NTNU

Norwegian University of
Science and Technology

Discontinous Galerkin Methods for Conservation Laws,

with and without fractional diffusion

Alexander N. Sigurdsson

Applied and Engineering Mathematics
Submission date: July 2018
Supervisor: Espen Robstad Jakobsen, IMF

Norwegian University of Science and Technology
Department of Mathematical Sciences

Preface

This thesis is the culmination of my period as a student at the Master of Science program in Industrial Mathematics at The Norwegian University of Science and Technology (NTNU). The work here reflects my current interests in mathematics, differential equations and numerical analysis. Not showcased in this thesis is my love for dynamical systems, bifurcations and chaos theory. I would like to thank my advisor, Professor Espen Robstad Jakobsen, for our weekly meetings and his inspiration and guidance in completing my thesis. I would also like to thank the Department of Mathematical Sciences, especially the administration, for their hospitality and the way they accommodated me for the final stretch. Lastly, I want to thank my parents, Grethe Sigurdsson and Eric Nævdal, for their unwavering support. Trondheim,

July 5. 2018
Alexander N Sigurdsson

Summary

This thesis was submitted on July 5'th 2018 as the Master's thesis for Alexander N Sigurdsson in Industrial Mathematics at the Department of Mathematical Sciences at The Norwegian University of Science and Technology (NTNU). In this thesis we apply the Discontinuous Galerkin (DG) methods on scalar conservation laws with and without fractional diffusion. The supporting theory of Discontinuous Galerkin methods is taken from [2] and [6] while the discretizing of the fractional Laplacian is shown in [1]. More general theory on numerical solutions of PDE's is found from [9] and [7]. The first part of the thesis gives an introduction to DG methods for scalar conservation laws. We then employ the methods to solve specific conservation laws with both linear and non-linear flux. Here we also introduce an explicit numerical time integration scheme, a Runge-Kutta method with TVD (Total Variation Diminishing) properties (RKTVD method) and show how to use slope-limiting techniques to avoid spurious oscillations that occur for higher-order methods. Numerical examples and results are shown for both linear and non-linear flux up to 2'nd order. The second part tackles the addition of a fractional diffusion operator specifically the fractional Laplacian. By using a result where the fractional Laplacian can be written as a singular integral we are able to discretize it and find solutions of the fractional conservation law. Numerical examples with fractional diffusion are shown for both linear and non-linear flux up to 1'st order. The appendix gives a more thorough walk-through of how the integrals are analytically calculated for the fractional Laplacian, and how one can calculate these integrals numerically.

Sammendrag

Denne oppgaven ble innlevert den 5. juli 2018 som avsluttende masteroppgave for Alexander N Sigurdsson i Industriell Matematikk ved Institutt for Matematiske Fag ved NTNU. I denne oppgaven anvendes Diskontinuerlig Galerkin (DG) metode på den skalare bevaringsloven med og uten fraksjonell diffusjon. Teorien back metoden er tatt fra [2] og [6], mens diskretiseringen av den fraksjonelle Laplace operatoren er vist i [1]. Mer generell teori når det kommer til løsningen av partielle differensialligninger er hentet fra [9] og [7]. Den første delen av oppgaven gir en introduksjon til DG metoder for skalare bevaringslover. Vi bruker så metoden metoden til å løse spesifikke bevaringslover med både lineære og ikke-lineær flux. Her introduserer vi også et spesielt, eksplisitt skjema for tidsintegrering, en Runge-Kutta metode med TVD (Total Variation Diminishing) egenskaper (RKTVD metode) og viser hvordan vi kan broke slope-limiting teknikker for å unngå falske svingninger som oppstår for høyere-ordens metoder. Numeriske eksempler og resultater er vist både for lineær og ikke-lineær flux for opp til kvadratisk basis. Den andre delen av oppgaven tar for seg hva som skjer når man inkluderer en fraksjonell diffusjos-operator i bevaringslovene, spesifikt den fraksjonelle Laplace operatoren. Ved å benytte oss av et resultat der den fraksjonelle Laplace operatoren kan representeres ved et singulært integral kan vi forsøke å diskretise den for å finne løsninger til den fraksjonelle bevaringsloven. Numeriske eksempler med fraksjonell diffusjon er vist for både lineær og ikke-lineær fluks opp til først orden. Appendikset gir en mer grundig gjennomgang av hvordan man regner ut integralene i diskretiseringen analytisk, og hvordan man regner ut integralene numerisk.

DISCONTINOUS GALERKIN METHODS FOR CONSERVATION LAWS, WITH AND WITHOUT FRACTIONAL DIFFUSION

ALEXANDER N SIGURDSSON

Abstract. We seek to apply the Discontinous Galerkin (DG) method to the scalar conservation law with and without fractional diffusion. Using the framework in [2] and [1], we show how total variation diminishing methods (TVD-methods) permits accurate higher-order solutions with the help of slope limiters, and how to discretize the fractional Laplacian, both analytically and numerically. Numerical examples include solving for the non-linear Burgers' equation.

Date: July 5, 2018.

Contents

1. Introduction

1.1. **The Discontinous Galerkin method.** The Discontinous Galerkin (DG) method enjoys certain properties that are familiar in the multitude of methods to discretize partial differential equations. The method divides its domain into elements, similarly to FEM, allowing for complex geometries. The difference is that instead of using piecewise continuous basis functions the basis functions used to reconstruct the solution are defined locally in each element, and they do not have to be continuous across element boundaries. This locality, which is something you can find in FDM, makes the method excel at capturing high gradient solutions/shock solutions.

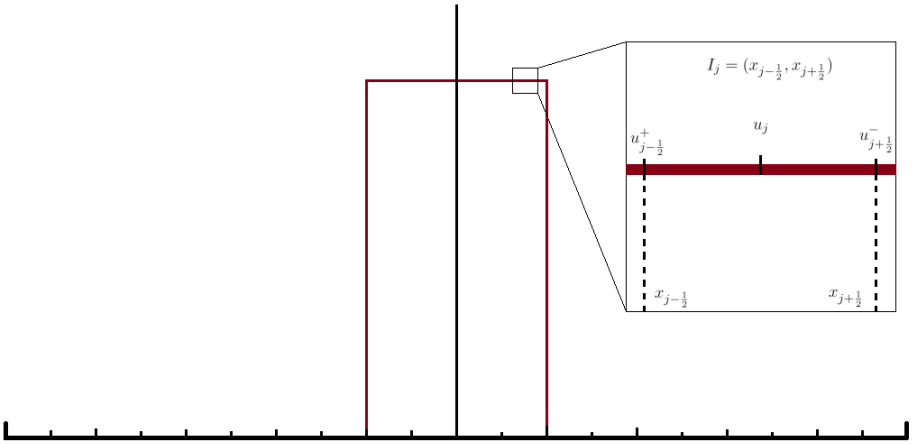

Figure 1. Initial block function and an example element I_j

Let us consider the scalar conservation law, excluding source terms

$$\begin{aligned} &\frac{\partial u}{\partial t} + \frac{\partial f(u)}{\partial x} = 0 \quad \text{in } \Omega \times (0, T), \\ &u(x, 0) = u_0(x) \end{aligned} \tag{1}$$

Here we have the differential equation posed on a physical domain $\Omega = [L, R]$, where L and R is the left and right boundary. In the same way as for conform (continuous) Galerkin methods we will provide a weak formulation, but here the weak formulation will given over each element $I_i = (x_{i-\frac{1}{2}}, x_{i+\frac{1}{2}})$, where I_i is the partition of Ω. We will use the notation $\frac{\partial}{\partial t} := \partial_t$. Figure 1 shows the block function,

$$u_0(x) = \begin{cases} 1 & \text{if } a < x < b \\ 0 & \text{otherwise} \end{cases} \tag{2}$$

and a close up of an example element, I_j. Here, one can see how u and x are defined on the interval. The numerical examples in this thesis uses the block function as initial data for the most part, except for when calculating convergence. Considering the block function it makes sense to use either Dirichlet boundary conditions, where the solution is 0 on the boundary, or periodic boundary conditions. However, we can add a non-local source term to our conservation law to obtain the fractional conservation law,

$$\begin{aligned} \partial_t u(x,t) + \partial_x f(u(x,t)) &= -\epsilon\,(-\partial_x^2)^{\frac{\alpha}{2}} u(x,t) \\ &\qquad \text{in } \Omega \times (0,T),\ \alpha \in (0,2),\ \epsilon > 0 \\ u(x,0) &= u_0(x). \end{aligned} \tag{3}$$

Here, the right side of the PDE is the fractional Laplacian, a non-local source of diffusion, meaning that the diffusion happens across the entire domain. The details and definition of this operator will be explained in section 6, for now it is enough to know that because of this diffusion using Dirichlet boundary conditions set to 0 will cause mass loss at the edges. Therefore, we have chosen periodic boundary conditions for our numerical results in sections 3.4, 6.2 and appendix C.

2. Background

2.1. **The Weak Formulation.** We seek to find an approximation, $u_h|_{I_i}$, that satisfies the differential equation in weak form within each element I_i. We are looking for approximations belonging to the space

$$\mathcal{V}_h = \{v \in L^1(\Omega) : \quad v|_{I_i} \in P^l(I_i),\ l = 0, ..., k\},$$

where P^l is the polynomial space up to a degree of k, a *basis*. Hence, we let our approximation be a linear combination of these polynomials:

$$u_h(x,t)|_{I_i} = \sum_{l=0}^{k} u_i^l(t) v_i^l(x) \tag{4}$$

and we take the direct sum of the numerical solution on each element to obtain the complete numerical solution,

$$u_h(x,t) = \bigoplus_{I_i} u_h(x,t)|_{I_i} = \bigoplus_{I_i} \sum_{l=0}^{k} u_i^l(t) v_i^l(x). \tag{5}$$

Here $\bigoplus_{I_i}$ just means that the complete solution is a sum of the numerical solutions on each element I_i. The reason for this notation, as opposed to simply $\sum_{I_i}$, is that the basis polynomials are not defined outside their respective elements. We can now multiply our conservation law by the test function v and integrate over the interval I_i to find

the weak form. Applying integration by parts to the integral containing the flux term yields the following,

$$\int_{I_i} \partial_t u(x,t)v(x)\,dx + \int_{I_i} \partial_x f(u(x,t))v(x)\,dx = 0$$

$$\int_{I_i} \partial_t u(x,t)v(x)\,dx + \Big[f(u(x,t))v(x)\Big]_{I_i} - \int_{I_i} f(u(x,t))\partial_x v(x)\,dx = 0.$$

Our conservation law can now be broken down into four parts, they are as follows,

$$\begin{aligned}
&\int_{I_i} \partial_t u(x,t)v(x)\,dx \quad \text{(mass matrix)} \\
&= \int_{I_i} f(u(x,t))\partial_x v(x)\,dx \quad \text{(stiffness matrix)} \\
&- \Big(f(u(x_{i+\frac{1}{2}},t))v(x^-_{i+\frac{1}{2}}) - f(u(x_{i-\frac{1}{2}},t))v(x^+_{i-\frac{1}{2}})\Big) \quad \text{(flux)} \\
&\int_{I_i} u(x,0)v(x)\,dx = \int_{I_i} u_0(x)v(x)\,dx \quad \text{(initial condition)}.
\end{aligned} \tag{6}$$

2.2. **Picking a basis.** When it comes to picking a basis, we will look at the following options and their advantages and limitations: Firstly, the monomials, the simplest polynomial basis, we will use the reference element $\mathsf{I} = (-1, 1)$,

$$\psi_n(x) := x^n$$

secondly, the Legendre polynomials

$$P_n(x) := 2^n \sum_{k=0}^{n} x^k \binom{n}{k} \binom{\frac{n+k-1}{2}}{n}$$

and lastly, Lagrange polynomials,

$$L_n(x) := \sum_{j=0}^{n} y_j \ell_j(x), \text{ where}$$

$$\ell_j(x) := \prod_{\substack{0 \le m \le k \\ m \ne j}} \frac{x - x_m}{x_j - x_m}.$$

Let us first calculate the mass matrix with a monomial basis of N'th order

$$\begin{aligned}
\int_{\mathsf{I}} \partial_t u(x,t)v(x)\,dx &= \int_{\mathsf{I}} \partial_t \sum_{i=0}^{N} u^i(t)x^i x^j\,dx \quad \text{for } j \in [0,...,N] \\
&= \sum_{i=0}^{N} \partial_t u^i \int_{\mathsf{I}} x^i x^j\,dx \quad \text{for } j \in [0,...,N] \\
&= \mathbf{M} \cdot \partial_t \mathbf{u}
\end{aligned}$$

$$M_{i,j} = \int_{\mathsf{I}} x^i x^j \, dx$$

When solving the discretized system one will have to calculate M^{-1} in order to find $\partial_t \mathbf{u}$. Using 2'nd order as an example we can find the condition number of this matrix, $\kappa(M^{3\times3}) = \frac{\lambda_{\max}(M^{3\times3})}{\lambda_{\min}(M^{3\times3})} \approx \frac{2.24}{0.16} \approx 14$. This is an ill-conditioned matrix, and one can show that the condition number increases exponentially for higher orders, see [6]. This makes the monomials an unsuitable basis for our discretization. Similarly, calculate the mass matrix using Legendre polynomials,

$$M_{i,j} = \int_{\mathsf{I}} P_i P_j \, dx = \frac{2}{2n+1}\delta_{ij}$$

Here we use the fact that Legendre polynomials are orthogonal, and δ_{ij} is the Kronecker delta. This gives a diagonal mass matrix which is trivial to invert. We can see that the orthogonal property makes the Legendre polynomials a very suitable basis. Having the solution on each element be approximated by a sum of polynomials multiplied by some coefficient is known as a *Modal* method. Compare this to regular Galerkin method where, for example using a piecewise linear basis e_i, we have the numerical approximation $u_h = \sum_i^N u_i \, e_i$. For a general problem this method attempts to solve $a(u_h, e_j) = f(e_j)$, which gives $\sum_i^N u_i a(e_j, e_i) = f(e_j)$. This can be rewritten as a linear system of equations $\boldsymbol{A}\boldsymbol{u} = \boldsymbol{f}$, where $A_{ij} = a(e_j, e_i)$. $\boldsymbol{A}$ is an $N \times N$ matrix that gives the coefficients for the entire system, while M, for a linear basis, is a 2×2 matrix giving the coefficients for just a single element. One advantage for DG methods is that one avoids the computationally taxing task of inverting large matrices, while the downside is an increase in the degrees of freedom. For linear elements there is a doubling as inverting $\boldsymbol{A}$ gives an ODE-system of size N, while inverting M gives 2 ode's per element, $2N$ in total. For a more thorough approach to the Galerkin finite element method, and a more complete look at the fauna of methods used to solve partial differential equations, see [10].

