INTERNSHIP REPORT

REGULARIZING DISCONTINUITIES BASED ON FILTERING USING DIRAC DELTA KERNELS

Author:
B.W. Wissink (S0199184)

Supervisor:
Prof. dr. ir. G.B. Jacobs

Home Institution
University of Twente
Enschede, The Netherlands
Faculty of Engineering Technology
Group of Engineering Fluid Dynamics
Prof. dr. ir. H.W.M. Hoeijmakers

Host Institution
San Diego State University
San Diego (CA), USA
Department of Aerospace Engineering
Computational Fluid Dynamics Laboratory
Prof. dr. ir. G.B. Jacobs

November 21, 2016

ABSTRACT

In this report the regularization of discontinuous initial conditions of the one-dimensional Advection Equation will be studied. The discrete initial conditions will be interpolated using polynomial interpolation. This polynomial interpolation is convoluted with a high order regularized Dirac-delta function. The equation will be solved using a spectral collocation method. The convolution with the polynomial-based Dirac-delta function is written in a matrix-vector multiplication for convenient implementation.

It is shown that this method yields stable results and higher order convergence away from the regularization zone for different discontinuous initial conditions. The influence of the variables of the regularized delta function is studied and explained.

Furthermore, the results are compared with the theoretical filter error. It is shown that the solution converges according to the theoretical filter error in the case of filtered boundary conditions and sufficiently wide regularization zones.

Keywords: Hyperbolic conservation laws, one-dimensional advection equation, regularization, Dirac-delta, spectral collocation matrix, filtering.

TABLE OF CONTENTS

LIST OF FIGURES

LIST OF TABLES

PAGE

CHAPTER 1
INTRODUCTION

1.1 BACKGROUND

The general one-dimensional, homogeneous hyperbolic partial differential equation (PDE) or conservation law is given as:

$$\frac{\partial}{\partial t}\vec{Q}(x,t) + \frac{\partial}{\partial x}\vec{F}(\vec{Q}) = 0, \tag{1.1}$$

In which $\vec{Q}$ represents the q conserved quantities and $\vec{F}$ is the flux function. This equation governs the behavior of a wide range of physical systems in which waves and convection are of importance such as gas dynamics, electromagnetism and traffic flow. This equation is able to form discontinuous solutions such as shocks in gas dynamics if the fluxes are nonlinear.

Shocks in gas dynamics occur in a wide range of applications, such as the flow over a supersonic airfoil, jet engines and explosions. One specific example is the combustion process in a SCRAMJET (Supersonic Combustion Ramjet) engine. The time the fluid is inside a SCRAMJET engine is in the order of milliseconds. In SCRAMJET combustion, the mixing is the limiting factor and is therefore an important aspect in improving the efficiency of this engine [10]. Therefore one is interested in high accuracy away from the shock where the mixing occurs.

A more fundamental example is the instability that occurs when two fluids of different density are impulsively accelerated by the passage of a shock wave. This instability is the so-called Richtmyer-Meshkov instability (RMI), which can be considered the impulsive-acceleration limit of the general Rayleigh-Taylor instability (RT). Supersonic combustion in a SCRAMJET may benefit from RMI as the fuel-oxidants interface is enhanced by the breakup of the fuel into finer droplets [2].

Commonly used numerical methods to accurately capture shock phenomena are high-resolution versions of Godunov's method in which Riemann problems are solved to determine the local wave structure and limiters are then applied to eliminate numerical oscillations. A variety of closely related approaches have also been developed for achieving high-resolution results. These include the second order Lax-Wendroff approach and higher order ENO (Essentially Non Oscillatory) and more recently WENO (Weighted Essentially

Non Oscillatory) methods to name a few. For nonlinear systems of equations, solving a Riemann problem can be expensive. A variety of approximate Riemann solvers have been developed to simplify this process, but algorithms based on Riemann problems are still typically expensive relative to approaches that only require evaluating the flux function [9].

Spectral methods are an efficient way of solving partial differential equations to high accuracy on simple domains if the data defining the problem are smooth. However, in case of non-smooth data, use of spectral methods leads to oscillations near the discontinuity. Grid refinement will not diminish the oscillations. The generation of oscillations near discontinuities is called Gibb's phenomenon [12].

