelen avsluttes med en diskusjon omkring viktige problemstillinger i implementasjonen, samt anbefalinger for mulig videre arbeid relatert til emnet.

Contents

Chapter 1

Introduction

1.1 Background and motivation for the project

In physics and the applied sciences, the principle of conserved quantities is fundamental to modeling a wide class of phenomena, ranging from chemical reactions to financial asset pricing. Examples of conserved quantities in physics are mass-energy, linear momentum, angular momentum and probability. Together with constitutive or empirical laws, a vast class of equations can be derived from the simple principle of conservation. A couple of well-known examples from fluid dynamics are the Navier-Stokes' equation and the Burgers' equation.

The equations that this project aims at solving numerically belong to a relatively new class of partial differential equations: conservation laws with nonlinear fractional diffusion. Other types of resemblant equations have been studied previously. These include equations with linear fractional diffusion and nonfractional nonlinear equations like the Boussinesq equation and the Burgers' equation. For these equations there exists developed theory of different numerical simulation techniques. For the nonlinear fractional equations, there is not so much literature available. In [4], Jakobsen and Cifani present a finite volume like scheme. In [2], they develop spectral methods for equations with linear fractional diffusion, but as far as the author knows, the general nonlinear diffusion case has not been approached to a great extent by using spectral methods.

An important goal for the project is to generalize the theory that is presented in [2] to also include more general diffusion functions, using spectral methods. The motivation for this is that creating solvers for more complicated models enables practitioners to describe physical phenomena more accurately. It is therefore also a goal to implement efficient solvers and as high accuracy as possible, so that there is also a practical use of the theory that is developed. In [16], we develop spectral methods for linear fractional option pricing, using spectral methods and obtaining spectral accuracy, so this project is a natural step in generalizing those results.

Extending to the nonlinear case is interesting both from a theoretical and practical point of view. In addition to the theoretical aspects of showing properties

of the method, several implementation concerns arise in the development of the practical solver. Examples include the calculation of difficult integrals and possible aliasing effects, both of which are addressed in this project.

1.2 Project overview

The project consists of two parts; one theoretical and one practical.

The theoretical part of the report consists of some background to the equations that are studied, an introduction to the spectral method where the vanishing and the spectral vanishing viscosity equations are introduced in Chapter 2, and finally an analysis that culminates in showing compactness of the family of numerical solutions of the spectral vanishing viscosity method in Chapter 3. Based on the a priori estimates that are derived, a convergence estimate is given, which shows that the spectral vanishing viscosity method converges to the vanishing viscosity solution.

The practical part consists of a derivation of the equations implemented in the numerical solvers in Chapter 4, a presentation and discussion of numerical simulations carried out in the project in Chapter 5, and in Chapter 6 a discussion of some important implementation concerns that have arisen in the development of the solvers.

In chapter 7, some concluding remarks and a discussion of possible further work are given. Appendix A consists of mathematical results that are used in the analyses, and Appendix B contains some software documentation for the solvers that were developed.

Chapter 2

Models and general problem

2.1 Analytical equation

The relevant class of equations can in a very general form be written

$$u_t + \frac{\mathrm{d}}{\mathrm{d}x} f(u,x) - \mathcal{L}[A(u,x)] = g(x), \tag{2.1}$$

where $\mathcal{L}$ is a Lévy type (nonlocal) operator and f and A are functions of u and possibly the spatial variable x. In this project, the class of problems under study is restricted to

$$u_t + \frac{\mathrm{d}}{\mathrm{d}x} f(u) + (-\Delta)^{\frac{\alpha}{2}} A(u) = 0 \tag{2.2}$$

i.e. $\mathcal{L} = -(-\Delta)^{\frac{\alpha}{2}}$ is the fractional Laplacian operator, and f and A are spatially invariant.

To obtain a well-posed formulation of the above equation, a domain and initial/boundary values must be specified. One possibility is to pose the Cauchy problem, seeking solutions on the whole line and specifying initial values only. Another option, which is perhaps most relevant for physical applications, is to seek solutions on a finite domain, specifying both initial values and an appropriate set of boundary conditions. This project is concerned about numerical solutions of the equation posed on a finite domain, so the Cauchy problem is not discussed further in the rest of the article. Theory for the general Cauchy problem in arbitrary dimensions is treated by among others Jacobsen and Cifani. For analytical theory and results concerning the Cauchy problem, the reader is referred to [4].

2.1.1 Fractional Laplacian operator

The fractional Laplacian operator $(-\Delta)^{\frac{\alpha}{2}}$ can be interpreted and defined in several ways. One way is to use a probabilistic interpretation and consider it as the generator of a Lévy process, cf. [1]. The physical interpretation of the operator

is that it models anomalous diffusion, cf. [10]. Mathematically, the easiest way to define it is probably using the Fourier transform. Then it is defined through

$$\mathcal{F}\big((-\Delta)^{\frac{\alpha}{2}}u\big)(\xi) = |\xi|^{\alpha}\mathcal{F}(u) \;\; \forall u \in H^{\frac{\alpha}{2}}(\mathbb{R}),\; 0<\alpha<2 \tag{2.3}$$

Observe that this definition is consistent in the limit $\alpha \to 2$, where the operator converges to the standard Laplacian (or at least a constant times Δ depending on the definition of the Fourier transform). Another definition is the following:

$$-(-\Delta)^{\frac{\alpha}{2}}u = c_\alpha\Big(P.V.\int_{|y|<r}\frac{u(x+y)-u(x)-yu_x(x)}{|y|^{1+\alpha}}\,\mathrm{d}y + \int_{|y|>r}\frac{u(x+y)-u(x)}{|y|^{1+\alpha}}\,\mathrm{d}y\Big)$$
$$\forall u \in H^{\frac{\alpha}{2}}(\mathbb{R}),\; 0<\alpha<2, \tag{2.4}$$

where $r>0$ can be chosen arbitrarily and *P.V.* denotes the Cauchy principal value defined by

$$P.V.\int_{|y|>0}\varphi(y)\,\mathrm{d}y = \lim_{b\to 0}\int_{|y|>b}\varphi(y)\,\mathrm{d}y$$

and

$$c_\alpha = \frac{\alpha\Gamma\big(\frac{1+\alpha}{2}\big)}{2\pi^{\frac{1}{2}+\alpha}\Gamma\big(1-\frac{\alpha}{2}\big)}.$$

The above value of c_α is used throughout the report. This definition reveals the nonlocal nature of the operator, since the integration is performed on the whole line. Note that the integral is singular, hence the principal value. It can be shown that under certain assumptions, (2.3) and (2.4) are equivalent up to a multiplicative constant, cf. Theorem 1 in [13]. In the rest of the article, they will be used interchangeably, depending on which is more convenient to use. By using the last definition, some properties of the operator can be deduced. These will become useful in the rest of the article, so we state them in the following

Lemma 1 (Properties of the fractional Laplacian operator)**.** *Let $\mathcal{L} = -(-\Delta)^{\frac{\alpha}{2}}$ be defined as in (2.4). Then, the following properties hold:*

- *i) Splitting*

$$\mathcal{L}(u) = c_\alpha\Big(P.V.\underbrace{\int_{|y|<r}\frac{u(x+y)-u(x)-yu_x(x)}{|y|^{1+\alpha}}\,dy}_{=:\mathcal{L}_r(u)} + \underbrace{\int_{|y|>r}\frac{u(x+y)-u(x)}{|y|^{1+\alpha}}\,dy}_{=:\mathcal{L}^r(u)}\Big), \tag{2.5}$$

  where $\mathcal{L}_r(u)$ is singular and $\mathcal{L}^r(u)$ is not.

- *ii) Principal value*

$$c_\alpha P.V.\int_{|y|<r}\frac{u(x+y)-u(x)-yu_x(x)\,\mathrm{d}y}{|y|^{1+\alpha}} = c_\alpha P.V.\int_{|y|<r}\frac{u(x+y)-u(x)\,\mathrm{d}y}{|y|^{1+\alpha}} \tag{2.6}$$

- *iii) Integrability of measure*

$$\begin{aligned}\int_{|y|>0} \frac{\min\{1,|y|\}\mathrm{d}y}{|y|^{1+\alpha}} &< \infty, \;\; 0<\alpha<1 \\ \int_{|y|>0} \frac{\min\{1,y^2\}\mathrm{d}y}{|y|^{1+\alpha}} &< \infty, \;\; 1<\alpha<2.\end{aligned} \tag{2.7}$$

Proof of iii). Consider

$$\begin{aligned}\int_{|y|>0} \frac{\min\{1,|y|\}\mathrm{d}y}{|y|^{1+\alpha}} &= \int_0^1 \frac{\mathrm{d}y}{|y|^{\alpha}} + \int_1^\infty \frac{\mathrm{d}y}{|y|^{1+\alpha}} \\ &= \frac{1}{1-\alpha} + \frac{1}{\alpha} < \infty, \;\; 0<\alpha<1\end{aligned}$$

and

$$\begin{aligned}\int_{|y|>0} \frac{\min\{1,y^2\}\mathrm{d}y}{|y|^{1+\alpha}} &= \int_0^1 \frac{\mathrm{d}y}{|y|^{\alpha-1}} + \int_1^\infty \frac{\mathrm{d}y}{|y|^{1+\alpha}} \\ &= \frac{1}{2-\alpha} + \frac{1}{\alpha} < \infty, \;\; 1<\alpha<2\end{aligned}$$

□

Properties i) and ii) in Lemma 1 correspond to Lemma 1 in [9], and a proof can be found there.

2.2 Numerical formulation

In this section, the analytical equation is casted to a numerical formulation on a finite domain. Due to the possibly non-linear terms, the analytical equation cannot be used directly, and a perturbed variety is used instead. The reasons for this are motivated in the following, and both the perturbed equation in strong form and the variational form are discussed.

2.2.1 Preliminaries

In the recasting from a Cauchy problem posed on an infinite domain to a numerical form posed on a compact domain with boundary conditions and a discrete function space, some modifications of the equation must be made. There are possibly several ways to overcome these issues. This project focuses on spectral vanishing viscosity methods with nonlinear interpolation.

Notation

In the following, two sets of parentheses are used interchangeably to distinguish between functional and function evaluation. For instance, $f(u)$ means the function f evaluated at u, which itself is a function of x. The notation $f[u](x)$ however means the number deriving from a functional f_x evaluated at the function u, namely the function $f(u)$ evaluated at the spatial point x.

Fourier spectral method

For the theoretical analysis that follows, it is necessary to define the method that is to be analyzed. Since the boundary conditions are periodic for the relevant problem, and the domain is chosen to $\Omega = (0, 2\pi)$, the method seeks solutions that can be written

$$u(x,t) = \sum_{\xi=-N}^{N} \hat{u}_\xi \mathrm{e}^{i\xi x}$$

for an integer N, which is a Fourier spectral method. This choice of method satisfies the boundary conditions intrinsically and is therefore practical also in terms of computations. If in contrast other boundary conditions were to be satisfied, a polynomial basis would be an appropriate choice. Furthermore, with this method, the numerical solutions are smooth, which is an assumption that is used in the following.

Nonlinear interpolation

When introducing a discretization of the equation, the associated discrete function space has a given, finite dimension. By expressing the numerical solution as a linear combination of its basis functions, one is confined within the boundaries of this space. In the case of a linear equation, this concept is straightforward.

However, allowing for nonlinear terms, i.e. terms that are nonlinear functions of the numerical solution, terms of higher order than the discrete space dimension will inevitably occur in the equation, since these no longer are linear combinations of the discrete basis functions. One way to remedy this is to introduce nonlinear interpolation in the equation. Instead of having the (possibly) nonlinear function $A(u)$, the Nth order interpolation of $A(u)$, $P_N A(u)$, is introduced instead. In Fourier space, this operator can be defined in the following way. Assume that the Fourier representation of $A[u]$ exists, hence

$$A[u](x) = \sum_{\xi=-\infty}^{\infty} A_\xi \mathrm{e}^{i\xi x}$$

with

$$A_\xi = \frac{1}{2\pi} \int_0^{2\pi} A[u](x) \cdot \mathrm{e}^{-i\xi x} \, \mathrm{d}x.$$

Then,

$$P_N A(u) := \sum_{\xi=-N}^{N} A_\xi \mathrm{e}^{i\xi x}.$$

The possibly nonlinear convection term is treated in a similar way. The Fourier representation of $f[u]$ is assumed to exist, and thereby

$$f[u](x) = \sum_{\xi=-\infty}^{\infty} f_\xi \mathrm{e}^{i\xi x}$$

where

$$f_\xi = \frac{1}{2\pi} \int_0^{2\pi} f[u](x) \cdot \mathrm{e}^{-i\xi x} \, \mathrm{d}x,$$

such that

$$P_N f(u) := \sum_{\xi=-N}^{N} f_\xi \mathrm{e}^{i\xi x}.$$

Observe that the interpolation operator P_N commutes with the differential operator ∂_x^r and the nonlocal operator $\mathcal{L}$. This will be utilized extensively throughout the analysis.

Spectral vanishing viscosity

When changing from physical domain to Fourier domain or vice versa, it can be convenient to use the well-known properties of convolution in the Fourier domain. In the following, the equations written in the physical domain will use convolutions to express sums in the Fourier domain. The high frequency components that are added as spectral vanishing viscosity terms in Fourier space can thus be expressed as a convolution in the time domain through the relation

$$\partial_x^2 Q_N * u = - \sum_{|\xi|=m_N}^{N} \xi^2 \hat{Q}_\xi(t) \hat{u}_\xi(t) \mathrm{e}^{i\xi x},$$

where

$$Q_N(x,t) = \sum_{p=m_N}^{N} \hat{Q}_p(t) \cdot \sum_{|\xi|=p} \mathrm{e}^{i\xi x}$$

and the $\hat{Q}_p$ satisfy certain requirements.

2.2.2 Strong vanishing viscosity formulation

In one dimension, introduce the spatial compact domain $\Omega = (0, 2\pi)$. The numerical strong equation for the periodic vanishing viscosity formulation of problem

(2.2) in Ω reads

$$\begin{cases} u_t^\epsilon = -\frac{\mathrm{d}}{\mathrm{d}x} f(u^\epsilon) + \mathcal{L}[A(u^\epsilon)] + \epsilon \partial_x^2 u^\epsilon, \ \ (x,t) \in D_T = \Omega \times [0,T] \\ u^\epsilon(0,t) = u^\epsilon(2\pi,t) \\ u^\epsilon(x,0) = u_0(x), \ \ x \in \Omega, \end{cases} \tag{2.8}$$

2.2.3 Strong spectral vanishing viscosity formulation

The spectral vanishing viscosity approximation of problem (2.2) in Ω reads

$$\begin{cases} u_t = -\frac{\mathrm{d}}{\mathrm{d}x}[P_N f(u)] + \mathcal{L}[P_N A(u)] + \epsilon_N \partial_x^2 Q_N * u, \ \ (x,t) \in D_T = \Omega \times [0,T] \\ u(0,t) = u(2\pi,t) \\ u(x,0) = P_N u_0(x), \ \ x \in \Omega, \end{cases} \tag{2.9}$$

where P_N denotes the interpolation operator defined above.

2.2.4 Numerical variational formulation

To obtain a variational formulation of (2.9), one must introduce a function space. For this problem, the natural space to search for solutions is

$$V = \{v \in H^{\frac{\alpha}{2}}(\mathbb{R}), \int_0^T \|v\|^2_{L^2(\Omega)} \, \mathrm{d}t < \infty\}.$$

The space is chosen to ensure that all integrals are well-defined. The nonlocal Lévy operator $\mathcal{L}$ induces the Sobolev norm of order $\frac{\alpha}{2}$. Now, multiply (2.9) with an arbitrary test function $v \in V$ and integrate the equation over the whole domain. The variational formulation reads

$$\begin{gathered} \text{For each } t > 0, \text{ find } u \in V \text{ such that} \\ \int_0^{2\pi} u_t \cdot v \, \mathrm{d}x = -\int_0^{2\pi} \frac{\mathrm{d}}{\mathrm{d}x}[P_N f(u)] \cdot v \, \mathrm{d}x + \int_0^{2\pi} \mathcal{L}[P_N A(u)] \cdot v \, \mathrm{d}x \\ + \epsilon_N \int_0^{2\pi} (\partial_x^2 Q_N * u) v \, \mathrm{d}x \ \ \forall v \in V \end{gathered} \tag{2.10}$$

Chapter 3

Convergence theory

3.1 Introduction

The aim of this section is to prove that the numerical solutions of the SVV approximation (3.16) converge towards the vanishing viscosity solutions of (2.8). For consistency, a brief introduction to the theory framework used in that respect is given. In the setting of linear equations, there are strong standard results available to prove existence and uniqueness of numerical solutions. For finite element methods and spectral methods, the well-known Lax-Milgram lemma is often applied in that regard, following a standard analysis of coercivity properties of the bilinear forms involved and so on. This method of analysis is very versatile and follows a quite standardized procedure.

For nonlinear equations, these tools cannot be used directly however. Instead there are a variety of different mathematical tools, each suited for special equations and numerical methods. For example in the case of conservation laws such as (2.1), a frequently used approach is to construct conservative numerical schemes, like finite volume methods. Then, one tries to prove that the method is monotone, i.e. that a maximum principle applies for the numerical solution. This will imply uniqueness of solutions (see for instance the introduction in [4] and the references therein).

It can be shown that the spectral vanishing viscosity method (3.16) is non-monotone, so convergence towards the entropy solution is not obvious without further analysis. To that end, there is a result available, originally due to Eduard Helly, and later modified for the purposes of PDE analysis. The theorem is proved in [12] and in one dimension it reads

Theorem 1 (Theorem A.8 in [12]: Helly compactness of the family of numerical solutions)**.** *Let $\{u_\eta\} : \mathbb{R} \times [0, \infty) \to \mathbb{R}$ be a family of functions such that for each positive T,*

$$|u_\eta(x,t)| \leq C_T, \;\; (x,t) \in \mathbb{R} \times [0,T]$$

for a constant C_T independent of η. Assume in addition that for all compact $B \subset \mathbb{R}$

and for $t \in [0,T]$ *that*

$$\sup_{|\xi|<|\rho|} \int_B |u_\eta(x+\xi,t) - u_\eta(x,t)| \; dx \leq \nu_{B,T}(|\rho|)$$

for a modulus of continuity ν*. Furthermore, assume for* s *and* t *in* $[0,T]$ *that*

$$\int_B |u_\eta(x,t) - u_\eta(x,s)| \; dx \leq \omega_{B,T}(|t-s|) \text{ as } \eta \to 0$$

for some modulus of continuity ω_T*. Then there exists a sequence* $\eta_j \to 0$ *such that for each* $t \in [0,T]$ *the function* $\{u_{\eta_j}(t)\}$ *converges to a function* $u(t)$ *in* $L^1_{loc}(\mathbb{R})$*. The convergence is in* $C([0,T]; L^1_{loc}(\mathbb{R}))$*.*

To use Theorem 1, we must prove boundedness and time regularity estimates of the numerical solution. We will apply the practical method that is outlined in [2] to do this. The estimates we end up with in this article are similar to those in [2], but some of them will differ slightly. This is both due to the nonlocal term in the equation, which is different here, and also because we do some different choices in the analysis. In the derivations, these differences will be elaborated to some extent. Note that in this article, the analysis is only carried out in one dimension, but it should be feasible to extend the analysis also to arbitrary dimensions, since Theorem 1 is valid also in arbitrary dimensions, cf. [12].

The procedure necessitates a series of additional, intermediate a priori estimates which are derived in the following sections. The chapter concludes with a convergence estimate that utilizes these stability estimates. This estimate establishes the convergence of the numerical solution of the spectral vanishing viscosity method towards the solution of the vanishing viscosity equation (2.8). The ultimate goal would be to prove that this limit function is the unique entropy solution of equation (2.2). This relies on whether the vanishing viscosity method converges towards the entropy solution, but showing that is outside the scope of this project. Jakobsen and Cifani outline a procedure to show this in the linear case in chapter 2, remark 2.6 in [2]. See also [3] for a discussion of the convergence of vanishing viscosity approximations in the case of nonlinear, possibly degenerate diffusion like here.

3.1.1 Prerequisites, required assumptions and results

The notation and constants that are introduced in [2] are used throughout the rest of the article if it is not written otherwise and commented in that specific regard. Also, note that constants may change value throughout an equation or estimate. This is to ease notation, since the precise value of constants is of less relevance in the asymptotic estimates of this chapter.

In the following sections, some additional assumptions and results that are derived or referred to in [2] are required. For the complete derivation of these, the reader is referred to the article, but a summary or reference is given below and otherwise where needed. Below, the most central assumptions and technical results required in the following analysis are stated.

Assumption 1 (SVV constants and variables)**.**

$$\begin{aligned} \epsilon_N &\propto N^{-\theta}, \ \ 0 < \theta < 1 \\ m_N &\propto N^{\frac{\theta}{2}} (\log N)^{-\frac{1}{2}} \\ \epsilon_N &> \frac{8\mathcal{K}_1}{N} \\ \epsilon_N m_N^2 \log N &\leq C \\ Q_N^{j,k}(\mathbf{x}, t) &= \sum_{p=m_N}^{N} \hat{Q}_p^{j,k}(t) \sum_{|\boldsymbol{\xi}|=p} \mathrm{e}^{i\boldsymbol{\xi}\cdot\mathbf{x}} \end{aligned}$$

with

$$\begin{aligned} &\hat{Q}_p^{j,k} \text{ is monotonically increasing with } p \\ &\hat{Q}_p^{j,k} \text{ spherically symmetric} \\ &|\hat{Q}_p^{j,k} - \delta_{ij}| \leq C m_N^2 p^{-2} \ \ \forall p \geq m_N, \end{aligned}$$

where δ_{ij} is the Kronecker delta.

Assumption 2 (Zero in origin)**.**

$$A(0) = 0 \tag{3.1}$$

Assumption 3 (Nondecreasing function)**.**

$$\frac{d}{dr} A(r) \geq 0 \ \ \forall r \in \mathbb{R} \tag{3.2}$$

Assumption 4 (Regularity of A)**.** *Assume that $A(u) \in C^s(\Omega)$ with*

$$s \geq \frac{4}{1-\theta}, \tag{3.3}$$

where $0 < \theta < 1$ is the constant used in Assumption 1.

Assumption 5 (Lipschitz continuity of A)**.** *Assume that A is locally Lipschitz, i.e. $\exists L_A \in \mathbb{R}$ such that*

$$|A(r) - A(s)| \leq L_A |r - s| \ \ \forall r, s \in \mathbb{R} \tag{3.4}$$

Assumption 6 (Zero in origin)**.**

$$f(0) = 0 \tag{3.5}$$

Assumption 7 (Regularity of f)**.** *Assume that $f(u) \in C^s(\Omega)$ with*

$$s \geq \frac{4}{1-\theta}, \tag{3.6}$$

where $0 < \theta < 1$ is the constant used in Assumption 1.

Assumption 8 (Regularity of initial data)**.**

$$u_0 \in L^1(\mathbb{R}) \cap L^\infty(\mathbb{R}) \cap BV(\mathbb{R}) \tag{3.7}$$

Assumption 9 (Bound for initial data)**.** *Assume that there is a constant C such that*

$$\epsilon_N^{2s} \|\partial_x^s u_0\|_{L^2(\Omega)}^2 \leq C \tag{3.8}$$

Lemma 2 (Lemma 3.1 in [2])**.** *For $0 \leq s \leq 2$,*

$$\|\partial_x^s R_N\|_{L^1(\Omega)} \leq C m_N^s \log N, \tag{3.9}$$

and for $0 \leq r \leq s \leq 2$, if $c_N \leq C\epsilon_N m_N^2 \log N \leq \hat{C}$, then for all $p \geq 1$, $\varphi \in L^p(\Omega)$,

$$\epsilon_N \|\partial_x^2 R * \varphi\|_{L^p(\Omega)} \leq c_N \|\varphi\|_{L^p(\Omega)}. \tag{3.10}$$

Lemma 3 (Gagliardo-Nirenberg L^2 bound for a function $g(u)$ (cf. (4.4) in [2]))**.** *Assume that $g \in C^s$. Then there exists a constant $\mathcal{K}_s$ such that*

$$\|\partial_x^s g(u)\|_{L^2(\Omega)} \leq \mathcal{K}_s \|\partial_x^s u\|_{L^2(\Omega)}, \;\; \mathcal{K}_s \leq C \sum_{k=1}^{s} |g|_{C^k} \|u\|_{L^\infty(\Omega)}^{k-1} \tag{3.11}$$

Lemma 4 (Estimate for projection error of a function $g(u)$)**.** *Assume that $g \in C^s$. By using the Fourier series expansion of $(I - P_N)g(u)$ and lemma 3, the inequality*

$$\|\partial_x^r (I - P_N) g(u)\|_{L^2(\Omega)} \leq \frac{\mathcal{K}_s}{N^{s-r}} \|\partial_x^s u\|_{L^2(\Omega)} \tag{3.12}$$

is obtained for $0 \leq r \leq s$.

L^∞ bound for L^2 norm and definition of B_r

First note that as in [2], according to Assumption 1, $c_N = C\epsilon_N m_N^2 \log N \leq \hat{C}$, where $\hat{C}$ is a constant. For $s = 0$, B_0 is defined through the following relation using the above mentioned assumption:

$$c_N \|u\|_{L^2(D_T)}^2 \leq C \|u\|_{L^\infty(D_T)}^2 =: B_0 \tag{3.13}$$

for some constant C. For $r > 0$, where r is an integer,

$$\mathcal{B}_r := C\mathcal{B}_{r-1} \cdot \mathcal{K}_r^2, \tag{3.14}$$

where C is some constant.

3.2 L^2 stability

3.2.1 Energy estimate of numerical solution

In the physical domain, the numerical formulation of the problem reads

$$u_t + \partial_x P_N f(u) - \mathcal{L}[P_N A(u)] - \epsilon_N \partial_x^2 Q_N * u = 0 \tag{3.15}$$

The last term on the left hand side is the spectral vanishing viscosity term. This includes the highest portion of the frequency spectrum, whereas the other part of an otherwise "full" Laplacian operator is left out. To obtain an expression including the Laplacian, one can add and subtract terms. This is also convenient to do for the convection and the fractional diffusion terms. The numerical equation can be written in the equivalent form

$$\begin{aligned} & u_t + \partial_x P_N f(u) - \mathcal{L}[A(u)] - \epsilon_N \partial_x^2 (Q_N * u + R_N * u) \\ & = -\epsilon_N \partial_x^2 R_N * u - \mathcal{L}\big[(I - P_N)A(u)\big] + \partial_x (I - P_N) f(u) \\ \Leftrightarrow \quad & u_t + \partial_x f(u) - \mathcal{L}[A(u)] - \epsilon_N \partial_x^2 u \\ & = -\epsilon_N \partial_x^2 R_N * u - \mathcal{L}\big[(I - P_N)A(u)\big] + \partial_x (I - P_N) f(u) \end{aligned} \tag{3.16}$$

By looking at the terms, one can interpret the equation. On the left hand side, there is the time derivative and the convection and the nonlocal terms. On the right hand side there is the interpolation errors of the convection and nonlocal operators. The second term is the residual part of the Laplacian operator, i.e. the lowest part of the spectrum. The reason for writing the equation like this is that it simplifies the calculations, as will become clear in the following. One can qualitatively expect that the right hand side consists of small terms measured in some norm.

Switching to the Fourier domain is often convenient in the analysis of Fourier-Galerkin methods, but this estimate will be performed in the physical domain.

Lemma 5. *The following energy estimate applies for the numerical solution:*

$$\|u\|^2_{L^2(\Omega)}(t) + 2\epsilon_N \|u_x\|^2_{L^2(D_T)} \leq B_0 + \|u_0\|^2_{L^2(\Omega)}, \tag{3.17}$$

where B_0 is the constant defined in (3.13).

Proof. Begin by multiplying (3.16) with the numerical solution u and integrate over Ω to obtain

$$\begin{aligned} & \int_0^{2\pi} u_t \cdot u \, \mathrm{d}x + \int_0^{2\pi} [f(u)]_x \cdot u \, \mathrm{d}x - \epsilon_N \int_0^{2\pi} u_{xx} \cdot u \, \mathrm{d}x - \int_0^{2\pi} \mathcal{L}\big[A(u)\big] u \, \mathrm{d}x \\ & = -\epsilon_N \int_0^{2\pi} (\partial_x^2 R_N * u) u \, \mathrm{d}x - \int_0^{2\pi} \mathcal{L}\big[(I - P_N)A(u)\big] u \, \mathrm{d}x + \int_0^{2\pi} \partial_x (I - P_N) f(u) \cdot u \, \mathrm{d}x \end{aligned} \tag{3.18}$$

Now consider each term separately, starting with the convection term on the left hand side. Intuitively, this term does not contribute in the energy estimate, and this is indeed the case. Use the chain rule to obtain $[f(u)]_x = f'(u) \cdot u_x$. Define the function $\tilde{f}(u) := f'(u) \cdot u$. This function is continuous from the assumptions made on f, and therefore has an anti-derivative which is labeled $\tilde{F}$. Now consider the contribution from the convection term and use periodicity of u to obtain

$$\begin{aligned}\int_0^{2\pi} [f(u)]_x \cdot u \, \mathrm{d}x &= \int_0^{2\pi} f'(u) \cdot u_x \cdot u \, \mathrm{d}x \\ &= \int_0^{2\pi} \tilde{f} \cdot u_x \, \mathrm{d}x \\ &= \int_0^{2\pi} \tilde{F}'(u) \cdot u_x \, \mathrm{d}x \\ &= \big[\tilde{F}[u](x)\big]_0^{2\pi} = 0\end{aligned} \tag{3.19}$$

Periodicity is also used to evaluate the diffusion term when integrating by parts, such that

$$\begin{aligned}&- \epsilon_N \int_0^{2\pi} u_{xx} \cdot u \, \mathrm{d}x \\ &= -\epsilon_N \Big(\big[u_x \cdot u\big]_0^{2\pi} - \int_0^{2\pi} u_x^2 \, \mathrm{d}x \Big) \\ &= \epsilon_N \|u_x\|^2.\end{aligned} \tag{3.20}$$

The nonlocal term is now treated. Start by defining a bilinear form B as in Definition 2 by

$$B(u,v) := -c_\alpha \int_0^{2\pi} \int_{|y|>0} \frac{\big(u(x+y) - u(x) - \mathbf{1}_{|y|<1} u_x(x)\big) v(x)}{|y|^{1+\alpha}} \, \mathrm{d}y\mathrm{d}x,$$

which by Lemma 16 in the appendix can be written as

$$B(u,v) = c_\alpha \int_0^{2\pi} \int_{|y|>0} \frac{\big(u(x+y) - u(x)\big)\big(v(x+y) - v(x)\big)}{|y|^{1+\alpha}} \, \mathrm{d}y\mathrm{d}x.$$

Then consider B evaluated at u and $A(u)$, which corresponds to the nonlocal term in (3.18). Remembering the assumption that A is an increasing function, cf. Assumption 3, it can be shown that this term is signed. That A is increasing means mathematically that the quotient of change is positive, i.e. that argument change and function change have the same sign. But if the quotient of two numbers is positive, it means that the product also is positive. Hence, $\big(A[u(x+y)] -$

$A[u(x)]\big)\big(u(x+y)-u(x)\big) \geq 0$ for all $x, y \in \mathbb{R}$. Therefore,

$$\begin{aligned} &-\int_0^{2\pi} u \cdot \mathcal{L}[A(u)] \, \mathrm{d}x = B(A(u), u) \\ &= c_\alpha \int_0^{2\pi} \int_{|y|>0} \frac{\big(A[u(x+y)] - A[u(x)]\big)\big(u(x+y)-u(x)\big)}{|y|^{1+\alpha}} \, \mathrm{d}x \geq 0 \end{aligned} \tag{3.21}$$

This means that the nonlocal term on the left hand side can be dropped in the energy estimate.