Lagrange polynomials are a little trickier,

$$M_{i,j} = \int_{\mathsf{I}} L_i L_j \, dx.$$

We need to calculate an integral for each element in M, and as we will see later the same applies to the stiffness matrix K which is the matrix form of the second line in (6). One way of doing this numerically is using quadrature rules,

$$\int_{-1}^{1} f(x) \, dx \approx \sum_{k=1}^{n} w_k f(x_k).$$

A smart way of setting up the discretization is to use the same points that define the Lagrange polynomial to calculate the quadrature, this

is referred to as a *Nodal* method. Let us use the points from the Gauss-Lobatto quadrature which uses the values at the end point of the interval,

$$\int_{\mathsf{I}} L_i L_j \, dx \approx \sum_{k=0}^{n} w_k L_i(x_k) L_j(x_k).$$

As an example, for a 2'nd order method we need 3 points to define our Lagrange polynomial. For the Lobatto quadrature we have the points $\mathbf{x} = (-1, 0, 1)$ and weights $\mathbf{w} = (\frac{1}{3}, \frac{4}{3}, \frac{1}{3})$. The values of $L_i(x_k)$ are easy to find as they are the value of our function u_h at those points, by definition. The results and analysis in this thesis utilizes a modal method with Legendre polynomials.

3. Burgers' equation

Let us consider Burgers' equation, named after the dutch physicist Jan Burgers, a non-linear conservation law,

$$\begin{aligned} &\partial_t u + \partial_x(\frac{1}{2}u^2) = 0 \quad \text{in } \Omega \times (0, T), \\ &u(x, 0) = u_0(x) \end{aligned} \tag{7}$$

We can see it is a conservation law of the form (1) with a non-linear flux, $f(u) = \frac{1}{2}u^2$.

3.1. Discontinous Galerkin Method with Legendre basis.

Now, we choose Legendre polynomials as our basis, and we pick our test functions v from the same basis. The first few Legendre polynomials are below.

l	$P_l(x)$
0	1
1	x
2	$\frac{1}{2}(3x^2 - 1)$

We want the polynomials transformed from the reference element $[-1, 1]$ to our elements, I_i. We denote this transformed polynomial by ϕ_i^l, $\Delta x = (x_{i+\frac{1}{2}} - x_{i-\frac{1}{2}})$.

$$\phi_i^l = P_l\Big(\frac{2(x - x_i)}{\Delta x}\Big)$$

For the transformed polynomials we have the orthogonal property:

$$\int_{I_i} \phi_i^l \phi_i^{l'} = \frac{\Delta x}{2l' + 1} \delta_{ll'} \tag{8}$$

From (4) we get that the our approximation with Legendre polynomials becomes:

$$u_h(x, t)|_{I_i} = \sum_{l=0}^{k} u_i^l(t) \phi_i^l(x)$$

Now looking at the Burgers' equation and with our approximate solution $u_h|_{I_i}$, we can revisit our weak formulation. From the orthogonality property (8), and the mass matrix (6) we have

$$\forall i = 1, ..., N \text{ and } l' = 0, ..., k$$

$$\int_{I_i} (\partial_t \sum_{l=0}^{k} u_i^l \phi_i^l) \phi_i^{l'} \, dx$$

$$\Rightarrow \partial_t \sum_{l=0}^{k} u_i^l \int_{I_i} \phi_i^l \phi_i^{l'} \, dx$$

$$\Rightarrow \frac{\Delta x}{2l'+1} \partial_t u_i^{l'}$$

Note that l refers to the order of the polynomials in the solution $u_h|_{I_i}$, while l' is the order for the test functions. From the stiffness matrix part of (6) we get

$$\forall i = 1, ..., N \text{ and } l' = 0, ..., k$$

$$\int_{I_i} f(\sum_{l=0}^{k} u_i^l \phi_i^l) \partial_x \phi_i^{l'} \, dx$$

$$\Rightarrow \int_{I_i} \frac{1}{2} (\sum_{l=0}^{k} u_i^l \phi_i^l)^2 \partial_x \phi_i^{l'} \, dx$$

The third line in (6) is referring to the flux. However, because we are dealing with an approximation that is discontinuous across the elements we have no way of evaluating $f(u(x_{i\pm\frac{1}{2}}, t))$. Instead we calculate a *numerical flux* that depends on the two values across each of the discontinuities. This numerical flux is defined as:

$$h(u_h)_{i\pm\frac{1}{2}} = h(u_h(x^-_{i\pm\frac{1}{2}}, t), u_h(x^+_{i\pm\frac{1}{2}}, t))$$

In [2] the motivation behind the Runge Kutta Discontinous Galerkin (RKDG) methods, which are the ones we use in this thesis, is explained. The idea is to make sure the method is monotone for constant basis functions, and then consider the higher-order solutions as perturbations of the constant basis solution. Monotone schemes are known to be stable and converge to the entropy solution for scalar conservation laws. A method $u_j^{n+1} = G(..., u_{j-1}^n, u_j^n, u_{j+1}^n, u_{j+2}^n, ...)$ is considered *monotone* if G is monotonically increasing in each of its arguments

The numerical flux is therefore chosen such that for a constant basis we have a monotone method. Two numerical fluxes that satisfies this are the Godunov flux,

$$h^G(a, b) = \begin{cases} \min_{a \le u \le b} f(u), & \text{if } a \le b \\ \max_{b \le u \le a} f(u), & \text{if otherwise} \end{cases}$$

and the local Lax-Friedrichs flux,

$$h^{LLF}(a,b) = \frac{1}{2}[f(a) + f(b) - C(b-a)],$$
$$C = \max_{\min(a,b)\le s\le\max(a,b)}|f'(s)|$$

For linear flux, $f(u) = cu$ these fluxes can be reduced to

$$h^{\text{linear}}(a,b) = \frac{c}{2}(a+b) - \frac{|c|}{2}(b-a) \tag{9}$$

This is simple to see for the Lax-Friedrichs flux where we have $|f'(s)| = |c|$. Substituting for the numerical flux and using the fact that $\phi_i^l(x_{i+\frac{1}{2}}) = 1$ and $\phi_i^l(x_{i-\frac{1}{2}}) = (-1)^l$, we get

$$\begin{aligned}
&- \Big(h(u_h)_{i+\frac{1}{2}}\phi^{l'}(x^-_{j+\frac{1}{2}}) - h(u_h)_{i-\frac{1}{2}}\phi^{l'}(x^+_{j-\frac{1}{2}})\Big) \\
&= - \Big(h(u_h(x^-_{i+\frac{1}{2}},t), u_h(x^+_{i+\frac{1}{2}},t))\phi^{l'}(x^-_{j+\frac{1}{2}}) \\
&\quad - h(u_h(x^-_{i-\frac{1}{2}},t), u_h(x^+_{i-\frac{1}{2}},t))\phi^{l'}(x^+_{j-\frac{1}{2}})\Big) \\
&= - \Big(h(u_h(x^-_{i+\frac{1}{2}},t), u_h(x^+_{i+\frac{1}{2}},t)) - (-1)^{l'}h(u_h(x^-_{i-\frac{1}{2}},t), u_h(x^+_{i-\frac{1}{2}},t))\Big) \\
&= - \Big(h(\sum_{l=0}^{k} u_i^l, \sum_{l=0}^{k}(-1)^l u_{i+1}^l) - (-1)^{l'}h(\sum_{l=0}^{k} u_{i-1}^l, \sum_{l=0}^{k}(-1)^l u_i^l)\Big)
\end{aligned}$$

The last part of the space-discretization is the initial condition from the last line in (6)

$$\int_{I_i} u_h(x,0)v(x)\,dx = \int_{I_i} u_0(x)v(x)\,dx$$
$$\forall i = 1,...,N \text{ and } l' = 0,...,k$$
$$\int_{I_i}(\sum_{l=0}^{k} u_i^l(0)\phi_i^l)\phi_i^{l'}\,dx = \int_{I_i} u_0(x)\phi_i^{l'}\,dx$$
$$\frac{\Delta x}{2l'+1}u_i^{l'}(0) = \int_{I_i} u_0(x)\phi_i^{l'}\,dx$$

Putting the pieces together we can write the method in matrix form with a vector $\boldsymbol{u}_i = [u_i^0(t), u_i^1(t), u_i^2(t), ...]$, mass matrix M, stiffness matrix K and numerical flux NF.

$$M \cdot \partial_t \boldsymbol{u}_i = K(\boldsymbol{u}_i) + NF$$
$$u_i^{l'}(0) = \frac{2l'+1}{\Delta x}\int_{I_i} u_0(x)\phi_i^{l'}\,dx \quad \forall l' = 0,...,k$$

Using up to 2'nd order Legendre polynomials we have:

$$M = \begin{bmatrix} \Delta x & 0 & 0 \\ 0 & \frac{\Delta x}{3} & 0 \\ 0 & 0 & \frac{\Delta x}{5} \end{bmatrix}$$

And we have for Burgers' equation $f(u) = \frac{1}{2}u^2$

$$K(\boldsymbol{u}_i)_{f(u)=\frac{1}{2}u^2} = \begin{bmatrix} 0 \\ (u_i^0)^2 + \frac{1}{3}(u_i^1)^2 + \frac{1}{5}(u_i^2)^2 \\ 2u_i^0 u_i^1 + \frac{4}{5}u_i^1 u_i^2 \end{bmatrix}$$

and for linear transport $f(u) = cu$

$$K(\boldsymbol{u}_i)_{f(u)=cu} = \begin{bmatrix} 0 \\ 2cu_i^0 \\ 2cu_i^1 \end{bmatrix}$$

and the numerical flux,

$$NF_{f(u)=\frac{1}{2}u^2} = \begin{bmatrix} -\Big(h(\sum_{l=0}^2 u_i^l, \sum_{l=0}^2 (-1)^l u_{i+1}^l) - h(\sum_{l=0}^2 u_{i-1}^l, \sum_{l=0}^2 (-1)^l u_i^l)\Big) \\ -\Big(h(\sum_{l=0}^2 u_i^l, \sum_{l=0}^2 (-1)^l u_{i+1}^l) + h(\sum_{l=0}^2 u_{i-1}^l, \sum_{l=0}^2 (-1)^l u_i^l)\Big) \\ -\Big(h(\sum_{l=0}^2 u_i^l, \sum_{l=0}^2 (-1)^l u_{i+1}^l) - h(\sum_{l=0}^2 u_{i-1}^l, \sum_{l=0}^2 (-1)^l u_i^l)\Big) \end{bmatrix}$$

$$NF_{f(u)=cu,\, c>0} = \begin{bmatrix} -(c(u_i^0 + u_i^1 + u_i^2) - c(u_{i-1}^0 + u_{i-1}^1 + u_{i-1}^2)) \\ -(c(u_i^0 + u_i^1 + u_i^2) + c(u_{i-1}^0 + u_{i-1}^1 + u_{i-1}^2)) \\ -(c(u_i^0 + u_i^1 + u_i^2) - c(u_{i-1}^0 + u_{i-1}^1 + u_{i-1}^2)) \end{bmatrix}$$

Our differential equation has now been completely discretized in space, and after multiplying our matrix form by M^{-1} on both sides we can see that we have obtained a system of first-order ODE's:

$$\frac{d}{dt}u_h = L_h(u_h), \quad \text{in } (0,T) \qquad u_h(t=0) = u_{0h} \tag{10}$$

Here $L_h(u_h) = M^{-1}[K(\boldsymbol{u}_i) + NF]$

3.2. **Time integration and Runge Kutta methods.** Now that we have a system of ODE's we must choose a suitable temporal discretization and method to solve the system. Total Variation Diminishing (TVD) and conservative schemes have been found to have desirable properties for solving the scalar conservation law. A scheme,

$$u_i^{n+1} = u_i^n - \lambda(\hat{f}_{i-\frac{1}{2}}^n - \hat{f}_{i+\frac{1}{2}}^n)$$

is TVD if

$$TV(u^{n+1}) \leq TV(u^n)$$

where we have the definition of Total Variation (TV),

$$TV(u) = \sum_i |u_{i+1} - u_i|.$$

From these definitions we can see that a TVD scheme can not have any spurious oscillations across discontinuities or shocks, as that would increase the TV. So far we have chosen our spatial discretization such

that it is monotone for constant basis functions. Harten showed in [5] that monotone schemes are TVD and that TVD schemes are monotonicity preserving. We will be using an explicit TVD Runge-Kutta method described in [11] and [2].