It is however possible to overcome the Gibb's phenomenon. It appears that in the solution there is still sufficient information to recover high-order accuracy using some form of postprocessing, for instance Gegenbauer postprocessing. In the nonlinear case, the Gibb's phenomon may cause a stable scheme to become unstable due to amplification of oscillations. A common way to prevent this is the use of an exponential filter. Again, some form of postprocessing can be used to recover high-order accuracy even in the case of nonlinear equations [5].

The aforementioned methods all emphasize sharp shock capturing abilities. In certain applications however, one is less interested in sharp shock capturing and more interested in high order accuracy away from the shock.

If this is the case, this leads to the possibility to use efficient spectral methods in combination with filters to regularize discontinuities (shocks), basically sacrificing the accuracy with which the shock is captured for high order convergence away from the shock.

1.2 OBJECTIVE

In [11], a high-order approximation of the Dirac-delta function is presented. This function is used to approximate singular source terms in the numerical solution of non-linear systems of hyperbolic conservation laws (Euler equations in gas dynamics) arising in the simulation of particle-laden flows with shocks [3], [4]. The reason for this approximation is the fact that singular source terms can induce nonphysical oscillations in the numerical solution [6], [7]. In [11], it is also mentioned that these delta-functions can be used to smoothen singular sources using the operation of convolution.

In this study, the linear homogeneous one-dimensional, hyperbolic partial differential equation (PDE) will be considered:

$$\frac{\partial}{\partial t}\vec{Q}(x,t) + \frac{\partial}{\partial x}\vec{Q}(x,t) = 0 \tag{1.2}$$

The high order regularized delta functions introduced in [11] will be used to smoothen discontinuous initial conditions using the operation of convolution. This operation is written as a matrix-vector product, yielding the so-called filter-matrix. The 1D advection equation will be solved using the spectral collocation method.

Firstly, chapter 1 presents the motivation and contributions of the present study. In chapter 2 the filter-matrix is derived and the spatial and time discretization is given. Next, chapter 3 presents the numerical results for two initial conditions, a top-hat and a sine with discontinuity. In chapter 4 filtered boundary conditions and (wider) support widths are used to show that the solution converges according to the theoretical error in this case. Finally chapter 5 summarizes the results and gives an outlook for future work.

CHAPTER 2
FILTERING AND DISCRETIZATION

2.1 FILTERMATRIX

The filtering of discontinuities is based on convolution with the regularized delta function as described by Suarez in [11]. The regularized delta function is a polynomial of degree $2(\frac{m}{2}+k+1)$, defined as:

$$\delta_{\varepsilon}^{m,k}(x) = \begin{cases} \frac{1}{\varepsilon} P^{m,k}\left(\frac{x}{\varepsilon}\right), & |x| \leq \varepsilon \\ 0, & |x| > \varepsilon \end{cases} \tag{2.1}$$

This is a $(m+1)$th order accurate delta-sequence with compact support $[-\varepsilon, \varepsilon]$. It is a mixed polynomial consisting of two single polynomials controlling the number of vanishing moments m and the number of continuous derivatives at the end of the support k respectively. The regularized delta function is uniquely determined by the following properties:

$$(i) \int_{-1}^{1} P^{m,k}(\xi) d\xi = 1 \tag{2.2}$$

$$(ii) \left(P^{m,k}\right)^{(i)}(\pm 1) = 0 \quad \text{for} \quad i = 1, ..., k \tag{2.3}$$

$$(iii) \int_{-1}^{1} \xi^{i} P^{m,k}(\xi) d\xi = 0 \quad \text{for} \quad i = 1, ..., m \tag{2.4}$$

In which (i) states that the area of the function equals one as the discrete Dirac-delta function, (ii) determines the number of continuous derivatives and (iii) the number of vanishing moments.

Suppose the data is given by the variable $f(x)$ on the domain $-1 < x < 1$. Then the filtered data, denoted as $\tilde{f}(x)$, follows from the convolution with the regularized delta-function:

$$\tilde{f}(x) = \int_{-1}^{1} f(\tau) \delta_{\varepsilon}^{m,k}(x-\tau) d\tau \tag{2.5}$$

Since the regularized delta function is only non-zero within its support width, the integration boundaries in the previous expression can be rewritten as:

$$\tilde{f}(x) = \int_{x-\varepsilon}^{x+\varepsilon} f(\tau)\delta_{\varepsilon}^{m,k}(x-\tau)d\tau \tag{2.6}$$