Turn to the right hand side, and look at the remainder term for the spectral vanishing viscosity. Use the Cauchy-Schwartz inequality and Lemma 2 to estimate the term:

$$\begin{aligned} \epsilon_N \int_0^{2\pi} (\partial_x^2 R_N * u) u \, \mathrm{d}x &\leq \epsilon_N \|u\|_{L^2(\Omega)} \|\partial_x^2 R_N * u\|_{L^2(\Omega)} \\ &\leq c_N \|u\|_{L^2(\Omega)}^2 \end{aligned} \tag{3.22}$$

Then consider the two projection error terms for the possibly nonlinear convection and diffusion fluxes. Now the very properties of the numerical method come into play. At this point, it should become obvious why it is convenient to write the equation in the form of (3.16), and it is useful to switch representation to Fourier space. Consider any function $\varphi(x)$ that has a Fourier representation, and let the representation be given by $\varphi(x) = \sum_{\xi=-\infty}^{\infty} \hat{\varphi}_\xi \mathrm{e}^{i\xi x}$. The residual from the projection of φ is given by

$$(I - P_N)\varphi(x) = (I - P_N) \sum_{\xi=-\infty}^{\infty} \hat{\varphi}_\xi \mathrm{e}^{i\xi x} = \sum_{|\xi|>N} \hat{\varphi}_\xi \mathrm{e}^{i\xi x}.$$

Define the space $X_{\mathcal{F}_N} := \{\mathrm{e}^{i\xi x}\}_{\xi=-N}^{N}$. If the residual is multiplied with a function $u = \sum_{p=-N}^{N} \hat{u}_p \mathrm{e}^{ipx}$, the product will be a series of nonconstant modes, since all modes of $(I - P_N)\varphi(x)$ are outside and all modes of u are inside $X_{\mathcal{F}_N}$. Therefore,

$$\int_0^{2\pi} (I - P_N)\varphi(x) \cdot u(x) \, \mathrm{d}x = \sum_{|\xi|>N} \sum_{p=-N}^{N} \int_0^{2\pi} \hat{\varphi}_\xi \cdot \hat{u}_p \cdot \mathrm{e}^{i(\xi+p)x} \, \mathrm{d}x = 0. \tag{3.23}$$

This fact is valid both in the case $\varphi = f(u)$ and $\varphi = \mathcal{L}[A(u)]$. Therefore, the contributions integrated against u, of both projection residuals, are zero. Note that this would not necessarily be the case if the residual were integrated against a nonlinear function of u or another function with components for higher frequencies than what is included in $X_{\mathcal{F}_N}$. This is important to remember in a more general setting, like in the estimate in Section 3.3. Note that $\int_0^{2\pi} u_t \cdot u \, \mathrm{d}x = \frac{1}{2}\frac{\mathrm{d}}{\mathrm{d}t}\|u\|_{L^2(\Omega)}^2$.

By applying (3.19)-(3.23), it is clear from (3.18) that

$$\begin{aligned}\frac{1}{2}\frac{\mathrm{d}}{\mathrm{d}t}\|u\|^2_{L^2(\Omega)} + \epsilon_N\|u_x^2\|_{L^2(\Omega)} &= \int_0^{2\pi} \mathcal{L}\big[A(u)\big]u \ \mathrm{d}x - \epsilon_N \int_0^{2\pi} (\partial_x^2 R_N * u)u \ \mathrm{d}x \\ &+ \int_0^{2\pi} (I - P_N)[f(u)]_x \cdot u \ \mathrm{d}x - \int_0^{2\pi} (I - P_N)\mathcal{L}[A(u)] \cdot u \ \mathrm{d}x \\ &\leq -\epsilon_N \int_0^{2\pi} (\partial_x^2 R_N * u)u \ \mathrm{d}x \leq c_N\|u\|^2_{L^2(\Omega)}\end{aligned} \tag{3.24}$$

With (3.22) in mind and applying the Cauchy-Schwartz inequality, (3.24) gives the inequality

$$\frac{1}{2}\frac{\mathrm{d}}{\mathrm{d}t}\|u\|^2_{L^2(\Omega)} + \epsilon_N\|u_x\|^2_{L^2(\Omega)} \leq c_N\|u\|^2_{L^2(\Omega)} \tag{3.25}$$

An energy estimate is now derived for the numerical solution by using 3.13 and integrating (3.25) in time:

$$\begin{aligned}\frac{1}{2}\int_0^T \frac{\mathrm{d}}{\mathrm{d}t}\|u\|^2_{L^2(\Omega)} \ \mathrm{d}t + \epsilon_N \int_0^T \|u_x\|^2_{L^2(\Omega)} \ \mathrm{d}x &\leq \int_0^T c_N\|u\|^2_{L^2(\Omega)} \ \mathrm{d}t \\ \Leftrightarrow \|u\|^2_{L^2(\Omega)} + 2\epsilon_N\|u_x\|^2_{L^2(D_T)} &\leq B_0 + \|u_0\|^2_{L^2(\Omega)}\end{aligned} \tag{3.26}$$

□

3.2.2 Energy estimate of derivatives of numerical solution

Lemma 6. *The following energy estimate holds for the derivatives of the numerical solution:*

$$\epsilon_N^{2r}\|\partial_x^r u\|^2_{L^2(\Omega)}(t) + \epsilon_N^{2r+1}\|\partial_x^{r+1}u\|^2_{L^2(D_T)} \leq B_r + \epsilon_N^{2r}\|\partial_x^r u_0\|^2_{L^2(\Omega)}, \ \ r \geq 0 \tag{3.27}$$

for a constant B_r as defined in 3.14.

Proof. The base case $r = 0$ is shown to hold in Lemma 5. Note that in the base case, the estimate is slightly stronger than what is implied by (3.27), since there is a factor $2\epsilon_N$ in front of the second term on the left hand side of equation (3.17). Instead of applying u, equation (3.16) is multiplied with $\partial_x^{2r}u$ for an arbitrary positive integer r and integrated over Ω. The goal is to establish an L^2 estimate for the rth derivative of the numerical solution. In the evaluation of terms, integration by parts is applied repeatedly together with the periodicity of u to obtain the desired form. Start with the time derivative on the left hand side:

$$\begin{aligned}\int_0^{2\pi} \partial_x^{2r} u \cdot u_t \, \mathrm{d}x &= \Big(\underbrace{[\partial_x^{2r-1} u \cdot u_t]_0^{2\pi}}_{=0} - \int_0^{2\pi} \partial_x^{2r-1} u \cdot \partial_t \partial_x u \, \mathrm{d}x \Big) \\ &= ... = (-1)^r \int_0^{2\pi} \partial_x^r u \cdot \partial_t \partial_x^r u \, \mathrm{d}x \\ &= (-1)^r \frac{1}{2} \frac{\mathrm{d}}{\mathrm{d}t} \int_0^{2\pi} (\partial_x^r u)^2 \, \mathrm{d}x \\ &= (-1)^r \frac{1}{2} \frac{\mathrm{d}}{\mathrm{d}t} \|\partial_x^r u\|^2_{L^2(\Omega)}.\end{aligned} \tag{3.28}$$

Further,

$$\begin{aligned}-\epsilon_N \int_0^{2\pi} \partial_x^2 u \cdot \partial_x^{2r} u \, \mathrm{d}x &= -\epsilon_N \Big(\underbrace{[\partial_x^2 u \cdot \partial_x^{2r-1} u]_0^{2\pi}}_{=0} - \int_0^{2\pi} \partial_x^{2+1} u \cdot \partial_x^{2r-1} u \, \mathrm{d}x \Big) \\ &= ... = -(-1)^{r-1} \epsilon_N \int_0^{2\pi} \partial_x^{2+r-1} u \cdot \partial_x^{2r-(r-1)} u \, \mathrm{d}x \\ &= (-1)^r \epsilon_N \|\partial_x^{r+1} u\|^2_{L^2(\Omega)}\end{aligned} \tag{3.29}$$

Moving on to the nonlocal operator, it is now more complex to treat the contribution than in the case $r = 0$. Above, it was possible to derive that

$$-\int_0^{2\pi} \mathcal{L}[A(u)] u \, \mathrm{d}x = B\big(A(u), u\big) \geq 0.$$

But in this case, it is not known whether $\partial_x^{2r} u$ increases or decreases, so this argument cannot be used directly. It seems convenient to switch to the Fourier representation. Due to Lemma 10, $\mathcal{L}[A(u)]$ can be represented by

$$\mathcal{L}[A(u)] = -\sum_{\xi=-\infty}^{\infty} C_\alpha |\xi|^\alpha \hat{A}_\xi \mathrm{e}^{i\xi x},$$

where C_α is some positive constant. Observe that

$$\partial_x^{2r} u = (-1)^r \sum_{\xi=-N}^{N} |i\xi|^{2r} \hat{u}_\xi \mathrm{e}^{i\xi x}.$$

Therefore,

$$\begin{aligned}\int_0^{2\pi} -\mathcal{L}[A(u)]\cdot\partial_x^{2r}u\ \mathrm{d}x &= (-1)^{r+1}(-C_\alpha)\int_0^{2\pi}\sum_{p=-\infty}^{\infty}\sum_{\xi=-N}^{N}|p|^\alpha\hat{A}_p\cdot|\xi|^{2r}\hat{u}_\xi \mathrm{e}^{i(p+\xi)x}\ \mathrm{d}x\\ &= 2\pi(-1)^{r+2}C_\alpha\sum_{\xi=-N}^{N}|\xi|^\alpha\hat{A}_{-\xi}\cdot|\xi|^{2r}\hat{u}_\xi\\ &\overset{\geq}{\leq} 2\pi(-1)^r C_\alpha\sum_{\xi=-N}^{N}|\xi|^\alpha\hat{A}_{-\xi}\cdot\hat{u}_\xi\\ &= (-1)^r\int_0^{2\pi}\mathcal{L}[A(u)]u\ \mathrm{d}x\ \overset{\geq}{\leq}\ 0\end{aligned} \tag{3.30}$$

The convection term gives a contribution

$$\begin{aligned}\int_0^{2\pi}\partial_x f(u)\cdot\partial_x^{2r}u\ \mathrm{d}x &= \underbrace{[f(u)\cdot\partial_x^{2r}u]_0^{2\pi}}_{=0} - \int_0^{2\pi}\partial_x^2 f(u)\cdot\partial_x^{2r-1}u\ \mathrm{d}x\\ &= \ldots = \int_0^{2\pi}\partial_x^{1+(r-1)}f(u)\cdot\partial_x^{2r-(r-1)}u\ \mathrm{d}x\\ &\overset{\text{C-S}}{\leq} \|\partial_x^{r+1}u\|_{L^2(\Omega)}\cdot\|\partial_x^r f(u)\|_{L^2(\Omega)}\\ &\overset{3}{\leq} \mathcal{K}_r\|\partial_x^{r+1}u\|_{L^2(\Omega)}\cdot\|\partial_x^r u\|_{L^2(\Omega)}\end{aligned} \tag{3.31}$$

On the right hand side, consider the residual from the SVV term. The term is integrated repeatedly by parts, and Lemma 2 is used together with the Cauchy-Schwartz' inequality to give a bound:

$$\begin{aligned}\epsilon_N\int_0^{2\pi}(\partial_x^2 R * u)\cdot\partial_x^{2r}u\ \mathrm{d}x &= \epsilon_N\Big(\underbrace{[(\partial_x^2 R * u)\cdot\partial_x^{2r-1}u]}_{=0} - \int_0^{2\pi}(\partial_x^2 R * \partial_x u)\cdot\partial_x^{2r-1}u\ \mathrm{d}x\Big)\\ &= \ldots = (-1)^r\epsilon_N\int_0^{2\pi}(\partial_x^2 R * \partial_x^r u)\cdot\partial_x^r u\ \mathrm{d}x\\ &\overset{\text{C-S}}{\overset{\geq}{\leq}} (-1)^r\epsilon_N\|(\partial_x^2 R * \partial_x^r u)\|_{L^2(\Omega)}\cdot\|\partial_x^r u\|_{L^2(\Omega)}\\ &\overset{2}{\overset{\geq}{\leq}} (-1)^r c_N\|\partial_x^r u\|^2_{L^2(\Omega)}\ \overset{\geq}{\leq}\ C\|\partial_x^r u\|^2_{L^2(\Omega)}\end{aligned} \tag{3.32}$$

The only terms remaining are the contributions from the projection residuals. Since the discrete space is closed under differentiation, it follows that $\partial_x^{2r}u \in X_{\mathcal{F}_N}$.

Therefore, by using the same argument as in Section 3.2.1, it can be concluded that

$$\begin{aligned}\int_0^{2\pi} \mathcal{L}\big[(I-P_N)A(u)\big]\cdot\partial_x^{2r}u\ \mathrm{d}x = 0\\ \int_0^{2\pi} \partial_x(I-P_N)f(u)\cdot\partial_x^{2r}u\ \mathrm{d}x = 0\end{aligned} \tag{3.33}$$

Also, assume that $C+\frac{\mathcal{K}_r^2}{2\epsilon_N}\leq\frac{\mathcal{K}_r^2}{\epsilon_N}$. By using this, (3.28)-(3.33), Young's inequality and integrating in time, one can obtain an estimate for the derivatives of the numerical solution by

$$\begin{aligned}&\frac{1}{2}\frac{\mathrm{d}}{\mathrm{d}t}\|\partial_x^r u\|^2_{L^2(\Omega)}+\epsilon_N\|\partial_x^{r+1}u\|^2_{L^2(\Omega)}\leq C\|\partial_x^r u\|^2_{L^2(\Omega)}+\mathcal{K}_r\|\partial_x^{r+1}u\|_{L^2(\Omega)}\cdot\|\partial_x^r u\|_{L^2(\Omega)}\\
&\Rightarrow\frac{1}{2}\frac{\mathrm{d}}{\mathrm{d}t}\|\partial_x^r u\|^2_{L^2(\Omega)}+\epsilon_N\|\partial_x^{r+1}u\|^2_{L^2(\Omega)}\leq C\|\partial_x^r u\|^2_{L^2(\Omega)}\\
&\qquad+\mathcal{K}_r\Big(\frac{\mathcal{K}_r}{2\epsilon_N}\|\partial_x^r u\|^2_{L^2(\Omega)}+\frac{\epsilon_N}{2\mathcal{K}_r}\|\partial_x^{r+1}u\|^2_{L^2(\Omega)}\Big)\\
&\Rightarrow\frac{1}{2}\frac{\mathrm{d}}{\mathrm{d}t}\|\partial_x^r u\|^2_{L^2(\Omega)}+\frac{\epsilon_N}{2}\|\partial_x^{r+1}u\|^2_{L^2(\Omega)}\leq\Big(C+\frac{\mathcal{K}_r^2}{2\epsilon_N}\Big)\|\partial_x^r u\|^2_{L^2(\Omega)}\leq\frac{K_r^2}{\epsilon_N}\|\partial_x^r u\|^2_{L^2(\Omega)}\\
&\Rightarrow\|\partial_x^r u\|^2_{L^2(\Omega)}(t)+\epsilon_N\|\partial_x^{r+1}u\|^2_{L^2(D_T)}\leq\frac{2\mathcal{K}_r^2}{\epsilon_N}\|\partial_x^r u\|^2_{L^2(D_T)}+\|\partial_x^r u_0\|^2_{L^2(\Omega)}\end{aligned} \tag{3.34}$$

From the induction hypothesis (3.27) one can derive the relation

$$\|\partial_x^s u\|^2_{L^2(D_T)}\leq C\mathcal{B}_{s-1}\epsilon_N^{-(2s-1)} \tag{3.35}$$

by inserting $r=s-1$ into the inequality and using Assumption 9. From (3.35) and the definition of $\mathcal{B}_r$, one can obtain that

$$\frac{2\mathcal{K}_r^2}{\epsilon_N}\|\partial_x^r u\|^2_{L^2(D_T)}\leq\frac{2\mathcal{K}_r^2}{\epsilon_N}\cdot C\mathcal{B}_{r-1}\epsilon_N^{-(2r-1)}=\epsilon_N^{-2r}\mathcal{B}_r,$$

which gives the desired result by multiplying the equation with ϵ_N^{2r} □

3.3 L^∞ estimate

The next step towards proving compactness is to show boundedness of the numerical solutions. In that regard, a stability estimate is given in

Lemma 7 (L^∞ stability of numerical solutions)**.** *In finite time, more specifically for $t<C\ln N$,*

$$\|u\|_{L^\infty(\Omega)}\leq C\|u_0\|_{L^\infty(\Omega)}. \tag{3.36}$$

Proof. Equation (3.16) is multiplied with a test function as in the energy estimate. It is convenient to use the function pu^{p-1}, where u is the numerical solution from the method, and p is an even number. Observe that

$$\begin{aligned}\frac{\mathrm{d}}{\mathrm{d}t}\|u\|^p_{L^p(\Omega)} &= p\|u\|^{p-1}_{L^p(\Omega)} \cdot \frac{\mathrm{d}}{\mathrm{d}t}\|u\|_{L^p(\Omega)} \\ &= \frac{\mathrm{d}}{\mathrm{d}t}\int_0^{2\pi} |u|^p \,\mathrm{d}x \\ &= \int_0^{2\pi} p|u|^{p-1} \cdot \frac{\mathrm{d}}{\mathrm{d}t}|u| \,\mathrm{d}x \\ &= p\int_0^{2\pi} u^{p-1} \cdot u_t \,\mathrm{d}x\end{aligned} \tag{3.37}$$

By performing spatial integration over Ω, the equation reads

$$\begin{aligned}&\int_0^{2\pi} pu^{p-1} \cdot u_t \,\mathrm{d}x - \epsilon_N \int_0^{2\pi} u_{xx} \cdot pu^{p-1} \,\mathrm{d}x + \int_0^{2\pi} \partial_x f(u) \cdot pu^{p-1} \,\mathrm{d}x \\ &\quad - \int_0^{2\pi} \mathcal{L}[A(u)] \cdot pu^{p-1} \,\mathrm{d}x = \epsilon_N \int_0^{2\pi} (\partial_x^2 R_N * u) pu^{p-1} \,\mathrm{d}x \\ &\quad + \int_0^{2\pi} \partial_x (I - P_N) f(u) \cdot pu^{p-1} \,\mathrm{d}x - \int_0^{2\pi} \mathcal{L}[(I - P_N)A(u)] \cdot pu^{p-1} \,\mathrm{d}x\end{aligned}$$

Each term on the left hand side is now treated separately, starting with the nonlocal term. Just as in the energy estimate, the contribution from the nonlocal operator can be rewritten in terms of the bilinear form. This is justified by using Lemma 16 with $A(u)$ and $v = u^{p-1}$ inserted into the bilinear form:

$$\begin{aligned}&-\int_0^{2\pi} u^{p-1}\mathcal{L}[A(u)] \,\mathrm{d}x = B(A(u), u^{p-1}) \\ &= c_\alpha \int_0^{2\pi} \int_{|y|>0} \frac{\big(A[u(x+y)] - A[u(x)]\big)\big(u^{p-1}(x+y) - u^{p-1}(x)\big)}{|y|^{1+\alpha}} \,\mathrm{d}y\mathrm{d}x \geq 0\end{aligned} \tag{3.38}$$

since both are increasing functions. Thus, this term is non-negative. Proceed to the convection term on the left hand side and define $F(u)$ to be the antiderivative of $f'(u) \cdot u^{p-1}$. This function exists since $f'(u)$ and u^{p-1} are assumed to be continuous. Therefore, due to periodicity of u,

$$\begin{aligned}p\int_0^{2\pi} \partial_x f(u) \cdot u^{p-1} \,\mathrm{d}x &= p\int_0^{2\pi} \underbrace{f_u(u) \cdot u^{p-1}}_{F(u)} \cdot u_x \,\mathrm{d}x \\ &= \int_0^{2\pi} F'(u) \cdot u_x \,\mathrm{d}x = \Big[F\big(u(x)\big)\Big]_0^{2\pi} = 0.\end{aligned} \tag{3.39}$$

As for the viscosity term, because p is even and u and u_x are periodic, one can show that this is non-negative:

$$-\epsilon_N \int_0^{2\pi} u_{xx}\cdot u^{p-1}\,\mathrm{d}x = -\epsilon_N\Big(\underbrace{\left[u_x\cdot u^{p-1}\right]_0^{2\pi}}_{=0} - (p-1)\int_0^{2\pi} u_x^2\cdot u^{p-2}\,\mathrm{d}x\Big) \geq 0\ \forall p\in 2\mathbb{N} \tag{3.40}$$

Observe that in this case, the result (3.23) that was used in the energy estimate, cannot be applied to the projection errors. The reason is that the function against which the equation is multiplied, pu^{p-1}, is potentially nonlinear. Therefore, the modes of the test function do not necessarily reside in $X_{\mathcal{F}_N}$. Now it is possible to proceed similarly as in [2], except that in this case there is also the projection error term from the nonlocal operator. The relations (3.38), (3.39), (3.40) and the Hölder inequality with p and $q=\frac{p}{p-1}$ together yield the inequality

$$\begin{aligned}\frac{\mathrm{d}}{\mathrm{d}t}\|u\|_{L^p(\Omega)}^p &\leq p\int_0^{2\pi} u^{p-1}\Big(\partial_x^2 R_N * u + \partial_x(I-P_N)f(u) + \mathcal{L}[(I-P_N)A(u)]\Big)\,\mathrm{d}x\\ &\leq p\|u^{p-1}\|_{L^{\frac{p}{p-1}}(\Omega)}\cdot\Big(\|\partial_x^2 R_N * u\|_{L^p(\Omega)} + \|\partial_x(I-P_N)f(u)\|_{L^p(\Omega)} +\\ &\quad \|\mathcal{L}[(I-P_N)A(u)]\|_{L^p(\Omega)}\Big)\end{aligned} \tag{3.41}$$

Observe that $\|u^{p-1}\|_{L^{\frac{p}{p-1}}(\Omega)} = \|u\|_{L^p(\Omega)}^{p-1}$. Divide (3.41) by $p\|u\|_{L^p(\Omega)}^{p-1}$, apply (3.37) and take the limit as $p\to\infty$, which gives

$$\begin{aligned}&\Rightarrow \frac{\frac{\mathrm{d}}{\mathrm{d}t}\|u\|_{L^p(\Omega)}^p}{p\|u\|_{L^p(\Omega)}^{p-1}} = \frac{p\|u\|_{L^p(\Omega)}^{p-1}\cdot\frac{\mathrm{d}}{\mathrm{d}t}\|u\|_{L^p(\Omega)}}{p\|u\|_{L^p(\Omega)}^{p-1}}\\ &\quad\leq \frac{p\|u^{p-1}\|_{L^{\frac{p}{p-1}}(\Omega)}}{p\|u\|_{L^p(\Omega)}^{p-1}}\\ &\qquad\cdot\Big(\|\partial_x^2 R_N * u\|_{L^p(\Omega)} + \|\partial_x(I-P_N)f(u)\|_{L^p(\Omega)} + \|\mathcal{L}[(I-P_N)A(u)]\|_{L^p(\Omega)}\Big)\\ &\Rightarrow \lim_{p\to\infty}\frac{\mathrm{d}}{\mathrm{d}t}\|u\|_{L^p(\Omega)} \leq \lim_{p\to\infty}\Big(\|\partial_x^2 R_N * u\|_{L^p(\Omega)} + \|\partial_x(I-P_N)f(u)\|_{L^p(\Omega)}\\ &\quad + \|\mathcal{L}[(I-P_N)A(u)]\|_{L^p(\Omega)}\Big)\\ &\Rightarrow \frac{\mathrm{d}}{\mathrm{d}t}\|u\|_{L^\infty(\Omega)} \leq \|\partial_x^2 R_N * u\|_{L^\infty(\Omega)} + \|\partial_x(I-P_N)f(u)\|_{L^\infty(\Omega)} + \|\mathcal{L}[(I-P_N)A(u)]\|_{L^\infty(\Omega)}\end{aligned} \tag{3.42}$$

The projection errors and the residual from the spectral vanishing viscosity term must be estimated. To estimate the contribution from the nonlocal operator, it is convenient to rewrite the operator $\mathcal{L}$. For brevity, introduce the function $\varphi = (I-P_N)A(u)$. Let $r>0$ be a variable for the moment. Lemma (14) is applied

to the operator, but there is a natural distinction between the cases $0 < \alpha < 1$ and $1 \leq \alpha < 2$ in the following. The estimate of the nonlocal operator will be performed in both cases. First consider the case $0 < \alpha < 1$:

$$\begin{aligned}
\|\mathcal{L}[\varphi(x)]\|_{L^\infty(\Omega)} &= \left\| P.V. \int_{|y|<r} \varphi(x+y) - \varphi(x) \, \mathrm{d}\mu(y) + \int_{|y|>r} \varphi(x+y) - \varphi(x) \, \mathrm{d}\mu(y) \right\|_{L^\infty(\Omega)} \\
&= \left\| P.V. \int_{|y|<r} \int_0^1 y \cdot \varphi_x(x+\theta y) \, \mathrm{d}\theta \mathrm{d}\mu(y) + \int_{|y|>r} \varphi(x+y) - \varphi(x) \, \mathrm{d}\mu(y) \right\|_{L^\infty(\Omega)} \\
&\leq \|\varphi_x\|_{L^\infty(\Omega)} \cdot \underbrace{\int_{|y|<r} y \, \mathrm{d}\mu(y)}_{=:\psi_1(r)} + 2\|\varphi\|_{L^\infty(\Omega)} \cdot \underbrace{\int_{|y|>r} \mathrm{d}\mu(y)}_{=:\psi_2(r)} \\
&\leq C_1(r)\|\varphi_x\|_{L^\infty(\Omega)} + C_0(r)\|\varphi\|_{L^\infty(\Omega)}
\end{aligned} \tag{3.43}$$

for some $C_0(r), C_1(r)$ depending on r. Note that both ψ_1 and ψ_2 are well-defined due to Lemma 1, point iii) For $1 \leq \alpha < 2$, the Taylor expansion must be made to second order, and to do so, Lemma , point ii) is used, and the estimate becomes

$$\begin{aligned}
&\|\mathcal{L}[\varphi(x)]\|_{L^\infty(\Omega)} \\
&= \left\| P.V. \int_{|y|<r} \varphi(x+y) - \varphi(x) - y \cdot \varphi_x(x) \, \mathrm{d}\mu(y) + \int_{|y|>r} \varphi(x+y) - \varphi(x) \, \mathrm{d}\mu(y) \right\|_{L^\infty(\Omega)} \\
&= \left\| P.V. \int_{|y|<r} \int_0^1 \int_0^\theta y^2 \cdot \varphi_x(x+ty) \, \mathrm{d}t\mathrm{d}\theta\mathrm{d}\mu(y) + \int_{|y|>r} \varphi(x+y) - \varphi(x) \, \mathrm{d}\mu(y) \right\|_{L^\infty(\Omega)} \\
&\leq \|\varphi_{xx}\|_{L^\infty(\Omega)} \cdot \underbrace{\int_{|y|<r} y \, \mathrm{d}\mu(y)}_{=:\psi_1(r)} + 2\|\varphi\|_{L^\infty(\Omega)} \cdot \underbrace{\int_{|y|>r} \mathrm{d}\mu(y)}_{=:\psi_2(r)} \\
&\leq C_2(r)\|\varphi_{xx}\|_{L^\infty(\Omega)} + C_0(r)\|\varphi\|_{L^\infty(\Omega)}
\end{aligned} \tag{3.44}$$

for some $C_0(r), C_2(r)$ depending on r. As before, ψ_2 is well-defined. ψ_1 is well-defined since $\alpha < 2$ and

$$|\psi_1(r)| \leq c_\alpha \int_{|y|<r} \frac{|y|^2}{|y|^{1+\alpha}} \, \mathrm{d}y < \infty.$$

This is not an optimal estimate, since r is not tuned to minimize the bound. To that end, the integrals in (3.43) and (3.44) are calculated:

$$\begin{aligned}
\int_{|y|>r} \mathrm{d}\mu &= C \int_r^\infty \frac{\mathrm{d}y}{y^{1+\alpha}} = \frac{Cr^{-\alpha}}{\alpha} \\
\int_{|y|<r} |y| \mathrm{d}\mu &= C \int_0^r \frac{y\mathrm{d}y}{y^{1+\alpha}} = \frac{Cr^{1-\alpha}}{1-\alpha} \\
\int_{|y|<r} y^2 \mathrm{d}\mu &= C \int_0^r \frac{y^2\mathrm{d}y}{y^{1+\alpha}} = \frac{Cr^{2-\alpha}}{2-\alpha}
\end{aligned}$$

Define $f_1(r) := \frac{Cr^{-\alpha}}{\alpha}\|\varphi\|_{L^\infty(\Omega)} + \frac{Cr^{1-\alpha}}{1-\alpha}\|\varphi_x\|_{L^\infty(\Omega)}$ and $f_2(r) := \frac{Cr^{-\alpha}}{\alpha}\|\varphi\|_{L^\infty(\Omega)} + \frac{Cr^{2-\alpha}}{2-\alpha}\|\varphi_{xx}\|_{L^\infty(\Omega)}$. Minimize these with respect to r to obtain

$$\begin{aligned}\frac{\mathrm{d}f_1}{\mathrm{d}r} = 0 \Rightarrow r_{\min} &= \frac{\|\varphi\|_{L^\infty(\Omega)}}{\|\varphi_x\|_{L^\infty(\Omega)}},\\ \frac{\mathrm{d}f_2}{\mathrm{d}r} = 0 \Rightarrow r_{\min} &= \sqrt{\frac{\|\varphi\|_{L^\infty(\Omega)}}{\|\varphi_{xx}\|_{L^\infty(\Omega)}}}.\end{aligned}$$

These minimized expressions can be inserted into the estimate, which gives

$$\begin{aligned}f_1(r_{\min}) &= C\frac{\|\varphi_x\|_{L^\infty(\Omega)}}{1-\alpha}\cdot\frac{\|\varphi\|^{1-\alpha}_{L^\infty(\Omega)}}{\|\varphi_x\|^{1-\alpha}_{L^\infty(\Omega)}} + C\frac{\|\varphi\|_{L^\infty(\Omega)}}{\alpha}\cdot\frac{\|\varphi_x\|^{\alpha}_{L^\infty(\Omega)}}{\|\varphi\|^{\alpha}_{L^\infty(\Omega)}}\\ &= C\|\varphi\|^{1-\alpha}_{L^\infty(\Omega)}\|\varphi_x\|^{\alpha}_{L^\infty(\Omega)},\\ f_2(r_{\min}) &= C\frac{\|\varphi_{xx}\|_{L^\infty(\Omega)}}{2-\alpha}\cdot\Big(\frac{\|\varphi\|_{L^\infty(\Omega)}}{\|\varphi_{xx}\|_{L^\infty(\Omega)}}\Big)^{\frac{2-\alpha}{2}} + C\frac{\|\varphi\|_{L^\infty(\Omega)}}{\alpha}\cdot\Big(\frac{\|\varphi_x\|_{L^\infty(\Omega)}}{\|\varphi\|_{L^\infty(\Omega)}}\Big)^{\frac{\alpha}{2}}\\ &= C\|\varphi\|^{1-\frac{\alpha}{2}}_{L^\infty(\Omega)}\|\varphi_{xx}\|^{\frac{\alpha}{2}}_{L^\infty(\Omega)}.\end{aligned} \tag{3.45}$$

What remains now is to obtain an estimate of the quantities $\|\partial_x(I-P_N)f(u)\|_{L^\infty(\Omega)}$, $\|(I-P_N)A(u)\|_{L^\infty(\Omega)}$, $\|\partial_x(I-P_N)A(u)\|_{L^\infty(\Omega)}$ and $\|\partial_x^2(I-P_N)A(u)\|_{L^\infty(\Omega)}$. These are estimated using the energy estimate for the derivatives of the solution in Lemma 6, the Gagliardo-Nirenberg Lemma 4, Assumption 1 and the Sobolev inequality $\|\partial_x^r\varphi\|_{L^\infty(\Omega)} \le \|\varphi^{r+[\frac{3}{2}]}\|_{L^2(\Omega)} = \|\varphi^{r+2}\|_{L^2(\Omega)}$ in one dimension (cf. theorem 6, chapter 5 in [6]). Consider a function $g(u)$ with smoothness s, and let $0 \le r \le s$. The energy estimate gives

$$\|\partial_x^s u\|^2_{L^2(\Omega)} \le C\epsilon_N^{-2s} \text{ and } \|\partial_x^{s+1}u\|^2_{L^2(D_T)} \le C\epsilon_N^{-(2s+1)}$$

The Gagliardo-Nirenberg inequality yields

$$\begin{aligned}\|\partial_x^r(I-P_N)g(u)\|_{L^2(\Omega)} &\le \frac{\mathcal{K}_s}{N^{s-r}}\|\partial_x^s u\|_{L^2(\Omega)}\\ &\le \frac{\mathcal{K}_s}{N^{s-r}}\cdot C\epsilon_N^{-s}\\ &\le \frac{\mathcal{K}_s}{N^{s-r}}\cdot CN^{\theta s} = C\mathcal{K}_s N^{r-s(1-\theta)}\\ &\le \mathcal{B}_s N^{r-s(1-\theta)}\end{aligned} \tag{3.46}$$

and similarly,

$$\|\partial_x^r(I-P_N)g(u)\|_{L^2(D_T)} \le \mathcal{B}_s N^{r-s(1-\theta)-\frac{\theta}{2}}. \tag{3.47}$$