3.2.1. *TVD Runge-Kutta algorithm.* We partition $[0, T]$ into N_t parts with size $\Delta t = t^{n+1} - t^n, n = 0, ..., N_t - 1$

- Set $u_h^0 = u_{0h}$
- For $n = 0, ..., N_t - 1$ compute u_h^{n+1} from u_h^n
  (1) set $u_h^{(0)} = u_h^n$
  (2) for $i = 1, ..., k + 1$

$$u_h^{(i)} = \Big\{ \sum_{l=0}^{i-1} \alpha_{il} u_h^{(l)} + \beta_{il} \Delta t^n L_h(u_h^{(l)}) \Big\};$$

  (3) set $u_h^{n+1} = u_h^{k+1}$

Runge-Kutta discretization parameters		
order	α_{il}	β_{il}
2	$\begin{matrix} 1 & \\ \frac{1}{2} & \frac{1}{2} \end{matrix}$	$\begin{matrix} 1 & \\ 0 & \frac{1}{2} \end{matrix}$
3	$\begin{matrix} 1 & & \\ \frac{3}{4} & \frac{1}{4} & \\ \frac{1}{2} & 0 & \frac{2}{3} \end{matrix}$	$\begin{matrix} 1 & & \\ 0 & \frac{1}{4} & \\ 0 & 0 & \frac{2}{3} \end{matrix}$

3.3. **Slope limiters.** As we shall see from the numerical results, using higher order polynomials to approximate a solution gives rise to spurious oscillation where our solution has sharp gradients/shocks. Burgers' equation gives rise to shocks, and is indeed suffering from this phenomenon. We can explain this by the fact that our method is only TVD for constant basis functions. Thus we need to create some adjustments to our solution to enforce TVD when using a higher-order basis. To do this we use a TVDM (total variation diminishing in the means) slope limiter. The principle is to compare the slope of our approximate solution on each element, given in equation (4) to that of the differences between the mean value of the element and the surrounding elements, for linear basis functions with Legendre polynomials,

$$u_i^1(t) = \text{minmod}\big(\upsilon_i^1(t),\ \upsilon_{i+1}^0(t) - \upsilon_i^0(t),\ \upsilon_i^0(t) - \upsilon_{i-1}^0(t)\big). \tag{11}$$

This limiter is denoted $\Lambda\Pi_h^1$ in [2] where it is given as function of means and slopes, here it is given as a function of the polynomial coefficients. υ is the approximation before limiting, and minmod is a function that picks the value closest to 0, given that they all have the same sign, or it returns 0. Mathematically,

$$\text{minmod}(a_1, ..., a_\nu) = \begin{cases} s \min_{1\le n\le \nu} |a_n| & \text{if } s = sign(a_1) = ... = sign(a_\nu), \\ 0 & \text{otherwise.} \end{cases}$$

When dealing with Legendre polynomials of higher order, the algorithm is to compare the function value at the edges to the surrounding means, this higher-order limiting is denoted $\Lambda\Pi_h^k$,

$$u^-_{i+\frac{1}{2}} = \bar{v}_i + \text{minmod}(v^-_{i+\frac{1}{2}} - \bar{v}_i,\ \bar{v}_i - \bar{v}_{i-1},\ \bar{v}_{i+1} - \bar{v}_i)$$
$$u^+_{i-\frac{1}{2}} = \bar{v}_i - \text{minmod}(\bar{v}_i - v^+_{i-\frac{1}{2}},\ \bar{v}_i - \bar{v}_{i-1},\ \bar{v}_{i+1} - \bar{v}_i).$$

If $u^-_{i+\frac{1}{2}} = v^-_{i+\frac{1}{2}}$ and $u^+_{i-\frac{1}{2}} = v^+_{i-\frac{1}{2}}$ then we set $u_h|_{I_i} = v_h|_{I_i}$, otherwise we reduce the order of the approximate solution to a linear basis in that element and set the new slope value by using the first-order limiter $\Lambda\Pi_h^1$, (11).

3.3.1. *TVD Runge-Kutta algorithm with slope limiters.* Including sloper limiters the modified RKTVD algorithm becomes, We partition $[0, T]$ into N_t parts with size $\Delta t = t^{n+1} - t^n, n = 0, ..., N_t - 1$

- Set $u_h^0 = \Lambda\Pi_h^k u_{0h}$
- For $n = 0, ..., N_t - 1$ compute u_h^{n+1} from u_h^n
  (1) set $u_h^{(0)} = u_h^n$
  (2) for $i = 1, ..., k + 1$

$$u_h^{(i)} = \Lambda\Pi_h^k \Big\{ \sum_{l=0}^{i-1} \alpha_{il} u_h^{(l)} + \beta_{il} \Delta t^n L_h(u_h^{(l)}) \Big\};$$

  (3) set $u_h^{n+1} = u_h^{k+1}$

Note that the slope limiting is applied for each intermediate step. Each intermediate step has the ability to create spurious oscillations that might propagate if not corrected for before completing a full step.

3.4. **Numerical results.** First a note that is relevant for all figures throughout this entire thesis. All calculations are done on the domain $\Omega = [-10, 10]$, while the figures only show the domain $[-2, 6]$. The reason behind this is to create consistency between the figures with and without fractional diffusion. As will be explained in section 6, the fractional Laplacian acts globally so we capture more of the diffusion with a larger domain.

Figures 2 through 6 shows the solution to Burgers' equation using different orders of polynomials. From figures 3 and 5 it is very apparent that without using limiting the solution suffers from spurious oscillations across all sharp gradients, and especially the shock. For the constant case in figure 2, there are no oscillations because of the monotonicity property discussed in section 3. A keen eye can perhaps

spot that for constant basis functions the solution seems to smooth out the edges of the initial data much more quickly than for linear and quadratic basis. Figure 7 uses constant basis functions, but with a much finer grid, it much closer mimics the solution from higher-order polynomials. The next section deals with how quickly the different methods converge to the exact solution.

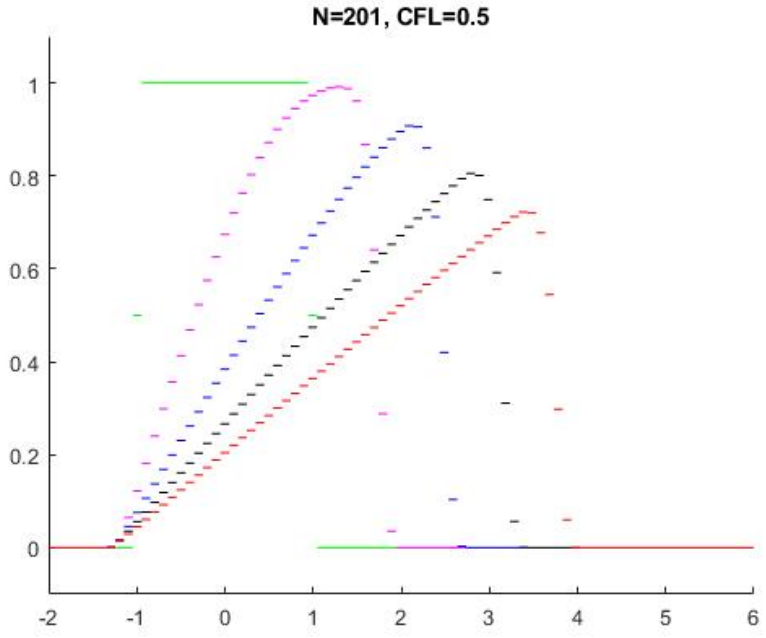

FIGURE 2. The evolution of Burgers' equation with constant basis functions and the block function as initial data

3.5. **Numerical convergence.** By refining our grid we can check the order of convergence for the different orders of the method. We can calculate the error between exact solutions and approximate solutions using the L_1 norm:

$$\text{Error} = \left|\left|u - u_h\right|\right|_1 = \int_{-\infty}^{\infty} |u - u_h| \, dx$$

We can look at approximate solutions using second order basis functions

$$\int_{-\infty}^{\infty} |u - u_h| \, dx = \int_{-\infty}^{\infty} |u - \bigoplus_{I_i} \sum_{l=0}^{k} u_i^l(t) v_i^l(x)| \, dx$$

For linear flux without the fractional diffusion the exact solutions are just translations of the initial data. The numerical convergence results will be looking at the convergence of the methods using linear flux with $c = 1$ on the Gauss function e^{-x^2}. Let us look at the solution after some

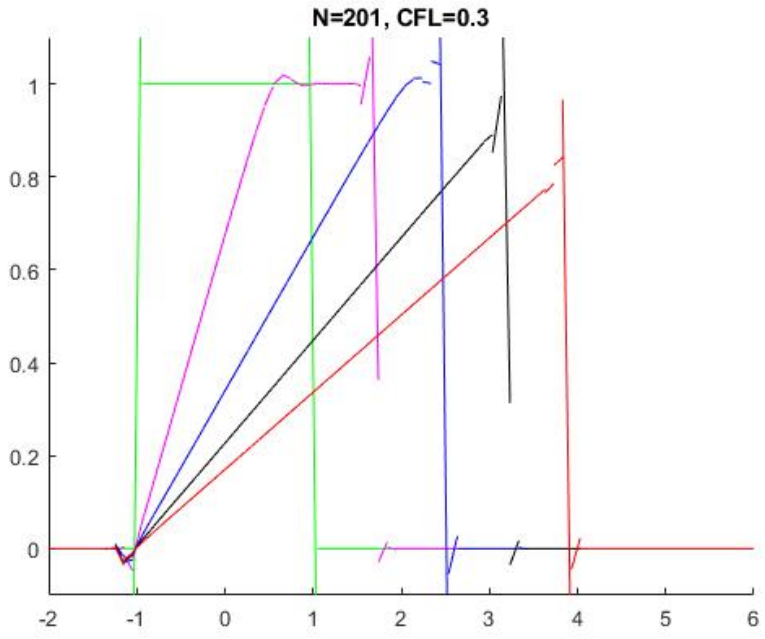

FIGURE 3. The evolution of Burgers' equation with up to linear basis functions and the block function as initial data, no slope limiting

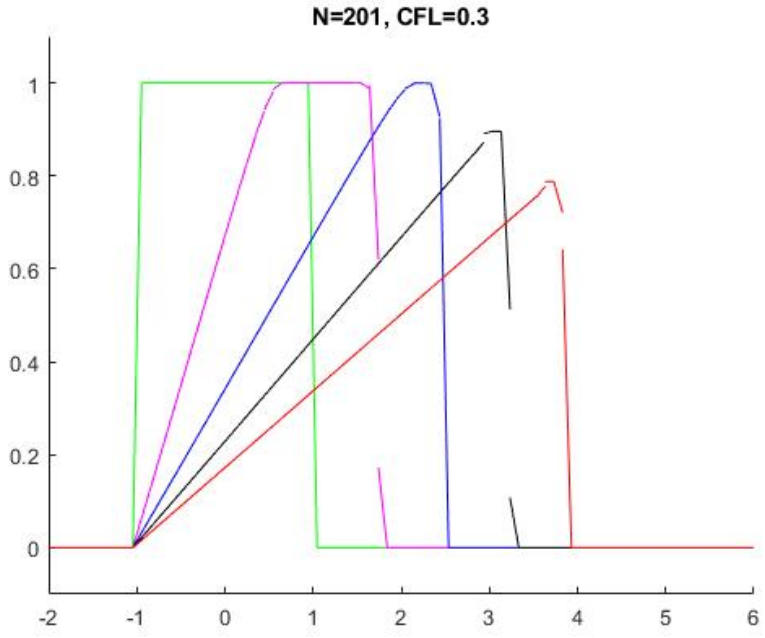

FIGURE 4. The same as figure 3, but with slope limiting

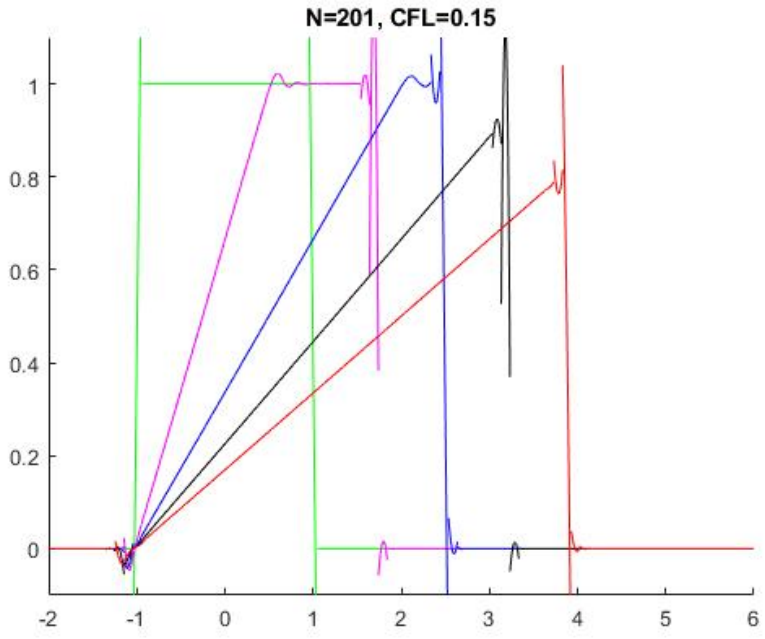

FIGURE 5. The evolution of Burgers' equation with up to quadratic basis functions and the block function as initial data, no slope limiting

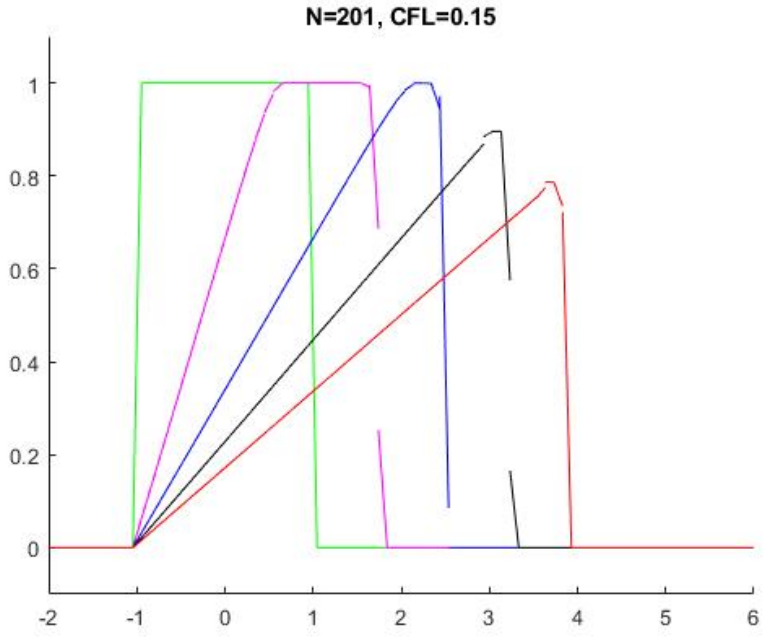

FIGURE 6. The same as figure 5, but with slope limiting

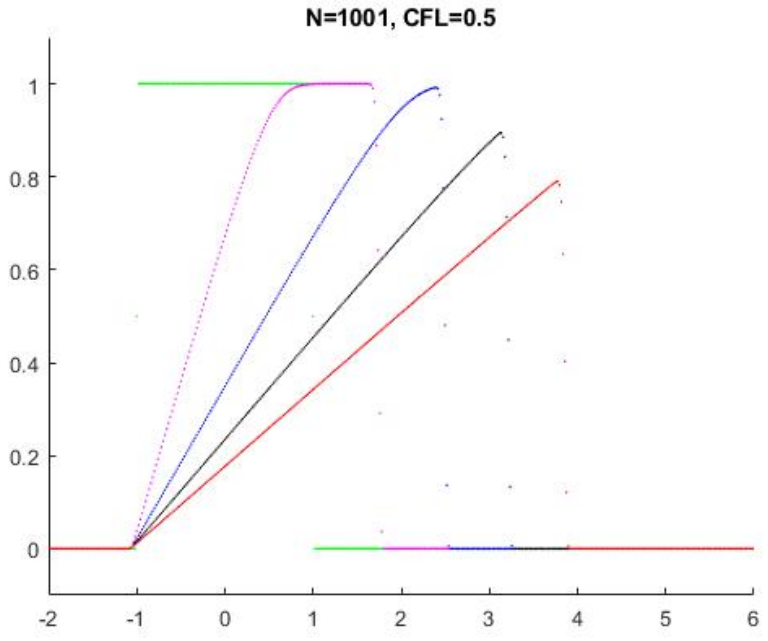

FIGURE 7. The same as figure 2, but with a finer grid

time T. Because the Gauss function is smooth the numerical results are without any slope limiting.

$$\sum_{i=1}^{N} \int_{I_i} |e^{-(T-x)^2} - \sum_{l}^{k} u_i^l(T)\phi_i^l(x)|\, dx$$

This integral can be solved numerically using quadratures, check the appendix for more details about quadrature integration.