However, in numerical methods, the data is only defined on a finite amount of discrete points x_i. Therefore the signal will be written using polynomial interpolation as described in [8]. For a given set of data points (x_i, y_i) there exist a polynomial of order n such that:

$$f(x_i) = y_i \quad 0 \leq i \leq n \tag{2.7}$$

This polynomial can be written in different forms, in Langrangian form:

$$f(x) = \sum_{i=0}^{N} f(x_i) l_i(x) \tag{2.8}$$

With the so-called Lagrange polynomials given as:

$$l_i(x) = \Pi_{j=0, j\neq i}^{N} \frac{x - x_j}{x_i - x_j} \qquad (0 \leq i \leq N) \tag{2.9}$$

Applying the convolution operation, 2.6, to the polynomial interpolation, 2.8, yields:

$$\tilde{f}(x) = \int_{x-\varepsilon}^{x+\varepsilon} \left[\sum_{i=0}^{N} f(x_i) l_i(\tau)\right] \delta_{\varepsilon}^{m,k}(x-\tau)d\tau \tag{2.10}$$

Expanding the summation gives:

$$\tilde{f}(x) = \int_{x-\varepsilon}^{x+\varepsilon} [f(x_0)l_0(\tau) + f(x_1)l_1(\tau) + ... + f(x_N)l_N(\tau)]\delta_{\varepsilon}^{m,k}(x-\tau)d\tau \tag{2.11}$$

This can be written as:

$$\begin{aligned} \tilde{f}(x) = f(x_0)\int_{x-\varepsilon}^{x+\varepsilon} l_0(\tau)\delta_{\varepsilon}^{m,k}(x-\tau)d\tau + f(x_1)\int_{x-\varepsilon}^{x+\varepsilon} l_1(\tau)\delta_{\varepsilon}^{m,k}(x-\tau)d\tau + ... \\ + f(x_N)\int_{x-\varepsilon}^{x+\varepsilon} l_N(\tau)\delta_{\varepsilon}^{m,k}(x-\tau)d\tau \end{aligned} \tag{2.12}$$

So, for some value of x the filtered version of $f(x)$ can be written as an inner product of two vectors, a vector containing the integral of the product of the l_n polynomial and the regularized delta function (which can be evaluated analytically) and a vector containing the data points $f(x_i)$:

$$\tilde{f}(x) = \begin{pmatrix} \int\limits_{x-\varepsilon}^{x+\varepsilon} l_0(\tau)\delta(x-\tau)d\tau & \int\limits_{x-\varepsilon}^{x+\varepsilon} l_1(\tau)\delta(x-\tau)d\tau & ... & \int\limits_{x-\varepsilon}^{x+\varepsilon} l_N(\tau)\delta(x-\tau)d\tau \end{pmatrix} \cdot \begin{pmatrix} f(x_0) \\ f(x_1) \\ ... \\ f(x_N) \end{pmatrix} \tag{2.13}$$

So, in general, the filtered vector can be written as a matrix vector multiplication of the 'filter matrix' with the original vector:

$$\begin{pmatrix} \tilde{f}(x_0) \\ ... \\ \tilde{f}(x_N) \end{pmatrix} = \begin{pmatrix} \int_{x_0-\varepsilon}^{x_0+\varepsilon} l_0(\tau)\delta_\varepsilon^{m,k}(x_0-\tau)d\tau & ... & \int_{x_0-\varepsilon}^{x_0+\varepsilon} l_N(\tau)\delta_\varepsilon^{m,k}(x_0-\tau)d\tau \\ ... & ... & ... \\ \int_{x_N-\varepsilon}^{x_N+\varepsilon} l_0(\tau)\delta_\varepsilon^{m,k}(x_N-\tau)d\tau & ... & \int_{x_N-\varepsilon}^{x_N+\varepsilon} l_N(\tau)\delta_\varepsilon^{m,k}(x_N-\tau)d\tau \end{pmatrix} \begin{pmatrix} f(x_0) \\ ... \\ f(x_N) \end{pmatrix} \tag{2.14}$$

2.2 Boundaries

In the filtering process, the discrete signal is written as a polynomial and is thus only a good representation on the domain $-1 < x < 1$. When filtering over the domain this leads to problems near the boundaries, since, in that case, the regularized delta function, $\delta_\varepsilon^{m,k}$ extends out of the domain. To resolve this, the data is not filtered if the delta function extends out of the domain. This means that:

if

$$x_i - \varepsilon < -1 \quad \text{or} \quad x_i + \varepsilon > 1 \tag{2.15}$$