Hence in the maximum norm,

$$\begin{aligned}\|\partial_x^r(I-P_N)g(u)\|_{L^\infty(\Omega)} &\le \mathcal{B}_s N^{r+2-s(1-\theta)}\\ \|\partial_x^r(I-P_N)g(u)\|_{L^\infty(D_T)} &\le \mathcal{B}_s N^{r+2-s(1-\theta)-\frac{\theta}{2}}.\end{aligned} \tag{3.48}$$

Let s_f and s_A denote the regularity of f and A respectively, and let the regularity constants satisfy assumptions (7) and (4). Now apply (3.48), substituting g with f and A and their respective smoothness constants. The interpolation error of the convection operator is then bounded by

$$\begin{aligned}\|\partial_x(I-P_N)f(u)\|_{L^\infty(\Omega)} &\leq B_{s_f}N^{-\left(1+2-\frac{2}{(1-\theta)}\cdot(1-\theta)\right)} = \frac{B_{s_f}}{N} \\ &\leq \frac{C_1}{N}\prod_{k=1}^{s_f}\mathcal{K}_{s_f} \leq \frac{\hat{C}_1}{N}\|u\|_{L^\infty(\Omega)}^{\frac{s_f^2}{2}}.\end{aligned} \tag{3.49}$$

With the required regularity of A, a bound is obtained for the minimized expressions in (3.45):

$$\begin{aligned}&0<\alpha<1 \Rightarrow C\|(I-P_N)A(u)\|_{L^\infty(\Omega)}^{1-\alpha}\|\partial_x(I-P_N)A(u)\|_{L^\infty(\Omega)}^{\alpha} \\ &\leq C(B_{s_A}N^{-s_A(1-\theta)})^{1-\alpha}(B_{s_A}N^{1-s_A(1-\theta)})^{\alpha} \\ &\leq \frac{C_A}{N^{s_A(1-\theta)(1-\alpha)+\alpha(s_A(1-\theta)-1)}}\|u\|_{L^\infty(\Omega)}^{\frac{s_A^2}{2}} = \frac{C_A}{N^{s_A(1-\theta)-\alpha}}\|u\|_{L^\infty(\Omega)}^{\frac{s_A^2}{2}},\end{aligned}$$

and

$$\begin{aligned}&1<\alpha<2 \Rightarrow C\|(I-P_N)A(u)\|_{L^\infty(\Omega)}^{1-\frac{\alpha}{2}}\|\partial_{xx}(I-P_N)A(u)\|_{L^\infty(\Omega)}^{\frac{\alpha}{2}} \\ &\leq C(B_{s_A}N^{-s_A(1-\theta)})^{1-\frac{\alpha}{2}}(B_{s_A}N^{2-s_A(1-\theta)})^{\frac{\alpha}{2}} \\ &\leq \frac{C_A}{N^{s_A(1-\theta)(1-\frac{\alpha}{2})+\frac{\alpha}{2}(s_A(1-\theta)-2)}}\|u\|_{L^\infty(\Omega)}^{\frac{s_A^2}{2}} = \frac{C_A}{N^{s_A(1-\theta)-\alpha}}\|u\|_{L^\infty(\Omega)}^{\frac{s_A^2}{2}}.\end{aligned}$$

Observe that the estimate turns out to be the same in both cases. To complete the estimate, apply Lemma 2 to obtain a bound for the residual of the SVV term:

$$\|\partial_x^2 R_N * u\|_{L^\infty(\Omega)} \overset{\text{Lemma2}}{\leq} c_N\|u\|_{L^\infty(\Omega)} \leq C\|u\|_{L^\infty(\Omega)} \tag{3.50}$$

Assume for simplicity that $s_f = s_A = s$. This assumption, together with the above results, yield the following inequality, valid for $0<\alpha<2$:

$$\begin{aligned}\frac{\mathrm{d}}{\mathrm{d}t}\|u\|_{L^\infty(\Omega)} &\leq c_N\|u\|_{L^\infty(\Omega)} + \frac{C_1}{N}\|u\|_{L^\infty(\Omega)}^{\frac{s^2}{2}} + \frac{C_2}{N^{s(1-\theta)-\alpha}}\|u\|_{L^\infty(\Omega)}^{\frac{s^2}{2}} \\ &= c_N\|u\|_{L^\infty(\Omega)} + \left(\frac{C_1}{N} + \frac{C_2}{N^{s(1-\theta)-\alpha}}\right)\|u\|_{L^\infty(\Omega)}^{\frac{s^2}{2}},\ \ 0<\alpha<2\end{aligned} \tag{3.51}$$

This is an inhomogeneous nonlinear ordinary differential inequality with initial condition $\|u\|_{L^\infty(\Omega)}(0) = \|u_0\|_{L^\infty(\Omega)}$. To solve it, proceed as in [2]: define $y :=$

$\mathrm{e}^{-c_N t}\|u\|_{L^\infty(\Omega)}$ and insert into (3.51). Observe that $y \geq 0$ and obtain

$$\begin{aligned}
(3.51) \Rightarrow \frac{\mathrm{d}y}{\mathrm{d}t} &\leq \Big(\frac{C_1}{N} + \frac{C_2}{N^{s(1-\theta)-\alpha}}\Big) y^{\frac{s^2}{2}} \mathrm{e}^{c_N\left(\frac{s^2}{2}-1\right)t} \\
\Rightarrow \int_{y(0)}^{y(t)} \frac{\mathrm{d}z}{z^{\frac{s^2}{2}}} &\leq \Big(\frac{C_1}{N} + \frac{C_2}{N^{s(1-\theta)-\alpha}}\Big) \int_0^t \mathrm{e}^{c_N(\frac{s^2}{2}-1)\tau}\, \mathrm{d}\tau \\
\Rightarrow y^{1-\frac{s^2}{2}}(t) &\leq y^{1-\frac{s^2}{2}}(0) - \underbrace{\frac{1}{c_N} \cdot \Big(\frac{C_1}{N} + \frac{C_2}{N^{s(1-\theta)-\alpha}}\Big)}_{=:h(N)} \big(\mathrm{e}^{c_N(\frac{s^2}{2}-1)t} - 1\big) \\
&= y^{1-\frac{s^2}{2}}(0)\Big(1 - h(N)\big(\mathrm{e}^{c_N(\frac{s^2}{2}-1)t} - 1\big) y^{\frac{s^2}{2}-1}(0)\Big) \\
\Rightarrow y(t) &\leq y(0)\Big(1 - h(N)\big(\mathrm{e}^{c_N(\frac{s^2}{2}-1)t} - 1\big) y^{\frac{s^2}{2}-1}(0)\Big)^{-\frac{1}{\frac{s^2}{2}-1}} \\
\Rightarrow \|u\|_{L^\infty(\Omega)}(t) &\leq \mathrm{e}^{c_N t}\Big(1 - h(N)\big(\mathrm{e}^{c_N\left(\frac{s^2}{2}-1\right)t} - 1\big)\|u_0\|_{L^\infty(\Omega)}^{\frac{s^2}{2}-1} \mathrm{e}^{c_N t\left(1-\frac{s^2}{2}\right)}\Big)^{-\frac{1}{\frac{s^2}{2}-1}} \|u_0\|_{L^\infty(\Omega)}
\end{aligned} \tag{3.52}$$

where the complicated factor in front of $\|u_0\|_{L^\infty(\Omega)}$ is bounded for $t < c \ln N$ for some constant c. □

3.4 BV estimate

Stability of the numerical solutions measured in the bounded variation norm is ensured by

Lemma 8. *In finite time, the numerical solution of the SVV approximation (3.16) is bounded in the BV norm by*

$$\|u\|_{BV(\Omega)}(t) \leq \begin{cases} \mathrm{e}^{c_N t}\Big(\|u_0\|_{BV(\Omega)} + \frac{C}{N^2}\Big), & 0 < \alpha < 1 \\ \mathrm{e}^{c_N t}\Big(\|u_0\|_{BV(\Omega)} + \frac{C_1}{N^2} + \frac{C_2}{N}\Big), & 1 \leq \alpha < 2 \end{cases} \tag{3.53}$$

Proof. The procedure used in [2] can be applied with a modification for the nonlocal term. Start by differentiating (3.16) spatially, yielding

$$\begin{aligned}
&\partial_t \partial_x u + \partial_x^2\big(f(u)\big) - \mathcal{L}\big[\partial_x A(u)\big] - \epsilon_N \partial_x^3 u \\
&= \partial_x^2 (I - P_N) f(u) - \mathcal{L}\big[\partial_x (I - P_N) A(u)\big] + \epsilon_N \partial_x^2 R_N * \partial_x u
\end{aligned} \tag{3.54}$$

Multiply (3.54) with $\mathrm{sgn}_\rho(\partial_x u)$, where sgn_ρ is a smooth approximation of the signum function, and integrate over Ω. Then, take the limit on each side of the equation as $\rho \to 0$. This is to obtain the L^1 norm. The first term gives the contribution

$$\lim_{\rho \to 0} \partial_t \int_0^{2\pi} \partial_x u \cdot \mathrm{sgn}_\rho(\partial_x u)\, \mathrm{d}x = \frac{\mathrm{d}}{\mathrm{d}t} \|\partial_x u\|_{L^1(\Omega)} \tag{3.55}$$

The convection term contribution is greater than zero. The argument for that is as follows. Define $\eta'_\rho(u_x) := \operatorname{sgn}_\rho(u_x)$ to ease notation. Use Lemma 12 in the appendix and the fact that $(\eta'_\rho(u_x) - \eta'(u_x)) = o(1)$ pointwise almost everywhere when $\rho \to 0$. This can be shown, but we omit the slightly technical proof here. For the interested reader, it can be noted that it is shown by applying Lebesgue's dominated convergence theorem and that the integral over the difference makes sense when passing to the limit $\rho \to 0$. With this in mind, we get

$$\begin{aligned}
\lim_{\rho\to 0}\int_0^{2\pi} \partial_x^2 f(u)\cdot \operatorname{sgn}_\rho(\partial_x u)\,\mathrm{d}x &= \lim_{\rho\to 0}\int_0^{2\pi} \partial_{xx} f(u)\cdot u_x\cdot \operatorname{sgn}_\rho(\partial_x u)\,\mathrm{d}x \\
&= \lim_{\rho\to 0}\int_0^{2\pi} \partial_{xx} f(u)\cdot u_x\big(\eta'(u_x) + \underbrace{(\eta'_\rho(u_x) - \eta'(u_x))}_{=o(1)}\big)\,\mathrm{d}x \\
&= \int_0^{2\pi} \partial_x\big(f'(u)\cdot\eta(u_x)\big)\,\mathrm{d}x \\
&= \big[f'(u)\cdot\eta(u_x)\big]_0^{2\pi} = 0.
\end{aligned} \tag{3.56}$$

The linear term can be shown to be greater than zero. Use the same notation as above, observe that η_ρ is a convex function and use Lemma 11 in the appendix to realize that

$$\begin{aligned}
\lim_{\rho\to 0} -\int_0^{2\pi} \partial_x^3 u\cdot\eta_\rho(\partial_x u)\,\mathrm{d}x &\geq \lim_{\rho\to 0} -\int_0^{2\pi} \partial_x^2(\eta_\rho(u_x))\,\mathrm{d}x \\
&= \big[\partial_x(\eta_\rho(u_x))\big]_0^{2\pi} = 0.
\end{aligned} \tag{3.57}$$

As in the preceding sections, the nonlocal operator can be eliminated due to its sign. To see this, we apply Lemma 16 and proceed as follows:

$$\begin{aligned}
&-\int_0^{2\pi} \mathcal{L}[\partial_x A(u)]\cdot\operatorname{sgn}_\rho(\partial_x u)\,\mathrm{d}x = B\big(\partial_x A(u), \operatorname{sgn}_\rho(\partial_x u)\big) \\
&= c_\alpha\int_0^{2\pi}\int_{|y|>0} \frac{\big(\partial_x A(u(x+y)) - \partial_x A(u(x))\big)\big(\operatorname{sgn}_\rho(\partial_x u(x+y)) - \operatorname{sgn}_\rho(\partial_x u(x))\big)}{|y|^{1+\alpha}}\,\mathrm{d}y\mathrm{d}x \\
&= c_\alpha\int_0^{2\pi}\int_{|y|>0} \frac{\Big(\big(\frac{\partial A}{\partial u}u_x\big)(x+y) - \big(\frac{\partial A}{\partial u}u_x\big)(x)\Big)\Big(\operatorname{sgn}_\rho(\partial_x u(x+y)) - \operatorname{sgn}_\rho(\partial_x u(x))\Big)}{|y|^{1+\alpha}}\,\mathrm{d}y\mathrm{d}x \\
&\geq 0,
\end{aligned} \tag{3.58}$$

since A is an increasing function. By using (3.55)-(3.58) and Lemma 2, an inequality for the time derivative of the norm of $\partial_x u$ in L^1 is obtained:

$$\begin{aligned}
&\frac{\mathrm{d}}{\mathrm{d}t}\|\partial_x u\|_{L^1(\Omega)} \leq c_N\|\partial_x u\|_{L^1(\Omega)} + C\|\partial_x^2(I-P_N)f(u)\|_{L^1(\Omega)} + \|\partial_x(I-P_N)\mathcal{L}[A(u)]\|_{L^1(\Omega)} \\
\Rightarrow &\frac{\mathrm{d}}{\mathrm{d}t}\Big(\mathrm{e}^{-c_N t}\|\partial_x u\|_{L^1(\Omega)}\Big) \leq \mathrm{e}^{-c_N t}\Big(C\|\partial_x^2(I-P_N)f(u)\|_{L^1(\Omega)} + \|\partial_x(I-P_N)\mathcal{L}[A(u)]\|_{L^1(\Omega)}\Big)
\end{aligned}$$

But it holds that $\|u\|_{BV(\Omega)} \leq \|\partial_x u\|_{L^1(\Omega)}$. Therefore,

$$\|u\|_{BV(\Omega)} \leq \mathrm{e}^{c_N t}\Big(\|u_0\|_{BV(\Omega)} + C\|\partial_x^2(I-P_N)f(u)\|_{L^1(D_T)} + \|\partial_x(I-P_N)\mathcal{L}[A(u)]\|_{L^1(D_T)}\Big) \tag{3.59}$$

What remains now is to estimate the projection error contributions on the right hand side. Once again, the energy estimate from Section 3.2 plays an important role. By using the energy estimate for the derivatives and the Gagliardo-Nirenberg inequality (3.12), the L^2 norm of the convection residual can be bounded gracefully:

$$\|\partial_x^2(I-P_N)f(u)\|_{L^2(D_T)} \leq B_s N^{2-s(1-\theta)}\sqrt{T} \tag{3.60}$$

As concerns the nonlocal residual, there are also here two cases of relevance; $0 < \alpha < 1$ and $1 \leq \alpha < 2$. To see how things turn out, it appears to be convenient to estimate the term in Fourier space. Write $A(u) = \sum_{\xi=-\infty}^{\infty} \hat{A}_\xi \mathrm{e}^{i\xi x}$ such that $(I-P_N)A(u) = \sum_{|\xi|>N} \hat{A}_\xi \mathrm{e}^{i\xi x}$. Apply the operators of interest to see that

$$\partial_x \mathcal{L}[(I-P_N)A(u)] = \sum_{|\xi|>N} (-G_\xi)(i\xi)\hat{A}_\xi \mathrm{e}^{i\xi x},$$

where $G_\xi = -C_\alpha|\xi|^\alpha$ is the Fourier symbol of $\mathcal{L}$, and C_α is a constant (see [2], proposition 7.1 and section 4.2 in this article for details). The term can now be estimated by using Parseval's identity and going back to the physical domain to recognize that

$$\begin{aligned}
\|\partial_x \mathcal{L}[(I-P_N)A(u)]\|_{L^2(\Omega)}^2 &= \sum_{|\xi|>N} |(-G_\xi)(i\xi)\hat{A}_\xi|^2 \\
&= \sum_{|\xi|>N} |G_\xi \xi \hat{A}_\xi|^2 \\
&\leq \begin{cases} C_\alpha \sum_{|\xi|>N} |\xi|^4 |\hat{A}_\xi|^2, & 0<\alpha<1 \\ C_\alpha \sum_{|\xi|>N} |\xi|^6 |\hat{A}_\xi|^2, & 1\leq\alpha<2 \end{cases} \\
&= \begin{cases} C_\alpha \|\partial_x^2(I-P_N)A(u)\|_{L^2(\Omega)}^2 & 0<\alpha<1 \\ C_\alpha \|\partial_x^3(I-P_N)A(u)\|_{L^2(\Omega)}^2 & 1\leq\alpha<2. \end{cases}
\end{aligned} \tag{3.61}$$

By using the energy estimate obtained previously,

$$\begin{aligned}
&\|\partial_x(I-P_N)\mathcal{L}[A(u)]\|_{L^2(D_T)} \leq \begin{cases} CB_s N^{2-s(1-\theta)}\sqrt{T}, & 0<\alpha<1 \\ CB_s N^{3-s(1-\theta)}\sqrt{T}, & 1\leq\alpha<2 \end{cases} \\
&\text{alternatively,} \\
&\|\partial_x(I-P_N)\mathcal{L}[A(u)]\|_{L^2(D_T)} \leq \begin{cases} CB_s N^{2-s(1-\theta)-\frac{\theta}{2}}, & 0<\alpha<1 \\ CB_s N^{3-s(1-\theta)-\frac{\theta}{2}}, & 1\leq\alpha<2. \end{cases}
\end{aligned} \tag{3.62}$$

Also, since the L^1 norm is bounded by a constant times the L^2 norm in the compact domain D_T due to Lemma 15 in the appendix,

$$\begin{aligned}
&\|\partial_x^2(I-P_N)f(u)\|_{L^1(D_T)} \leq CB_sN^{2-s(1-\theta)}\sqrt{T} \\
&\|\partial_x(I-P_N)\mathcal{L}[A(u)]\|_{L^1(D_T)} \leq \begin{cases} CB_sN^{2-s(1-\theta)}\sqrt{T}, & 0<\alpha<1 \\ CB_sN^{3-s(1-\theta)}\sqrt{T}, & 1\leq\alpha<2 \end{cases} \\
&\text{alternatively,} \qquad (3.63) \\
&\|\partial_x^2(I-P_N)f(u)\|_{L^1(D_T)} \leq CB_sN^{2-s(1-\theta)-\frac{\theta}{2}} \\
&\|\partial_x(I-P_N)\mathcal{L}[A(u)]\|_{L^1(D_T)} \leq \begin{cases} CB_sN^{2-s(1-\theta)-\frac{\theta}{2}}, & 0<\alpha<1 \\ CB_sN^{3-s(1-\theta)-\frac{\theta}{2}}, & 1\leq\alpha<2. \end{cases}
\end{aligned}$$

By merging constants and using Assumptions 4 and 7 for the regularity, the desired BV stability bound

$$\begin{aligned}
\|u\|_{BV(\Omega)}(t) &\leq \begin{cases} \mathrm{e}^{c_N t}\Big(\|u_0\|_{BV(\Omega)} + CN^{2-s(1-\theta)}\Big), & 0<\alpha<1 \\ \mathrm{e}^{c_N t}\Big(\|u_0\|_{BV(\Omega)} + C_1N^{2-s(1-\theta)} + C_2N^{3-s(1-\theta)}\Big), & 1\leq\alpha<2 \end{cases} \\
&\leq \begin{cases} \mathrm{e}^{c_N t}\Big(\|u_0\|_{BV(\Omega)} + CN^{-2}\Big), & 0<\alpha<1 \\ \mathrm{e}^{c_N t}\Big(\|u_0\|_{BV(\Omega)} + C_1N^{-2} + C_2N^{-1}\Big), & 1\leq\alpha<2 \end{cases}
\end{aligned}$$

is obtained. □

3.5 Time regularity estimate

The procedure from lemma 5.3 in [2] is used to derive time regularity estimates. The nonlocal term must be treated differently however. The estimate is concluded with

Lemma 9. *Let the assumptions in section 3.1 hold and let u be the numerical solution of the SVV approximation (3.16). Then*

$$\|u(\cdot,t_1)-u(\cdot,t_2)\|_{L^1(\Omega)} \leq C\sqrt{|t_1-t_2|} \qquad (3.64)$$

Proof. Let $\epsilon>0$ initially be undetermined. Introduce the mollifier unit function $\omega_\epsilon(x)=\frac{1}{\epsilon}\cdot\omega(\frac{x}{\epsilon})$, where $\omega\in C_c^\infty(\mathbb{R})$ is nonnegative and satisfies

$$\omega(-x)=\omega(x), \quad \omega(x)=0 \ \forall|x|>1 \quad \text{and} \int_{\mathbb{R}}\omega(x)\,\mathrm{d}x=1.$$

Also, introduce $u^\epsilon = u*\omega_\epsilon$. Add and subtract terms and use the triangular inequality to see that

$$\begin{aligned}
&\|u(\cdot,t_1)-u(\cdot,t_2)\|_{L^1(\Omega)} \leq \|u(\cdot,t_1)-u^\epsilon(\cdot,t_1)\|_{L^1(\Omega)} \\
&+\|u^\epsilon(\cdot,t_2)-u(\cdot,t_2)\|_{L^1(\Omega)} + \|u^\epsilon(\cdot,t_1)-u^\epsilon(\cdot,t_2)\|_{L^1(\Omega)}
\end{aligned} \qquad (3.65)$$

By using the definition of the convolution, the first and second terms on the right hand side are bounded in the BV seminorm:

$$\begin{aligned}\|u(\cdot,t)-u^\epsilon(\cdot,t)\|_{L^1(\Omega)} &= \int_0^{2\pi}\Big|\int_{-\infty}^{\infty}\omega_\epsilon(y-x)\big(u(x,t)-u(y,t)\big)\,\mathrm{d}y\Big|\,\mathrm{d}x\\ &\le \int_0^{2\pi}\Big|\int_{-\infty}^{\infty}\omega_\epsilon(s)\big(u(x,t)-u(x+s,t)\big)\,\mathrm{d}s\Big|\,\mathrm{d}x\\ &\le |u|_{BV(\Omega)}\int_{-\infty}^{\infty}|s|\cdot\omega_\epsilon(s)\,\mathrm{d}s\\ &\le \epsilon|u|_{BV(\Omega)}\end{aligned}$$

The tricky third term is bounded using Taylor's formula with integral remainder:

$$\begin{aligned}&\|u^\epsilon(\cdot,t_1)-u^\epsilon(\cdot,t_2)\|_{L^1(\Omega)}\\ &\le |t_1-t_2|\int_0^{2\pi}\int_0^1|\partial_t u^\epsilon(x,t_1+\tau(t_2-t_1))|\,\mathrm{d}\tau\mathrm{d}x\end{aligned}$$

It is now obvious that what is needed to conclude the estimate is a bound for $\|\partial_t u\|_{L^1(\Omega)}$. In [2], this is achieved by convolving the numerical equation with the mollifier ω_ϵ and integrating over Ω. The difference is that in this case, there is also the term from the projection of the nonlocal operator since it is possibly nonlinear. It seems natural to deal with this in the same manner as the convection term. Hence, convolve the numerical equation (3.16) with ω_ϵ and use the triangle inequality to obtain

$$\begin{aligned}\|\partial_t u*\omega_\epsilon\|_{L^1(\Omega)} = \|\partial_t u^\epsilon\|_{L^1(\Omega)} &\le \|\partial_x f(u)*\omega_\epsilon\|_{L^1(\Omega)} + \|\partial_x(I-P_N)f(u)*\omega_\epsilon\|_{L^1(\Omega)}\\ &\quad+\epsilon_N\|\partial_x^2 u*\omega_\epsilon\|_{L^1(\Omega)} + \epsilon_N\|(\partial_x^2 R_N*u)*\omega_\epsilon\|_{L^1(\Omega)}\\ &\quad+\|\mathcal{L}[A(u)]*\omega_\epsilon\|_{L^1(\Omega)} + \|\mathcal{L}[(I-P_N)A(u)]*\omega_\epsilon\|_{L^1(\Omega)}\end{aligned}$$

The four first terms on the right hand side can be estimated as in [2] using the triangle inequality, Lemma 2 and Young's inequality for convolutions to obtain that

$$\begin{aligned}\|\partial_x f(u)*\omega^\epsilon\|_{L^1(\Omega)} + \|\partial_x(I-P_N)f(u)*\omega^\epsilon\|_{L^1(\Omega)} &\le C\Big(|u|_{BV(\Omega)}+\frac{1}{N}\Big)\\ \epsilon_N\|\partial_x^2 u*\omega^\epsilon\|_{L^1(\Omega)} &\le \|\partial_x u*\partial_x\omega_\epsilon\|_{L^1(\Omega)} \le \epsilon^{-1}|u|_{BV(\Omega)}\\ \epsilon_N\|\big(\partial_x^2 R_N*u\big)*\omega^\epsilon\|_{L^1(\Omega)} &\le C\|u\|_{L^1(\Omega)}\end{aligned} \tag{3.66}$$

The last two terms are treated in a similar manner, but using an arbitrary function. Consider $\|\mathcal{L}[\varphi]*\omega_\epsilon\|_{L^1(\Omega)}$ for a function $\varphi(x)$. Then split the nonlocal operator in two and use the triangle inequality to obtain

$$\begin{aligned}|\mathcal{L}[\varphi]*\omega_\epsilon\|_{L^1(\Omega)} &\le \int_0^{2\pi}\Big|\int_{-\infty}^{\infty}\Big(\int_{|y|<1}\varphi(x+y)-\varphi(x)-y\partial_x\varphi(x)\mathrm{d}\mu(y)\Big)\omega_\epsilon(x-p)\Big|\mathrm{d}p\mathrm{d}x\\ &\quad+\int_0^{2\pi}\Big|\int_{-\infty}^{\infty}\Big(\int_{|y|>1}\varphi(x+y)-\varphi(x)\mathrm{d}\mu(y)\Big)\omega_\epsilon(x-p)\Big|\mathrm{d}p\mathrm{d}x\end{aligned}$$

The first term is expressed as a Taylor expansion with integral remainder. By using $\|\partial_x \omega_\epsilon\|_{L^1} \leq C\epsilon^{-1}$ and integrating by parts, it can be seen that the first term is bounded by

$$\begin{aligned}
&\int_0^{2\pi} \Big| \int_{-\infty}^{\infty} \Big(\int_{|y|<1} \varphi(x+y) - \varphi(x) - y\partial_x\varphi(x) \mathrm{d}\mu(y) \Big) \omega_\epsilon(x-p) \mathrm{d}p \Big| \mathrm{d}x \\
&= \int_0^{2\pi} \Big| \int_{-\infty}^{\infty} \Big(\int_{|y|<1} \int_0^1 (1-\tau) y^2 \partial_x\varphi(x) \ \mathrm{d}\tau \mathrm{d}\mu(y) \Big) \omega_\epsilon(x-p) \mathrm{d}p \Big| \mathrm{d}x \\
&\leq \int_0^{2\pi} \int_{-\infty}^{\infty} \int_{|y|<1} \int_0^1 (1-\tau) y^2 |\partial_x\varphi(x)| |\omega_\epsilon(x-p)| \ \mathrm{d}\tau \mathrm{d}\mu(y) \mathrm{d}p \mathrm{d}x \\
&= \int_0^{2\pi} \int_{-\infty}^{\infty} \int_{|y|<1} \int_0^1 (1-\tau) y^2 |\partial_x\varphi(x)| |\partial_x\omega_\epsilon(x-p)| \ \mathrm{d}\tau \mathrm{d}\mu(y) \mathrm{d}p \mathrm{d}x \\
&\leq C\epsilon^{-1} |\varphi|_{BV(\Omega)}
\end{aligned}$$

The second term is bounded by

$$\int_0^{2\pi} \Big| \int_{-\infty}^{\infty} \Big(\int_{|y|>1} \varphi(x+y) - \varphi(x) \mathrm{d}\mu(y) \Big) \omega_\epsilon(x-p) \Big| \mathrm{d}p \mathrm{d}x \leq C\|\varphi\|_{L^1(\Omega)}$$

So by letting $\varphi(x) = A\big(u(x)\big)$ we get the bound

$$\|\mathcal{L}[A(u)] * \omega_\epsilon\|_{L^1(\Omega)} \leq C\epsilon^{-1} |A(u)|_{BV(\Omega)} + C\|A(u)\|_{L^1(\Omega)}$$

Now is the time to use the properties of A. Firstly, observe that the BV seminorm of a differentiable function φ can be written

$$|\varphi|_{BV(\Omega)} = \sup_{h \neq 0} \int_0^{2\pi} \frac{|\varphi(x+h) - \varphi(x)|}{|h|} \ \mathrm{d}x,$$

cf. Appendix A in [12]. Use Assumption 5 to discover that

$$\begin{aligned}
|A(u)|_{BV(\Omega)} &= \sup_{h \neq 0} \int_0^{2\pi} \frac{|A[u](x+h) - A[u](x)|}{|h|} \ \mathrm{d}x \\
&\leq \sup_{h \neq 0} \int_0^{2\pi} \frac{L_A |u(x+h) - u(x)|}{|h|} \ \mathrm{d}x \\
&= L_A |u|_{BV(\Omega)}
\end{aligned}$$

Secondly, use Assumption 2 and consider the second term above:

$$\begin{aligned}
\|A(u)\|_{L^1(\Omega)} = \int_0^{2\pi} |A[u](x)| \ \mathrm{d}x &= \int_0^{2\pi} |A[u](x) - \underbrace{A[0](x)}_{=0}| \ \mathrm{d}x \\
&\leq \int_0^{2\pi} L_A |u(x) - 0(x)| \ \mathrm{d}x \\
&= L_A \|u\|_{L^1(\Omega)}
\end{aligned}$$

In conclusion,

$$\|\mathcal{L}[A(u)] * \omega_\epsilon\|_{L^1(\Omega)} \leq L_A \cdot C(\epsilon^{-1}|u|_{BV(\Omega)} + \|u\|_{L^1(\Omega)}).$$

With $\varphi(x) = (I - P_N)A(u)$, observe that the residual is bounded by

$$\begin{aligned}\|\mathcal{L}[(I - P_N)A(u)] * \omega_\epsilon\|_{L^1(\Omega)} &\leq C\epsilon^{-1}|(I - P_N)A(u)|_{BV(\Omega)} + C\|(I - P_N)A(u)\|_{L^1(\Omega)} \\ &\leq C\epsilon^{-1}\|\partial_x(I - P_N)A(u)\|_{L^1(\Omega)} + C\|(I - P_N)A(u)\|_{L^1(\Omega)}\end{aligned}$$

Now we can use (3.46) to obtain

$$\begin{aligned}\|(I - P_N)A(u)\|_{L^1(\Omega)} &\leq B_s N^{-s(1-\theta)}, \\ \|\partial_x(I - P_N)A(u)\|_{L^1(\Omega)} &\leq B_s N^{1-s(1-\theta)}\end{aligned} \tag{3.67}$$

such that, when using Assumption 4, we have that

$$\begin{aligned}\|\mathcal{L}[(I - P_N)A(u)] * \omega_\epsilon\|_{L^1(\Omega)} &\leq C\epsilon^{-1}B_s N^{1-s(1-\theta)} + CB_s N^{-s(1-\theta)} \\ &\leq C(\epsilon^{-1}N^{-3} + N^{-4}),\end{aligned}$$

The conclusion we get by using the BV stability proved in Lemma 8, the energy estimate from Lemma 5, (3.66) and the fact that $\|u\|_{L^1(\Omega)} \leq C\|u\|_{L^2(\Omega)}$, is that

$$\begin{aligned}\|\partial_t u^\epsilon\|_{L^1(\Omega)} \leq \|\partial_t u\|_{L^1(\Omega)} &\leq C\Big(|u|_{BV(\Omega)} + \frac{1}{N}\Big) + C(\epsilon^{-1}N^{-3} + N^{-4}) \\ &\quad + C(\epsilon^{-1}|u|_{BV(\Omega)} + \|u\|_{L^1(\Omega)}) \\ &\leq C\Big(1 + \frac{1}{\epsilon}\Big)\end{aligned}$$