Table 1 gives an overview for convergence rate for the DG method using different orders. For reference, $\left|\left|e^{-x^2}\right|\right|_1 = \sqrt{\pi} \approx 1.77$. The constant basis solution converges very slowly, while the quadratic can use a very coarse grid and still give an accurate solution. The trade-off being that the semi-discretized system (10) needed to be solved for each time step has the size $N \cdot (k+1)$, where N are the number of elements, and k is the order of polynomials used. The lower limit of 0.0004 is related to the time-discretization, and can be further reduced with more time steps. Figures 8 through 10 show the solution of the scalar conservation law with $f(u) = cu$, $c = 1$ using a Gauss function as initial data with $\Delta x = 0.4$. The initial condition wave is in green while the translated solution at $t = 2$ is in red. Using a quadratic basis is clearly superior for such a coarse grid.

Δx	Constant	Linear	Quadratic
2	2.0092	0.8044	0.3317
1.33	1.6131	0.5340	0.1219
1	1.3944	0.3120	0.0443
0.667	1.1536	0.1340	0.0107
0.4	0.8418	0.0416	0.0020
0.25	0.6096	0.0142	0.0006
0.167	0.4484	0.0057	0.0004
0.1	0.2943	0.0019	0.0004
0.0625	0.1947	0.0008	0.0004
0.04	0.1292	0.0005	0.0004
0.02	0.0669	0.0004	0.0004

TABLE 1. This table shows L_1 norm for the difference between the numerical solution and the exact solution for the scalar conservation law with linear flux at time $T = 2$. The Gauss function is used as initial data.

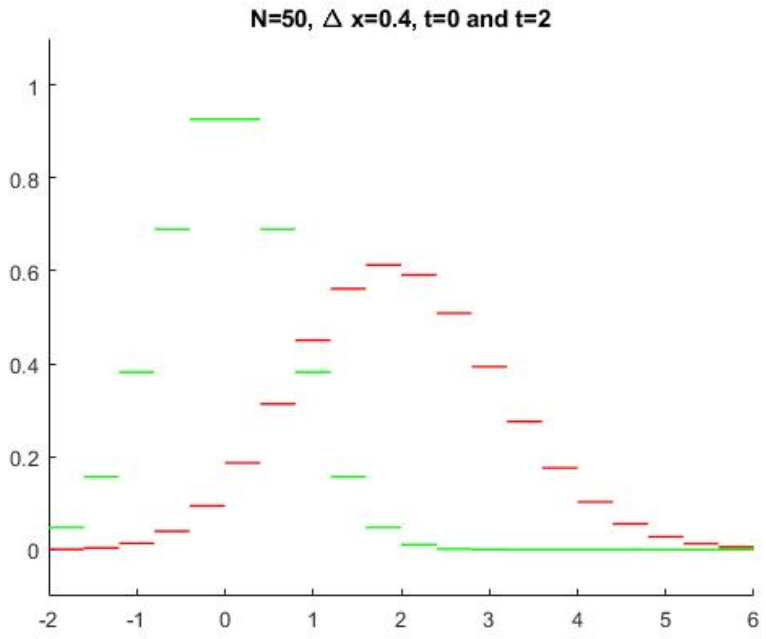

FIGURE 8. Constant basis: It is clearly causing some unwanted dissipation. Here the initial condition ($t = 0$) is shown in green, while the solution for $t = 2$ is in red.

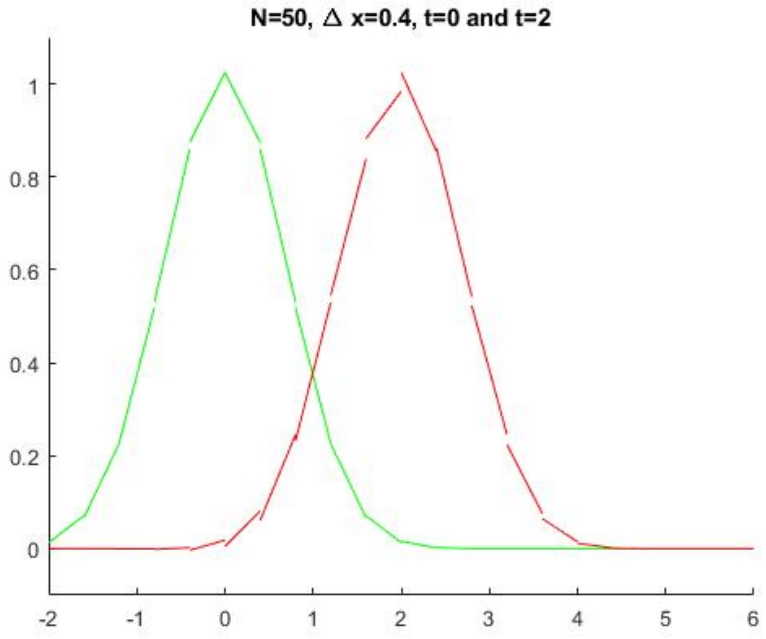

FIGURE 9. Linear basis: No longer has the unwanted dissipation, but it is still far from the smooth solution we would like.

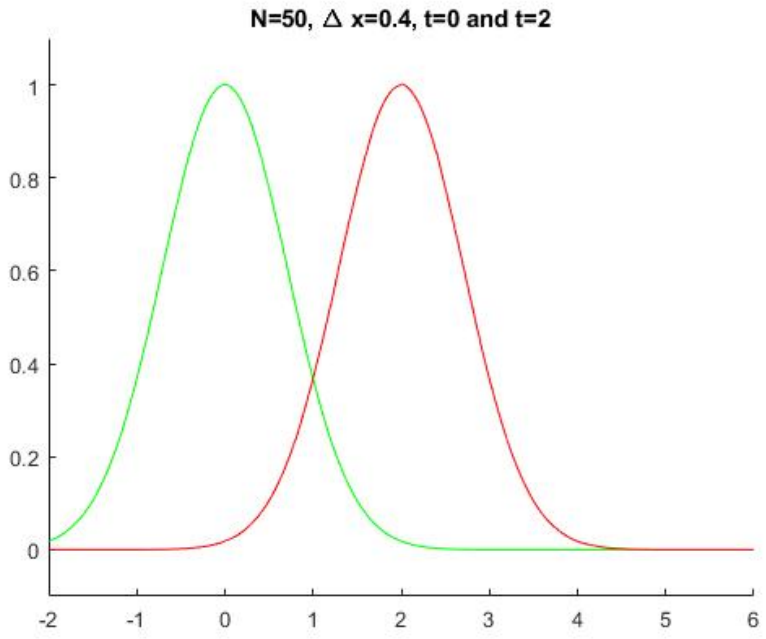

FIGURE 10. Quadratic basis: Almost impossible to tell that this is a numerical approximation and not the exact Gauss function.

4. A (super) short introduction to fractional calculus

Before adding fractional diffusion to our conservation law we need to understand exactly what is meant by fractional derivatives. One way to understand them is through the Fourier transform. The Fourier transform and inverse Fourier transform are given below,

$$\hat{f}(\xi) = \int_{-\infty}^{\infty} f(x) e^{-2\pi i x \xi} dx$$
$$f(x) = \int_{-\infty}^{\infty} \hat{f}(\xi) e^{2\pi i x \xi} d\xi.$$

One can calculate the Fourier transform of the derivative of a function as well,

$$\widehat{f'}(\xi) = \int_{-\infty}^{\infty} f'(x) e^{-2\pi i x \xi} dx = \Big[f(x) e^{-2\pi i x \xi} \Big]_{-\infty}^{\infty} - \int_{-\infty}^{\infty} (-2\pi i \xi) f(x) e^{-2\pi i x \xi} dx$$
$$= (2\pi i \xi) \int_{-\infty}^{\infty} f(x) e^{-2\pi i x \xi} dx = (2\pi i \xi) \hat{f}(\xi),$$

and by extension,

$$\widehat{f^{(n)}}(\xi) = (2\pi i \xi)^n \hat{f}(\xi).$$

By allowing n to be any number and not just integers, the Fourier transform gives a simple definition of fractional derivatives. This simple definition is enough to understand how a specific fractional operator behaves when Fourier transformed, namely the fractional Laplacian. All that is needed is some simple manipulation of the second derivative,

$$\widehat{f^{(2)}}(\xi) = -(2\pi\xi)^2 \hat{f}(\xi)$$
$$\widehat{-\partial_x^2 f}(\xi) = (2\pi\xi)^2 \hat{f}(\xi)$$
$$-\widehat{(-\partial_x^2)^{\frac{\alpha}{2}} f}(\xi) = -(2\pi)^\alpha |\xi|^\alpha \hat{f}(\xi),$$

the last equation is the fractional Laplacian and its Fourier transform.

5. The Fourier transform of fractional conservation laws

5.1. **Transforming the fractional Burgers' equation.** Because of this definition we can transform entire fractional differential equations into the Fourier space. One idea behind this is that the differential equations might be easier to solve in the Fourier space, and even though transforming the solution back again might be difficult, the L_2-norm preserving property of the Fourier space might let us do some simpler convergence analysis. A requirement of the transformation is of course that the L_2-norm of our initial condition is $< \infty$, the fact that we're

dealing with conservation laws let's us know that the solution $u(x,t)$ will stay bounded.

$$\partial_t u(x,t) + \partial_x f(u(x,t)) = -\epsilon(-\partial_x^2)^{\frac{\alpha}{2}} u(x,t)$$
$$F\Big\{\partial_t u(x,t)\Big\} + F\Big\{\partial_x f(u(x,t))\Big\} = F\Big\{ -\epsilon(-\partial_x^2)^{\frac{\alpha}{2}} u(x,t)\Big\}$$
$$\partial_t \hat{u}(k,t) + \int_{-\infty}^{\infty} \partial_x (\frac{1}{2} u(x,t)^2) e^{-2\pi i k x} dx = \int_{-\infty}^{\infty} -\epsilon(-\partial_x^2)^{\frac{\alpha}{2}} u(x,t) e^{-2\pi i k x}$$
$$\partial_t \hat{u}(k,t) + \Big[(\frac{1}{2}u^2)e^{-2\pi i k x}\Big]_{-\infty}^{\infty} - \int_{-\infty}^{\infty} (\frac{1}{2}u(x,t)^2)\frac{1}{-2\pi i k} e^{-2\pi i k x} dx$$
$$= -\epsilon(2\pi)^{\alpha}|k|^{\alpha}\hat{u}(k,t)$$
$$\partial_t \hat{u}(k,t) + \frac{1}{4\pi i k}\int_{-\infty}^{\infty} (u(x,t)^2) e^{-2\pi i k x} dx = -\epsilon(2\pi)^{\alpha}|k|^{\alpha}\hat{u}(k,t)$$
$$\partial_t \hat{u}(k,t) + \frac{1}{4\pi i k}(\hat{u}(k,t) * \hat{u}(k,t)) = -\epsilon(2\pi)^{\alpha}|k|^{\alpha}\hat{u}(k,t)$$

5.2. **Transforming the linear fractional equation, $f(u) = cu$.**

$$\partial_t u(x,t) + \partial_x f(u(x,t)) = -\epsilon(-\partial_x^2)^{\frac{\alpha}{2}} u(x,t)$$
$$F\Big\{\partial_t u(x,t)\Big\} + F\Big\{\partial_x f(u(x,t))\Big\} = F\Big\{ -\epsilon(-\partial_x^2)^{\frac{\alpha}{2}} u(x,t)\Big\}$$
$$\partial_t \hat{u}(k,t) + \int_{-\infty}^{\infty} \partial_x (c \cdot u(x,t)) e^{-2\pi i k x} dx = \int_{-\infty}^{\infty} -\epsilon(-\partial_x^2)^{\frac{\alpha}{2}} u(x,t) e^{-2\pi i k x}$$
$$\partial_t \hat{u}(k,t) + c \cdot (2\pi i k)\hat{u}(k,t) = -\epsilon(2\pi)^{\alpha}|k|^{\alpha}\hat{u}(k,t)$$
$$\partial_t \hat{u}(k,t) = -(\epsilon(2\pi)^{\alpha}|k|^{\alpha} + c \cdot (2\pi i k))\hat{u}(k,t)$$
$$\hat{u}(k,t) = C \cdot e^{-(\epsilon(2\pi)^{\alpha}|k|^{\alpha} + c\cdot(2\pi i k))t}$$

6. The fractional conservation law

Now that we have an understanding of fractional calculus we can entertain the idea of fractional differential equations. By adding the fractional Laplacian to our scalar conservation law we get what is known as the fractional conservation law, as mentioned in equation (1) in the introduction.

$$\partial_t u(x,t) + \partial_x f(u(x,t)) = -\epsilon\,(-\partial_x^2)^{\frac{\alpha}{2}} u(x,t)$$
$$\text{in } \Omega \times (0,T),\ \alpha \in (0,2),\ \epsilon > 0$$
$$u(x,0) = u_0(x)$$

Another way to represent fractional derivatives, other than through the Fourier transform, is with singular integrals. The integral representation of fractional Laplacian is given in [3] for two cases of α in one dimension.