The value of $f(x_i)$ should be returned, i.e. $\tilde{f}(x_i) = f(x_i)$, so the i-th row of the filtermatrix should simply become a zero-row with a one at the i-th column, for instance for x_0:

$$\tilde{f}(x_0) = \begin{pmatrix} 1 & 0 & ... & 0 \end{pmatrix} \cdot \begin{pmatrix} f(x_0) \\ f(x_1) \\ ... \\ f(x_N) \end{pmatrix} \tag{2.16}$$

2.3 CLENSHAW-CURTIS QUADRATURE

The general expression of the terms in the filtermatrix, 2.14, is as follows:

$$\int_{x_i-\varepsilon}^{x_i+\varepsilon} l_n(\tau)\delta_\varepsilon^{m,k}(x_i-\tau)d\tau \tag{2.17}$$

Analytical evaluation of these integrals is time-consuming and therefore an accurate numerical method is preferred. For this purpose Clenshaw-Curtis quadrature will be used.

Clenshaw-Curtis quadrature are methods for numerical integration that are based on an expansion of the integrand in terms of Chebyshev polynomials. Equivalently, they employ a change of variables $x = \cos\theta$ and use a discrete cosine transform (DCT) approximation for the cosine series.

Briefly, the function $f(x)$ to be integrated is evaluated at the N extrema or roots of a Chebyshev polynomial (Chebyshev points) and these values are used to construct a polynomial approximation for the function. This polynomial is then integrated exactly. In practice, the integration weights for the value of the function at each node are precomputed. In that case an integral can easily be computed as follows [1]:

$$\int_{-1}^{1} f(x)dx \approx \sum_{q=0}^{Q} w_q f(x_q) \tag{2.18}$$

Applying this to the general expression of the term in the filtermatrix, 2.17, yields:

$$\int_{x_i-\varepsilon}^{x_i-\varepsilon} l_n(\tau)\delta_\varepsilon^{m,k}(x_i-\tau)d\tau \approx \sum_{q=0}^{Q} w_q l_n(x_q)\delta_\varepsilon^{m,k}(x_i-x_q) \tag{2.19}$$

Substitution into the expression for the filtermatrix yields:

$$\overline{\overline{F}} = \begin{pmatrix} \sum\limits_{q=0}^{Q} w_q l_0(x_q)\delta(x_0-x_q) & \sum\limits_{q=0}^{Q} w_q l_1(x_q)\delta(x_0-x_q) & \dots & \sum\limits_{q=0}^{Q} w_q l_N(x_q)\delta(x_0-x_q) \\ \sum\limits_{q=0}^{Q} w_q l_0(x_q)\delta(x_1-x_q) & \sum\limits_{q=0}^{Q} w_q l_1(x_q)\delta(x_1-x_q) & \dots & \sum\limits_{q=0}^{Q} w_q l_N(x_q)\delta(x_1-x_q) \\ \dots & \dots & \dots & \dots \\ \sum\limits_{q=0}^{Q} w_q l_0(x_q)\delta(x_N-x_q) & \sum\limits_{q=0}^{Q} w_q l_1(x_q)\delta(x_N-x_q) & \dots & \sum\limits_{q=0}^{Q} w_q l_N(x_q)\delta(x_N-x_q) \end{pmatrix} \tag{2.20}$$

In order to use the Clenshaw-Curtis quadrature as shown in the filtermatrix, a Chebyshev subgrid has to be defined at every gridpoint x_i. These subdomains are thus defined as:

$$x_i - \varepsilon < x_q < x_i + \varepsilon \tag{2.21}$$

If we define the number of quadrature points as Q, then the subgrids become:

$$x_q = x_i - \varepsilon \cdot \cos\left(\frac{\pi \cdot q}{Q}\right), \quad k = 0, 1, ..., Q \tag{2.22}$$