The whole proof is now concluded by inserting the above estimates into (3.65), yielding

$$\|u(\cdot, t_1) - u(\cdot, t_2)\|_{L^1(\Omega)} \leq C\Big(\epsilon + |t_1 - t_2|(1 + \epsilon^{-1})\Big),$$

and by taking $\epsilon = \sqrt{|t_1 - t_2|}$,

$$\|u(\cdot, t_1) - u(\cdot, t_2)\|_{L^1(\Omega)} \leq C\sqrt{|t_1 - t_2|},$$

and the proof is complete. □

3.6 Compactness

Due to the compactness theorem discussed in section 3.1 and the results obtained in sections 3.2 through 3.5, we can conclude that the family of numerical solutions of the SVV approximation (3.16) is compact with the following theorem:

Theorem 2 (Compactness of SVV approximation)**.** *Suppose that Assumptions 2-9 hold, and let u_N be the solution of (3.16) for a given space dimension N. Then there exists a subsequence $\{u_N\}$ converging in $C([0,T]; L^1(\Omega))$ to a limit $u \in C([0,T]; L^1(\Omega)) \cap L^\infty(D_T) \cap L^\infty([0,T]; BV(\Omega))$.*

3.7 Convergence of the numerical method

By synthesizing the results that are obtained in the previous sections, the convergence analysis can now be concluded with the following

Theorem 3. *Let Assumptions 2-9 hold, let u_N be the solution of the SVV approximation (3.16), and let v_{ϵ_N} be the solution of (2.8). Then the following convergence estimate holds:*

$$\|u_N(\cdot,T) - v_{\epsilon_N}(\cdot,T)\|_{L^1(\Omega)} \leq C\sqrt{\epsilon_N}(1 + N^{-3} + N^{-3+\mathbf{1}_{1\leq\alpha<2}})$$

Proof. The direct argument used in [2] can be applied. v_{ϵ_N} is smooth, so equation (2.8) can be subtracted from (3.16), such that

$$\begin{aligned}\partial_t(u_N - v_{\epsilon_N}) + \partial_x\big(f(u_N) - f(v_{\epsilon_N})\big) - \mathcal{L}[A(u_N) - A(v_{\epsilon_N})] - \epsilon_N\partial_x^2(u_N - v_{\epsilon_N})\\ = -\epsilon_N\partial_x R * \partial_x u_N + \partial_x(I - P_N)f(u_N) + \partial_x(I - P_N)A(u_N)\end{aligned} \tag{3.68}$$

Multiply (3.68) with $\operatorname{sgn}_\rho(u_N - v_{\epsilon_N})$, a smooth approximation of $\operatorname{sgn}(u_N - v_{\epsilon_N})$, integrate over Ω and go to the limit $\rho \to 0$ to obtain

$$\begin{aligned}\frac{\mathrm{d}}{\mathrm{d}t}\|u_n - v_{\epsilon_N}\|_{L^1(\Omega)} \leq \epsilon_N\|\partial_x R_N * \partial_x u_N\|_{L^1(\Omega)} + \|\partial_x(I - P_N)f(u)\|_{L^1(\Omega)}\\ + \|(I - P_N)\mathcal{L}[A(u)]\|_{L^1(\Omega)}\end{aligned}$$

The first term on the right hand side is estimated by applying Lemma 8, equation (3.9) in Lemma 2 with $s = 2$ and Assumption 1:

$$\begin{aligned}\|\partial_x R_N * \partial_x u\|_{L^1(\Omega)} &\leq \|\partial_x R_N\|_{L^1(\Omega)}\|\partial_x u\|_{L^1(\Omega)} \leq Cm_N \log N\|u\|_{BV(\Omega)}\\ &\leq C\epsilon_N^{-\frac{1}{2}}\end{aligned}$$

Integrate in time to obtain

$$\begin{aligned}\|u_n - v_{\epsilon_N}\|_{L^1(\Omega)} &\leq C\sqrt{\epsilon_N} + \|\partial_x(I - P_N)f(u)\|_{L^1(D_T)} + \|(I - P_N)\mathcal{L}[A(u)]\|_{L^1(D_T)}\\ &\leq C(\sqrt{\epsilon_N} + \|\partial_x(I - P_N)f(u)\|_{L^2(D_T)} + \|(I - P_N)\mathcal{L}[A(u)]\|_{L^2(D_T)})\end{aligned}$$

Now perform an estimate as in (3.61) without the derivative to obtain an estimate like (3.63):

$$\|\mathcal{L}[(I - P_N)A(u)]\|_{L^1(D_T)} \leq \begin{cases} CB_s N^{1-s(1-\theta)-\frac{\theta}{2}}, & 0 < \alpha < 1\\ CB_s N^{2-s(1-\theta)-\frac{\theta}{2}}, & 1 \leq \alpha < 2\end{cases} \tag{3.69}$$

Use (3.47), (3.69) and Assumptions1, 4 and 7 to estimate the last two terms:

$$\begin{aligned}\|\partial_x(I - P_N)f(u_N(\cdot,T))\|_{L^2(\Omega)} &\leq C\mathcal{K}_s N^{1-s(1-\theta)-\frac{\theta}{2}} \leq C\mathcal{K}_s N^{-3-\frac{\theta}{2}}\\ &\leq CN^{-3}\sqrt{\epsilon_N}\end{aligned}$$

$$\|(I-P_N)\mathcal{L}[A(u_N(\cdot,T))]\|_{L^2(\Omega)} \leq C\mathcal{K}_s N^{1+\mathbf{1}_{1\leq\alpha<2}-s(1-\theta)-\frac{\theta}{2}} \leq C\mathcal{K}_s N^{-3+\mathbf{1}_{1\leq\alpha<2}-\frac{\theta}{2}} \leq CN^{-3+\mathbf{1}_{1\leq\alpha<2}}\sqrt{\epsilon_N}$$

such that

$$\|u_n(\cdot,T)-v_{\epsilon_N}(\cdot,T)\|_{L^1(\Omega)} \leq C\sqrt{\epsilon_N}(1+N^{-3}+N^{-3+\mathbf{1}_{1\leq\alpha<2}}),$$

which concludes the proof. □

Chapter 4

Discretization and numerical solvers

4.1 Introduction - overview of solvers

One of the goals for the thesis was to generalize the framework of known numerical methods for fractional conservation laws, by analyzing a new method for solving the quite general model equation (2.2). When a solver is implemented in practice, there occur some additional obstacles that one does not think of when analyzing the problem theoretically. For instance, the nonlinear interpolations require calculation of integrals that are not known before the equation is determined. That is, the integrals are not method-specific, but equation-specific.

A generic solver must address this problem in a good way such that the equation can be specified arbitrarily. However, the cost of making such a generic solver is that quadrature must be applied, and this can affect the accuracy of the method. This is because the numerical scheme is then a perturbation of the original equation. For special cases, like $f = u^p$ and $A = u^p$, the use of quadrature for the interpolations can be avoided by using convolutions in Fourier space, and thus an "exact" method is obtained.

We have developed two solvers in this project for comparison; one generic solver that can process arbitrarily specified f and A, and one which is an exact method for linear and quadratic f and A. The latter uses convolutions to calculate the nonlinear projections exactly, whereas the former applies the discrete Fourier transform to approximate the integrals occurring in the interpolations. The two methods also differ in the choice of discrete space. An elegant feature of both methods is that they diagonalize all discrete operators, including the discretization of the nonlocal operator. This greatly reduces computational costs, since the solvers are inherently matrixless due to this fact.

The derivation of the differential equations that are implemented in code will be given in the following. For simplicity and clarity, the details are given in one and

two dimensions, and these are also the cases treated in the numerical experiments of the project. Extension to arbitrary dimensions is a matter of book-keeping, and the interested reader can therefore easily implement extensions of the discretizations derived in this report. The notation is kept on a general level so as to include both solvers in the derivations. Where there are differences between the two solvers, this is commented particularly.

4.2 Discrete space

In this project, the spatial domain used in the discretizations of the Fourier-Galerkin method is $\Omega = (0, 2\pi)^d$, where d is the number of dimensions of the problem. Let $\mathbf{p} \in \mathbb{Z}^d, \mathbf{x} \in \mathbb{R}^d$. The discrete space is defined in terms of a basis as

$$X_{\mathcal{F}_N} = \text{span } \{\mathrm{e}^{i\mathbf{p}\cdot\mathbf{x}}\}.$$

The basis is orthogonal with respect to the standard L^2 inner product, but it is not normalized. It is convenient to drop the normalization factor in the basis, since this will vanish in the discretized equations anyway.

N is assumed to be an even number in the case of the generic solver. For the generic solver, it is required that $\min_j p_j \geq -\frac{N}{2}$, $\max_j p_j \leq \frac{N}{2} - 1$ and for the exact solver, that $\min_j p_j \geq -N$, $\max_j p_j \leq N$. For generality, introduce the index set $\mathcal{I}$, which contains all the indices of the discrete space $X_{\mathcal{F}_N}$. This eases the notation in the following, as it covers both solvers, and will be used where it is appropriate.

Choosing $-\frac{N}{2}$ and $\frac{N}{2} - 1$ as the lowest and highest frequency components is a common convention in the Fourier-Galerkin framework to create an N^d dimensional space. The reason for this is that one often uses quadrature to evaluate integrals in weak formulations, and in particular, the discrete Fourier transform. The generic solver uses the DFT/FFT as quadrature, and the discrete Fourier transform is defined in such a way that this basis corresponds to the components of the transform. It is therefore practical to induce the same basis in the discrete solution space. For the exact solver, it is more convenient to use $-N$ and N as the lowest and highest frequency components.

4.3 Discretization in one spatial dimension

For each $t > 0$, the discrete solution is expressed as a linear combination of the trial functions, i.e. $u(x,t) = \sum_{p\in\mathcal{I}} \hat{u}_p(t)\mathrm{e}^{ipx}$. To obtain the semi-discretized system of ODEs, the solution is inserted into the numerical variational equation (2.10). In (2.10), let the test function be given by $v = \mathrm{e}^{-i\xi x}$. In the following, each term is treated specifically, and the orthogonality of the chosen basis space is utilized to obtain a decoupled system of ODEs. In the derivation of the terms in the exact solver below, the equation

$$u_t + \frac{1}{2}\frac{\mathrm{d}}{\mathrm{d}x}u^2 = -(-\Delta)^{\frac{\alpha}{2}}u^2$$

is implemented, but the reader should easily be able to extend this also to other equations with $f(u), A(u)$ as for instance u^p.

Mass term

The mass term is trivial. For both solvers, the contribution is

$$\int_0^{2\pi} u_t \cdot v \, \mathrm{d}x = \sum_{p\in\mathcal{I}} \hat{u}_p \int_0^{2\pi} \mathrm{e}^{ipx}\mathrm{e}^{-i\xi x} \, \mathrm{d}x = 2\pi\hat{u}_\xi(t), \;\; \xi \in \mathcal{I}$$

Convection term - generic solver

The convection term is possibly nonlinear, and its projection onto $X_{\mathcal{F}_N}$ is

$$P_N f(u) = \sum_{p=-\frac{N}{2}}^{\frac{N}{2}-1} \hat{f}_p \mathrm{e}^{ipx},$$

where

$$\hat{f}_p = \frac{1}{2\pi}\int_0^{2\pi} f[u](x)\mathrm{e}^{-ipx} \, \mathrm{d}x$$

is the Fourier expansion of $f(u)$. These integrals are approximated numerically using the discrete Fourier transform (DFT). One can show that the DFT corresponds to the trapezoidal quadrature rule for periodic functions. Although this modification is a "variational crime", the quadrature error in this approach is of the same magnitude as the discretization error for periodic functions under certain conditions, but we do not prove this here (see for instance [7], p. 300). Let $\tilde{f}$ denote the DFT approximation of the Fourier coefficient. The contribution from the convection term is thus calculated as follows:

$$\begin{aligned}\int_0^{2\pi} \frac{\mathrm{d}P_N f(u)}{\mathrm{d}x} \cdot v \, \mathrm{d}x &= \sum_{p=-\frac{N}{2}}^{\frac{N}{2}-1} \int_0^{2\pi} ip\hat{f}_p \mathrm{e}^{ipx}\mathrm{e}^{-i\xi x} \;\; \mathrm{d}x \\ &= 2\pi i\xi \hat{f}_\xi \\ &\approx 2\pi i\xi \tilde{f}_\xi\end{aligned}$$

Convection term - exact solver

Consider

$$\begin{aligned}\frac{1}{2}\frac{\mathrm{d}}{\mathrm{d}x}u^2 &= \frac{1}{2}\frac{\mathrm{d}}{\mathrm{d}x}\Big(\sum_{p=-N}^{N} \hat{u}_p \mathrm{e}^{ipx}\Big)\Big(\sum_{q=-N}^{N} \hat{u}_q \mathrm{e}^{iqx}\Big) \\ &= \frac{1}{2}\sum_{p=-N}^{N}\sum_{q=-N}^{N} i(p+q)\hat{u}_p\hat{u}_p\mathrm{e}^{i(p+q)x}\end{aligned}$$

Integrate against the test function and take the projection to obtain the convection contribution:

$$\begin{aligned}\frac{1}{2}\int_0^{2\pi} \frac{\mathrm{d}}{\mathrm{d}x}u^2 \cdot \mathrm{e}^{-i\xi x}\,\mathrm{d}x &= \frac{1}{2}\sum_{p=-N}^{N}\sum_{q=-N}^{N} i(p+q)\hat{u}_p\hat{u}_q \int_0^{2\pi} \mathrm{e}^{i(p+q-\xi)x}\,\mathrm{d}x \\ &= \pi \sum_{p=-N}^{N}\sum_{q=-N}^{N} i(p+q)\hat{u}_p\hat{u}_q,\ \ p+q=\xi \\ \Rightarrow \frac{1}{2}\int_0^{2\pi} P_N\frac{\mathrm{d}}{\mathrm{d}x}u^2 \cdot \mathrm{e}^{-i\xi x}\,\mathrm{d}x &= \pi i\xi \sum_{p=-N}^{N} \hat{u}_p\hat{u}_{\xi-p}\mathbf{1}_{-N\le\xi-p\le N},\ \ \xi\in\{-N,...,N\}\end{aligned}$$

Fractional diffusion term - generic solver

In the numerical equation, the projection of the possibly non-linear term onto the discrete space $X_{\mathcal{F}_N}$ is calculated as

$$P_N A[u](x) = \sum_{p\in\mathcal{I}} \hat{A}_p \mathrm{e}^{ipx},$$

where

$$\hat{A}_p = \frac{1}{2\pi}\int_0^{2\pi} A[u](x)\mathrm{e}^{-ipx}\,\mathrm{d}x,\ \ p\in\mathcal{I}.$$

Essentially, the nonlinear projection is a truncated Fourier representation of the nonlinear function $A[u]$.

As with the convection term, these integrals are calculated using the DFT. The choice can be motivated also from a computational point of view. For the DFT, there are very efficient algorithms and highly optimized off-the-shelf computer software. A naïve implementation of the DFT has a complexity of $\mathcal{O}(n^2)$, whereas the Fast Fourier Transform (FFT) requires $\mathcal{O}(n\log n)$ flop. Let $\tilde{A}_p$ denote the DFT approximation of $\hat{A}_p$. The nonlocal term can now be evaluated using the (approximate) nonlinear projection:

$$\begin{aligned}&\int_0^{2\pi} \mathcal{L}(P_N A[u])\cdot v\,\mathrm{d}x \\ &= \sum_{p\in\mathcal{I}} \hat{A}_p \int_0^{2\pi} \mathrm{e}^{-i\xi x}\int_{|y|>0} \mathrm{e}^{ip(x+y)} - \mathrm{e}^{ipx} - ipy\mathbf{1}_{|y|<1}\mathrm{e}^{ipx}\,\mathrm{d}\mu(y)\mathrm{d}x \\ &= 2\pi\hat{A}_\xi \underbrace{\int_{|y|>0} \mathrm{e}^{ipy} - 1 - ipy\mathbf{1}_{|y|<1}\,\mathrm{d}\mu(y)\mathrm{d}x}_{G(\xi)} \\ &\approx 2\pi\tilde{A}_\xi G(\xi)\end{aligned} \tag{4.1}$$

Depending on the measure $\mathrm{d}\mu(y)$, the integral $G(\xi)$ can be calculated (semi-)analytically. In [2], when $\mathcal{L} = -(-\Delta)^{\frac{\alpha}{2}}$ like here, Jakobsen and Cifani give an explicit closed-form expression for $0<\alpha\le 1$ and a semi-analytical expression for $1<\alpha<2$.

According to their Proposition 7.1,

$$G(\xi) = -C_\alpha |\xi|^\alpha,$$
where
$$C_\alpha = \frac{2c_\alpha}{\alpha} \underbrace{\int_0^\infty x^{-\alpha} \sin x \, \mathrm{d}x}_{\Theta_\alpha} \tag{4.2}$$

For $1 < \alpha < 2$, no closed-form expression is known for Θ_α, and it must be approximated with numerical quadrature. For $0 < \alpha \leq 1$, it can be calculated analytically as

$$\Theta_\alpha = \begin{cases} \Gamma(1-\alpha) \sin \frac{\pi(1-\alpha)}{2}, & 0 < \alpha < 1 \\ \frac{\pi}{2}, & \alpha = 1 \end{cases}$$

See also Section 6.2 for more details of the implementation in code.

Fractional diffusion term - exact solver

Use the same approach as with the convection term to calculate the square of u. Then apply the nonlocal operator, multiply with the test function, take the projection and integrate to obtain

$$\begin{aligned}
&\int_0^{2\pi} \mathcal{L}[P_N u^2] \cdot \mathrm{e}^{-i\xi x} \, \mathrm{d}x \\
&= \int_0^{2\pi} \mathrm{e}^{-i\xi x} \int_{|y|>0} P_N u^2(x+y) - P_N u^2(x) - y \mathbf{1}_{|y|<1} P_N \left(u^2\right)_x \, \mathrm{d}\mu(y) \mathrm{d}x \\
&= \int_0^{2\pi} \sum_{p=-N}^{N} \sum_{q=-N}^{N} \hat{u}_p \hat{u}_q \mathbf{1}_{|p|,|q| \leq N} \mathrm{e}^{i(p+q-\xi)x} \underbrace{\int_{|y|>0} \mathrm{e}^{i(p+q)y} - 1 - i(p+q)\mathbf{1}_{|y|<1} y \, \mathrm{d}\mu(y)}_{G_{p+q}} \, \mathrm{d}x \\
&= 2\pi \sum_{p=-N}^{N} \hat{u}_p \hat{u}_{\xi-p} \mathbf{1}_{-N \leq \xi - p \leq N} G(\xi), \;\; p+q = \xi, \;\; \xi \in \{-N, ..., N\}
\end{aligned}$$

SVV term

For both solvers, the spectral vanishing viscosity term gives a contribution

$$\begin{aligned}
\int_0^{2\pi} (\partial_x^2 Q_N * u) v \, \mathrm{d}x &= \int_0^{2\pi} \Big(- \sum_{p \in \mathcal{I}, |p| \geq m_N} \xi^2 \hat{Q}_p(t) \hat{u}_p(t) \mathrm{e}^{ipx} \Big) \mathrm{e}^{i\xi x} \, \mathrm{d}x \\
&= -2\pi \xi^2 \hat{Q}_\xi(t) \hat{u}_\xi(t)
\end{aligned}$$

The viscosity kernels $\hat{Q}_\xi(t)$ are chosen according to the following scheme, which is used by Maday and Tadmor in [5]:

$$\begin{cases} \hat{Q}_\xi = 0, \ |\xi| < m \\ \hat{Q}_\xi = \frac{|\xi|-m}{m}, \ m \leq |\xi| \leq 2m \\ \hat{Q}_\xi = 1, \ 2m < |\xi| \leq \max \mathcal{I} \end{cases} \tag{4.3}$$

Source term

In the equation treated theoretically, there is no source term. However, the implementation in this project was made with the possibility of having a source term on the right hand side of the equation for comparison and testing. The contribution from a source term can be calculated as

$$\int_0^{2\pi} \mathrm{e}^{-i\xi x} \cdot g(x) \ \mathrm{d}x.$$

In the practical implementation, this integral is calculated using quadrature to allow for arbitrary source functions g. In the other terms of the variational equation, there is a geometric factor of 2π due to the choice of basis. This factor does not appear in the source term contribution. In the final equation, this factor is divided away, so for notation it is convenient to define

$$S(\xi) := \frac{1}{2\pi} \int_0^{2\pi} \mathrm{e}^{-i\xi x} \cdot g(x) \ \mathrm{d}x.$$

4.3.1 ODE system

Having evaluated all the terms, the numerical solution can be computed by solving the following ODE systems. After dividing the equation by 2π, the equation for the generic solver reads

$$\frac{\mathrm{d}\hat{u}_\xi}{\mathrm{d}t} = \tilde{A}_\xi(\hat{u}_\xi)G(\xi) - i\xi\tilde{f}_\xi(\hat{u}_\xi) - \epsilon_N\xi^2\hat{Q}_\xi\hat{u}_\xi + S(\xi) \ \ \forall\xi \in \{-\frac{N}{2}, ..., \frac{N}{2} - 1\}, \tag{4.4}$$

whereas the equation for the exact solver reads

$$\frac{\mathrm{d}\hat{u}_\xi}{\mathrm{d}t} = \sum_{p=-N}^{N} \hat{u}_p\hat{u}_{\xi-p}\mathbf{1}_{-N\leq\xi-p\leq N}\big(G(\xi) - \frac{1}{2}i\xi\big) - \epsilon_N\xi^2\hat{Q}_\xi\hat{u}_\xi + S_(\xi) \ \ \forall\xi \in \{-N, ..., N\} \tag{4.5}$$

Note that the sum on the right hand side is a discrete convolution, and can therefore be implemented efficiently as such.

4.4 Discretization in two spatial dimensions

The solution is represented by $u(x_1, x_2, t) = \sum_{p_1 \in \mathcal{I}} \sum_{p_2 \in \mathcal{I}} \hat{u}_{p_1 p_2}(t) \mathrm{e}^{i\mathbf{p}\cdot\mathbf{x}}$, where $\mathbf{p} = [p_1, p_2]$, $\mathbf{x} = [x_1, x_2]$ and $\mathcal{I}$ is the one-dimensional index set defined in section 4.2. In two dimensions the test function is $v = \mathrm{e}^{-i\boldsymbol{\xi}\cdot\mathbf{x}}$ with $\boldsymbol{\xi} = [\xi_1, \xi_2]$. The procedure in the following is the same as in one dimension, and orthogonality of the basis is used in each variable. In the derivation of the terms in the exact solver, the equation

$$u_t + \frac{1}{2}\nabla \cdot [u^2, u^2] = -(-\Delta)^{\frac{\alpha}{2}} u^2$$

is implemented as an example (but as in one dimension, it can easily be generalized to for instance u^p in the f and A).

Mass term

The mass contribution for both solvers is

$$\begin{aligned}\int_0^{2\pi}\int_0^{2\pi} u_t \cdot v \, \mathrm{d}x_1\mathrm{d}x_2 &= \sum_{p_1=-\frac{N}{2}}^{\frac{N}{2}-1} \sum_{p_2=-\frac{N}{2}}^{\frac{N}{2}-1} \hat{u}_{p_1 p_2} \int_0^{2\pi}\int_0^{2\pi} \mathrm{e}^{i\mathbf{p}\cdot\mathbf{x}} \mathrm{e}^{-i\boldsymbol{\xi}\cdot\mathbf{x}} \, \mathrm{d}x_1\mathrm{d}x_2 \\ &= (2\pi)^2 \hat{u}_{\xi_1\xi_2}(t)\end{aligned}$$

Convection term - generic solver

In dimensions higher than one, the convection function is vector-valued. Here it is therefore denoted by $\mathbf{f} = [f_1, f_2]$. It is possibly nonlinear, and its component-wise projection onto $X_{\mathcal{F}_N}$ is

$$P_N f_j(u) = \sum_{p_1=-\frac{N}{2}}^{\frac{N}{2}-1} \sum_{p_2=-\frac{N}{2}}^{\frac{N}{2}-1} \hat{f}^j_{p_1 p_2} \mathrm{e}^{i(p_1 x_1 + p_2 x_2)},$$

where

$$\hat{f}^j_{p_1 p_2} = \frac{1}{4\pi^2} \int_0^{2\pi}\int_0^{2\pi} f_j[u](\mathbf{x}) \mathrm{e}^{-i\mathbf{p}\cdot\mathbf{x}} \, \mathrm{d}x_1\mathrm{d}x_2, \;\; j \in \{1, 2\}$$

is the two-dimensional Fourier expansion of $f_j(u)$. As in one dimension, these integrals are approximated numerically with quadrature, essentially DFT. In MATLAB there is an efficient FFT routine also for higher dimensions. In two dimensions it is called `fft2()`. Let $\tilde{f}$ denote the FFT approximation of the Fourier coefficient.

The contribution from the convection term is thus calculated as follows:

$$\begin{aligned}
&\int_0^{2\pi} vP_N\nabla\cdot\mathbf{f}[u](x)\ \mathrm{d}x_1\mathrm{d}x_2\\
&=\sum_{p_1=-\frac{N}{2}}^{\frac{N}{2}-1}\sum_{p_2=-\frac{N}{2}}^{\frac{N}{2}-1}\int_0^{2\pi}\int_0^{2\pi} i(p_1\hat{f}^1_{p_1p_2}+p_2\hat{f}^2_{p_1p_2})\mathrm{e}^{i\left((p_1-\xi_1)x_1+(p_2-\xi_2)x_2\right)}\ \mathrm{d}x_1\mathrm{d}x_2\\
&=4\pi^2 i(\xi_1\hat{f}^1_{\xi_1\xi_2}+\xi_2\hat{f}^2_{\xi_1\xi_2})=4\pi^2 i\boldsymbol{\xi}\cdot\hat{\mathbf{f}}_{\xi_1\xi_2}\\
&\approx 4\pi^2 i\boldsymbol{\xi}\cdot\tilde{\mathbf{f}}_{\xi_1\xi_2}
\end{aligned}$$

Convection term - exact solver

Consider

$$\begin{aligned}
&\frac{1}{2}\Big(\frac{\partial}{\partial x_1}+\frac{\partial}{\partial x_2}\Big)u^2=\frac{1}{2}\Big(\frac{\partial}{\partial x_1}+\frac{\partial}{\partial x_2}\Big)\Big(\sum_{p_1=-N}^{N}\sum_{p_2=-N}^{N}\hat{u}_{p_1p_2}\mathrm{e}^{i\mathbf{p}\cdot\mathbf{x}}\Big)\Big(\sum_{q_1=-N}^{N}\sum_{q_2=-N}^{N}\hat{u}_{q_1q_2}\mathrm{e}^{i\mathbf{q}\cdot\mathbf{x}}\Big)\\
&=\frac{1}{2}\sum_{p_1=-N}^{N}\sum_{p_2=-N}^{N}\sum_{q_1=-N}^{N}\sum_{q_2=-N}^{N} i(p_1+p_2+q_1+q_2)\hat{u}_{p_1p_2}\hat{u}_{q_1q_2}\mathrm{e}^{i(\mathbf{p}+\mathbf{q})\cdot x}
\end{aligned}$$

Integrate against the test function and take the projection to obtain the convection contribution:

$$\begin{aligned}
&\frac{1}{2}\int_0^{2\pi}\int_0^{2\pi}\Big(\frac{\partial}{\partial x_1}+\frac{\partial}{\partial x_2}\Big)u^2\cdot\mathrm{e}^{-i\boldsymbol{\xi}\cdot\mathbf{x}}\ \mathrm{d}x_1\mathrm{d}x_2\\
&=\frac{1}{2}\sum_{p_1=-N}^{N}\sum_{p_2=-N}^{N}\sum_{q_1=-N}^{N}\sum_{q_2=-N}^{N} i(p_1+p_2+q_1+q_2)\hat{u}_{p_1p_2}\hat{u}_{q_1q_2}\int_0^{2\pi}\int_0^{2\pi}\mathrm{e}^{i(\mathbf{p}+\mathbf{q}-\boldsymbol{\xi})\cdot\mathbf{x}}\ \mathrm{d}x_1\mathrm{d}x_2\\
&=2\pi^2\sum_{p_1=-N}^{N}\sum_{p_2=-N}^{N}\sum_{q_1=-N}^{N}\sum_{q_2=-N}^{N} i(p_1+p_2+q_1+q_2)\hat{u}_{p_1p_2}\hat{u}_{q_1q_2},\ \ \mathbf{p}+\mathbf{q}=\boldsymbol{\xi}\\
&\Rightarrow\frac{1}{2}\int_0^{2\pi}\int_0^{2\pi}P_N\Big(\frac{\partial}{\partial x_1}+\frac{\partial}{\partial x_2}\Big)u^2\cdot\mathrm{e}^{-i\boldsymbol{\xi}\cdot\mathbf{x}}\ \mathrm{d}x_1\mathrm{d}x_2\\
&=2\pi^2 i(\xi_1+\xi_2)\sum_{p_1=-N}^{N}\sum_{p_2=-N}^{N}\hat{u}_{\mathbf{p}}\hat{u}_{\boldsymbol{\xi}-\mathbf{p}}\mathbf{1}_{-N\le\xi_1-p_1\le N}\mathbf{1}_{-N\le\xi_2-p_2\le N},\ \ \xi_1,\xi_2\in\{-N,...,N\}
\end{aligned}$$

Fractional diffusion term - generic solver

The projection of A onto the discrete space is also a two-dimensional Fourier expansion,

$$P_NA[u](x)=\sum_{p_1=-\frac{N}{2}}^{\frac{N}{2}-1}\sum_{p_2=-\frac{N}{2}}^{\frac{N}{2}-1}\hat{A}_{p_1p_2}\mathrm{e}^{i(p_1x_1+p_2x_2)},$$

where

$$\hat{A}_{p_1p_2} = \frac{1}{4\pi^2}\int_0^{2\pi}\int_0^{2\pi} A[u](\mathbf{x})\mathrm{e}^{-i\mathbf{p}\cdot\mathbf{x}}\,\mathrm{d}x_1\mathrm{d}x_2,\;\; p_1,p_2 \in \Big\{-\frac{N}{2},...,\frac{N}{2}-1\Big\}.$$

Let $\tilde{A}$ be the FFT approximation. The nonlocal term reads

$$\begin{aligned}
&\int_0^{2\pi}\int_0^{2\pi} \mathcal{L}(P_N A[u])\cdot v\;\mathrm{d}x_1\mathrm{d}x_2 \\
&= \sum_{p_1=-\frac{N}{2}}^{\frac{N}{2}-1}\sum_{p_2=-\frac{N}{2}}^{\frac{N}{2}-1} \hat{A}_{\mathbf{p}}\int_0^{2\pi}\int_0^{2\pi} \mathrm{e}^{-i\boldsymbol{\xi}\cdot\mathbf{x}}\int_{|\mathbf{y}|>0} \mathrm{e}^{i\mathbf{p}\cdot(\mathbf{x}+\mathbf{y})} - \mathrm{e}^{i\mathbf{p}\cdot\mathbf{x}} - i\mathbf{p}\cdot\mathbf{y}\mathbf{1}_{|\mathbf{y}|<1}\mathrm{e}^{i\mathbf{p}\cdot\mathbf{x}}\;\mathrm{d}\mu(\mathbf{y})\mathrm{d}x_1\mathrm{d}x_2 \\
&= 4\pi^2\hat{A}_{\boldsymbol{\xi}}\underbrace{\int_{|\mathbf{y}|>0} \mathrm{e}^{i\mathbf{p}\cdot\mathbf{y}} - 1 - i\mathbf{p}\cdot\mathbf{y}\mathbf{1}_{|\mathbf{y}|<1}\;\mathrm{d}\mu(\mathbf{y})}_{G(\boldsymbol{\xi})} \\
&\approx 4\pi^2\tilde{A}_{\boldsymbol{\xi}}G(\boldsymbol{\xi}),\;\; \xi_1,\xi_2 \in \{-\frac{N}{2},...,\frac{N}{2}-1\}
\end{aligned} \tag{4.6}$$

In [2], the semi-closed-form expression for G is given also in higher dimensions. The general case in d dimensions is given by

$$G(\boldsymbol{\xi}) = -C_\alpha|\boldsymbol{\xi}|^\alpha\int_{|\mathbf{y}|=1}\mathrm{d}S, \tag{4.7}$$

where Θ_α is defined as before,

$$C_\alpha = 2c_\alpha\Theta_\alpha$$

with

$$c_\alpha = \alpha\Gamma\Big(\frac{d+\alpha}{2}\Big)\Big(2\pi^{\frac{d}{2}+\alpha}\Gamma\big(1-\frac{\alpha}{2}\big)\Big)^{-1},$$

and the surface integral can be calculated as

$$\int_{|\mathbf{y}|=1}\mathrm{d}S = \frac{2\pi^{\frac{d}{2}}}{\Gamma\big(\frac{d}{2}\big)}.$$