For $\alpha \in (0,1)$ we have

$$-(-\partial_x^2)^{\frac{\alpha}{2}}\phi(x) = g_\alpha[\phi(x)] = c(\alpha)\int_{\mathbb{R}} \frac{\phi(x+z)-\phi(x)}{|z|^{1+\alpha}}dz$$

For $\alpha \in (1,2)$ we have

$$-(-\partial_x^2)^{\frac{\alpha}{2}}\phi(x) = g_\alpha[\phi(x)] = c(\alpha)\int_{\mathbb{R}} \frac{\phi(x+z)-\phi(x)-\partial_x\phi(x)\cdot z}{|z|^{1+\alpha}}dz$$

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

Now let's look at how we can discretize such an operator with the Discontinous Galerkin method. For the case where $\alpha \in (0,1)$ we do the same procedure as previously, we find the weak formulation by multiplying by a test function and integrating. For a method using $\alpha \in (1,2)$ see [13].

Using a constant basis the approximation is given by

$$u_h(x,t) = \bigoplus_{I_i} u_i^0(t)\phi_i^0(x)$$

The weak formulation of the fractional Laplacian becomes

$$c(\alpha)u_i^0(t)\int_{\mathbb{R}}\int_{\mathbb{R}} \frac{(\phi_i^0(x+z)-\phi_i^0(x))\phi_j^0(x)}{|z|^{1+\alpha}}dx\,dz$$

The order of integration is arbitrary as z and x are independent variables. Where we have the constant functions

$$\phi_i^0(x) = \begin{cases} 1 & \text{for } x \in [x_{i-\frac{1}{2}}, x_{i+\frac{1}{2}}] \\ 0 & \text{otherwise} \end{cases}$$

$$\phi_i^0(x+z) = \begin{cases} 1 & \text{for } x+z \in [x_{i-\frac{1}{2}}, x_{i+\frac{1}{2}}] \\ 0 & \text{otherwise} \end{cases}$$

$$G_i^j = c(\alpha)u_i^0(t)\int_{\mathbb{R}}\int_{\mathbb{R}} \frac{(\phi_i^0(x+z)-\phi_i^0(x))\phi_j^0(x)}{|z|^{1+\alpha}}dx\,dz$$

$$G_i^j = c(\alpha)u_i^0(t)\int_{\mathbb{R}}\int_{x_{j-\frac{1}{2}}}^{x_{j+\frac{1}{2}}} \frac{(\phi_i^0(x+z)-\phi_i^0(x))}{|z|^{1+\alpha}}dx\,dz$$

If $i=j$ we have two different cases for the inner integral

$$\xi(z)_{(i=j)} = \int_{x_{j-\frac{1}{2}}}^{x_{j+\frac{1}{2}}} \frac{(\phi_i(x+z)-\phi_i(x))}{|z|^{1+\alpha}}dx = \begin{cases} -\frac{1}{|z|^\alpha} & \text{for } |z| \le \Delta x \\ -\frac{\Delta x}{|z|^{1+\alpha}} & \text{for } |z| > \Delta x \end{cases}$$

Further simplifications can be made with a change of variable $z = \hat{z}\cdot\Delta x$

$$\xi(\hat{z})_{(i=j)} = \begin{cases} -\frac{1}{|\hat{z}|^{\alpha}\cdot\Delta x^{\alpha}} & \text{for } |\hat{z}| \leq 1 \\ -\frac{1}{|\hat{z}|^{1+\alpha}\cdot\Delta x^{\alpha}} & \text{for } |\hat{z}| > 1 \end{cases}$$

leading to the integral

$$\begin{aligned} G_j^j(\alpha) =& c(\alpha)u_i^0(t)\Big(\int_{\mathbb{R}} \xi(\hat{z})_{(i=j)}\, d\hat{z}\Big)\cdot\Delta x \\ =& c(\alpha)u_i^0(t)\Big(\int_{|\hat{z}|\leq 1} -\frac{1}{|\hat{z}|^{\alpha}}\, d\hat{z} + \int_{|\hat{z}|>1} -\frac{1}{|\hat{z}|^{1+\alpha}}\, d\hat{z}\Big)\cdot\Delta x^{1-\alpha} \\ =& c(\alpha)u_i^0(t)\Big(\int_0^1 -\frac{2}{\hat{z}^{\alpha}}\, d\hat{z} + \int_1^{\infty} -\frac{2}{\hat{z}^{1+\alpha}}\, d\hat{z}\Big)\cdot\Delta x^{1-\alpha} \\ =& c(\alpha)u_i^0(t)\Big(\Big[\frac{-2}{1-\alpha}z^{1-\alpha}\Big]_0^1 + \Big[\frac{2}{\alpha}z^{-\alpha}\Big]_1^{\infty}\Big)\cdot\Delta x^{1-\alpha} \\ =& c(\alpha)u_i^0(t)\Big(\frac{2}{\alpha-1} - \frac{2}{\alpha}\Big)\cdot\Delta x^{1-\alpha} \\ =& c(\alpha)\Big(\frac{2}{\alpha(\alpha-1)}\Big)u_i^0(t)\cdot\Delta x^{1-\alpha} \end{aligned}$$

Because of the condition $\alpha \in (0,1)$ we have convergence of the integrals as the inner integral removes one order of $|z|$ across $z = 0$. Solving for $\alpha = 0.5$ we get $c(0.5) = \frac{1}{4\pi}$ and $\frac{2}{\alpha(\alpha-1)} = -8$ leading to

$$G_j^j(0.5) = -\frac{2}{\pi}u_i^0(t)\cdot\Delta x^{0.5}$$

In the case that $i \neq j$ then we have $\phi_i^0 = 0$ leading to,

$$G_i^j = -c(\alpha)u_i^0(t)\int_{\mathbb{R}}\int_{x_{j-\frac{1}{2}}}^{x_{j+\frac{1}{2}}} \frac{\phi_i^0(x+z)}{|z|^{1+\alpha}} dx\, dz.$$

We start by assuming $j > i$ and we'll use symmetry to find the values for when $j < i$.
If $j > i$ we have to have $z > 0$ for the following integrals to be non-zero.

$$\begin{aligned} \xi(z)_{(j>i)} &= \int_{x_{j-\frac{1}{2}}}^{x_{j+\frac{1}{2}}} \frac{\phi_i(x+z)}{z^{1+\alpha}} dx \\ &= \begin{cases} \frac{z-(j-i-1)\Delta x}{z^{1+\alpha}} & \text{for } (j-i-1)\Delta x < z < (j-i)\Delta x \\ 0 & \text{otherwise} \end{cases} \end{aligned}$$

$$G_i^j = c(\alpha)u_i(t)\int_{(j-i-1)\Delta x}^{(j-i)\Delta x} \Big(\frac{1}{z^{\alpha}} - \frac{(j-i-1)\Delta x}{z^{1+\alpha}}\Big)\, dz$$

Again with a variable change $z = \hat{z} \cdot \Delta x$.

$$\begin{aligned} G_i^j &= c(\alpha) u_i^0(t) \int_{(j-i-1)}^{(j-i)} \frac{1}{\hat{z}^\alpha} - \frac{(j-i-1)}{\hat{z}^{1+\alpha}} \, d\hat{z} \cdot \Delta x^{1-\alpha} \\ &= c(\alpha) u_i^0(t) \Big(\Big[\frac{1}{1-\alpha} \hat{z}^{1-\alpha} \Big]_{j-i-1}^{j-1} - \Big[\frac{j-i-1}{\alpha} \hat{z}^{-\alpha} \Big]_{j-i-1}^{j-1} \Big) \cdot \Delta x^{1-\alpha} \\ &= c(\alpha) u_i^0(t) \Big(\frac{1}{1-\alpha} ((j-i)^{1-\alpha} - (j-i-1)^{1-\alpha}) \\ &+ \frac{(j-i-1)}{\alpha} ((j-i)^{-\alpha} - (j-i-1)^{-\alpha}) \Big) \cdot \Delta x^{1-\alpha} \end{aligned}$$

Note that $G_{i+1}^{j+1} = G_i^j$, this means that the full matrix $\boldsymbol{G}$ will be Toeplitz, meaning one can construct the entire matrix from just one row or column. One also needs to careful that for $j = i + 1$ the inner integral is $\xi(z)_{(j=i+1)} = \frac{1}{z}$ for $0 \leq z \leq \Delta x$, otherwise one risks dividing by 0 when calculating the outer integral. Again, using the value $\alpha = 0.5$ we can calculate the super-diagonal and, similarly, all other diagonals. $G_i^j = G_j^i$.

$$G_i^{i+1} = \frac{1}{2\pi} u_i^0(t) \cdot \Delta x^{0.5} \tag{12}$$

From this we can construct the matrix $\boldsymbol{G}$ for the discretized fractional Laplacian. The table below shows the values for each diagonal of the matrix, a symmetric Toeplitz matrix.

$$\begin{bmatrix} c(\alpha)\frac{-2}{\alpha(1-\alpha)} \\ c(\alpha)\frac{1}{1-\alpha} \\ c(\alpha)\big(\frac{1}{1-\alpha}(2^{1-\alpha}-1) + \frac{1}{\alpha}(2^{-\alpha}-1)\big) \\ c(\alpha)\big(\frac{1}{1-\alpha}(3^{1-\alpha}-2^{1-\alpha}) + \frac{1}{\alpha}(3^{-\alpha}-2^{-\alpha})\big) \\ c(\alpha)\big(\frac{1}{1-\alpha}(4^{1-\alpha}-3^{1-\alpha}) + \frac{1}{\alpha}(4^{-\alpha}-3^{-\alpha})\big) \\ \vdots \\ c(\alpha)\big(\frac{1}{1-\alpha}(N^{1-\alpha}-(N-1)^{1-\alpha}) + \frac{1}{\alpha}(N^{-\alpha}-(N-1)^{-\alpha})\big) \end{bmatrix}$$

and for $\alpha = 0.5$

$$\begin{bmatrix} -2/\pi \\ 0.5/\pi \\ \frac{1}{\pi}\big(0.5(\sqrt{2}-1) + 0.5(\frac{1}{\sqrt{2}}-1)\big) \\ \frac{1}{\pi}\big(0.5(\sqrt{3}-\sqrt{2}) + 0.5(\frac{1}{\sqrt{3}}-\frac{1}{\sqrt{2}})\big) \\ \frac{1}{\pi}\big(0.5(\sqrt{4}-\sqrt{3}) + 0.5(\frac{1}{\sqrt{4}}-\frac{1}{\sqrt{3}})\big) \\ \vdots \\ \frac{1}{\pi}\big(0.5(\sqrt{N}-\sqrt{N-1}) + 0.5(\frac{1}{\sqrt{N}}-\frac{1}{\sqrt{N-1}})\big) \\ \frac{1}{\pi}\big(0.5(\frac{N+1}{\sqrt{N}}-\frac{N}{\sqrt{N-1}})\big) \end{bmatrix}$$

Here we see that the fractional Laplacian has a global scope, the matrix is a full matrix, meaning that the diffusion process is applied on the whole domain. Ideally, for complete accuracy the entire $\mathbb{R}$ would need to be discretized, but this is not feasible, however when i and j are far apart, the diffusion diminishes rapidly, meaning that as long as the discretized domain includes the entire initial condition wave as well as some buffer there shouldn't be any significant loss of accuracy. This is the reason why the numerical examples are calculated for a larger domain, $[-10, 10]$, than what is shown in the figures.

6.1. **The linear case.** Using a linear basis we have the approximation

$$u_h(x,t) = \bigoplus_{I_i} \left(u_i^0(t)\phi_i^0(x) + u_i^1(t)\phi_i^1(x) \right)$$

For the two different test functions $v = \phi_i^0$ and $v = \phi_i^1$ the weak form of the fractional Laplacian becomes

$$\begin{aligned} c(\alpha) \Big(u_i^0(t) \int_{\mathbb{R}} \int_{\mathbb{R}} \frac{(\phi_i^0(x+z) - \phi_i^0(x))\phi_j^0(x)}{|z|^{1+\alpha}} dx\, dz \\ + u_i^1(t) \int_{\mathbb{R}} \int_{\mathbb{R}} \frac{(\phi_i^1(x+z) - \phi_i^1(x))\phi_j^0(x)}{|z|^{1+\alpha}} dx\, dz \Big) \end{aligned}$$

and

$$\begin{aligned} c(\alpha) \Big(u_i^0(t) \int_{\mathbb{R}} \int_{\mathbb{R}} \frac{(\phi_i^0(x+z) - \phi_i^0(x))\phi_j^1(x)}{|z|^{1+\alpha}} dx\, dz \\ + u_i^1(t) \int_{\mathbb{R}} \int_{\mathbb{R}} \frac{(\phi_i^1(x+z) - \phi_i^1(x))\phi_j^1(x)}{|z|^{1+\alpha}} dx\, dz \Big) \end{aligned}$$

Where we have the same constant functions as before, with the addition of the linear basis functions

$$\phi_i^1(x) = \begin{cases} \frac{2}{\Delta x}(x - x_i) & \text{for } x \in [x_{i-\frac{1}{2}}, x_{i+\frac{1}{2}}] \\ 0 & \text{otherwise} \end{cases}$$

$$\phi_i^1(x+z) = \begin{cases} \frac{2}{\Delta x}(x + z - x_i) & \text{for } x + z \in [x_{i-\frac{1}{2}}, x_{i+\frac{1}{2}}] \\ 0 & \text{otherwise} \end{cases}$$

There are 4 inner integrals that we need to solve.

$$\zeta_{0,0} = \int_{\mathbb{R}} (\phi_i^0(x+z) - \phi_i^0(x))\phi_j^0(x)\,dx$$
$$\zeta_{1,0} = \int_{\mathbb{R}} (\phi_i^1(x+z) - \phi_i^1(x))\phi_j^0(x)\,dx$$
$$\zeta_{0,1} = \int_{\mathbb{R}} (\phi_i^0(x+z) - \phi_i^0(x))\phi_j^1(x)\,dx$$
$$\zeta_{1,1} = \int_{\mathbb{R}} (\phi_i^1(x+z) - \phi_i^1(x))\phi_j^1(x)\,dx$$

Refer to appendix A for the solutions of these integrals analytically, and appendix B for how to solve them numerically.