If the cosine-series is known, Clenshaw-Curtis quadrature that evaluates the integrand on Q points integrates polynomials exactly up to degree $Q-1$. The regularized delta function is a polynomial of order $2(\frac{m}{2}+k+1)$:

$$\delta_\varepsilon^{m,k} = P^{(2(\frac{m}{2}+k+1))} \tag{2.23}$$

The polynomials in the polynomial interpolation, 2.9 are of order $N-1$:

$$l_n = P^{(N-1)} \tag{2.24}$$

Therefore the terms in the filtermatrix, are polynomials of order:

$$l_n \delta_\varepsilon^{m,k} = P^{(2(\frac{m}{2}+k+1)+N-1)} \tag{2.25}$$

So, the optimal number of quadrature points is:

$$Q = 2(\frac{m}{2}+k+1) + N \tag{2.26}$$

The Clenshaw-Curtis weights w_q are calculated using an existing algorithm which uses fast Fourier transform (FFT). Since the weights are the same on the different subdomains, as they are shifted in space, the weights only have to be calculated on a subgrid once.

2.4 Spatial Discretization

The spatial derivatives will be discretized using the so-called spectral method. Spectral methods are commonly used in the discretization of spatial derivatives in PDE's due to the exponential convergence rate in the case of smooth functions [12]. In the following the spectral collocation method will be briefly explained.

Spectral methods are based on a global approximation of the derivative, instead of a local approximation as is the case for finite difference methods. Spectral methods can be divided into Galerkin, Tau and Collocation methods. Of these methods, collocation methods are the simplest when it comes to the treatment of non-linear terms. Typically Fourier spectral methods are used in the case of periodic boundary conditions whereas polynomial spectral methods are used for non-periodic boundary conditions [5]. The latter will be used in the remainder of this report.

The collocation method is based on polynomial interpolation as used in section 2.1 and is repeated for convenience:

$$f(x) = \sum_{i=0}^{N} f(x_i) l_i(x) \tag{2.27}$$

With the so-called Lagrange polynomials given as:

$$l_i(x) = \Pi_{j=0, j\neq i}^{N} \frac{x - x_j}{x_i - x_j} \qquad (0 \leq i \leq N) \tag{2.28}$$

To determine the derivatives at the points x_i, the derivative of the interpolating polynomial is taken:

$$f(x_i)' \approx \sum_{j=0}^{N} f(x_j) l_j(x_i)' \tag{2.29}$$

The derivatives of the Lagrange polynomials are commonly denoted as $l_j(x_i)' = D_{i,j}$. The previous expression can be written as a matrix vector multiplication:

$$\boldsymbol{f}' = \overline{\overline{D}}\boldsymbol{f} + \mathcal{O}(N^{-r}) \tag{2.30}$$

With $\overline{\overline{D}}$ the so-called differentiation matrix and r a constant that depends on the order of approximation and the smoothness of the solution.

Spectral collocation methods usually don't use uniform grids. Typically Chebyshev points are used. Using these points minimizes the oscillatory behavior near the edges of the interval known as the Runge phenomenon [12]. The Chebyshev points are defined as:

$$x_i = -cos(i\pi/N), \qquad i = 0, 1, ..., N \tag{2.31}$$

From the above formula it follows that the points are more densely spaced near the edges of the interval.

2.5 Time Integration

For the time integration, the 4th order Runge-Kutta scheme will be used:

$$\begin{aligned} k1 &= \Delta t L(q_n, t_n) \\ k2 &= \Delta t L(q_n - \frac{1}{2}k_1, t_n + \frac{1}{2}\Delta t) \\ k3 &= \Delta t L(q_n - \frac{1}{2}k_2, t_n + \frac{1}{2}\Delta t) \\ k4 &= \Delta t L(q_n - k_3, t_n + \Delta t) \\ u_{n+1} &= u_n - \frac{1}{6}(k_1 + 2k_2 + 2k_3 + k_4) \end{aligned} \tag{2.32}$$

In order to obtain stable results, the CFL condition has to be satisfied:

$$\frac{u\Delta t}{\Delta x} \leq C \tag{2.33}$$

Since a Chebyshev grid has non-equispaced grid points, Δx is the minimum value between two points. Because the points are closely spaced near the edges of the domain this leads to a restriction of the allowable time-step that is smaller compared to an equispaced grid.

CHAPTER 3
NUMERICAL RESULTS

3.1 Numerical Variables

In this chapter the solutions of the one-dimensional advection equation will be presented in which the initial condition is filtered using the filtermatrix as defined in 2.20. First the results for a top-hat initial condition will be shown, followed by the results for a sinus with a discontinuity as initial condition. In the filtering process, the support width ε of the delta function $\delta_\varepsilon^{m,k}$ has to be chosen.