Fractional diffusion term - exact solver

As in one dimension, calculate the square of u, apply the nonlocal operator, multiply with the test function, take the projection and integrate to obtain

$$\begin{aligned}
&\int_0^{2\pi} \mathcal{L}[P_N u^2] \cdot \mathrm{e}^{-i\boldsymbol{\xi}\cdot\mathbf{x}} \,\mathrm{d}x \\
&= \int_0^{2\pi} \mathrm{e}^{-i\xi x} \int_{|y|>0} P_N u^2(\mathbf{x}+\mathbf{y}) - P_N u^2(\mathbf{x}) - \mathbf{y}\mathbf{1}_{|\mathbf{y}|<1} \cdot \nabla P_N u^2 \,\mathrm{d}\mu(y)\mathrm{d}x \\
&= \int_0^{2\pi}\int_0^{2\pi} \sum_{p_1=-N}^{N}\sum_{p_2=-N}^{N}\sum_{q_1=-N}^{N}\sum_{q_2=-N}^{N} \hat{u}_{\mathbf{p}}\hat{u}_{\mathbf{q}}\mathbf{1}_{|\mathbf{p}|,|\mathbf{q}|\le N}\mathrm{e}^{i(\mathbf{p}+\mathbf{q}-\boldsymbol{\xi})\cdot\mathbf{x}} \\
&\quad \cdot \underbrace{\int_{|\mathbf{y}|>0} \mathrm{e}^{i(\mathbf{p}+\mathbf{q})\cdot\mathbf{y}} - 1 - i(\mathbf{p}+\mathbf{q})\cdot\mathbf{y}\mathbf{1}_{|\mathbf{y}|<1} \,\mathrm{d}\mu(y)}_{G(\mathbf{p}+\mathbf{q})} \,\mathrm{d}x_1\mathrm{d}x_2 \\
&= 4\pi^2 \sum_{p_1=-N}^{N}\sum_{p_2=-N}^{N} \hat{u}_{\mathbf{p}}\hat{u}_{\boldsymbol{\xi}-\mathbf{p}}\mathbf{1}_{-N\le\xi_1-p_1\le N}\mathbf{1}_{-N\le\xi_2-p_2\le N}G(\boldsymbol{\xi}), \\
&\mathbf{p}+\mathbf{q}=\boldsymbol{\xi},\ \ \xi_1,\xi_2\in\{-N,...,N\}.
\end{aligned}$$

SVV term

For both solvers, the integral from the spectral vanishing viscosity term becomes

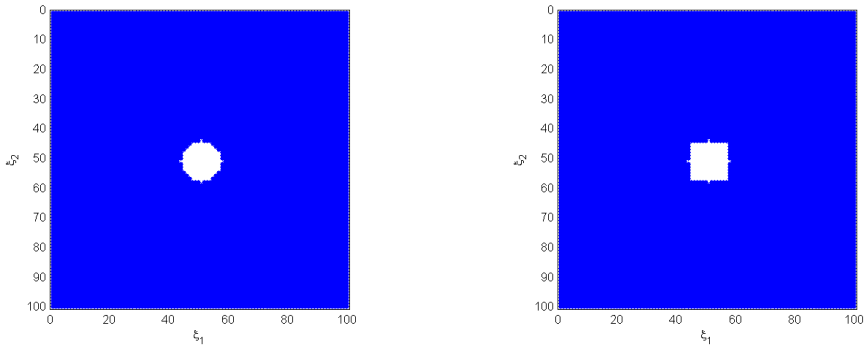
$$\begin{aligned}
&\int_0^{2\pi}\int_0^{2\pi} \big((\partial_{x_1}^2+\partial_{x_2}^2)Q_N * u\big)v \,\mathrm{d}x_1\mathrm{d}x_2 \\
&= \sum_{|p_1|=m_N}^{\frac{N}{2}}\sum_{|p_2|=m_N}^{\frac{N}{2}} \int_0^{2\pi}\int_0^{2\pi} (p_1^2+p_2^2)\hat{Q}_{p_1p_2}(t)\hat{u}_{p_1p_2}(t)\mathrm{e}^{i\mathbf{p}\cdot\mathbf{x}}\mathrm{e}^{i\boldsymbol{\xi}\cdot\mathbf{x}} \,\mathrm{d}x_1\mathrm{d}x_2 \\
&= -4\pi^2(\xi_1^2+\xi_2^2)\hat{Q}_{\xi_1\xi_2}(t)\hat{u}_{\xi_1\xi_2}(t)
\end{aligned}$$

The viscosity kernels $\hat{Q}_{\xi_1\xi_2}(t)$ are chosen according to the same scheme as in one dimension:

$$\begin{cases}
\hat{Q}_{\xi_1\xi_2} = 0, & |\boldsymbol{\xi}| < m \\
\hat{Q}_{\xi_1\xi_2} = \frac{|\boldsymbol{\xi}|-m}{m}, & m \le |\boldsymbol{\xi}| \le 2m \\
\hat{Q}_{\xi_1\xi_2} = 1, & 2m < |\boldsymbol{\xi}| \le \max\mathcal{I},
\end{cases} \tag{4.8}$$
$$\text{where } \boldsymbol{\xi} = [\xi_1, \xi_2]$$

However, in dimensions greater than one, there is freedom related to the distance measure. In the simulations performed in this study, the standard Euclidean 2-norm is used, i.e. $|\boldsymbol{\xi}| = \|\boldsymbol{\xi}\|_2 = \sqrt{\xi_1^2+\xi_2^2}$, but other norms are also possible. The requirement is that the kernels are spherically symmetric. In the implementation of the method, the kernel coefficients are gathered in a viscosity matrix $\hat{Q}$. In figure

4.1, typical such matrices are shown. The viscosity free spectrum is the white portion of the matrix.



(a) 2-norm (b) ∞-norm

Figure 4.1: Viscosity matrices with different norms

Source term

The source term contribution becomes a two-dimensional integral:

$$\int_0^{2\pi}\int_0^{2\pi} \mathrm{e}^{-i\xi_1 x_1}\mathrm{e}^{-i\xi_2 x_2}\cdot g(x_1,x_2)\,\mathrm{d}x_1\mathrm{d}x_2.$$

In two dimensions, the geometric factor in the other terms is $4\pi^2$, hence define

$$S(\boldsymbol{\xi}) := \frac{1}{4\pi^2}\int_0^{2\pi}\int_0^{2\pi} \mathrm{e}^{-i\xi_1 x_1}\mathrm{e}^{-i\xi_2 x_2}\cdot g(x_1,x_2)\,\mathrm{d}x_1\mathrm{d}x_2.$$

4.4.1 ODE system

Divide the equation by $4\pi^2$. The numerical solution is given in terms of the following ODE systems. The equation for the generic solver reads

$$\begin{aligned}
\frac{\mathrm{d}\hat{u}_{\xi_1\xi_2}}{\mathrm{d}t} &= \tilde{A}_{\xi_1\xi_2}(\hat{u}_{\xi_1\xi_2})G(\xi_1,\xi_2) - i(\xi_1\tilde{f}^1_{\xi_1\xi_2} + \xi_2\tilde{f}^2_{\xi_1\xi_2}) + \epsilon_N(\xi_1^2+\xi_2^2)\hat{Q}_{\xi_1\xi_2}(t)\hat{u}_{\xi_1\xi_2}(t) \\
&\quad + S(\xi_1,\xi_2) \\
\Leftrightarrow \frac{\mathrm{d}\hat{u}_{\boldsymbol{\xi}}}{\mathrm{d}t} &= \tilde{A}_{\boldsymbol{\xi}}(\hat{u}_{\boldsymbol{\xi}})G(\boldsymbol{\xi}) - i\boldsymbol{\xi}\cdot\tilde{\mathbf{f}}_{\boldsymbol{\xi}}(\hat{u}_{\boldsymbol{\xi}}) + \epsilon_N\|\boldsymbol{\xi}\|_2^2\hat{Q}_{\boldsymbol{\xi}}\hat{u}_{\boldsymbol{\xi}} + S(\boldsymbol{\xi}) \\
&\quad \forall\boldsymbol{\xi} = [\xi_1,\xi_2] \in \left\{-\frac{N}{2},...,\frac{N}{2}-1\right\}^2
\end{aligned} \tag{4.9}$$

while the equation for the exact solver is

$$\begin{aligned}
\frac{\mathrm{d}\hat{u}_{\xi_1\xi_2}}{\mathrm{d}t} &= \sum_{p_1=-N}^{N}\sum_{p_2=-N}^{N}\big[\hat{u}_{p_1p_2}\hat{u}_{(\xi_1-p_1,\xi_2-p_2)}\big(G(\xi_1,\xi_2)-\frac{1}{2}i(\xi_1+\xi_2)\big)\mathbf{1}_{-N\leq\xi_1-p_1\leq N} \\
&\quad\cdot\mathbf{1}_{-N\leq\xi_2-p_2\leq N}\big]+\epsilon_N(\xi_1^2+\xi_2^2)\hat{Q}_{\xi_1\xi_2}(t)\hat{u}_{\xi_1\xi_2}(t)+S(\xi_1,\xi_2)) \\
\Leftrightarrow \frac{\mathrm{d}\hat{u}_{\boldsymbol{\xi}}}{\mathrm{d}t} &= \sum_{p_1=-N}^{N}\sum_{p_2=-N}^{N}\hat{u}_{\mathbf{p}}\hat{u}_{\boldsymbol{\xi}-\mathbf{p}}\mathbf{1}_{-N\leq\xi_1-p_1\leq N}\mathbf{1}_{-N\leq\xi_2-p_2\leq N}\Big(G(\boldsymbol{\xi})-\frac{1}{2}i\|\boldsymbol{\xi}\|_1\Big) \\
&\quad+\epsilon_N\|\boldsymbol{\xi}\|_2^2\hat{Q}_{\boldsymbol{\xi}}\hat{u}_{\boldsymbol{\xi}}+S(\boldsymbol{\xi})\ \ \forall\boldsymbol{\xi}=[\xi_1,\xi_2]\in\Big\{-N,...,N\Big\}^2
\end{aligned} \tag{4.10}$$

Chapter 5

Numerical experiments

5.1 Introduction

The preceding sections have grounded a theoretical fundament for the Fourier-Galerkin method. In this project, we have implemented in code and tested the two different methods mentioned in Chapter 4; both an "exact" convolution solver for the special quadratic cases and a generic FFT solver where f and A can be specified arbitrarily.

For both methods, a series of simulations have been conducted to verify convergence and analyze qualitative properties of the numerical solutions. Testing the code is an important step in any software development process, and in particular for mathematical software, since small bugs in the code quickly can lead to inaccurate and unstable results.

The software is written in MATLAB for the convenience of easy prototyping, except a quadrature module for calculating the integrals deriving from the nonlocal operator. The quadrature module is written in C, and called as a mex plugin in MATLAB (see also Section 6.2 and Appendix B for more details and example code). All simulations have been carried out on a Windows 8 (64 bit) Lenovo T500 laptop, with a duo-core processor running at 2.8 GHz and 8 GB of memory.

5.2 Qualitative behavior of solutions

By having a mathematical understanding of the effect of terms involved in the equation, one has a qualitative feeling of how the numerical solutions should look like. For instance, the nonlocal operator acts as a smoothing, averaging agent, which tears down sharp corners and discontinuities. The effect of this however depends on the parameter α, and it is therefore interesting to run simulations and see how it affects the smoothing effect of the operator. In the introduction of [2], Jakobsen and Cifani refer to recent works that show that for nonlinear fractional conservation laws, solutions are smooth when $1 \leq \alpha < 2$, and that shock

discontinuities can occur, even from regular initial data, when $0 < \alpha < 1$.

With many parameters to vary, there are a vast number of interesting cases to test and simulate. We give some examples of different equations with discontinuous initial data to illustrate the effect of different types of diffusion. Some of the test cases presented below are with and some are without convection to highlight the difference. Also, each equation is simulated with different values of α, with and without the spectral vanishing viscosity so that the effect of including this term in the equation becomes clear. The simulations in Section 5.2.4 and the 2D simulations are performed using the generic FFT solver, whereas the others are performed using the convolution solver. In all simulations in this section, the dimension of the discrete space is 512 for both the convolution solver and the generic FFT solver. Also, all simulations are run up to $T = 0.5$. The function H denotes the heaviside step function.

5.2.1 Quadratic convection and diffusion

The plots in figure 5.1 show solutions of the problem

$$\begin{aligned} &u_t + \frac{1}{2}\frac{\mathrm{d}}{\mathrm{d}x}u^2 = -(-\Delta)^{\frac{\alpha}{2}}u^2, \;\; (x,t) \in (0,2\pi) \times [0,T], \\ &u(x,0) = H\Big(x - \frac{\pi}{2}\Big) - H\Big(x - \frac{3\pi}{2}\Big) - \frac{1}{2} \end{aligned} \tag{5.1}$$

for different values of α, with and without SVV added to the equation. The plots show that for $\alpha < 1$, the Gibbs oscillations that are caused by the discontinuous initial data are scattered in the whole solution curve with peaks around the initial discontinuities, and they do not vanish over time unless SVV is added. Interestingly, the solution becomes increasingly unstable for increasing α, and diverges rapidly for $\alpha = 1.0$ and $\alpha = 1.5$. Notice that there is a certain under- and overshooting in the solutions, and hence the maximum principle is violated. This is an indication that the method is non-monotone, which we should expect for these spectral methods contrary to finite volume schemes like in [4].

5.2.2 Quadratic convection, linear diffusion

The solutions of the problem

$$\begin{aligned} &u_t + \frac{1}{2}\frac{\mathrm{d}}{\mathrm{d}x}u^2 = -(-\Delta)^{\frac{\alpha}{2}}u, \;\; (x,t) \in (0,2\pi) \times [0,T], \\ &u(x,0) = H\Big(x - \frac{\pi}{2}\Big) - H\Big(x - \frac{3\pi}{2}\Big) - \frac{1}{2} \end{aligned} \tag{5.2}$$

displayed in figure 5.2 show more stability than with quadratic diffusion. As expected, there are nonvanishing Gibbs' oscillations for $\alpha = 0.5$, but for $\alpha \geq 1$, these are smoothed out over time. Observe how the solutions behave qualitatively differently with and without SVV. The SVV term clearly reduces gradients near the discontinuities. Solutions of (5.1) and (5.2) with SVV are compared for $\alpha = 0.5$

in figure 5.4. With quadratic diffusion, we can see that more of the energy is conserved, and gradients are steeper.

Equation (5.2) was also simulated in 2D, and the plots are shown in figure (5.3). In these simulations, the discrete dimension is $32^2 = 1024$.

5.2.3 No convection, quadratic diffusion

With no convection the problem reads

$$\begin{aligned} &u_t = -(-\Delta)^{\frac{\alpha}{2}} u^2, \ \ (x,t) \in (0,2\pi) \times [0,T], \\ &u(x,0) = H\Big(x - \frac{\pi}{2}\Big) - H\Big(x - \frac{3\pi}{2}\Big) - \frac{1}{2}, \end{aligned} \tag{5.3}$$

and the solutions are displayed in figure 5.5. From the figure, we see that the behavior is similar to that of 5.1 except that the curve is symmetric since the convection term is missing. Also here, the solutions diverge for $\alpha \geq 1$ without SVV. This indicates that it is the quadratic diffusion which is troublesome, not the convection term, contrary to the linear diffusion case.

5.2.4 Fully degenerate diffusion

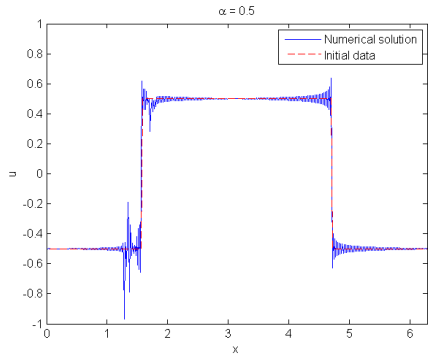
Another interesting case is strongly degenerate diffusion, i.e. A vanishes on a set of positive measure. To illustrate this case, we apply one of the degenerate functions used in [4], and the solutions are shown in figure 5.6. Note that this diffusion function is not covered in the theory that we have derived since it is not smooth, but it is interesting to try numerical simulations still. Observe that the solutions are not smoothed out without SVV for $\alpha \geq 1$ unlike with linear diffusion.

$$\begin{aligned} &u_t = -(-\Delta)^{\frac{\alpha}{2}} A(u), \ \ (x,t) \in (0,2\pi) \times [0,T], \\ &u(x,0) = H\Big(x - \frac{\pi}{2}\Big) - H\Big(x - \frac{3\pi}{2}\Big) - \frac{1}{2}, \\ &A(u) = \begin{cases} 0, & u \leq 0.5 \\ 5(2.5u - 1.25)(u - 0.5), & 0.5 < u \leq 0.6 \\ 1.25 + 2.5(u - 2.6), & u > 0.6 \end{cases} \end{aligned} \tag{5.4}$$

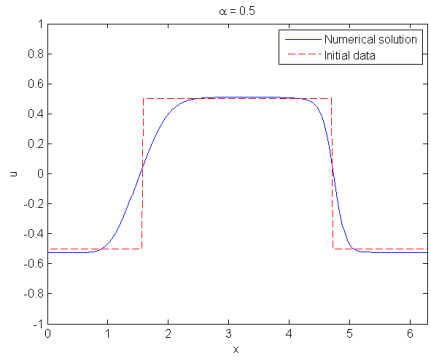
5.2.5 Comparison with the standard Laplacian

Although the case $\alpha = 2$ is not treated theoretically in this project, the numerical solvers are written to also handle this. Note that according to [9], Section 4.2.2, when $\alpha \to 2$, the fractional diffusion operator defined in 2.3 converges to $\frac{\Delta}{4\pi^2}$ and not Δ. The reason for this is related to which definition of the Fourier transform that is used. We have performed some simulations in the double-quadratic case 5.1 with different values of α approaching 2 and compared them to the numerical solution when $\alpha = 2$. The results are shown in figure 5.7. In 5.7b the difference

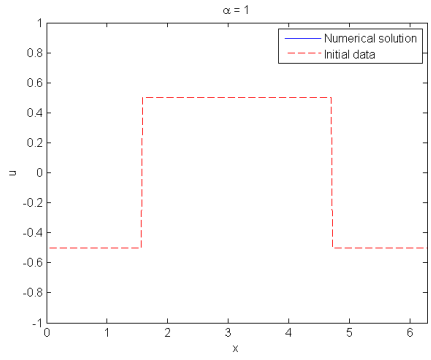
between the numerical solution at $\alpha = 2$ and values of α approaching 2 indicates that the solver converges to the "standard" Laplacian solution when $\alpha \to 2$.



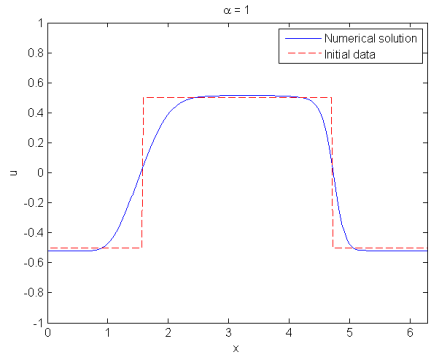
(a) Without SVV



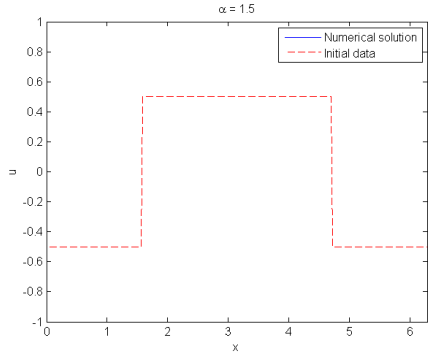
(b) With SVV



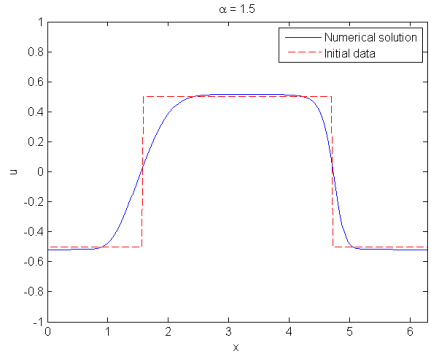
(c) Without SVV



(d) With SVV

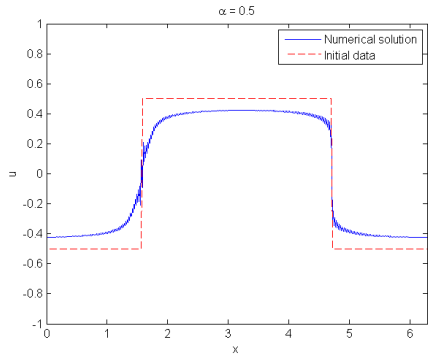


(e) Without SVV

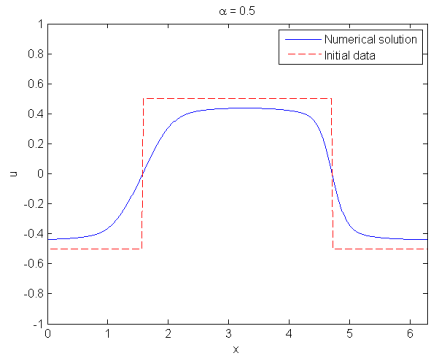


(f) With SVV

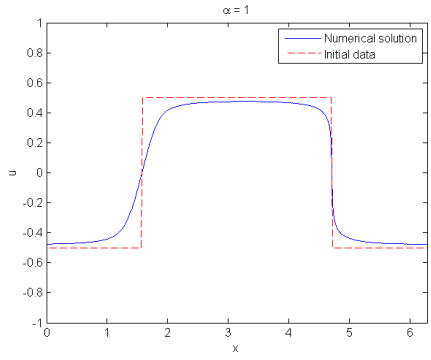
Figure 5.1: Numerical solution of (5.1)



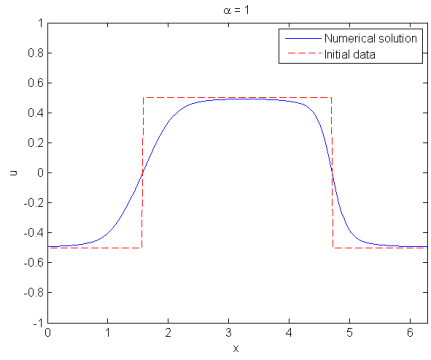
(a) Without SVV



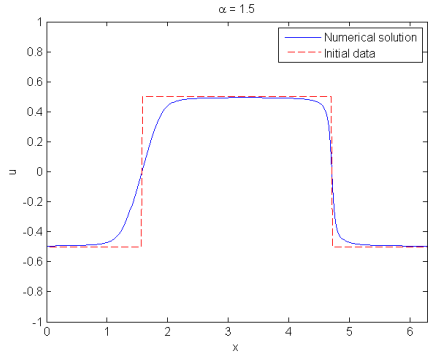
(b) With SVV



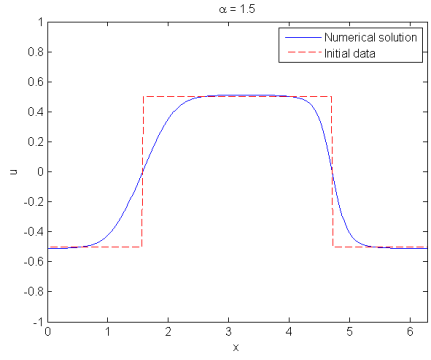
(c) Without SVV



(d) With SVV

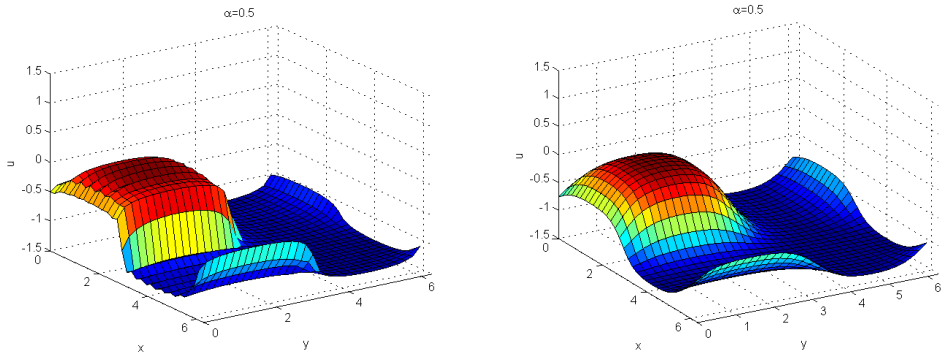


(e) Without SVV



(f) With SVV

Figure 5.2: Numerical solution of (5.2)



(a) Without SVV (b) With SVV

Figure 5.3: Numerical solution of (5.2) in 2 dimensions

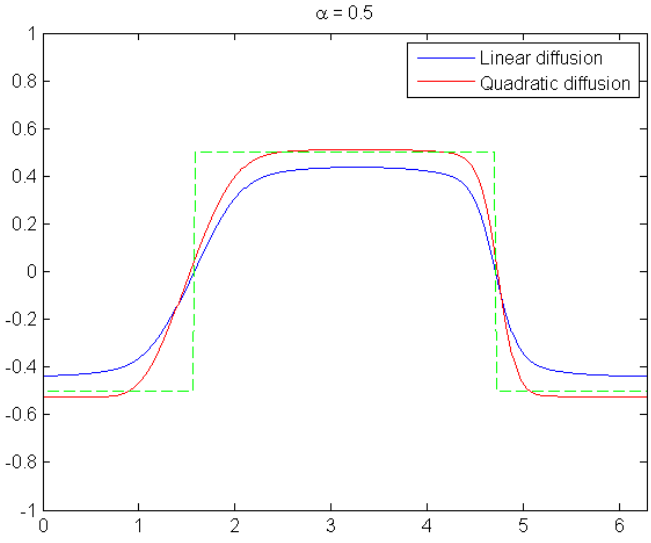
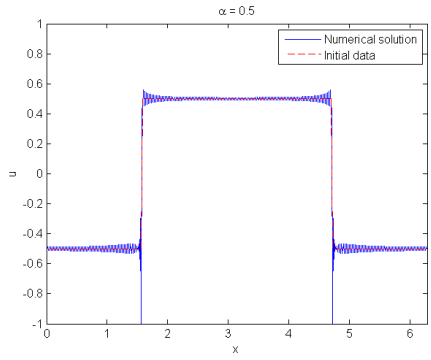
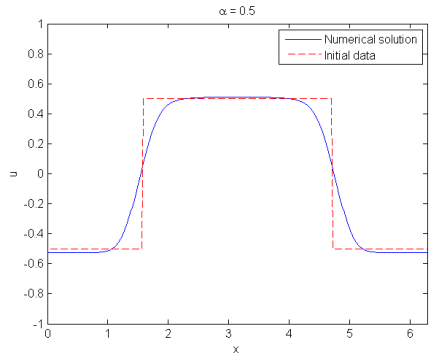


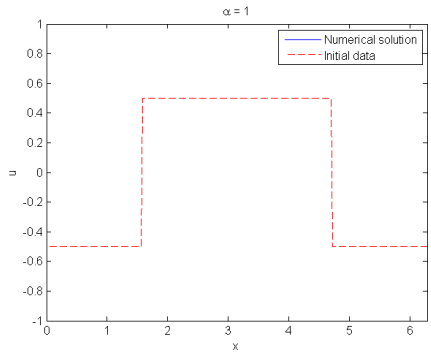
Figure 5.4: Comparison between (5.1) and (5.2)



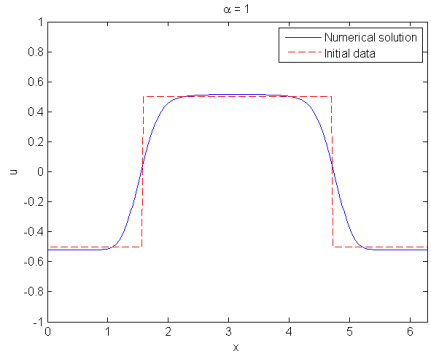
(a) Without SVV



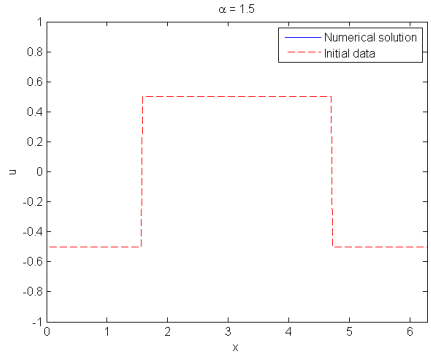
(b) With SVV



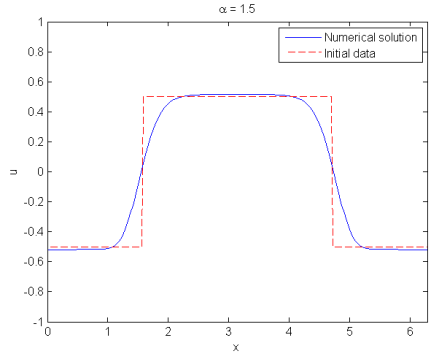
(c) Without SVV



(d) With SVV

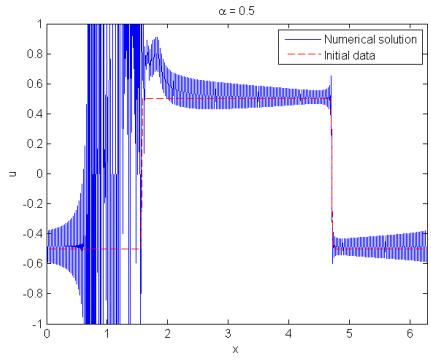


(e) Without SVV

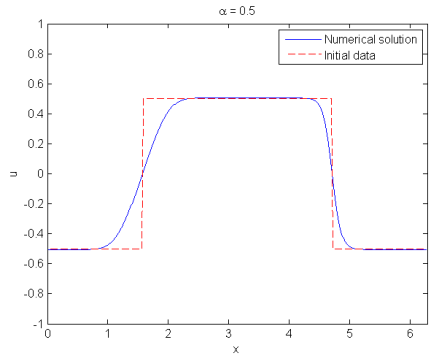


(f) With SVV

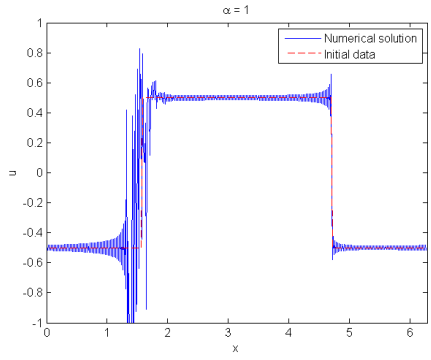
Figure 5.5: Numerical solution of (5.3)



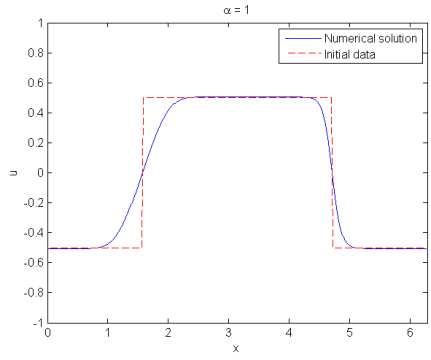
(a) Without SVV



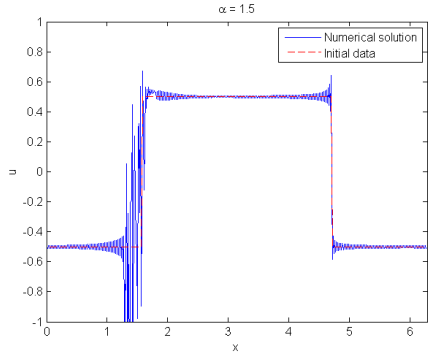
(b) With SVV



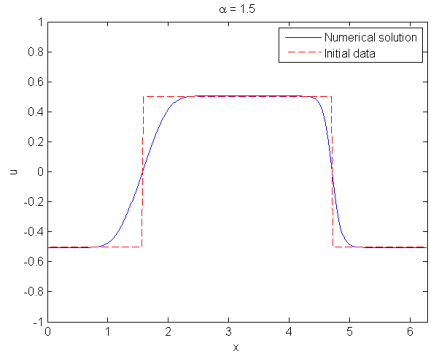
(c) Without SVV



(d) With SVV

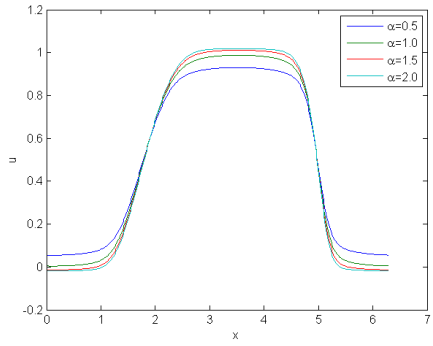


(e) Without SVV

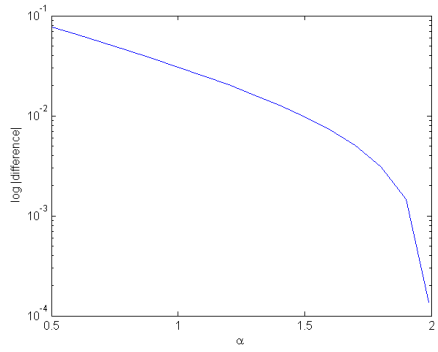


(f) With SVV

Figure 5.6: Numerical solution of (5.4)



(a) Comparison for different α



(b) Convergence towards the solution with $\alpha = 2$

Figure 5.7: Comparison of solutions when $\alpha \to 2$

5.3 Convergence

A measure of the numerical convergence is the most important verification that the method is implemented correctly, and also an indication that the theoretical analyses performed in the previous chapters are valid. We have performed convergence tests for quadratic, linear and degenerated diffusion functions. The new case is of course the quadratic one, and it is interesting to compare these results to the linear case. The fully degenerated case is also included, as it is of interest to see for instance whether it has different convergence properties than other types of nonlinear, non-degenerated diffusion.