6.2. **Numerical results.** Figures 11 and 12 shows the solution to Burgers' equation for a constant and linear basis. Comparing to the solutions of Burgers' equation without diffusion we see that the peak of the wave is getting smaller quicker, as well as some smoothing of the edge of the wave. This is evidence of the non-locality, diffusion is happening over the entire domain. Refer to appendix C.2 for results with other values of α and ϵ.

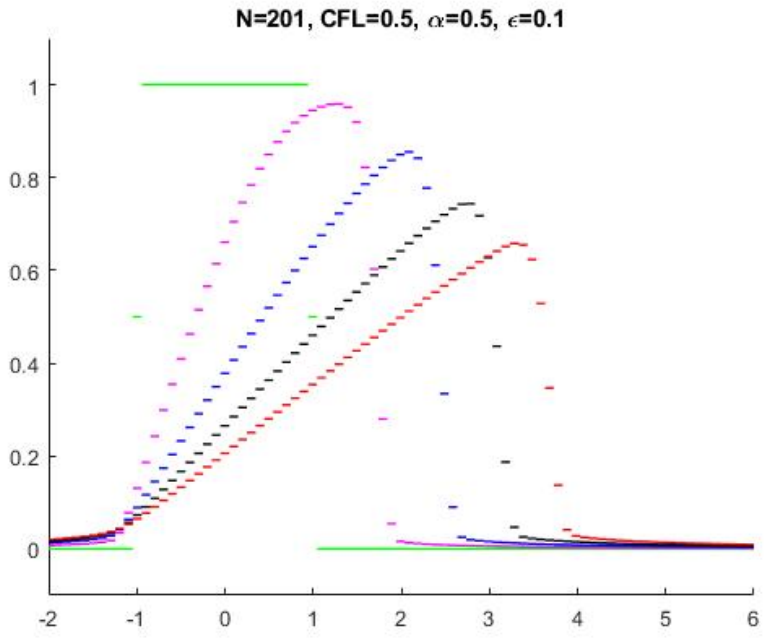

FIGURE 11. The evolution of Burgers' equation with fractional diffusion using a constant basis

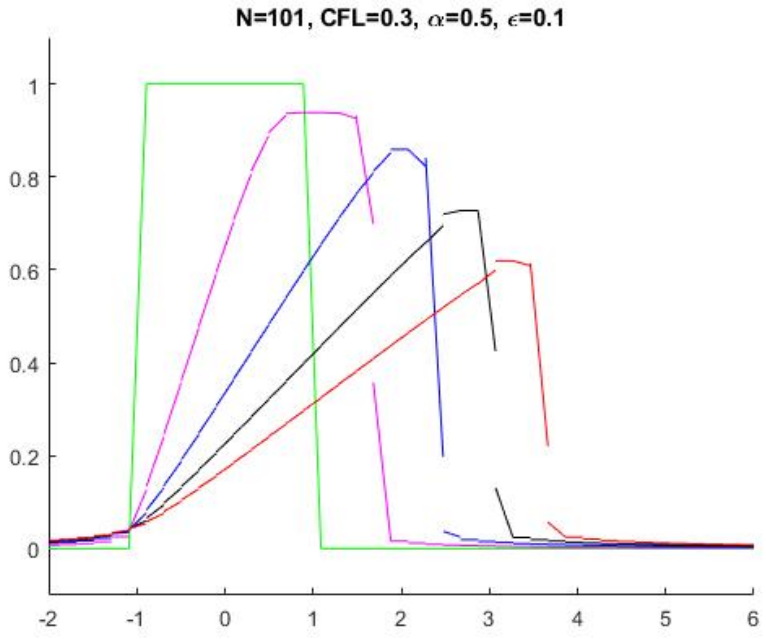

FIGURE 12. The evolution of Burgers' equation with fractional diffusion and slope limiting using a linear basis

7. CFL-condition

One can find the CFL condition for the semi-discrete system. $\frac{\partial u}{\partial t} = L(u)$. For non-linear flux the system $L(u)$ becomes very complex, but we can do some analysis using the linear case $F(u) = cu$. As we saw in section 3.1, the numerical fluxes for the linear case reduces to

$$h(a,b) = \frac{c}{2}(a+b) - \frac{|c|}{2}(b-a)$$

Using this flux, we can see that $\frac{\partial u}{\partial t} = L(u)$ becomes a linear system $\frac{\partial u}{\partial t} = \Lambda u$, where we have

$$\Lambda = -\frac{1}{2\Delta x}\begin{bmatrix} 2|c| & c-|c| & 0 & \dots & \dots & 0 \\ -c-|c| & 2|c| & c-|c| & \ddots & & \vdots \\ 0 & -c-|c| & \ddots & \ddots & \ddots & \vdots \\ \vdots & \ddots & \ddots & \ddots & c-|c| & 0 \\ \vdots & & \ddots & -c-|c| & 2|c| & c-|c| \\ 0 & \dots & \dots & 0 & -c-|c| & 2|c| \end{bmatrix}^{N\times N}$$

From [12], the eigenvalues of a tridiagonal, Toeplitz matrix, with diagonal, superdiagonal and subdiagonal d_1, d_2, d_3, is given by

$$\lambda_n = d_1 + 2\sqrt{d_2 d_3}\cdot\cos\left(\frac{n\pi}{N+1}\right) \text{ for } n = 1, ..., N$$

This let's us know that the largest eigenvalue in the system is max $\lambda_n = d_1+2\sqrt{d_2 d_3} = -\frac{1}{\Delta x}(|c|+\sqrt{(c-|c|)(-c-|c|)}) = -\frac{1}{\Delta x}(|c|+\sqrt{|c|^2-c^2}) = -\frac{|c|}{\Delta x}$. From this we can use the forward Euler step to find a stability condition.

$$\begin{aligned}
\partial_t u &= -\frac{|c|}{\Delta x}u \\
\Rightarrow u_{n+1} &= u_n - \Delta t\frac{|c|}{\Delta x}u_n \\
\Rightarrow u_n &= (1-|c|\frac{\Delta t}{\Delta x})^n u_0
\end{aligned}$$

For this to be stable we need

$$\begin{aligned}
\left|1-|c|\frac{\Delta t}{\Delta x}\right| &\leq 1 \\
-1 \leq 1-\frac{\Delta t}{\Delta x}|c| &\leq 1 \\
-2 \leq -\frac{\Delta t}{\Delta x}|c| &\leq 0 \\
0 \leq \frac{\Delta t}{\Delta x}|c| &\leq 2 \\
|c|\frac{\Delta t}{\Delta x} &\leq 2
\end{aligned}$$

The Gershgorin circle theorem also tells us that the eigenvalues of Λ all lie within $-\frac{|c|}{\Delta x} \pm \frac{2|c|}{\Delta x}$, which in the worst case is $\frac{|c|}{\Delta x}$ giving us the same stability condition as the one we arrived at.

8. Conclusion and future work

We have shown how the DG methods can be used to numerically solve the scalar conservation law. This includes both linear and non-linear flux. By making sure the method is Total Variation Diminishing (TVD) we could ensure a stable solution for constant basis functions. For higher-order basis functions and on differential equations that give rise to shocks we were able to salvage the accuracy through slope limiting techniques.

A direct continuation of the work in this thesis would be to analytically solve the integrals needed for the discretization of the fractional Laplacian for even higher orders, as the numerical integration is very inefficient. One could also investigate how diffusion will affect the CFL-condition, one would probably expect it to allow for a more lenient condition. Additional challenges include applying the methods to irregular grids and 2D and 3D domains. One could also use the framework here to discretize conservation laws with other types of singular non-local sources or even the larger class of partial integro-differential equations (PIDE).

Appendix A. Analytically solving the integrals for the discretized fractional Laplacian

$$\zeta_{0,0} = \int_{\mathbb{R}} (\phi_i^0(x+z) - \phi_i^0(x))\phi_j^0(x)\,dx$$
$$\zeta_{1,0} = \int_{\mathbb{R}} (\phi_i^1(x+z) - \phi_i^1(x))\phi_j^0(x)\,dx$$
$$\zeta_{0,1} = \int_{\mathbb{R}} (\phi_i^0(x+z) - \phi_i^0(x))\phi_j^1(x)\,dx$$
$$\zeta_{1,1} = \int_{\mathbb{R}} (\phi_i^1(x+z) - \phi_i^1(x))\phi_j^1(x)\,dx$$

The first one is the same integral we solved using basis functions. But now we'll solve it a little differently, first for $j = i$

$$\begin{aligned}
&\int_{\mathbb{R}} (\phi_i^0(x+z) - \phi_i^0(x))\phi_i^0(x)\,dx \\
&= \int_{\mathbb{R}} \phi_i^0(x+z)\phi_i^0(x)\,dx - \int_{\mathbb{R}} \phi_i^0(x)\phi_i^0(x)\,dx \\
&= \begin{cases} \int_{x_{i-\frac{1}{2}}}^{x_{i+\frac{1}{2}}-z} dx - \Delta x & \text{for } 0 < z \le \Delta x \\ \int_{x_{i-\frac{1}{2}}-z}^{x_{i+\frac{1}{2}}} dx - \Delta x & \text{for } -\Delta x \le z < 0 \\ -\Delta x & \text{otherwise} \end{cases} \\
&= \begin{cases} -z & \text{for } 0 < z \le \Delta x \\ z & \text{for } -\Delta x \le z < 0 \\ -\Delta x & \text{otherwise} \end{cases}
\end{aligned}$$

For $j = i - n$

$$\begin{aligned}
&\int_{\mathbb{R}} (\phi_i^0(x+z) - \phi_i^0(x))\phi_{i-n}^0(x)\,dx \\
&= \int_{\mathbb{R}} \phi_i^0(x+z)\phi_{i-n}^0(x)\,dx \\
&= \begin{cases} \int_{x_{i-\frac{1}{2}}-z}^{x_{i-n+\frac{1}{2}}} dx & \text{for } (i-j-1)\Delta x < z \le (i-j)\Delta x \\ \int_{x_{i-n-\frac{1}{2}}}^{x_{i+\frac{1}{2}}-z} dx & \text{for } (i-j)\Delta x \le z < (i-j+1)\Delta x \\ 0 & \text{otherwise} \end{cases} \\
&= \begin{cases} (n-1)\Delta x + z & \text{for } (i-j-1)\Delta x < z \le (i-j)\Delta x \\ (n+1)\Delta x - z & \text{for } (i-j)\Delta x \le z < (i-j+1)\Delta x \\ 0 & \text{otherwise} \end{cases}
\end{aligned}$$

for $j = i + n$

$$
\begin{aligned}
&\int_{\mathbb{R}} (\phi_i^0(x+z) - \phi_i^0(x))\phi_{i+n}^0(x)\,dx \\
&= \int_{\mathbb{R}} \phi_i^0(x+z)\phi_{i+n}^0(x)\,dx \\
&= \begin{cases} \int_{x_{i+n-\frac{1}{2}}}^{x_{i+\frac{1}{2}}-z} dx & \text{for } (i-j)\Delta x < z \leq (i-j+1)\Delta x \\ \int_{x_{i-\frac{1}{2}}-z}^{x_{i+n+\frac{1}{2}}} dx & \text{for } (i-j-1)\Delta x \leq z < (i-j)\Delta x \\ 0 & \text{otherwise} \end{cases} \\
&= \begin{cases} (n-1)\Delta x - z & \text{for } (i-j)\Delta x < z \leq (i-j+1)\Delta x \\ (n+1)\Delta x + z & \text{for } (i-j-1)\Delta x \leq z < (i-j)\Delta x \\ 0 & \text{otherwise} \end{cases}
\end{aligned}
$$

and now for solving the complete integral

$$
\begin{aligned}
&\int_{\mathbb{R}} \frac{\zeta_{0,0}}{|z|^{1+\alpha}}\,dz \quad \text{for } j = i \\
&= \int_0^{\Delta x} \frac{-z}{|z|^{1+\alpha}}\,dz + \int_{-\Delta x}^0 \frac{z}{|z|^{1+\alpha}}\,dz + \int_{\Delta x}^{\infty} \frac{-\Delta x}{|z|^{1+\alpha}}\,dz + \int_{-\infty}^{-\Delta x} \frac{-\Delta x}{|z|^{1+\alpha}}\,dz \\
&= \Big(\int_0^1 \frac{-1}{z^{\alpha}}\,dz - \int_{-1}^0 \frac{1}{(-z)^{\alpha}}\,dz + \int_1^{\infty} \frac{-1}{z^{1+\alpha}}\,dz + \int_{-\infty}^{-1} \frac{-1}{(-z)^{1+\alpha}}\,dz \Big) \Delta x^{1-\alpha} \\
&= \Big(\frac{1}{\alpha - 1} + \frac{1}{\alpha - 1} + \frac{-1}{\alpha} + \frac{-1}{\alpha} \Big) \Delta x^{1-\alpha} \\
&= \Big(\frac{2}{\alpha(\alpha - 1)} \Big) \Delta x^{1-\alpha}
\end{aligned}
$$

This is the same solution that was arrived at in section 5.1. Solving the outer integral is similar for all the other cases and has been omitted. Continuing with the inner integrals, the second can be solved as follows,

first for $i = j$,

$$\begin{aligned}
\int_{\mathbb{R}} (\phi_i^1(x+z) - \phi_i^1(x))\phi_i^0(x)\,dx &= \int_{x_{i-\frac{1}{2}}}^{x_{i+\frac{1}{2}}} \phi_i^1(x+z) - \phi_i^1(x)\,dx \\
&= \int_{x_{i-\frac{1}{2}}}^{x_{i+\frac{1}{2}}-z} \phi_i^1(x+z)\,dx \\
&= \begin{cases} \int_{x_{i-\frac{1}{2}}}^{x_{i+\frac{1}{2}}-z} \frac{2}{\Delta x}(x+z-x_i) & \text{for } 0 < z \le \Delta x \\ \int_{x_{i-\frac{1}{2}-z}}^{x_{i+\frac{1}{2}}} \frac{2}{\Delta x}(x+z-x_i) & \text{for } -\Delta x \le z < 0 \\ 0 & \text{otherwise} \end{cases} \\
&= \begin{cases} z - \frac{z^2}{\Delta x} & \text{for } 0 < z \le \Delta x \\ z + \frac{z^2}{\Delta x} & \text{for } -\Delta x \le z < 0 \\ 0 & \text{otherwise.} \end{cases}
\end{aligned}$$

Here we have used the fact that $\int_{x_{j-\frac{1}{2}}}^{x_{j+\frac{1}{2}}} \phi_i^1(x)\,dx = 0$ and the identities $(x_{j+\frac{1}{2}} - x_{j-\frac{1}{2}}) = \Delta x$ and $(\frac{1}{2}x_{j+\frac{1}{2}}^2 - \frac{1}{2}x_{j-\frac{1}{2}}^2) = x_i \cdot \Delta x$

and then for $i \neq j$, fixing i we calculate the integrals for $j = i + n$ and $j = i - n$

$$\begin{aligned}
&\int_{\mathbb{R}} (\phi_i^1(x+z) - \phi_i^1(x))\phi_{i-n}^0(x)\,dx \\
&= \int_{x_{i-n-\frac{1}{2}}}^{x_{i-n+\frac{1}{2}}} \phi_i^1(x+z)\,dx \\
&= \begin{cases} \int_{x_{i-\frac{1}{2}}-z}^{x_{i-n+\frac{1}{2}}} \frac{2}{\Delta x}(x+z-x_i)\,dx & \text{for } (i-j-1)\Delta x \le z \le (i-j)\Delta x \\ \int_{x_{i-n-\frac{1}{2}}}^{x_{i+\frac{1}{2}}-z} \frac{2}{\Delta x}(x+z-x_i)\,dx & \text{for } (i-j)\Delta x \le z < (i-j+1)\Delta x \\ 0 & \text{otherwise} \end{cases} \\
&= \begin{cases} (n^2-n)\Delta x - (2n-1)z + \frac{z^2}{\Delta x} & \text{for } (i-j-1)\Delta x \le z \le (i-j)\Delta x \\ -(n^2+n)\Delta x + (2n+1)z - \frac{z^2}{\Delta x} & \text{for } (i-j)\Delta x \le z < (i-j+1)\Delta x \\ 0 & \text{otherwise} \end{cases}
\end{aligned}$$

and similarly for $j = i + n$

$$\begin{aligned}
&\int_{\mathbb{R}} (\phi_i^1(x+z) - \phi_i^1(x))\phi_{i+n}^0(x)\,dx \\
&= \begin{cases} -(n^2-n)\Delta x - (2n-1)z - \frac{z^2}{\Delta x} & \text{for } (i-j)\Delta x \le z \le (i-j+1)\Delta x \\ (n^2+n)\Delta x + (2n+1)z + \frac{z^2}{\Delta x} & \text{for } (i-j-1)\Delta x \le z < (i-j)\Delta x \\ 0 & \text{otherwise} \end{cases}
\end{aligned}$$

The third integral, for $j = i - n$:

$$\int_{\mathbb{R}} (\phi_i^0(x+z) - \phi_i^0(x))\phi_j^1(x)\,dx$$
$$= \begin{cases} (n^2+n)\Delta x - (2n+1)z + \frac{z^2}{\Delta x} & \text{for } (i-j)\Delta x \le z \le (i-j+1)\Delta x \\ -(n^2-n)\Delta x + (2n-1)z - \frac{z^2}{\Delta x} & \text{for } (i-j-1)\Delta x \le z < (i-j)\Delta x \\ 0 & \text{otherwise} \end{cases}$$

and for $j = i + n$:

$$\int_{\mathbb{R}} (\phi_i^0(x+z) - \phi_i^0(x))\phi_j^1(x)\,dx$$
$$= \begin{cases} (n^2-n)\Delta x + (2n-1)z + \frac{z^2}{\Delta x} & \text{for } (i-j)\Delta x \le z \le (i-j+1)\Delta x \\ -(n^2+n)\Delta x - (2n+1)z - \frac{z^2}{\Delta x} & \text{for } (i-j-1)\Delta x \le z < (i-j)\Delta x \\ 0 & \text{otherwise} \end{cases}$$

The fourth integral is by far the hardest, and the answers have been checked with Mathematica,

$$\int_{\mathbb{R}} (\phi_i^1(x+z) - \phi_i^1(x))\phi_j^1(x)\,dx = \int_{\mathbb{R}} \phi_i^1(x+z)\phi_j^1(x)\,dx - \int_{\mathbb{R}} \phi_i^1(x)\phi_j^1(x)\,dx.$$

The second part is easy, using the diagonal properties of Legendre polynomials

$$-\int_{\mathbb{R}} \phi_i^1(x)\phi_j^1(x)\,dx = -\frac{\Delta x}{3} \quad \text{for } i = j$$

The first part for $z > 0$ or in other words $j = i - n$

$$\int_{\mathbb{R}} \phi_i^1(x+z)\phi_j^1(x)\,dx = \int_{\mathbb{R}} \phi_i^1(x+z)\phi_{i-n}^1(x)\,dx$$
$$= \begin{cases} \int_{x_{i-\frac{1}{2}}-z}^{x_{i-n+\frac{1}{2}}} \frac{4}{\Delta x^2}(x+z-x_i)(x-x_{i-n}) & \text{for } (i-j-1)\Delta x \le z \le (i-j)\Delta x \\ \int_{x_{i-n-\frac{1}{2}}}^{x_{i+\frac{1}{2}}-z} \frac{4}{\Delta x^2}(x+z-x_i)(x-x_{i-n}) & \text{for } (i-j)\Delta x \le z < (i-j+1)\Delta x \\ 0 & \text{otherwise} \end{cases}$$
$$= \begin{cases} -\frac{2}{3}\frac{z^3}{\Delta x^2} + \Delta x(-\frac{4}{3}n^3 + 4n^2 - 3n + \frac{1}{3}) + \frac{2}{\Delta x}z^2 + (2n^2 - 6n + 3)z \\ \qquad \text{for } (i-j)\Delta x \le z \le (i-j+1)\Delta x \\ \frac{2}{3}\frac{z^3}{\Delta x^2} + \Delta x(\frac{4}{3}n^3 - n + \frac{1}{3}) - \frac{2}{\Delta x}z^2 - (2n^2 - 2n - 1)z \\ \qquad \text{for } (i-j-1)\Delta x \le z < (i-j)\Delta x \\ 0 \qquad \text{otherwise} \end{cases}$$

and the first part for $z < 0$, in other words $j = i + n$

$$\int_{\mathbb{R}} \phi_i^1(x+z)\phi_j^1(x)\,dx = \int_{\mathbb{R}} \phi_i^1(x+z)\phi_{i+n}^1(x)\,dx$$

$$= \begin{cases} \int_{x_{i+n-\frac{1}{2}}}^{x_{i+\frac{1}{2}}-z} \frac{4}{\Delta x^2}(x+z-x_i)(x-x_{i+n}) & \text{for } (i-j)\Delta x \le z \le (i-j+1)\Delta x \\ \int_{x_{i-\frac{1}{2}}-z}^{x_{i+n+\frac{1}{2}}} \frac{4}{\Delta x^2}(x+z-x_i)(x-x_{i+n}) & \text{for } (i-j-1)\Delta x \le z < (i-j)\Delta x \\ 0 & \text{otherwise} \end{cases}$$

$$= \begin{cases} \frac{2}{3}\frac{z^3}{\Delta x^2} + \Delta x(-\frac{4}{3}n^3 + 4n^2 - 3n + \frac{1}{3}) + \frac{2}{\Delta x}z^2 - (2n^2 - 6n + 3)z \\ \qquad \text{for } (i-j)\Delta x \le z \le (i-j+1)\Delta x \\ -\frac{2}{3}\frac{z^3}{\Delta x^2} + \Delta x(\frac{4}{3}n^3 - n + \frac{1}{3}) - \frac{2}{\Delta x}z^2 + (2n^2 - 2n - 1)z \\ \qquad \text{for } (i-j-1)\Delta x \le z < (i-j)\Delta x \\ 0 \qquad \text{otherwise} \end{cases}$$

Appendix B. Numerically solving the integrals for the discretized fractional Laplacian

B.1. **Numerically solving the integrals.** As the order of our method increases, so do the number of integrals. Instead of solving them analytically as in the previous section, an idea could be do them numerically using a quadrature method. Let's look at our integral for constant basis functions as an example, $\forall i, j$ we have

$$G_i^j = -c(\alpha)u_i(t)\int_{\mathbb{R}}\int_{\mathbb{R}} \frac{(\phi_i^0(x+z) - \phi_i^0(x))\phi_j^0(x)}{|z|^{1+\alpha}}dx\,dz$$

A generic quadrature method has the form

$$\int_a^b f(x)\,dx \approx \frac{b-a}{2}\sum_{i=1}^{n} f(\frac{b-a}{2}x_i + \frac{a+b}{2})\cdot w_i$$

Again using the fact that $\phi_j(x)$ is 0 except for $x_{j-\frac{1}{2}} \le x \le x_{j+\frac{1}{2}}$, all integrals will take the form

$$\begin{aligned} G_i^j =& -c(\alpha)u_i(t)\int_{\mathbb{R}}\int_{x_{j-\frac{1}{2}}}^{x_{j+\frac{1}{2}}} \frac{(\phi_i^0(x+z) - \phi_i^0(x))\phi_j^0(x)}{|z|^{1+\alpha}}dx\,dz \\ \approx& -c(\alpha)u_i(t)\int_{\mathbb{R}}\sum_{l=1}^{n}\frac{\Delta x}{2}(\phi_i^0(\frac{\Delta x}{2}x_l + x_j + z) - \phi_i^0(\frac{\Delta x}{2}x_l + x_j)) \\ & \cdot \phi_j^0(\frac{\Delta x}{2}x_l + x_j)\cdot w_l\frac{dz}{|z|^{1+\alpha}} \end{aligned}$$

Here we have used $\frac{a+b}{2} = \frac{x_{j-\frac{1}{2}}+x_{j+\frac{1}{2}}}{2} = x_j$ and $\frac{b-a}{2} = \frac{x_{j+\frac{1}{2}}-x_{j-\frac{1}{2}}}{2} = \frac{\Delta x}{2}$. We know that ϕ are polynomials, so our numerical integration will be accurate. When it comes to integrating over z we run into the problem that it is a singular integral and not a polynomial. When solving the

integrals analytically, we found that the integrals are convergent for $\alpha \in (0,1)$, however we still need to be careful to pick non-zero nodes for z. One way to do this, and it is the one we have used for my numerical results, is to discretize in space such that 0 is in the middle of an element, and then using a Gauss-Lobatto quadrature, pick an even number of quadrature points, that way 0 is not one of the nodes.

Keeping this in mind, the complete numerical integration is, $\forall i, j$

$$G_i^j \approx -c(\alpha)u_i(t)\sum_{p=1}^{N}\sum_{k=1}^{\text{even } n}\sum_{l=1}^{n}\frac{\Delta x^2}{4}\phi_j^0(\frac{\Delta x}{2}x_l + x_j)\frac{w_l w_k}{|(\frac{\Delta x}{2}z_k + x_p)|^{1+\alpha}}$$
$$\cdot(\phi_i^0(\frac{\Delta x}{2}x_l + x_j + (\frac{\Delta x}{2}z_k + x_p)) - \phi_i^0(\frac{\Delta x}{2}x_l + x_j))$$

Note that x_l and z_k are the quadrature nodes, while x_j and x_p are values on the domain. For fine grids this numerical integration is extremely inefficient, and so, the analytical solutions from appendix A are to be preferred. However, there are some tricks one can do to reduce the number of calculations. Firstly, because of the symmetric properties of the Legendre polynomials $\boldsymbol{G}$ has either a symmetric Toeplitz structure or a Toeplitz structure where the lower triangular part is the negative of the upper triangular part. This means only one row/column needs to be calculated. Secondly, for most of the node values, the function ϕ_i^l is 0. For further insight into optimizing quadrature calculations, see [8].

Appendix C. Figures

This appendix gives some further figures and Numerical results

C.1. **The effect of α in the fractional Laplacian.** The following figures 13 through 15 shows numerical solutions for different values of α. All the figures in this section uses $\epsilon = 1$ in order to clearer see the effect of the fractional Laplacian and how it changes with α. We have used a linear basis.

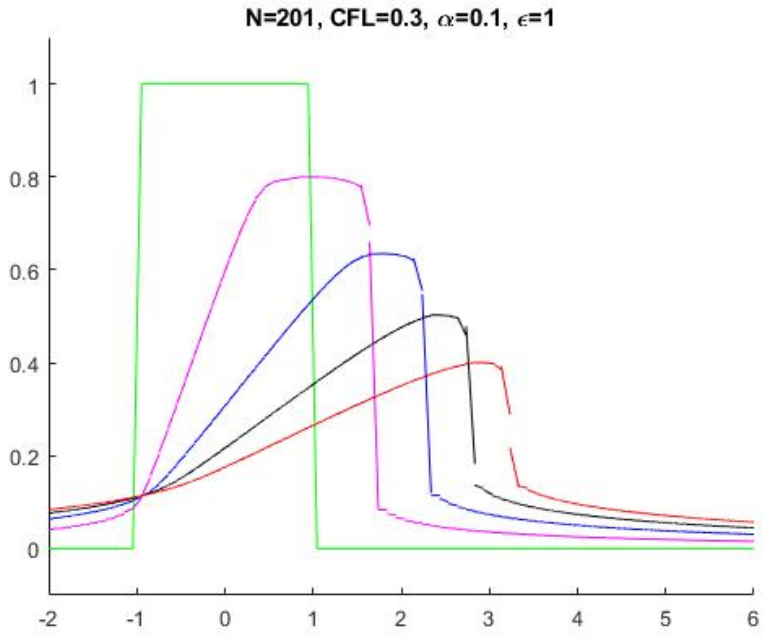

Figure 13. For very small α the shock looks to be preserved, even though compared to Figure 4 there is a considerable diffusion effect.