In each case, the numerical convergence has been measured with and without SVV for both $\alpha < 1$ and $\alpha > 1$, for both discontinuous initial data and infinitely smooth, periodic initial data. The non-smooth initial data used in the simulations are

$$u(x, 0) = \operatorname{sgn}(\pi - x), \tag{5.5}$$

and the smooth initial data are

$$u(x, 0) = \sin(\cos x). \tag{5.6}$$

For linear equations and "standard" spectral methods, the convergence depends on the regularity of solutions. If the solutions are infinitely smooth and periodic, the convergence is expected to be exponential, but otherwise not. From the theoretical analysis in Chapter 3, we should expect exponential convergence with the SVV method.

There are however some problematic issues related to the measure of convergence. The initial data need to be projected onto the discrete space before the solver

can start. The projection of these is carried out using the built-in quadrature in MATLAB to calculate the integrals

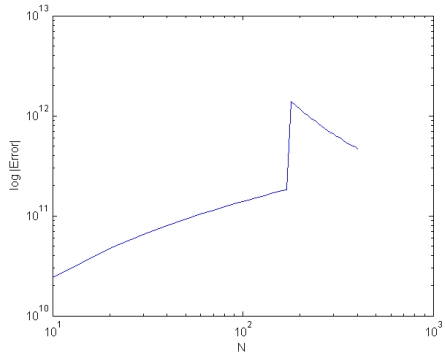
$$\hat{u}_0^\xi = \int_0^{2\pi} u_0(x) \cdot \mathrm{e}^{-i\xi x} \, \mathrm{d}x.$$

For large ξ, and especially if $u_0(x)$ is non-smooth, the quadrature is not able to integrate the expressions to machine precision and issues warnings about this. The precision limit was typically approximately 10^{-3} in these experiments. So when calculating a reference solution with a high space dimension, the integrals become inexact. This limits the accuracy which is possible to obtain with the method, and the effect is that it is not feasible to reach machine precision in the simulations.

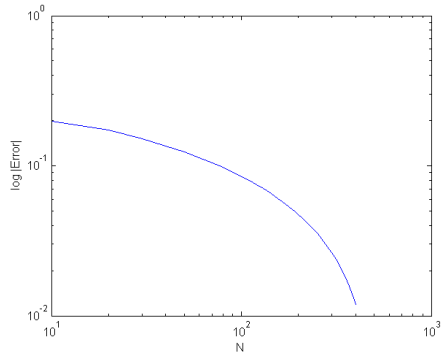
Analytic solutions of the problem posed on a finite domain are hard to get hold of (for the Cauchy problem, there exist for instance Barrenblatt solutions, cf. [11]). Therefore, a numerical solution with high refinement is used as a reference point instead. The numerical solutions are computed for increasing dimension of the discrete space. For each N, the error is measured in the standard discrete l^2 norm for the difference between the coefficients in the test solution and the appropriate, corresponding Fourier coefficients in the high order numerical solution. The results are plotted with logarithmic axes and shown in figures 5.8 through 5.13. A downward bending curve indicates exponential convergence, whereas a straight line with negative slope indicates polynomial convergence.

The conclusion is that there is strong evidence of numerical convergence for the SVV method in all cases. The results indicate that the asymptotic convergence of the method is exponential with linear and quadratic diffusion, but only polynomial with degenerate diffusion (and nonsmooth data). This result is in line with the theoretical expectations. The absolute error obtained is however not as small as one could desire (like machine precision) due to the above mentioned quadrature inaccuracy. There is also probably considerable room for tuning, for instance with regard to the choice of ϵ_N and m_N, to further improve the rate of convergence.

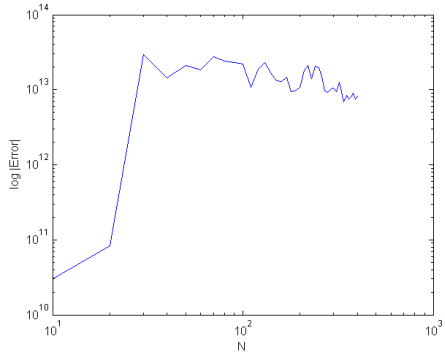
In all cases, the value of α does not affect the convergence when SVV is applied. However, when the SVV term is dropped, there are some interesting effects occurring. For instance, with quadratic diffusion and smooth initial data, the method converges for $\alpha = 0.5$, but not for $\alpha = 1.5$ and neither with nonsmooth initial data. With linear diffusion, it converges with smooth and nonsmooth initial data at least when $\alpha = 1.5$. With degenerate diffusion, it diverges with nonsmooth data and appears to converge polynomially with smooth data. Also not that when the method converges without SVV, the rate is considerably higher than with SVV. This is probably a part of the price that must be paid to guarantee convergence.



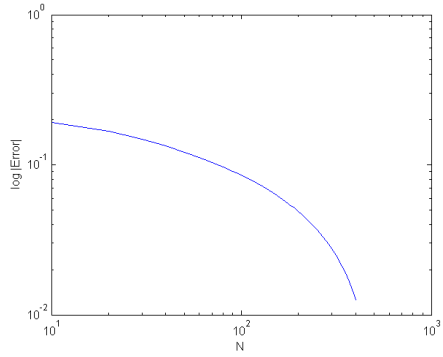
(a) $\alpha = 0.5$, without SVV



(b) $\alpha = 0.5$, with SVV

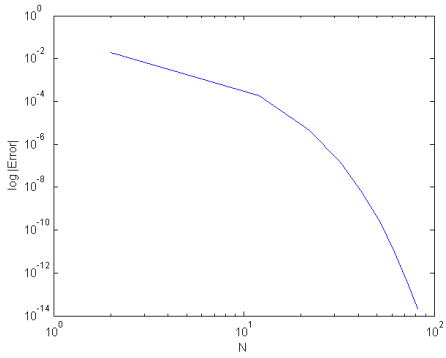


(c) $\alpha = 1.5$, without SVV

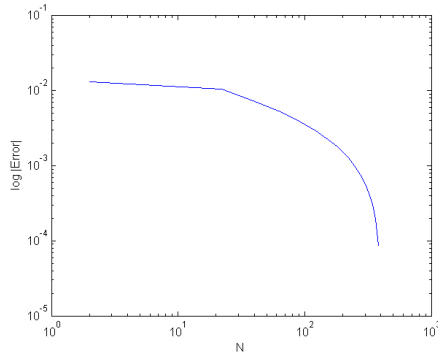


(d) $\alpha = 1.5$, with SVV

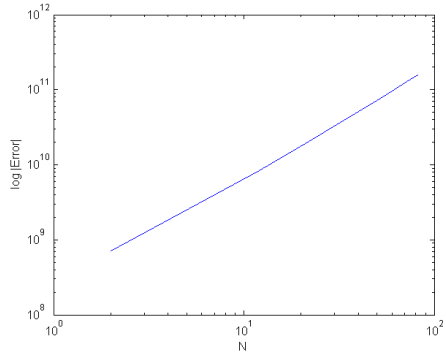
Figure 5.8: Convergence plots for equation (5.1) with initial data (5.5)



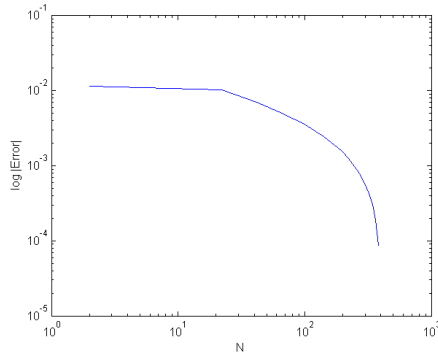
(a) $\alpha = 0.5$, without SVV



(b) $\alpha = 0.5$, with SVV

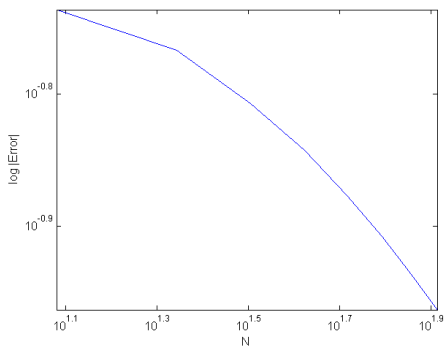


(c) $\alpha = 1.5$, without SVV

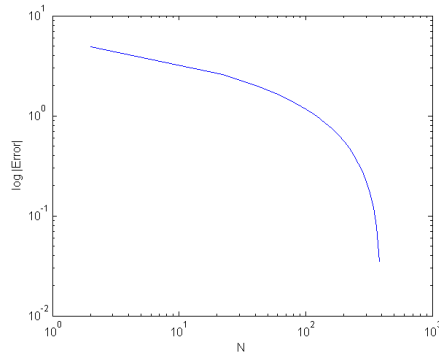


(d) $\alpha = 1.5$, with SVV

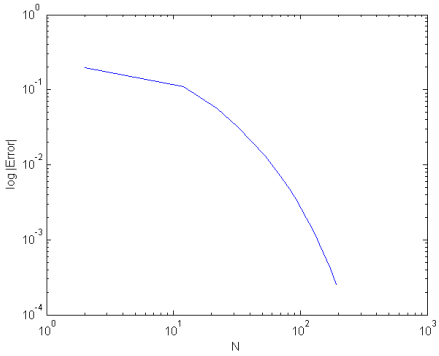
Figure 5.9: Convergence plots for equation (5.1) with initial data (5.6)



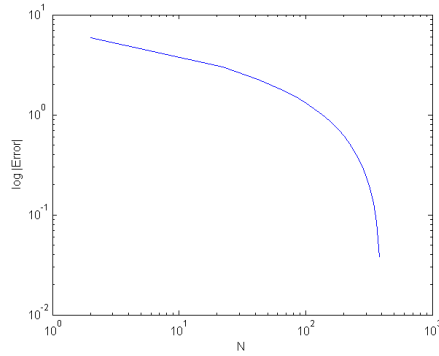
(a) $\alpha = 0.5$, without SVV



(b) $\alpha = 0.5$, with SVV

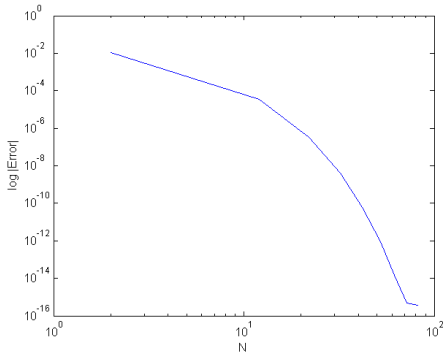


(c) $\alpha = 1.5$, without SVV

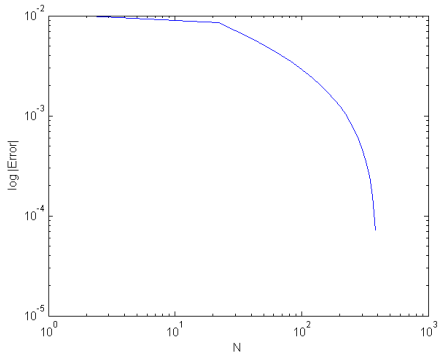


(d) $\alpha = 1.5$, with SVV

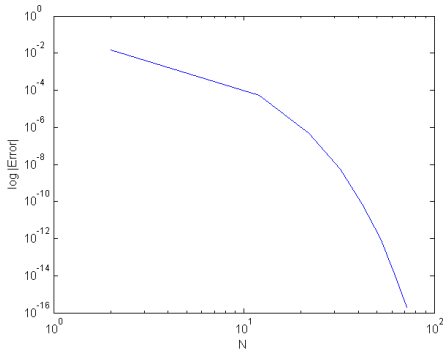
Figure 5.10: Convergence plots for equation (5.2) with initial data (5.5)



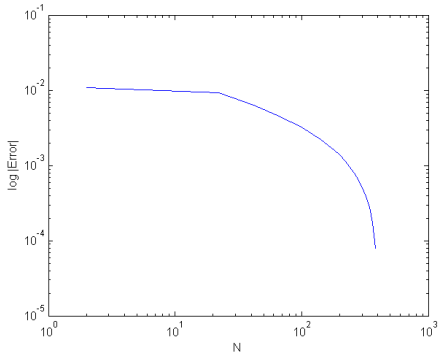
(a) $\alpha = 0.5$, without SVV



(b) $\alpha = 0.5$, with SVV

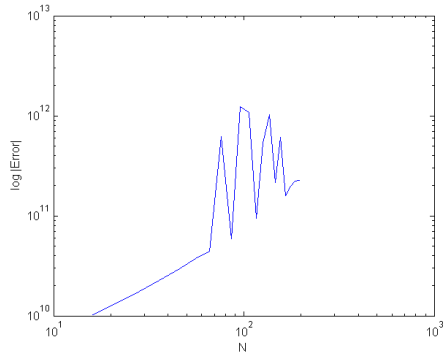


(c) $\alpha = 1.5$, without SVV

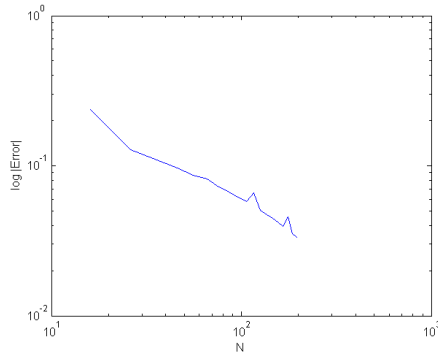


(d) $\alpha = 1.5$, with SVV

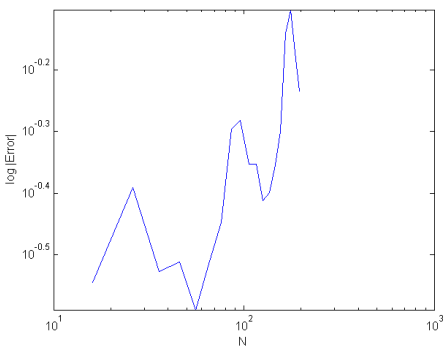
Figure 5.11: Convergence plots for equation (5.2) with initial data (5.6)



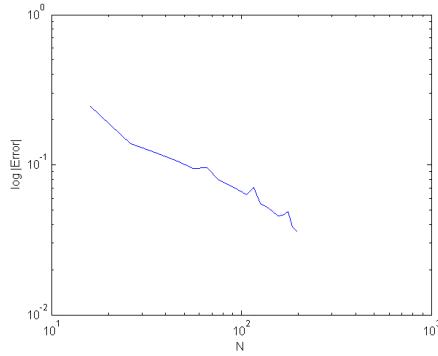
(a) $\alpha = 0.5$, without SVV



(b) $\alpha = 0.5$, with SVV

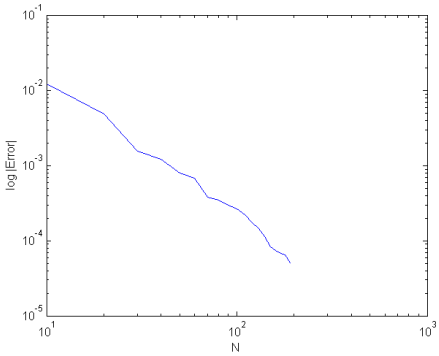


(c) $\alpha = 1.5$, without SVV

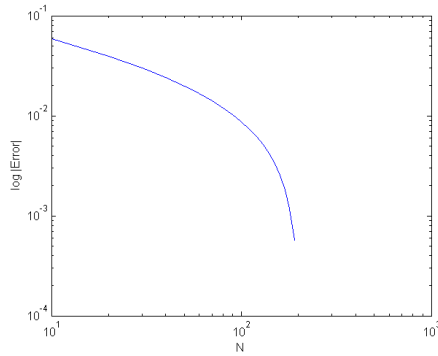


(d) $\alpha = 1.5$, with SVV

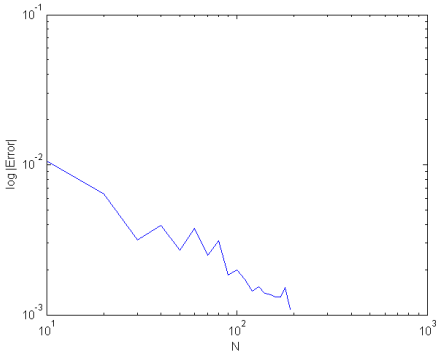
Figure 5.12: Convergence plots for equation (5.4) with initial data (5.5)



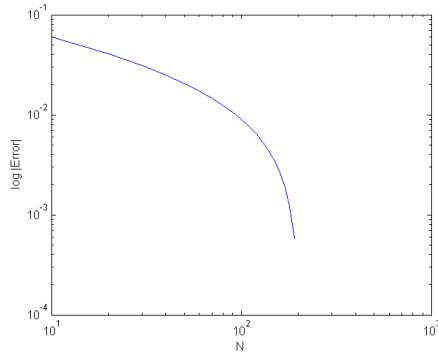
(a) $\alpha = 0.5$, without SVV



(b) $\alpha = 0.5$, with SVV



(c) $\alpha = 1.5$, without SVV



(d) $\alpha = 1.5$, with SVV

Figure 5.13: Convergence plots for equation (5.4) with initial data (5.6)

5.4 Computational complexity

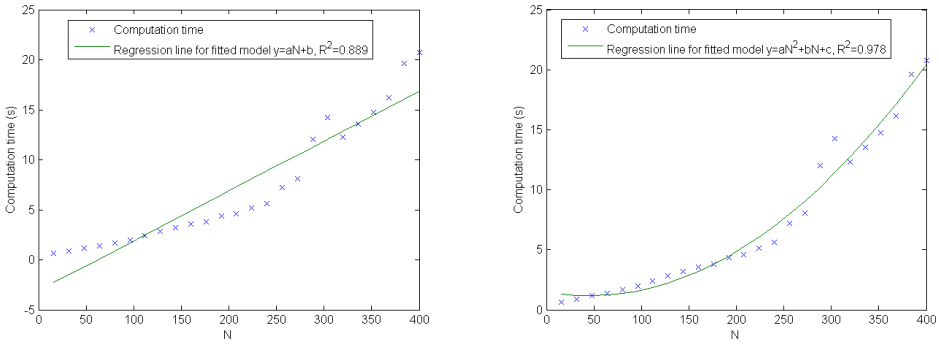
Another very important aspect of a numerical solver is the computational costs that it induces. It is of little practical use to develop a solver which does not have a reasonable tradeoff between the computational costs and the accuracy that is possible to obtain. The computation time of both solvers is therefore measured as a function of the discrete space dimension. Both solvers are tested using the same test setup, which is the computation of solutions to the problem

$$u_t + \frac{1}{2}\frac{\mathrm{d}}{\mathrm{d}x}u^2 = -(-\Delta)^{\frac{1}{2}}u^2 + \partial_x^2 R_N * u, \quad (x,t) \in \Omega \times [0, 0.5]$$
$$u(x,0) = \mathrm{sgn}(\pi - x)$$

(i.e. using SVV). In addition, the same equation is solved and the computation time measured in two spatial dimensions using the FFT solver, since this is a priori expected to be the fastest solver and therefore most interesting to implement in higher dimensions. The simulations were carried out several times, and the average of the simulations is used as the measurement. These measurements are plotted together with least-squares-fits of asymptotic computation time models and the statistical coefficient of determination, R^2.

5.4.1 Exact solver

The results are shown in figure 5.14. Not surprisingly, the linear fit in 5.14a seems to be inferior to the quadratic fit in 5.14b, with the latter having a coefficient of determination of around 98%. This is in line with what we can expect when using naïve convolution products to calculate the quadratic projections. It is possible to improve this by using fast convolution algorithms, but that has not been implemented in this project.



(a) Linear fit

(b) Loglinear fit

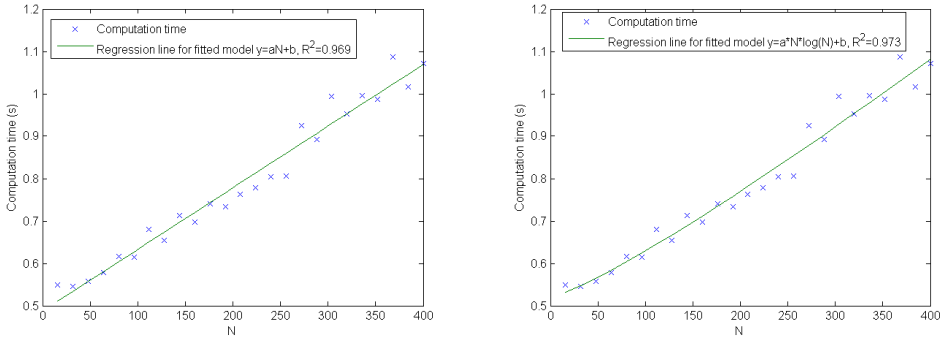
Figure 5.14: Computation time simulations for the exact solver in one dimension

5.4.2 FFT solver

One dimension

The results are shown in figure 5.15. Theoretically, the computational complexity is expected to be loglinear since the fast fourier transform is used to evaluate approximations of the quadratic projections. The loglinear fit in 5.15b has the highest coefficient of determination, as we could expect, but the linear fit in 5.15a is also very good. This is an indication that the FFT algorithm in MATLAB is very efficient and has low coefficients in the asymptotic expression. Also note that the absolute computation time in seconds is very low compared to the exact solver.

In [16], the measurements indicated linear computation time for a similar linear fractional diffusion equation solver, so the additional costs for computing nonlinear terms is very low with the FFT approach.



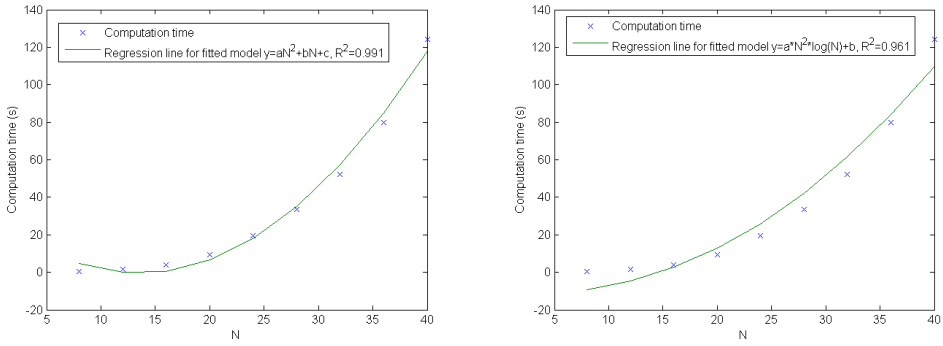
(a) Linear fit

(b) Log-linear fit

Figure 5.15: Computation time simulations for the FFT solver in one dimension

Two dimensions

The diagonal structure of the discrete operators and the inherently low computational complexity makes the FFT solver very attractive to implement in higher dimensions also. In d dimensions, The theoretical complexity is $\mathcal{O}\big(N^d \log N\big)$, compared to $\mathcal{O}\big(N^{2d}\big)$ for a naïve solver. This makes the algorithm almost optimal, in the sense that it requires $\mathcal{O}\big(\log N\big)$ operations per unknown (an optimal solver would require a constant amount of work per unknown). The results from computation time simulations with the 2D solver are shown in 5.16.



(a) Quadratic fit

(b) Log-quadratic fit

Figure 5.16: Computation time simulations for the FFT solver in two dimensions

Somewhat surprisingly perhaps, the quadratic model has a slightly higher value of R^2. In any case, the results show that the complexity is attractively low also in two dimensions.

Chapter 6

Implementation concerns

6.1 Aliasing errors

6.1.1 Introduction

In the field of numerical solution of nonlinear partial differential equations, it is often necessary to introduce some kind of linearization of the nonlinear terms to be able to calculate an approximative solution. One way could be to solve the linearization of the equation around a known solution, if available. In the case of finite difference methods, another way is to use iterative optimization algorithms to search for solutions of the discretized nonlinear equation.

When working with spectral methods, or finite element methods for that matter, one uses a discrete function space of a certain, finite dimension to approximate the solution. Any linear combination of the basis functions resides in this space, but nonlinear combinations of these do not necessarily reside in the approximation space. Take for instance a quadratic term. In the setting of the FFT solver of this project, the example term becomes

$$u^2 = \Big(\sum_{\xi=-\frac{N}{2}}^{\frac{N}{2}-1} a_\xi \mathrm{e}^{i\xi x} \Big)^2 = \sum_{\xi=-N}^{N-2} M_\xi \mathrm{e}^{i\xi x}$$

for some M_ξ depending nonlinearly on the a_ξ. Hence, the approximation space needed to represent this term has dimension $2N-1$, not N. To make an approximation within the discrete space, one can use the closest-point projection of the nonlinear term onto the discrete space. In this project, the L^2 inner product is used, and the projection is hence essentially a truncated Fourier series representation of the nonlinear term, such that

$$u^2 = \sum_{\xi=-N}^{N-2} M_\xi \mathrm{e}^{i\xi x} \approx P_N u^2 = \sum_{\xi=-\frac{N}{2}}^{\frac{N}{2}-1} b_\xi \mathrm{e}^{i\xi x}$$

with

$$b_\xi = \frac{1}{2\pi}\int_0^{2\pi} u^2(x)\cdot \mathrm{e}^{-i\xi x}\,\mathrm{d}x = \frac{1}{2\pi}\int_0^{2\pi}\Big(\sum_{p=-N}^{N-2} M_\xi \mathrm{e}^{ipx}\Big)\cdot \mathrm{e}^{-i\xi x}\,\mathrm{d}x \tag{6.1}$$

This projection representation is not exact. In this project, the projections are shown to be spectrally accurate, i.e. the error decays exponentially fast with increasing dimension number N, but it still constitutes a source of error. Generally speaking, since this type of error is of the same asymptotical order as other error sources, it needs no special treatment with regard to the convergence properties of the numerical method. Also, when using quadrature to approximate the projection, the additional error due to the quadrature is no worse than the quadrature error committed elsewhere in the method (for instance for source terms). If the quadrature is spectrally accurate, the method does not lose its asymptotic convergence rate. It should therefore be noted that attempting to diminish this particular error source is a matter of making the solution look more smooth, and does not affect the convergence rate of the method. In the solution of some equations, it turns out that the particular type of error due to the discrete approximation of nonlinear projections manifests itself as ripples which are not aesthetically pleasing. Hence, using a technique to reduce this phenomenon is of interest to improve the appearance of the numerical solution.

This particular numerical error is a purely nonlinear phenomenon, since linear equations do not impose the need for projections. As Rønquist discusses in [15], one can interpret the error in terms of the frequency components. The nonlinear term has higher frequency components than what is included in the discrete space of the numerical method. When calculating the projection of the nonlinear term onto the discrete space of lower dimension, the higher frequencies do interfere with the lower frequency components. This is only possible in the nonlinear case, since there are cross terms and interactions between the frequency components. For linear combinations of frequencies, there is no interaction between different frequency components. In a discrete sampling, a high frequency mode can appear to be a lower frequency mode by having the same sampled values on the chosen grid. This phenomenon is called aliasing and is illustrated in figure 6.1, where an equidistant grid is used to sample the functions $\sin \pi x$ and $\sin -9\pi x$ as in [15].

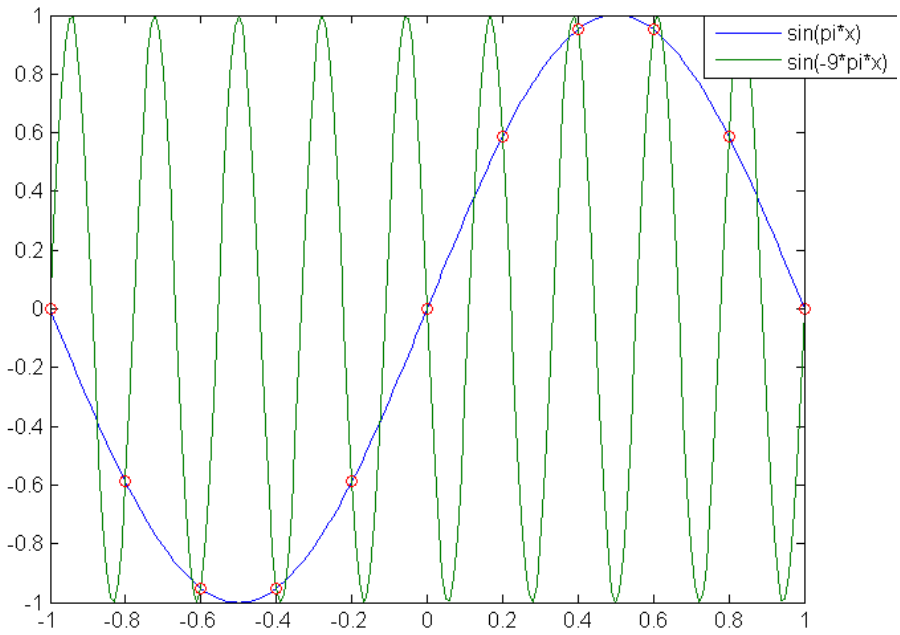


Figure 6.1: Sampled points are indistinguishable

The FFT is applied in practice to evaluate the integral (6.1). The FFT is a discrete sampling on an equidistant grid, so for instance the two components used as an example in figure 6.1 would experience constructive interference, and the low frequency mode is thus overestimated. This would not be a problem if the integral in (6.1) is calculated analytically, but with quadrature, the discrete sampling can cause this phenomenon. The higher frequency components, that ought to have zero amplitude in the approximation, look like they have lower frequency and therefore pollute the lower frequency modes, causing ripples in the solution.

6.1.2 Dealiasing

A practical remedy to the aliasing problem is presented in [15]. Historically, the problem of aliasing was first addressed in numerical methods developed for meteorological models in the 1950s and 1960s, according to [14]. The idea is to calculate an expanded sample in the time domain by temporarily extending the discrete space of basis functions with zero modulated higher frequencies. Then, the nonlinear term is calculated at the expanded grid using the larger sample. Lastly the nonlinear coefficients are obtained from the sampled nonlinear term and truncated to the original dimension of the discrete space. Let $f(u)$ be the nonlinear function

of u, N be the original discrete dimension, $M > N$ be the extended discrete dimension, $\mathbf{a}_\xi$ be a vector of length N with the coefficients of u as above, and let $\tilde{\mathbf{b}}_\xi$ be a vector of length N with the numerical (DFT) approximation of (6.1). Using this notation, the scheme in figure 6.2 illustrates the dealiasing procedure.