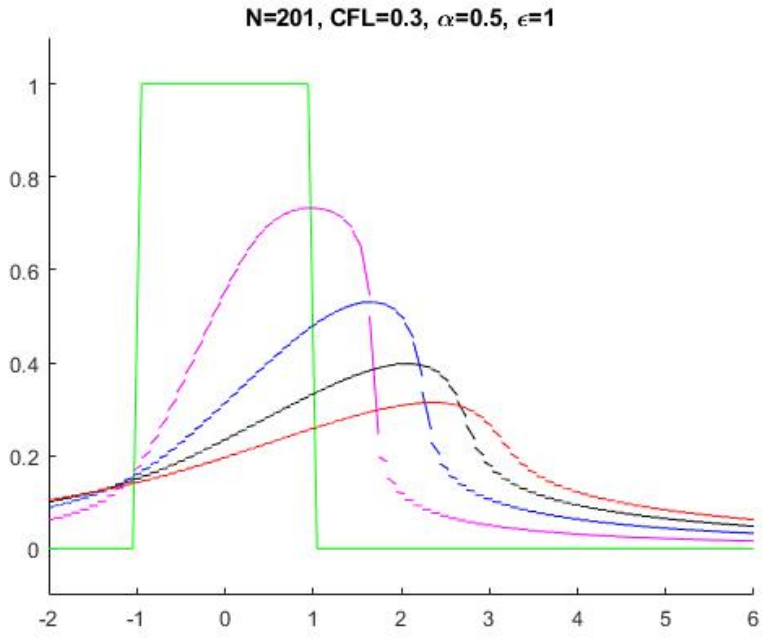

FIGURE 14. For $\alpha = 0.5$ the shock front is being lost in the diffusion. Note that the peak diminishes quicker here than for α closer to 0 or 1.

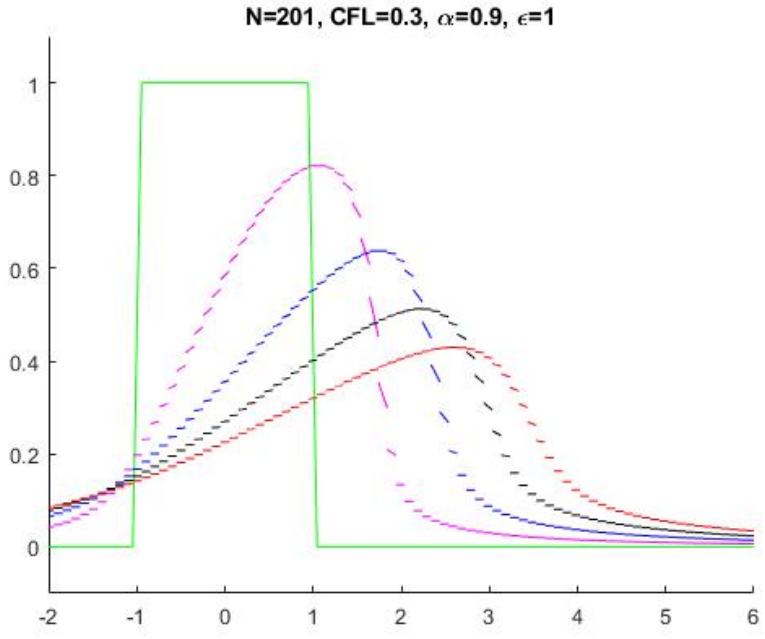

FIGURE 15. For $\alpha = 0.9$ all sharp gradients are quickly smoothed.

C.2. **The Buckley-Leverett equation.** The Buckley-Leverett equation is a scalar conservation law of the form (1) with $f(u) = \frac{4u^2}{4u^2+(1-u)}$. It is used to describe fluid displacement, see [4]. One thing to note is that the flux is non-convex and the solution consists of both a rarefaction and a shock, in other words, a compound wave. This effect can clearly be seen in figures 16 through 19, where the block function is being stretched out followed by a sharp shock at the wave front. The Buckley-Leverett solutions have all been computed using a constant basis and with the Godunov flux. Calculating the Godunov flux for the Buckley-Leverett equation is very simple when using a constant basis and as long as we use initial data such that $0 \leq u_0(x) \leq 1$. In this domain the Buckley-Leverett flux is monotonically increasing, meaning that finding the minimum and maximum of $f(u)$ is trivial. The numerical flux reduces simply to $h^G(a, b) = f(a)$.

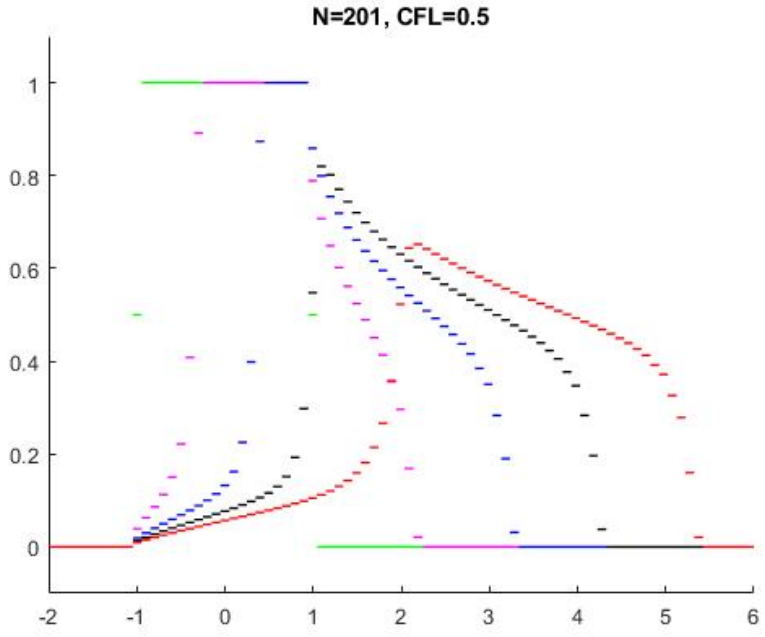

FIGURE 16. The evolution of the Buckley-Leverett equation without fractional diffusion.

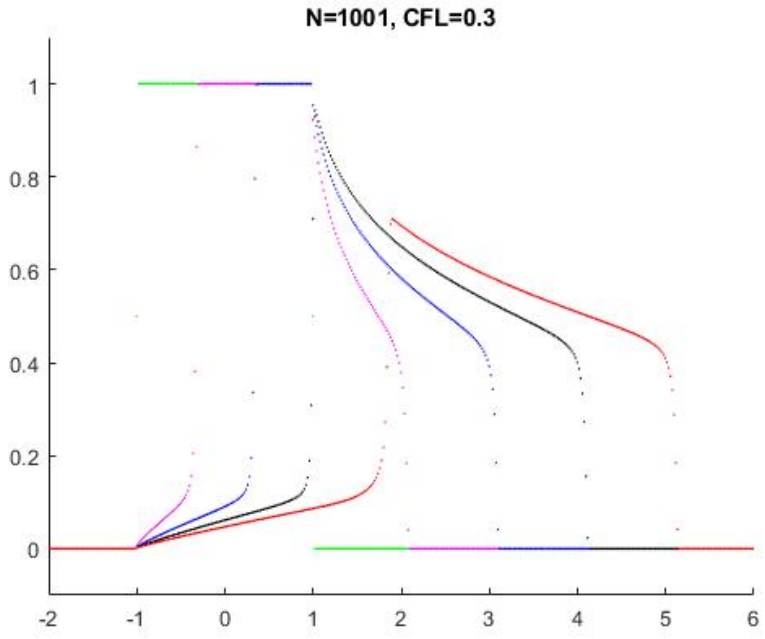

FIGURE 17. A higher resolution evolution of the Buckley-Leverett equation without fractional diffusion.

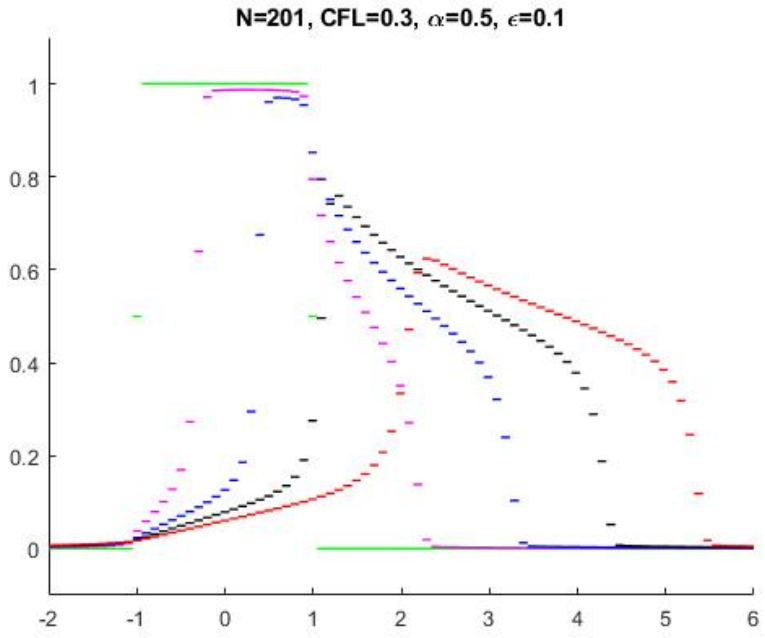

FIGURE 18. The evolution of the Buckley-Leverett equation with slight ($\epsilon = 0.1$) fractional diffusion.

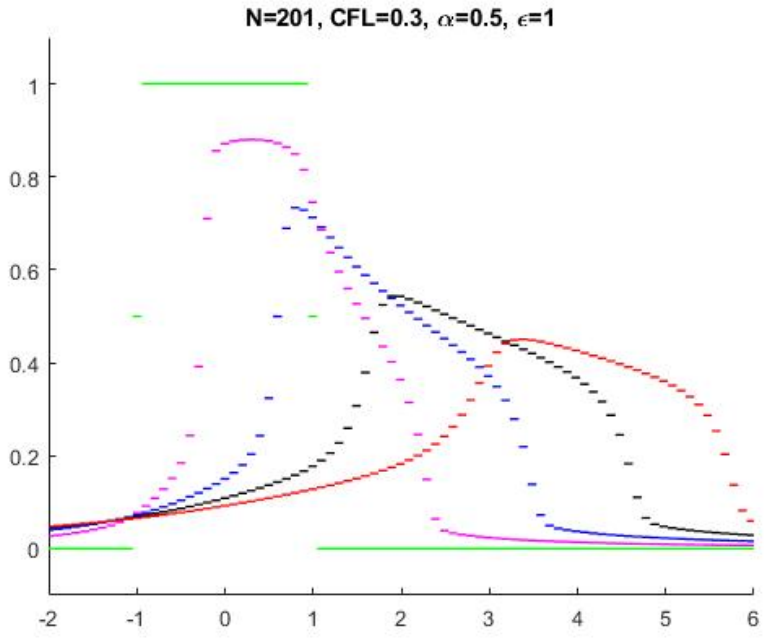

FIGURE 19. The evolution of the Buckley-Leverett equation with significant ($\epsilon = 1$) fractional diffusion.

Appendix D. On the usage of MATLAB

All plots in this thesis were created using `MATLAB`. The code has been included in the attachments as a .zip file. Here one will find 3 folders, one for each basis order. The main scripts are named `constantDGscript.m`, `linearDGscript.m` and `quadraticDGscript.m`. Inside them one can choose grid size, domain, limiting and/or adding fractional diffusion. For calculating the numerical convergence rates, the scripts `convscript0.m`, `convscript1.m` and `convscript2.m` were used. In them one can find a vector containing information as to how fine one wants the spatial grid to be.

References

[1] S. Cifani, E. R. Jakobsen, and K. H. Karlsen, *The discontinuous galerkin method for fractal conservation laws*, IMA Journal of Numerical Analysis, 31 (2011), pp. 1090–1122.

[2] B. Cockburn, *Discontinuous Galerkin Methods for Convection-Dominated Problems*, Springer Berlin Heidelberg, Berlin, Heidelberg, 1999, pp. 69–224.

[3] J. Droniou and C. Imbert, *Fractal first-order partial differential equations*, Archive for Rational Mechanics and Analysis, 182 (2006), pp. 299–331.

[4] S. E. Buckley and M. C. Leverett, *Mechanism of fluid displacement in sands*, Society of Petroleum Engineers, 146 (1942).

[5] A. Harten, *High resolution schemes for hyperbolic conservation laws*, Journal of Computational Physics, 135 (1997), pp. 260 – 278.

[6] J. Hesthaven and T. Warburton, *Nonlinear problems*, Springer New York, New York, NY, 2008, pp. 115–168.

[7] R. J. Leveque, *Finite Volume Methods for Hyperbolic Problems*, Cambridge University Press, New York, 2002.

[8] K. B. Ølgaard and G. N. Wells, *Quadrature representation of finite element variational forms*, Springer Berlin Heidelberg, Berlin, Heidelberg, 2012, pp. 147–158.

[9] A. Quarteroni, *Discontinuous element methods (DG and mortar)*, Springer Milan, Milano, 2014, pp. 267–289.

[10] ——, *The Galerkin finite element method for elliptic problems*, Springer Milan, Milano, 2014, pp. 61–119.

[11] C.-W. Shu, *Total-variation-diminishing time discretizations*, SIAM J. Sci. and Stat. Comput.,, 9 (1988), pp. 1073–1084.

[12] N. Silvia, P. Lionello, and R. Lothar, *Tridiagonal toeplitz matrices: properties and novel applications*, Numerical Linear Algebra with Applications, 20, pp. 302–326.

[13] Q. Xu and J. Hesthaven, *Discontinuous galerkin method for fractional convection-diffusion equations*, SIAM Journal on Numerical Analysis, 52 (2014), pp. 405–423.