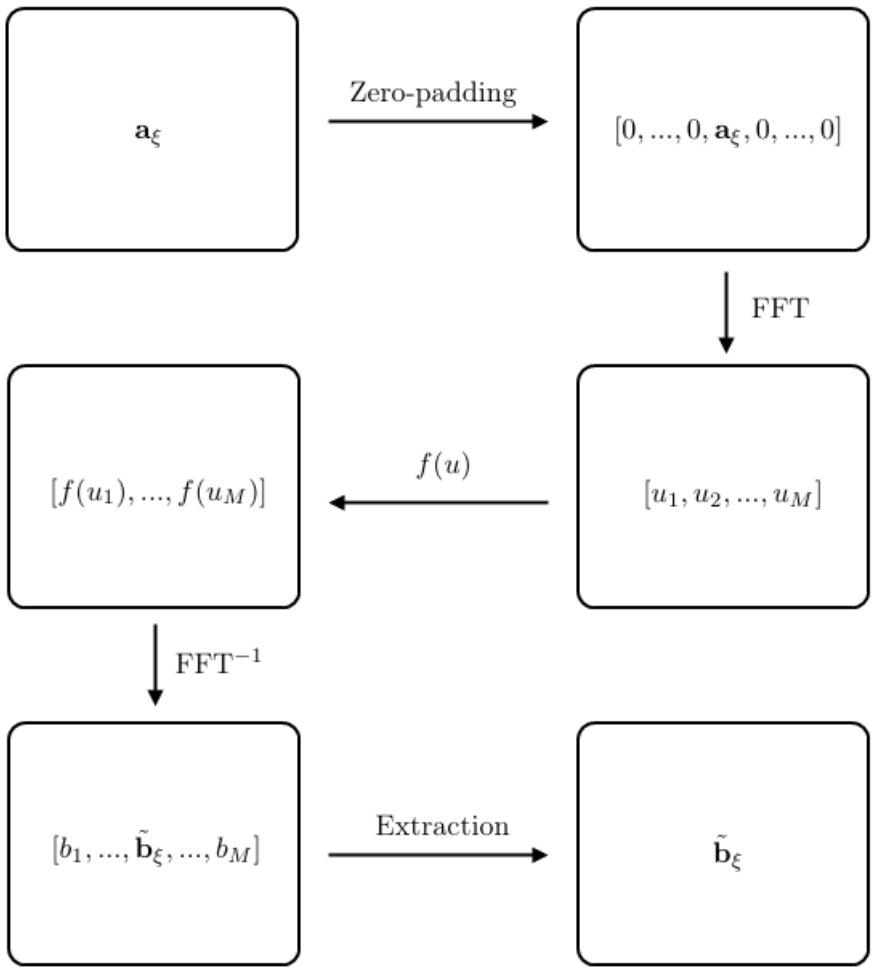


Figure 6.2: Dealiasing principle

It can be shown that the zero-padding procedure in the Fourier domain corresponds to polynomial interpolation in the physical domain. This fact can be seen in figure 6.2. Here, the square of a function u is calculated directly from u and sampled on N points. u^2 is also calculated using the zero-padding technique, sampling it on $M = \frac{3}{2}$ points. The dealiased approximation of u^2 has an extra point between two values of the nondealiased u^2, interpolating in between.

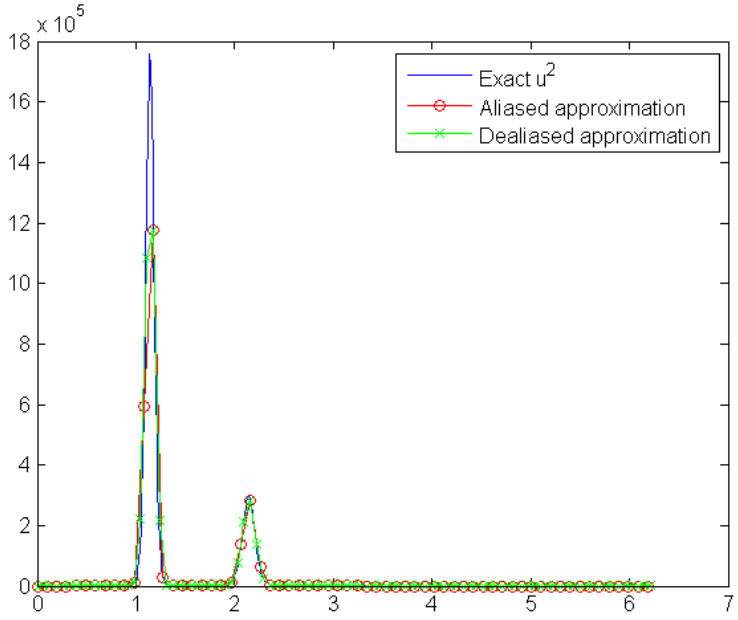


Figure 6.3: Interpolation in the time domain

There is freedom related to the choice of M, i.e. the size of the extended, zero-padded spectrum. This is a tradeoff issue between the computational complexity and the degree to which ripples in the solution are to be smoothed out. The optimal size is likely to depend on the nonlinearity of the term to be approximated. If the nonlinearity is not too strong, several authors report that the so-called $\frac{2}{3}$ rule attributed to Orszag suffices. According to [14], section 11.5. Orszag showed that $M = \frac{3}{2}N$ is the optimal choice for a quadratic nonlinearity, i.e. u^2, in the sense that this ensures N alias free coefficients.

6.1.3 Numerical results

Korteweg-de-Vries' equation

In [15], Rønquist treats a standard Fourier-Galerkin spectral method for the Korteweg-de-Vries' (KdV) equation $u_t + \frac{1}{2}\frac{\mathrm{d}}{\mathrm{d}x}u^2 - u_{xxx} = 0$, posed on a compact domain with soliton waves as initial data and periodic boundary conditions ($u_0(x) = 3\alpha_1^2\mathrm{sech}^2\left(\alpha_1\frac{(x-x_1)}{2} - \alpha_1^3 t\right)) + 3\alpha_2^2\mathrm{sech}^2\left(\alpha_2\frac{(x-x_2)}{2} - \alpha_2^3 t\right)$, $x_1 = \pi - 2, x_2 = \pi - 1, \alpha_1 = 25, \alpha_2 = 16$). With the given initial data, the analytic solution of the Cauchy problem are traveling soliton waves, so similar behavior is to be expected also on a finite domain with periodic boundary conditions. In the project, a solver was built for the KdV equation to explore the dealiasing technique for an equation with known behavior, and to compare the results with the effects of the technique applied to the main equation of study, namely the fractional conservation law.

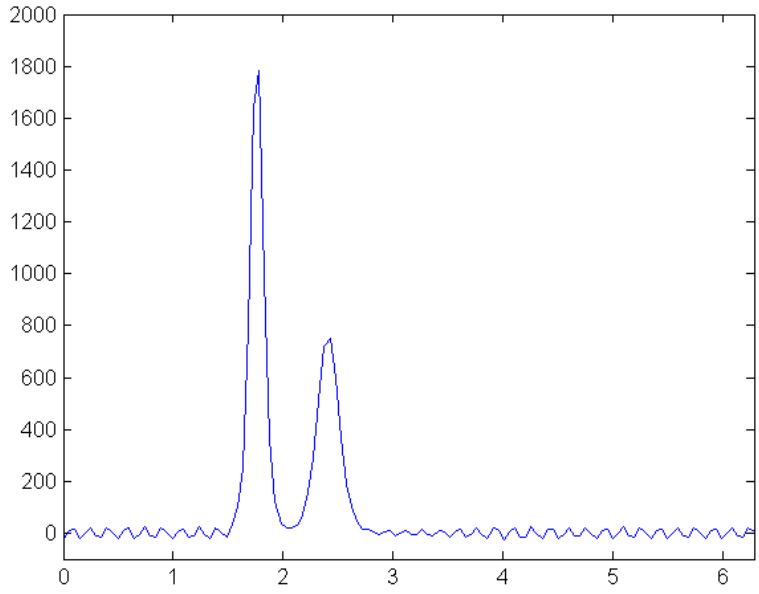
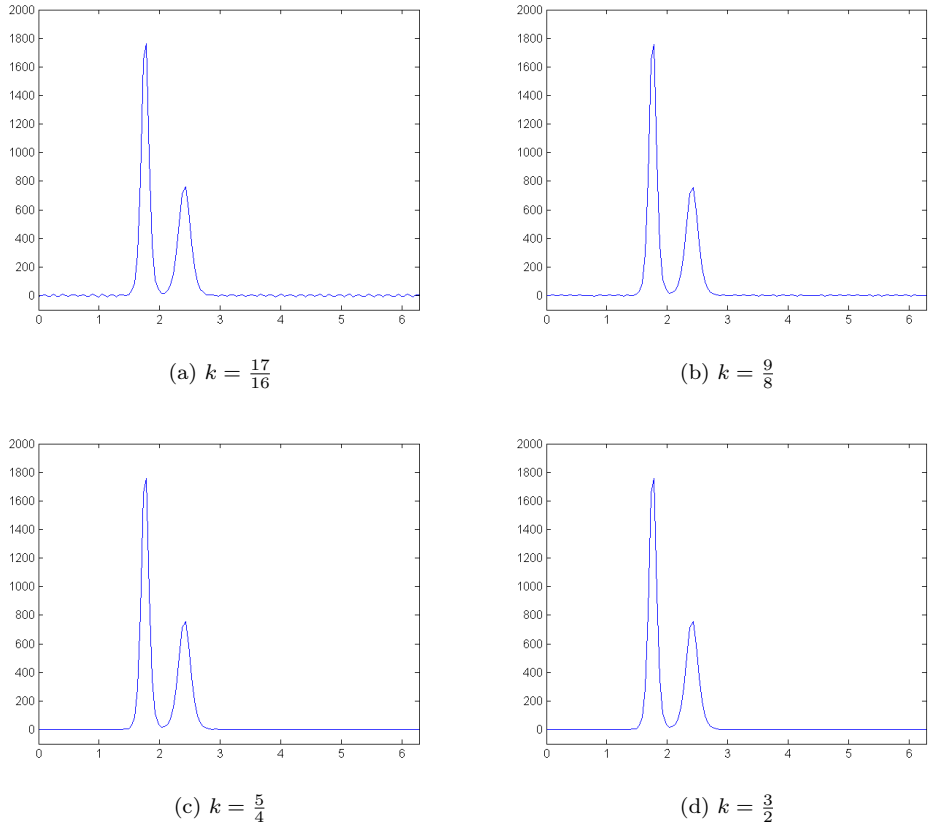


Figure 6.4: Numerical solution of KdV equation - no dealiasing applied

The aliased numerical solution is shown in figure 6.4, where the aliasing ripples are quite marked. In figure 6.5 the dealiasing procedure is applied to the solution at each timestep. The constant k denotes the size of the expanded spectrum by the relation $M = kN$, such that there are $\frac{(k-1)N}{2}$ zeros padded on each side of the spectrum. One can clearly see how the dealiasing procedure quickly improves the appearance of the solution. For $k = \frac{9}{8}$ the ripples are negligible, and they appear

to be completely smoothed out for $k = \frac{5}{4}$, since there is no vast improvement compared to $k = \frac{3}{2}$. Thus it seems that although Orszag's rule is theoretically correct, it can suffice with an even smaller spectrum. Heuristically speaking, this should be tested for the equation, initial data and nonlinearity in question.



(a) $k = \frac{17}{16}$ (b) $k = \frac{9}{8}$

(c) $k = \frac{5}{4}$ (d) $k = \frac{3}{2}$

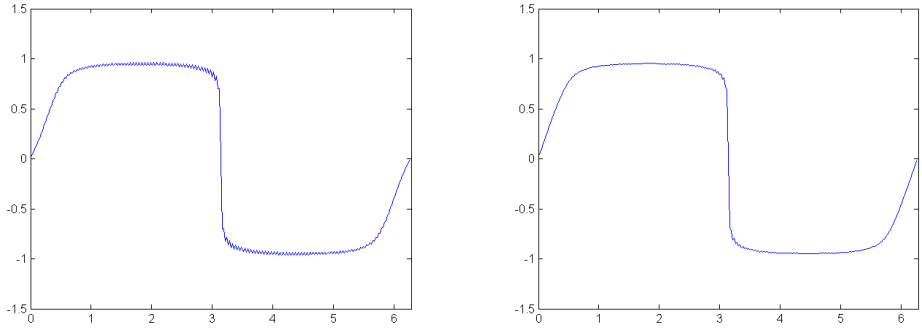
Figure 6.5: Numerical solution of KdV equation - dealiasing applied

Fractional conservation laws

For the fractional conservation law under study, there are possibly nonlinear terms, projections of these and quadrature approximations of the projections. In the light of the above analysis, it is therefore a legitimate hypothesis that there could occur aliasing errors in the numerical method. However, the Korteweg-de-Vries' equation has qualitatively different properties compared to the fractional conservation law. One central difference is the (possible) presence of fractional diffusion in equation (2.2). In the numerical method, spectral vanishing viscosity (SVV) is also present.

The qualitatively perhaps most important property of diffusion is that it regularizes solutions. Sharp corners and discontinuities are smoothed, and eventually solutions are "smeared" out under regular diffusion, and the fractional diffusion has similar properties, at least for $\alpha > 1$. A natural question to ask is then whether the diffusion can remove numerical ripples caused by the aliasing errors by itself, thereby making the dealiasing procedure redundant in the numerical method.

To test the numerical solver, different equations and initial data were posed. The solver was then configured with and without spectral viscosity and dealiasing routines to identify the cases where aliasing ripples occurred and whether fractional diffusion and/or spectral viscosity could remove these.



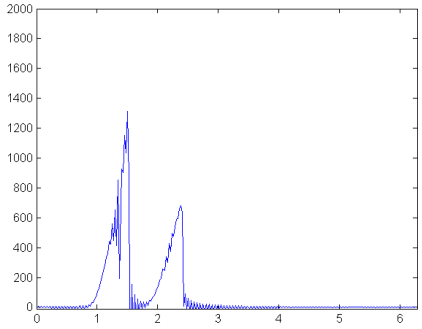
(a) No dealiasing

(b) Dealiasing applied

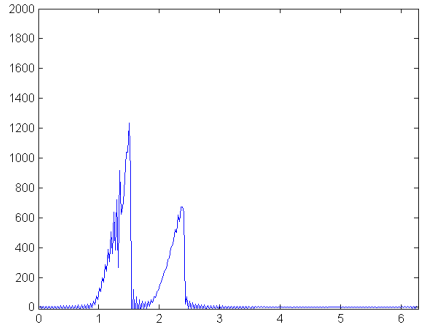
Figure 6.6: Numerical solution of $u_t + 0.5[u^2]_x = -(-\Delta)^{\frac{1}{2}}u + \epsilon_N \partial_x^2 Q * u$.

In figure 6.6a, weak SVV is applied. In figure 6.6b, the same amount of SVV is applied, but dealiasing is also enabled, using the Orszag $\frac{2}{3}$ rule. In both simulations, ripples are visibly present in the solution. However, the dealiased solution has markedly smaller ripples. If the ripples were purely caused by aliasing errors, no ripples should be present at all, according to Orszag's rule (since the nonlinearity is quadratic). This is evidence supporting that the ripples are a combination of aliasing errors and spurious Gibbs oscillations. The Gibbs oscillations occur when there are discontinuities in the initial data. The dealiasing technique does not remove such oscillations, so the remaining ripples in figure 6.6b are caused by the Gibbs' phenomenon.

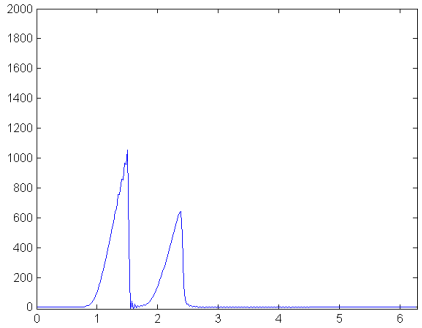
For continuous initial data with steep gradients, numerical experiments indicate that the module of SVV term must be fairly strong to stabilize the solution. To compare the solver to the one developed for the KdV equation, the same, steep initial data as in figure 6.5 were used.



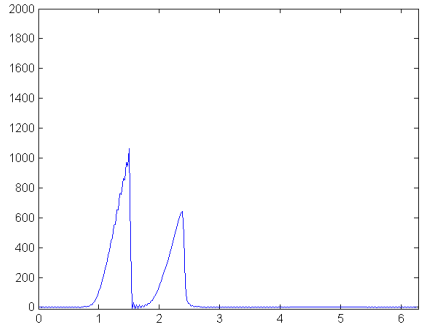
(a) Strong SVV, no dealiasing



(b) Strong SVV, dealiasing



(c) Even stronger SVV, no dealiasing



(d) Even stronger SVV, dealiasing

Figure 6.7: Numerical solution of $u_t + 0.01[u^2]_x = -(-\Delta)^{\frac{1}{2}}u + \epsilon_N \partial_x^2 Q * u$

In the experiments shown in figure 6.7, the module of the SVV term is much stronger than in the experiments in figure 6.6, even in the figures 6.7a and 6.7b. The difference between the dealiased solution and the other one is very small in these experiments. This indicates that the relatively strong SVV term smoothes these ripples fairly well, in contrast with the case of weaker SVV. In any case, the dealiasing procedure does not seem to destabilize or decrease the accuracy of the method. To sum up, applying it is useful when the required module of the SVV term is small, and not so much in the case of strong diffusion (large SVV term or fractional diffusion with $\alpha > 1$). So long as it does not constitute a

huge computational cost, it should be enabled. Alternatively, one can make some heuristic or a posteriori algorithm to determine the need for it based on the degree of diffusion present in the equation and the regularity of the initial data.

6.2 Computation of Fresnel integrals

In the numerical implementation of the method, the integrals deriving from the nonlocal operator pose a computational challenge. For $1 < \alpha < 2$, the generalized Fresnel integral $\int_0^\infty x^{-\alpha} \sin x \, \mathrm{d}x$ has to be evaluated numerically, cf. 4.2. This is challenging, not only because the integrand is singular in the origin, but also because it is an improper integral. The built-in quadrature routines in MATLAB are, as far as the author knows, not suited for such integrals (this was also verified during tests in the project). Therefore, a special quadrature package which is based on the so-called double exponential formula is applied. This is built to calculate the improper integral efficiently. The reader is referred to appendix B for further details about the package. However, it turns out that the singularity in the origin is problematic for the quadrature as $\alpha \to 2$. To remedy this, we have to modify the quadrature slightly, and we therefore developed a routine for calculating the integral analytically near the origin, and combined this with the quadrature package.

6.2.1 Semi-analytical algorithm

The integrand of the Fresnel integral changes rapidly in a boundary layer due to the singularity in the origin. As $\alpha \to 2$, the integrand changes increasingly rapidly. At a point along this sequence of α, the numerical quadrature breaks down and diverges, or at least the number of required evaluations becomes infeasible.

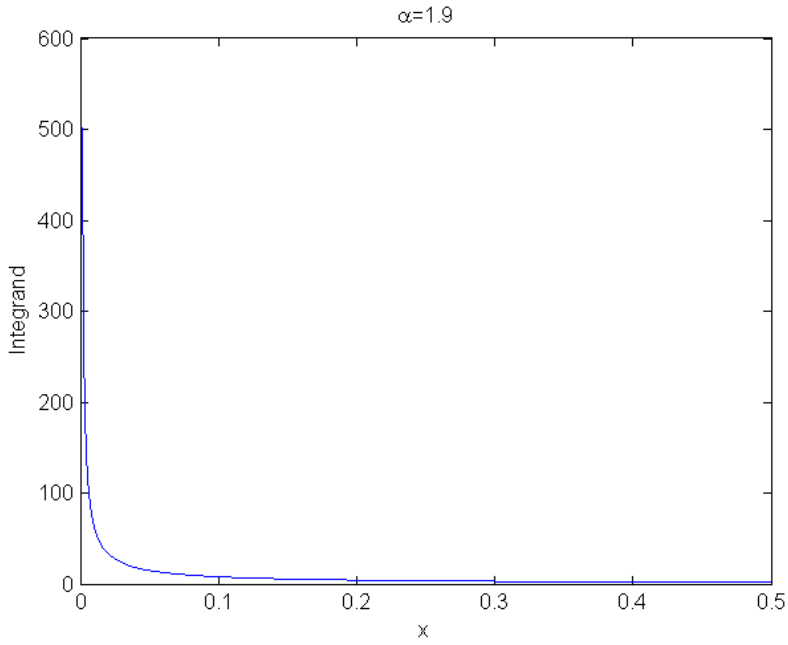


Figure 6.8: Boundary layer for integrand $f(x) = x^{-1.9} \sin x$

To overcome this issue, we propose a semi-analytical algorithm. The idea is to Taylor expand the integrand in the origin and integrate a truncated sum in the boundary layer. The number of terms in the sum is decided based on the error tolerance requirement, and the end point for the boundary layer is chosen heuristically. More concretely, the end point is chosen where the integrand equals some fixed number that is not too large (since the integrand blows off to infinity as $x \to 0$). The residual of the integral is calculated with the numerical quadrature, with required precision greater than the tolerance level. Denote the end point of the boundary layer with b. Then,

$$I = \int_0^\infty x^{-\alpha} \sin x \, \mathrm{d}x = \underbrace{\int_0^b x^{-\alpha} \sin x \, \mathrm{d}x}_{I_1} + \underbrace{\int_b^\infty x^{-\alpha} \sin x \, \mathrm{d}x}_{I_2}$$

The integral I_1 is now treated analytically. Expand the integrand in a truncated

Taylor series with remainder:

$$\begin{aligned} f(x) = x^{-\alpha}\sin x &= x^{-\alpha}\sum_{k=1}^{\infty}\frac{(-1)^{k+1}x^{2k-1}}{(2k-1)!} \\ &= \sum_{k=1}^{\infty}\frac{(-1)^{k+1}x^{2k-1-\alpha}}{(2k-1)!} \\ &= \sum_{k=1}^{K}\frac{(-1)^{k+1}x^{2k-1}}{(2k-1)!} + r_K, \end{aligned}$$

To calculate I_1, integrate the Taylor expansion to obtain

$$\begin{aligned} \int_0^b x^{-\alpha}\sin x\ \mathrm{d}x &= \sum_{k=1}^{K}\int_0^b \frac{(-1)^{k+1}x^{2k-1}}{(2k-1)!}\ \mathrm{d}x + R_K \\ &= \sum_{k=1}^{K}\frac{(-1)^{k+1}}{(2k-1)!}\Big[\frac{x^{2k-\alpha}}{2k-\alpha}\Big]_0^b + R_K \\ &= \sum_{k=1}^{K}\frac{(-1)^{k+1}b^{2k-\alpha}}{(2k-\alpha)(2k-1)!} + R_K \end{aligned}$$

Since this is an alternating series, the remainder term is smaller in magnitude than the next term of the expansion, i.e.

$$|R_K| \le \frac{b^{2K+2-\alpha}}{(2K+2)-\alpha)(2K+1)!}$$

Given an error tolerance ϵ, K can be determined by solving the inequality $\epsilon \le \frac{b^{2K+2-\alpha}}{(2K+2-\alpha)(2K+1)!}$. Example code in MATLAB for implementing the algorithm is given below. In the code, the parameter `alphaLimit` denotes the point where the quadrature breaks down and is set based on experiments. For $\alpha <$ `alphaLimit`, I is calculated with quadrature only, but for $\alpha >$ `alphaLimit`, the semi-analytical approach is used. The function `oscillatorQuadrature(alpha, b, errorTolerance)` is the quadrature routine, which calculates integrals of the type $\int_b^\infty x^{-\alpha}\sin x\ \mathrm{d}x$ numerically upto a precision `errorTolerance`. This routine is written in C, and is invoked in MATLAB using a so-called Mex file. For details about the implementation of this, the reader is referred to appendix B.

```
function I = fresnelIntegralQuadrature(alpha, errorTolerance)
    alphaLimit = 1.85;
    if(alpha <= 0 || alpha >= 2)
        error('0<alpha<2 is required!');
    end
    if(alpha<alphaLimit)
        [I, errorEstimate] = oscillatorQuadrature(alpha,
            errorTolerance, 0);
    else
        integrandUpperLimit = 1;
        searchStartingPoint = 1e-4;
        fsolveFunction = @(x) abs(x.^(-alpha).*sin(x)) -
            integrandUpperLimit;
        a = fsolve(fsolveFunction,searchStartingPoint, optimset('
            Display','off'));
        I1 = getBoundaryLayerIntegral(alpha, a, errorTolerance);
        [I2, errorEstimate] = oscillatorQuadrature(alpha,
            errorTolerance, a);
        I = I1 + I2;
    end
end

function I = getBoundaryLayerIntegral(alpha, b, tol)
    I=0;
    N=1;
    upperNBound=1e6;
    residualBound=b^(2*N+1-alpha)/((factorial(2*(N+1)-1)*(2*(N+1)-
        alpha)));
    while(residualBound>tol && N<upperNBound)
       N=N+1;
       residualBound=b^(2*N+1-alpha)/((factorial(2*(N+1)-1)*(2*(N+1)-
           alpha)));
    end
    for n=1:N
       I=I+((-1)^(n+1)*b^(2*n-alpha))/((factorial(2*n-1)*(2*n-alpha)))
           ;
    end
end
```

Chapter 7

Discussion and further work

7.1 Conclusions

The main goal of the project was to generalize the theory for spectral vanishing viscosity solvers to also include nonlinear fractional diffusion. Given the results and estimates in section 3, this goal has been fulfilled. By enforcing a sufficient set of assumptions enforced on A, we have shown that the numerical solutions converge towards the vanishing viscosity solutions and that the convergence is of exponential order. It remains to show that such a solution is the unique entropy solution of the problem, but that has not been a goal for this project.

The secondary goal was to develop efficient and versatile solvers in practice. We have developed and tested solvers in one and two dimensions, where one of them makes approximate projections, but can take arbitrarily specified convection and diffusion functions and has an indicated computational complexity of $\mathcal{O}(N \log N)$ in one dimension. The other one makes exact projections with an indicated complexity of $\mathcal{O}(N^2)$ in one dimension. The numerical results indicate that the solutions from the solvers converge exponentially when spectral vanishing viscosity is applied and sometimes also when it is omitted. This goal must therefore also be said to be fulfilled, although there is room for improvement both in terms of computational efficiency and convergence rate.

7.2 Suggestions for further work

In light of the above discussion, an obvious extension of the theoretical results is to show that the spectral vanishing viscosity method converges to the unique entropy solution by showing that the vanishing viscosity equation has the entropy solution as a unique solution. As mentioned in Chapter 3, Jakobsen and Cifani give some suggestions to how this can be done in [2] for the linear diffusion case, and the procedure in our case is probably resemblant. They argue that necessary modifications consist of using interpolation inequalities for the nonlocal term, which

we already have looked into in this project (see also their remark 2.6).

We have only considered the fractional Laplacian operator in this project. Many Lévy type operators are resemblant, but have asymmetric measures, like the operator arising from the CGMY model in finance (see also [1] and [16]). A very natural extension of this project would be to modify the convergence theory to also incorporating such measures. In [2] this is done for the linear diffusion case in Chapter 8. Also, the analysis in this article is only performed in one spatial dimension for simplicity. Extending the arguments to arbitrary dimensions should be very feasible (but perhaps technical).

Furthermore, the case $\alpha = 2$ is not treated here, but could definitely be of interest. The solvers we built can handle this, but we have not shown the theoretical results for this case. There exists some theory for porous medium type of equations, see for instance [10], but as far as the author is aware of, not for spectral methods on bounded domains.

Also other types of related problems could be of interest. In this project, periodic boundary conditions are considered, and this is convenient for modeling Cauchy problems by extending to a large domain, or to model system behavior in the middle of some flow field, for instance. However, these methods are not suited for modeling behavior at the boundaries of a system. Physical applications typically require some Dirichlet or Neumann boundary conditions to satisfy flow conditions and the like. In finance, some option types like barrier options also require boundary conditions to be specified. In that regard, spectral methods can also work, but the basis should for instance be some high order polynomial instead of Fourier components. From a modeling perspective, it is also interesting to develop more sophisticated physical or financial models since the mathematical tools become available for a wider class of equations.

With a polynomial spectral method, which essentially is a spectral element method, the theoretical analysis must be modified quite heavily, since for instance no periodicity can be assumed. It also induces some practical considerations that must be addressed, for instance the calculation of nonlocal integrals. It will typically not induce sparse discrete operators either, so computational tuning and various tricks, like tensor product forms would then have to be considered.

As regards other practical issues that can be studied further, we can mention the calculation of projections of initial data. As was mentioned in chapter 5, the built-in quadrature in MATLAB fails to produce approximations to arbitrary precision for high frequency components, so increasing the dimension of the discrete space does not improve the accuracy of the method beyond some limit. Quadrature issues were faced also during the calculation of Fresnel integrals in this project, and it was solved by applying a quadrature that was built especially for this purpose. Improving the accuracy of the initial data projections should also be feasible, but one might have to build or find some special quadrature for this purpose.

We conclude this section by mentioning that the computational complexity of the exact solver probably also can become log-linear by using a fast convolution algorithm, and the implementation of such should not require too much work.

Bibliography

[1] Schoutens, W. (2003): *Lévy Processes in Finance*. Wiley series in probability and statistics.

[2] Jakobsen, E., Cifani, S. (2013): *On the spectral vanishing viscosity method for periodic fractional conservation laws*. Math. Comp. 82, pp. 1489-1514.

[3] Jakobsen, E., Cifani, S., Alibaud, N. (2012): *Continuous dependence estimates for nonlinear fractional convection-diffusion equations*. SIAM J. Math. Anal. 44, pp. 603-632.

[4] Jakobsen, E., Cifani, S. (2012): *Entropy solution theory for fractional degenerate convection-diffusion equations*. Ann. Inst. H. Poincare Anal. Non Lineaire 28, pp. 413-441.

[5] Maday, Y., Tadmor, E. (1989): *Analysis of the Spectral Vanishing Viscosity method for periodic conservation laws*. SIAM Numer. anal. vol. 26, no. 4 pp. 856-870.

[6] Evans, L.C. (2010): *Partial differential equations*. AMS.

[7] Dahlquist, G., Björk, Å. (1974): *Numerical methods*. Dover publications.

[8] Ooura T., Mori M. (1991): *The double exponential formula for oscillatory functions over the half infinite interval, pp. 353-360*. Journal of Computational and Applied Mathematics 38.

[9] Sømme, Ø. (2012): *The fractal Burgers' equation - theory and numerics*. Master's thesis in the course TMA4900 at NTNU.

[10] Vazquez, J.L, Rodriguez A., Quiros F., de Pablo A. (2010): *A fractional porous medium equation*. Preprint.

[11] Vazquez, J.L (2012): *Barrenblatt solutions and asymptotic behaviour for a nonlinear fractional heat equation of porous medium type*. Preprint.

[12] Holden, H., Risebro, N.H. (2007): *Front tracking for Hyperbolic Conservation Laws*. Applied Mathematical Sciences, 152, Springer.

[13] Drouniou, J., Imbert C. (2006): *Fractal first-order partial differential equations*. Arch. Rational Mech. Anal. 182

[14] Boyd, J. (2001): *Chebyshev and Fourier Spectral Methods*. Dover publications.

[15] Rønquist, E. (2012): *Time dependent problems*. Note in the course MA8502 at NTNU.

[16] Davidsen, S-O. (2012): *Fractional diffusion equations in option pricing - numerical solution using spectral methods*. Project thesis in the course TMA4500 at NTNU.

Appendix A

Mathematics

A.1 Technical results

Note that in most of the following definitions and results, the nonlocal operator is written using a measure function m in the representations. This is to keep most of the results as general as possible. In this project, $\mathcal{L} = -(-\Delta)^{\frac{\alpha}{2}}$, and therefore the measure function that applies is $m(y) = \frac{c_\alpha}{|y|^{1+\alpha}}$.

Definition 1 (Nonlocal operator)**.** *The nonlocal operator $\mathcal{L}$ is defined as*

$$\mathcal{L}(u) := \int_{|y|>0} \big(u(x+y) - u(x) - y\mathbf{1}_{|y|<1}u_x(x)\big)m(y)\ dy,$$

i.e. r is chosen as 1 in (2.3).

Definition 2 (Nonlocal bilinear form)**.** *The nonlocal bilinear form B is defined as*

$$B(u,v) := -\int_0^{2\pi}\int_{|y|>0} \big(u(x+y) - u(x) - y\mathbf{1}_{|y|<1}u_x(x)\big)v(x)m(y)\ dydx,$$

Lemma 10 (Representation of diffusion term)**.** *Let $A(u)$ be the diffusion function. Assume that $A(u)$ has a Fourier series representation $A(u) = \sum_{\xi=-\infty}^{\infty}\hat{A}_\xi e^{i\xi x}$. Then,*

$$-(-\Delta)^{\frac{\alpha}{2}}[A(u)] = -C_\alpha \sum_{\xi=-\infty}^{\infty} |\xi|^\alpha \hat{A}_\xi e^{i\xi x},$$

where C_α is some positive constant.

Proof. Observe that $-(-\Delta)^{\frac{\alpha}{2}}$ is a linear operator and apply it to the representation

of $A(u)$. Use (4.2) to obtain that

$$\begin{aligned}\mathcal{L}[A(u)] &= -(-\Delta)^{\frac{\alpha}{2}}\Big(\sum_{\xi=-\infty}^{\infty}\hat{A}_\xi \mathrm{e}^{i\xi x}\Big) = -\sum_{\xi=-\infty}^{\infty}\hat{A}_\xi(-\Delta)^{\frac{\alpha}{2}}(\mathrm{e}^{i\xi x}) \\ &= \sum_{\xi=-\infty}^{\infty}\hat{A}_\xi \mathrm{e}^{i\xi x}\underbrace{\int_0^{2\pi}\mathrm{e}^{i\xi y}-1-y\mathbf{1}_{|y|<1}\,\mathrm{d}y}_{=G(\xi)} = -C_\alpha\sum_{\xi=-\infty}^{\infty}|\xi|^\alpha\hat{A}_\xi\mathrm{e}^{i\xi x}\end{aligned}$$

□

Lemma 11 (Inequality for convex functions)**.** *Let η be a differentiable, convex function and u_x be some differentiable function. Then, $\eta'' \geq 0$ and*

$$\partial_x^2(u_x)\cdot\eta_\rho'(u_x) \leq \partial_x^2(\eta_\rho(u_x)).$$

Proof. Consider

$$\begin{aligned}\big(\eta(u_x)\big)_{xx} &= \big(\eta'(u_x)\cdot u_{xx}\big)_x = \eta'(u_x)u_{xxx}+\eta''(u_x)u_{xx}^2 \\ \Rightarrow \eta'(u_x)u_{xxx} &= \big(\eta(u_x)\big)_{xx} - \underbrace{\eta''(u_x)}_{\geq 0}u_x^2 \\ &\leq \big(\eta(u_x)\big)_{xx}\end{aligned}$$

□

Lemma 12 (Divergence equality)**.** *Let $\eta'(x) := \operatorname{sgn}(x)$, and let f and u be twice differentiable functions. The convention $\operatorname{sgn}(0) = 0$ is used to give meaning to expressions. Then,*

$$\partial_x^2 f(u)\cdot\eta'(u_x) = \partial_x\big(f'(u)\cdot\eta(u_x)\big)$$

Proof. Observe that $s\eta'(s) = \eta(s)$ and apply the chain rule in reverse:

$$\begin{aligned}\partial_{xx}f(u)\cdot\eta'(u_x) &= \big(f''(u)u_x^2 + f'(u)u_{xx}\big)\eta'(u_x) \\ &= \underbrace{f''(u)u_x}_{=\partial_x f'(u)}\cdot\underbrace{u_x\eta'(u_x)}_{=\eta(u_x)} + f'(u)\underbrace{u_{xx}\eta'(u_x)}_{=\partial_x\eta(u_x)} \\ &= \partial_x\big(f'(u)\cdot\eta(u_x)\big)\end{aligned}$$

□

Note that the property $s\eta'(s) = \eta(s)$ is needed, so this is not a trick which is easy to generalize.

Lemma 13 (Symmetrization of integration domain for the nonlocal bilinear form)**.** *Let $u \in C^2[0,2\pi]$ be 2π-periodic. Then,*

$$\begin{aligned} B(u,v) &= -\int_0^{2\pi} \int_{|y|>0} [u(x+y) - u(x) - y\mathbf{1}_{|y|<1} u_x(x)] v(x) m(y)\ dy dx \\ &= -\sum_{k\in\mathbb{Z}} \int_0^{2\pi} \int_0^{2\pi} [u(x+y) - u(x) - (y+2\pi k)\mathbf{1}_{|y|<1} u_x(x)] v(x) m(y)\ dy dx \end{aligned} \tag{A.1}$$

Proof. Consider the nonlocal operator and use that $u(x+y'+2\pi k) = u(x+y')\ \forall k \in \mathbb{Z}$ because of periodicity.

$$\begin{aligned} &\int_{|y|>0} [u(x+y) - u(x) - y\mathbf{1}_{|y|<1} u_x(x)] m(y)\ \mathrm{d}y \\ &= \sum_{k\in\mathbb{Z}} \int_{2\pi k}^{2\pi k+2\pi} [u(x+y) - u(x) - y\mathbf{1}_{|y|<1} u_x(x)] m(y)\ \mathrm{d}y \\ &\overset{y'=y-2\pi k}{=} \sum_{k\in 2\mathbb{Z}} \int_0^{2\pi} [u(x+y'+2\pi k) - u(x) - (y'+2\pi k)\mathbf{1}_{|y|<1} u_x(x)] m(y'+2\pi k)\ \mathrm{d}y' \\ &\overset{y'\to y}{=} \sum_{k\in\mathbb{Z}} \int_0^{2\pi} [u(x+y) - u(x) - (y+2\pi k)\mathbf{1}_{|y|<1} u_x(x)] m(y+2\pi k)\ \mathrm{d}y \end{aligned}$$

Multiply by $v(x)$ and integrate in x to obtain

$$\begin{aligned} &\int_0^{2\pi} \int_{|y|>0} [u(x+y) - u(x) - y\mathbf{1}_{|y|<1} u_x(x)] v(x) m(y)\ \mathrm{d}y\mathrm{d}x \\ &= \sum_{k\in\mathbb{Z}} \int_0^{2\pi} \int_0^{2\pi} [u(x+y) - u(x) - (y+2\pi k)\mathbf{1}_{|y|<1} u_x(x)] m(y+2\pi k) v(x)\ \mathrm{d}y\mathrm{d}x \end{aligned} \tag{A.2}$$

□

Lemma 14 (Taylor expansion of integrals)**.** *For any $u \in C^2(\mathbb{R})$, we have that*

$$u(x+y) - u(x) = \int_0^1 u_x(x+\theta y) y\, d\theta$$

and

$$u(x+y) - u(x) - y u_x(x) = y \int_0^1 u_x(x+\theta y) - u_x(x)\, d\theta = \int_0^1 \int_0^\theta u_{xx}(x+ty) y^2\, dt\, d\theta.$$

Proof. Introduce the new variables $z := x + \theta y$ and $z' := x + ty$. By fixing x and y,

$$\begin{aligned} \mathrm{d}z &= y\mathrm{d}\theta \\ \frac{\mathrm{d}u}{\mathrm{d}x} &= \frac{\mathrm{d}u}{\mathrm{d}z}\frac{\mathrm{d}z}{\mathrm{d}x} = \frac{\mathrm{d}u}{\mathrm{d}z} \\ \Rightarrow \int_0^1 u_x(x+\theta y)y\mathrm{d}\theta &= \int_x^{x+y} u_z \frac{1}{y} y \mathrm{d}z \\ &= \big(u(x+y) - u(x)\big) \end{aligned} \tag{A.3}$$

and

$$\begin{aligned} \frac{\mathrm{d}^2u}{\mathrm{d}x^2} &= \frac{\mathrm{d}^2u}{\mathrm{d}z^2}\Big(\frac{\mathrm{d}^2z}{\mathrm{d}x^2}\Big)^2 = \frac{\mathrm{d}^2u}{\mathrm{d}z^2} \\ \int_0^\theta u_{xx}(x+ty)y^2\mathrm{d}t &= \int_x^{x+\theta y} u_{zz}\frac{1}{y}y^2\mathrm{d}z \\ &= y\big(u_x(x+\theta y) - u_x(x)\big) \end{aligned} \tag{A.4}$$

Hence, using (A.3) and integrating once more,

$$\begin{aligned} \int_0^1\int_0^\theta u_{xx}(x+ty)y^2\mathrm{d}t\mathrm{d}\theta &= y\int_0^1 u_x(x+\theta y) - u_x(x)\mathrm{d}\theta \\ &= \int_0^1 u_x(x+\theta y)y\mathrm{d}\theta - yu_x\int_0^1 \mathrm{d}\theta \\ &= \big(u(x+y) - u(x)\big) - yu_x(x), \end{aligned} \tag{A.5}$$

□

Corollary 1. *The bilinear form B in definition 2 can be written*

$$\begin{aligned} B(u,v) = &-\int_0^{2\pi}\int_{|y|<1}\int_0^1\int_0^\theta u_{xx}(x+ty)y^2\,dt\,d\theta v(x)m(y)\,dy\,dx \\ &-\int_0^{2\pi}\int_{|y|\geq 1}\int_0^1 u_x(x+\theta y)y\,d\theta v(x)m(y)\,dy\,dx \end{aligned} \tag{A.6}$$

Proof. Apply lemma 14 to $\mathcal{L}(u)$, multiply with the test function $v(x)$ and integrate in x from 0 to 2π. □

Lemma 15 (L^2 bound for L^1 norm)**.** *In the compact domain $\Omega = (0, 2\pi)$, it is valid for any function $u \in L^2(\Omega)$ that*

$$\|u\|_{L^1(\Omega)} \leq \sqrt{2\pi}\|u\|_{L^2(\Omega)} \tag{A.7}$$

Proof. Apply the Cauchy-Schwartz inequality to obtain

$$\begin{aligned}\|u\|_{L^1(\Omega)} = \int_0^{2\pi} |u(x)| \cdot 1 \, \mathrm{d}x &\leq \||u(x)|\|_{L^2(\Omega)} \cdot \|1\|_{L^2(\Omega)} \\ &= \Big(\int_0^{2\pi} 1^2 \, \mathrm{d}x \Big)^{\frac{1}{2}} \Big(\int_0^{2\pi} |u(x)|^2 \Big)^{\frac{1}{2}} \\ &= \sqrt{2\pi} \|u\|_{L^2(\Omega)}\end{aligned} \tag{A.8}$$

□

Lemma 16 (Symmetrization of B when $m(y) = m(-y)$)**.** *Let B be as defined in definition 2, and let $u, v \in C^2_{per}$ be 2π-periodic functions. Then*

$$\begin{aligned}B(u,v) &= -\int_0^{2\pi} \int_{|y|>0} u(x+y) - u(x) - y\mathbf{1}_{|y|<1} u_x(x)]v(x)m(y) \; dydx \\ &= \frac{1}{2} \int_0^{2\pi} \int_{|y|>0} [u(x+y) - u(x)][v(x+y) - v(x)]m(y) \; dydx\end{aligned} \tag{A.9}$$

Proof. Introduce the new variable $z = x + y$ and rewrite the form. Apply Fubini's theorem, lemma 13 and interchange variables. These manipulations are justified by periodicity of u, v and the fact that the integrals in the form are well-defined:

$$\begin{aligned}B(u,v) &= -\int_0^{2\pi} \int_{|z|>0} [u(z) - u(x) - (z-x)\mathbf{1}_{|z-x|<1} u_x(x)]v(x)m(z-x) \, \mathrm{d}z\mathrm{d}x \\
&= -\sum_{k\in\mathbb{Z}} \int_0^{2\pi} \int_0^{2\pi} [u(z) - u(x) - (z-x+2\pi k)\mathbf{1}_{|z-x|<1} u_x(x)]v(x)m(z-x+2\pi k) \, \mathrm{d}z\mathrm{d}x \\
&\overset{m(z)=m(-z)}{=} -\sum_{k\in\mathbb{Z}} \int_0^{2\pi} \int_0^{2\pi} [u(z) - u(x) - (z-x+2\pi k)\mathbf{1}_{|z-x|<1} u_x(x)]v(x) \\
&\qquad \cdot m(x-z-2\pi k) \, \mathrm{d}z\mathrm{d}x \\
&\overset{(x,z)\to(z,x)}{=} \sum_{k\in\mathbb{Z}} \int_0^{2\pi} \int_0^{2\pi} [u(x) - u(z) - (x-z-2\pi k)\mathbf{1}_{|x-z|<1} u_x(x)]v(x) \\
&\qquad \cdot m(x-z-2\pi k) \, \mathrm{d}z\mathrm{d}x \\
&= \sum_{k\in\mathbb{Z}} \int_0^{2\pi} \int_0^{2\pi} [u(z) - u(x) - (z-x-2\pi k)\mathbf{1}_{|z-x|<1} u_x(z)]v(z)m(z-x-2\pi k) \, \mathrm{d}x\mathrm{d}z \\
&\overset{\text{Fubini}}{=} \sum_{k\in\mathbb{Z}} \int_0^{2\pi} \int_0^{2\pi} [u(z) - u(x) - (z-x-2\pi k)\mathbf{1}_{|z-x|<1} u_x(z)]v(z)m(z-x-2\pi k) \, \mathrm{d}z\mathrm{d}x \\
&= \int_0^{2\pi} \int_{|z|>0} [u(z) - u(x) - (z-x)\mathbf{1}_{|z-x|<1} u_x(z)]v(z)m(z-x) \, \mathrm{d}z\mathrm{d}x.\end{aligned} \tag{A.10}$$

Since u, v are assumed to be in C^2_{per}, we can use the above computation to see that

$$\begin{aligned}
B(u,v) &= \frac{1}{2}\int_0^{2\pi}\int_{|z|>0}[u(z)-u(x)-(z-x)\mathbf{1}_{|z-x|<1}u_x(z)]v(z)m(z-x)\ \mathrm{d}z\mathrm{d}x \\
&- \frac{1}{2}\int_0^{2\pi}\int_{|z|>0}[u(z)-u(x)-(z-x)\mathbf{1}_{|z-x|<1}u_x(x)]v(x)m(z-x)\ \mathrm{d}z\mathrm{d}x \\
&= \underbrace{\frac{1}{2}\int_0^{2\pi}\int_{|z|>0}[u(z)-u(x)][v(z)-v(x)]m(z-x)\ \mathrm{d}z\mathrm{d}x}_{B_1(u,v)} \\
&- \underbrace{\frac{1}{2}\int_0^{2\pi}\int_{|z|>0}\mathbf{1}_{|z-x|<1}(z-x)[u_x(z)v(z)-u_x(x)v(x)]m(z-x)\ \mathrm{d}z\mathrm{d}x}_{B_2(u,v)}
\end{aligned} \tag{A.11}$$

An alternative is to introduce $\sin y$ instead of y by adding and subtracting terms. This would lead to the same conclusion, but perhaps with simpler arguments since $\sin y$ is 2π-periodic. Also note that the splitting of B into $B_1 + B_2$ is justified by the fact that the integrand is $\mathcal{O}(y^2)m(y)$. This can be verified by using the regularity assumptions made on u, v:

Integrand in B_1 :

$$\begin{aligned}
&[u(x+y)-u(x)][v(x+y)-v(x)]m(y) \\
&= [u(x)+yu_x(x)+\mathcal{O}(y^2)-u(x)] \\
&\qquad \cdot [v(x)+yv_x(x)+\mathcal{O}(y^2)-v(x)]m(y) \\
&= \big(y^2u_x(x)v_x(x)+\mathcal{O}(y^3)\big)m(y) \\
&= \mathcal{O}(y^2)m(y)
\end{aligned}$$

Integrand in B_2 :

$$\begin{aligned}
&y[u_x(x+y)v(x+y)-u_x(x)v(x)]m(y) \\
&= y[\big(u_x(x)+yu_{xx}(x)+\mathcal{O}(y^2)\big) \\
&\qquad \cdot \big(v(x)+yv_x(x)+\mathcal{O}(y^2)\big)-u_x(x)v(x)]m(y) \\
&= y[u_x(x)v(x)+yu_x(x)v_x(x)+yu_{xx}v(x)+\mathcal{O}(y^2)-u_x(x)v(x)]m(y) \\
&= \big(y^2u_x(x)v(x)+y^2u_{xx}v(x)+\mathcal{O}(y^3)\big)m(y) \\
&= \mathcal{O}(y^2)m(y)
\end{aligned}$$

With some intuition and symmetry considerations, one can expect B_2 to be zero. To show this, apply Fubini's theorem to the double integral. Changing the integration order and using the periodicity of the functions and their derivatives enables us to

evaluate the inner integral by translating x by $-y$:

$$
\begin{aligned}
(-2)\cdot B_2(u,v) &= \int_0^{2\pi}\int_{|z|>0} \mathbf{1}_{|z-x|<1}(z-x)[u_x(z)v(z)-u_x(x)v(x)]m(z-x)\ \mathrm{d}z\mathrm{d}x \\
&= \sum_{k\in\mathbb{Z}}\int_0^{2\pi}\int_0^{2\pi} \mathbf{1}_{|z-x|<1}(z-x+2\pi k)[u_x(z)v(z)-u_x(x)v(x)]m(z-x+2\pi k)\ \mathrm{d}z\mathrm{d}x \\
&= \sum_{k\in\mathbb{Z}}\int_0^{2\pi}\int_0^{2\pi} \mathbf{1}_{|y|<1}(y+2\pi k)[u_x(x+y)v(x+y)-u_x(x)v(x)]m(y+2\pi k)\ \mathrm{d}y\mathrm{d}x \\
&= \sum_{k\in\mathbb{Z}}\int_0^{2\pi}\Big(\int_0^{2\pi}[u_x(x+y)v(x+y)-u_x(x)v(x)]\ \mathrm{d}x\Big)\mathbf{1}_{|y|<1}(y+2\pi k)m(y+2\pi k)\mathrm{d}y \\
&= \sum_{k\in\mathbb{Z}}\int_0^{2\pi}\Big(\int_0^{2\pi}u_x(x+y)v(x+y)\ \mathrm{d}x-\int_0^{2\pi}u_x(x)v(x)\ \mathrm{d}x\Big)\mathbf{1}_{|y|<1} \\
&\qquad\qquad \cdot(y+2\pi k)m(y+2\pi k)\mathrm{d}y \\
&= \sum_{k\in\mathbb{Z}}\int_0^{2\pi}\Big(\int_y^{2\pi+y}u_x(x)v(x)\ \mathrm{d}x-\int_0^{2\pi}u_x(x)v(x)\ \mathrm{d}x\Big)\mathbf{1}_{|y|<1}(y+2\pi k)m(y+2\pi k)\mathrm{d}y \\
&= \sum_{k\in\mathbb{Z}}\int_0^{2\pi}\Big(\int_y^{2\pi}u_x(x)v(x)\ \mathrm{d}x+\int_{2\pi}^{2\pi+y}u_x(x)v(x)\ \mathrm{d}x \\
&\quad -\int_0^{2\pi}u_x(x)v(x)\ \mathrm{d}x\Big)\mathbf{1}_{|y|<1}(y+2\pi k)m(y+2\pi k)\mathrm{d}y \\
&= \sum_{k\in\mathbb{Z}}\int_0^{2\pi}\Big(\int_y^{2\pi}u_x(x)v(x)\ \mathrm{d}x+\int_{2\pi-2\pi}^{2\pi+y-2\pi}u_x(x)v(x)\ \mathrm{d}x \\
&\quad -\int_0^{2\pi}u_x(x)v(x)\ \mathrm{d}x\Big)\mathbf{1}_{|y|<1}(y+2\pi k)m(y+2\pi k)\mathrm{d}y \\
&= \sum_{k\in\mathbb{Z}}\int_0^{2\pi}\Big(\int_0^{y}u_x(x)v(x)\ \mathrm{d}x+\int_y^{2\pi}u_x(x)v(x)\ \mathrm{d}x \\
&\quad -\int_0^{2\pi}u_x(x)v(x)\ \mathrm{d}x\Big)\mathbf{1}_{|y|<1}(y+2\pi k)m(y+2\pi k)\mathrm{d}y \\
&= \sum_{k\in\mathbb{Z}}\int_0^{2\pi}\Big(\int_0^{2\pi}u_x(x)v(x)\ \mathrm{d}x-\int_0^{2\pi}u_x(x)v(x)\ \mathrm{d}x\Big)(y+2\pi k)m(y+2\pi k)\mathrm{d}y \\
&= \sum_{k\in\mathbb{Z}}\int_0^{2\pi}0\cdot\mathbf{1}_{|y|<1}(y+2\pi k)m(y+2\pi k)\mathrm{d}y \\
&= 0
\end{aligned}
\tag{A.12}
$$

□

Appendix B

Software and implementation

B.1 Numerical quadrature

The numerical quadrature which is used in this project to calculate the Fresnel integrals Θ_α in (4.2), is made especially for this purpose. The software package is written by assistant professor Takuya Ooura at Research Institute for Mathematical Sciences at the Kyoto University. It is based on the so-called double exponential formula for oscillatory functions over the half infinite interval. The interested reader is referred to the original article on the subject, cf. [8]. See also the homepage of Ooura for further information and to download the original source code: http://www.kurims.kyoto-u.ac.jp/~ooura/index.html.

The software package consists of three methods; `intde()`, `intdei()` and `intdeo()`, to calculate integrals over finite intervals, half open intervals for non-oscillatory integrands and half open intervals for oscillatory integrands, respectively. In this project, only the latter method is of interest. The methods are given as both C and FORTRAN implementations.

In this project, the C implementation is used, since it is possible to compile, link and invoke C code from MATLAB. This feature makes it easier to prototype software also in C. The procedure to invoke C code in MATLAB requires a bridge written in C and some extra steps. Since this could be of interest for others wanting to implement the numerical methods in this project, we give an outline of how to do this and an excerpt from the code used in this project.

1. Write a C file called for instance "oscillatoryQuadrature.c" where the header file "mex.h" is referred with an include statement. In this file, there must be a void function called `mexFunction()`, which works as the gateway function, which takes the parameters `int nlhs, mxArray *plhs[], int nrhs, const mxArray *prhs[]`. Also, the code that is to be invoked must be written in this file. Alternatively, it can be referred to (but then it is necessary to compile and link the code separately).

2. In `mexFunction()`, write C code to perform computations and invoke other

code as necessary. The pointers to the output parameters resulting from the computations, that are supposed to be passed back to MATLAB, must be put in the `plhs` array. This can be done with the built-in mex function `mxGetPr()` (see the example code below for details).

3. In the MATLAB command line or window, type "mex coscillatoryQuadrature.c" in the folder where the oscillatoryQuadrature.c file lies. Then the C code will be compiled, and a .mex~ file (for instance .mexw64 in Windows 64) will be generated in the same folder.

4. To invoke the code in MATLAB, call the function `oscillatoryQuadrature()` (must be the same name as the .c file) directly as a MATLAB command with the appropriate set of input and output variables. In the example below, if the .c file is called "oscillatoryQuadrature.c", the appropriate call would be `[integral, errorEstimate] = oscillatoryQuadrature(alpha, errorLimit, b)`.

```c
//Example file called "oscillatoryQuadrature.c"
#include <math.h>
#include "mex.h"

double f(double x, double alpha);
void intdeo(double b, double alpha, double omega, double eps, double *
    i, double *err);

//The quadrature function.
void intdeo(double b, double alpha, double omega, double eps, double *
    i, double *err)
{
  //See the home page of the author given above for the source code
}

//The integrand
double f(double x, double alpha)
{
    return pow(x,-alpha)*sin(x);
}


/*The mex gateway function.
nrhs indicates the number of input variables
nlhs indicates the number of output variables
prhs corresponds to the input variables from Matlab
plhs corresponds to the input variables that are passed to the C
    function */

void mexFunction( int nlhs, mxArray *plhs[], int nrhs, const mxArray *
    prhs[])
{
    double alpha, b, epsilon;    // input parameters
    double *integral;            // output parameters
    double *errorEstimate;

```

```
33      double omega; // auxiliary parameter
34
35      /* check for proper number of arguments */
36      if(nrhs!=3) {
37          mexErrMsgIdAndTxt("MyToolbox:arrayProduct:nrhs","3 inputs
                required.");
38      }
39      if(nlhs!=2) {
40          mexErrMsgIdAndTxt("MyToolbox:arrayProduct:nlhs","2 outputs
                required.");
41      }
42      /* make sure the first input argument is real scalar double */
43      if( !mxIsDouble(prhs[0]) ||
44           mxIsComplex(prhs[0]) ||
45           mxGetNumberOfElements(prhs[0])!=1 ) {
46          mexErrMsgIdAndTxt("MyToolbox:arrayProduct:notScalar","Input
                multiplier must be a scalar.");
47      }
48
49      /* make sure the second input argument is real scalar double */
50      if( !mxIsDouble(prhs[1]) ||
51           mxIsComplex(prhs[1]) ||
52           mxGetNumberOfElements(prhs[1])!=1 ) {
53          mexErrMsgIdAndTxt("MyToolbox:arrayProduct:notScalar","Input
                multiplier must be a scalar.");
54      }
55
56      /* make sure the third input argument is real scalar double */
57      if( !mxIsDouble(prhs[2]) ||
58           mxIsComplex(prhs[2]) ||
59           mxGetNumberOfElements(prhs[2])!=1 ) {
60          mexErrMsgIdAndTxt("MyToolbox:arrayProduct:notScalar","Input
                multiplier must be a scalar.");
61      }
62
63      /* get the value of the scalar input  */
64      alpha = mxGetScalar(prhs[0]);    //The fractionality parameter
65      epsilon = mxGetScalar(prhs[1]); //The error tolerance
66      b=mxGetScalar(prhs[2]);          //Lower integration limit
67      omega=1;                         //Frequency of sine factor
68
69      /* create the output matrices: scalar, 1x1 */
70      plhs[0] = mxCreateDoubleMatrix(1,1,mxREAL);
71      plhs[1] = mxCreateDoubleMatrix(1,1,mxREAL);
72
73      /* get a pointer to the real data in the output matrix */
74      integral = mxGetPr(plhs[0]);
75      errorEstimate = mxGetPr(plhs[1]);
76
77      /* call the computational routine */
78      intdeo(b, alpha, omega, epsilon, integral, errorEstimate);
79  }
```

B.2 Software library documentation

Below follows a brief documentation of the generic FFT solver for 1D that was developed in MATLAB as part of the project. The exact solver is almost identical in the structure of the code except that there are a few extra parameters, so documentation of this is omitted. The call structure is not explained in detail since it is built as a modular solver, and hence the great number of subroutines. The only thing that needs to be run is the script `simulation`, in which some parameters must be set. The functions f, A (and a possible source function g) can be specified arbitrarily in their respective subroutines.

- `simulation`:

  The very script that runs the simulations. Parameters such as the discrete dimension, the simulation time and quadrature error tolerance can be specified in it. The computed solution is plotted as an animation.

- `solver(N, T, alpha, eps, dealiasing, SVV)`:

  Takes parameters for the discrete dimension, simulation end point, `alpha`$= \alpha$, quadrature tolerance `eps`, and the booleans `dealiasing` and `SVV` that indicate whether or not dealiasing and SVV are to be applied. Returns the solution computed at all timesteps between 0 and `T` with a default spacing of $dt = 0.01$. The solution is returned as a matrix with columns ordered from left to right corresponding to the solution vectors.

- `initialCondition(N)`:

  Returns the initial condition for an `N` dimensional discrete space as a column vector. It is computed using standard MATLAB quadrature and may therefore suffer from instability for very high frequencies (which will induce console warnings).

- `getSpatial_odefunction(alpha, N, eps, dealiasing, SVV)`:

  Takes parameters for the discrete dimension, simulation end point, `alpha` $= \alpha$, quadrature tolerance `eps`, and the booleans `dealiasing` and `SVV` that indicate whether or not dealiasing and SVV are to be applied. Returns the function that corresponds to the time derivative u_t in the semidiscretization (4.4). The function is returned as a function handle that takes the solution and time as parameters like `spatial_odefunction`(y, t) (which is needed when using for instance ODE45) and returns the time derivative.

- `timeIntegrator(c0, t0, dt, spatial_odefunction)`:

  Takes parameters for an initial condition `c0`, the starting time `t0`, the time leap `dt` and a function handle `spatial_odefunction` which should return the time derivative of the solution and be callable as `spatial_odefunction`(y, t). It returns the solution at time `t0` + `dt` as a column vector. It computes the solution/integrates in time using `ode45_solver`(), but it is possible to use

another time integrator by writing a function as `ode45_solver` but with another solver (like ODE15s).

- `ode45_solver(c0, t0, dt, spatial_odefunction)`:

  Takes parameters for an initial condition `c0`, the starting time `t0`, the time leap `dt` and a function handle `spatial_odefunction` which should return the time derivative of the solution and be callable as `spatial_odefunction`(y, t). It returns the computed solution, using ODE45, as a column vector.

- `getSourceTermIntegrals(N)`:

  Computes the source integrals corresponding to the term $S(\xi)$ in 4.4 for an `N` dimensional discrete space, by using standard MATLAB quadrature. Returns the integrals as a column vector.

- `getNonlocalIntegrals(N, alpha, eps)`:

  Computes the integrals from the nonlocal operator corresponding to the factor $G(\xi)$ in 4.4 for an `N` dimensional discrete space, by using a semi-analytical algorithm. Returns the integrals as a column vector.

- `getNonlocalOperatorCoefficients(c, dealiasing)`:

  Computes the approximate projection of the nonlocal operator corresponding to the factor $\tilde{A}_\xi$ in 4.4 for the coefficient vector `c` by using FFT. Returns the projections in a column vector. If the boolean `dealiasing` is set to true, dealiasing will be applied in the calculations, otherwise not.

- `getViscosityCoefficients(N)`:

  Returns the viscosity kernel coefficients $\hat{Q}_\xi$ in (4.4) corresponding to the discrete dimension `N`. The scheme in 4.3 is used.

- `getViscositySpectrumLimit(N)`:

  Returns the viscosity spectrum limit m_N in (4.3) corresponding to the discrete dimension `N`. The scheme in assumption 1 is used.

- `getViscosityModule(N, SVV)`:

  Returns the viscosity module ϵ_N in 4.4 corresponding to the discrete dimension `N`. The boolean SVV indicates whether SVV is to be used or not. If SVV is set to true, the module corresponds to the asymptotic scheme given in assumption 1, otherwise it is set to zero

- `getConvectionOperatorCoefficients(c, dealiasing)`:

  Computes the approximate projection of the convection operator corresponding to the factor $\tilde{f}_\xi$ in 4.4 for the coefficient vector `c` by using FFT. Returns the projections in a column vector. If the boolean `dealiasing` is set to true, dealiasing will be applied in the calculations, otherwise not.

- `plotSimulationResults(C)`:

  Assumes that `C` is a matrix consisting of coefficient column vectors and plots the evolution of the function values corresponding to the coefficients in it in an animation

- `plotFunctionFromCoefficients(c)`:

  Assumes that `c` is a coefficient vector and plots the function values corresponding to the coefficients in it

- `setAxes()`:

  Sets the axes in simulation plots to a view which is suited for the default initial condition

- `getCoefficientsFromFunctionValues(u)`:

  Calculates the coefficients corresponding to the function values `u` using the discrete Fourier transform and returns the corresponding coefficients in a column vector

- `getFunctionValuesFromCoefficients(c)`:

  Calculates the function values corresponding to the coefficients `c` using the inverse discrete Fourier transform and returns the corresponding function values in a column vector

- `fresnelIntegralQuadrature(alpha, eps)`:

  Calculates the generalized Fresnel integral corresponding to Θ_α in 4.2 with a value of `alpha` for α and an error tolerance of `eps` using quadrature. Depending on how close `alpha` is to two, the integral is calculated using pure quadrature, calling `oscillatorQuadrature`, or divided in two and calculated using the functions `oscillatorQuadrature` and `getBoundaryLayerIntegral` `oscillatorQuadrature` and `getBoundaryLayerIntegral`

- `getBoundaryLayerIntegral(alpha, a, eps)`:

  Evaluates the innermost part, from 0 to `a`, of the generalized Fresnel integral corresponding to Θ_α in 4.2 with a value of `alpha` for α and an error tolerance of `eps`. The integral is calculated analytically using an adaptively truncated Taylor series

- `oscillatorQuadrature(alpha, eps, a)`:

  Calculates the outermost part, from `a` to ∞, of the generalized Fresnel integral corresponding to Θ_α in 4.2 with a value of `alpha` for α and an error tolerance of `eps`. The integral is calculated using the numerical quadrature routine described in section 6.2

- `evaluateSinIntegral(alpha, eps)`:

  Evaluates the generalized Fresnel integral corresponding to Θ_α in 4.2 with a value of `alpha` for α and a quadrature error tolerance of `eps`. Depending on the value of `alpha`, the integral is calculated analytically or semi-analytically using the function `fresnelIntegralQuadrature`

- `A(u)`:

  Calculates the diffusion function from the function values `u` and returns a column vector with the corresponding diffusion function values

- `f(u)`:

  Calculates the convection function from the function values `u` and returns a column vector with the corresponding diffusion function values

- `g(x)`:

  Calculates the value of the source term at the vector `x` and returns a column vector with the corresponding source function values