In [11] the optimal scaling parameter for smoothing is derived in case the numerical integration is done using a composite Newton-Cotes quadrature rule. Under the assumption that $m, k \geq 2$ and $s \leq \min(m, k) - 1$, in which s is the degree of exactness of the Newton-Cotes quadrature rule, the optimal scaling parameter is given as:

$$\varepsilon = \mathcal{O}\left(\left(\sum_{i=0}^{N_p-1} h_i^{s+2}\right)^{1/(m+s+3)}\right) \tag{3.1}$$

However, as we are using Clenshaw-Curtis quadrature in the smoothing process, $s > \min(m, k) - 1$ and the optimal scaling parameter can't be used.

Therefore the optimal scaling parameter for delta-sequences will be used. This is defined as:

$$\varepsilon = \mathcal{O}(N^{-k/(m+k+2)}) = q \cdot N^{-k/(m+k+2)} \tag{3.2}$$

The factor q is determined empirically such that the support width ε is sufficiently wide to get stable converging results. The factor depends on the delta function variables m and k. For our numerical tests the values used are summarized below:

		$N = 32$	$N = 64$	$N = 128$	$N = 256$
$m = 1, k = 5$	$q = 2.1$	$\varepsilon = 0.241$	$\varepsilon = 0.156$	$\varepsilon = 0.101$	$\varepsilon = 0.066$
$m = 3, k = 8$	$q = 2.4$	$\varepsilon = 0.284$	$\varepsilon = 0.186$	$\varepsilon = 0.121$	$\varepsilon = 0.079$
$m = 5, k = 8$	$q = 2.2$	$\varepsilon = 0.347$	$\varepsilon = 0.239$	$\varepsilon = 0.165$	$\varepsilon = 0.114$
Timestep	$\Delta t = 0.00001$				

Table 3.1: Variables used in numerical experiments

3.2 1D ADVECTION EQUATION, TOP-HAT

The 1D advection equation with a top-hat initial condition on the domain $-1 < x < 1$ is defined as follows:

$$\begin{aligned}\frac{\partial u}{\partial t} + \frac{1}{2}\frac{\partial u}{\partial x} &= 0 \\ u(x,0) &= 0 \qquad x \leq -0.25 \\ &= 1 \qquad -0.25 < x < 0.25 \\ &= 0 \qquad x \leq 0.25 \\ u(-1,t) &= 0\end{aligned} \tag{3.3}$$

The analytical solution is given as:

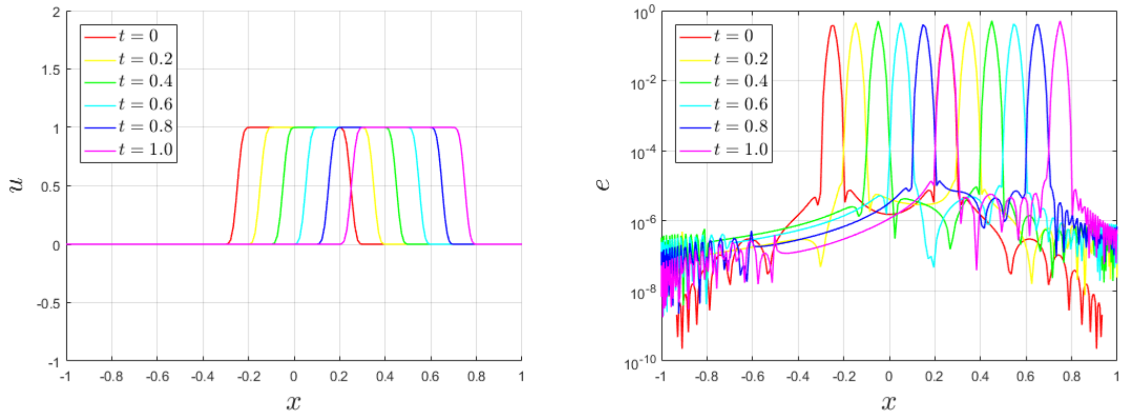
$$u = u_0(x - 1/2t) \tag{3.4}$$

The initial condition will be filtered using the filter matrix, 2.14:

$$\tilde{\boldsymbol{f}}_0 = \overline{\overline{F}}\boldsymbol{f}_0 \tag{3.5}$$

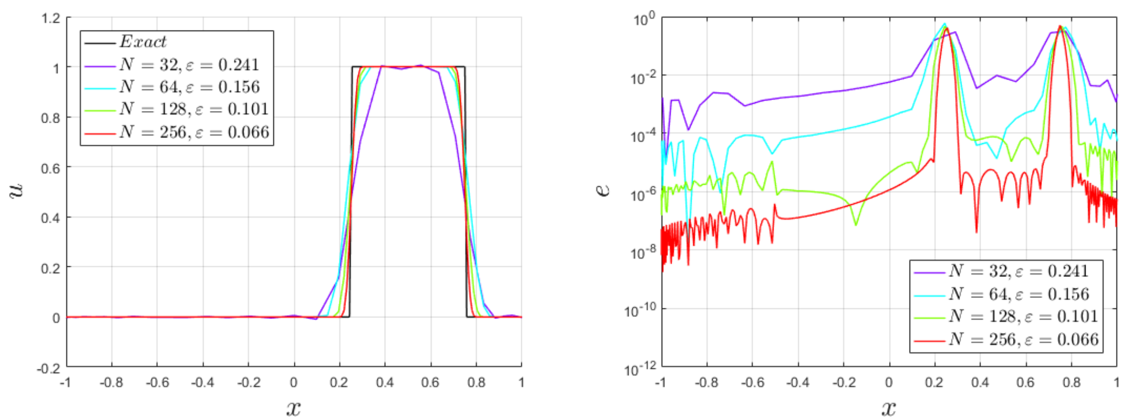
After this, the filtering isn't used again. Next, the partial differential equation will be solved using the discretizations described in 2.30 and 2.32.

3.2.1 Results, $m = 1$, $k = 5$



(a) (b)

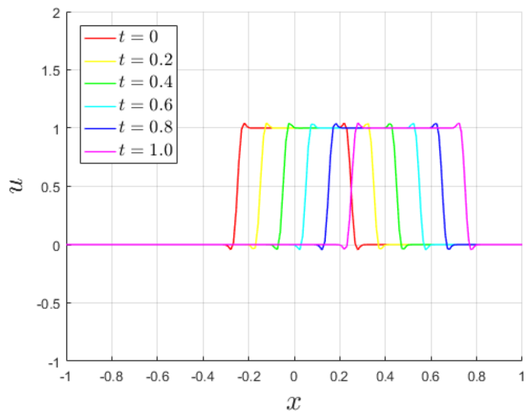
Figure 3.1: Time evolving numerical solution (a) and error (b) for $N = 256$, $m = 1$, $k = 5$, $\varepsilon = 0.066$



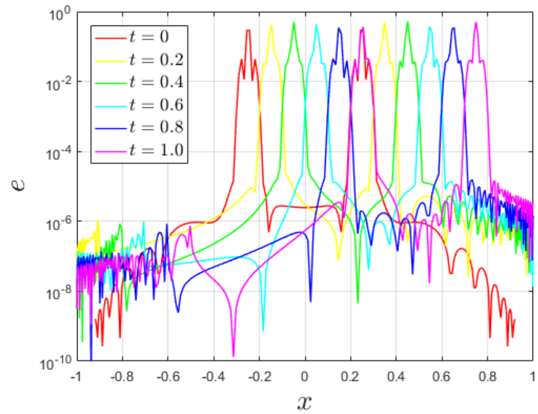
(a) (b)

Figure 3.2: Numerical solution (a) and error (b) at $t = 1$, for the four different grids for $m = 1$, $k = 5$

3.2.2 Results, $m = 3$, $k = 8$

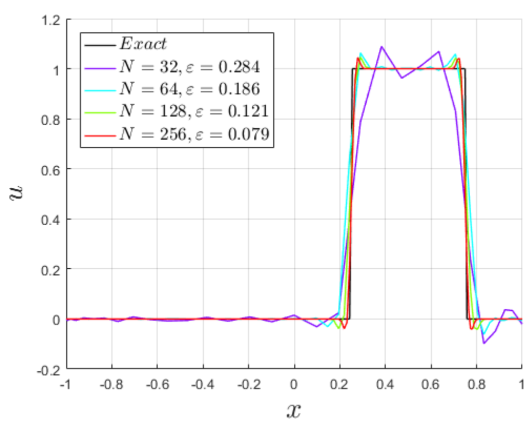


(a)

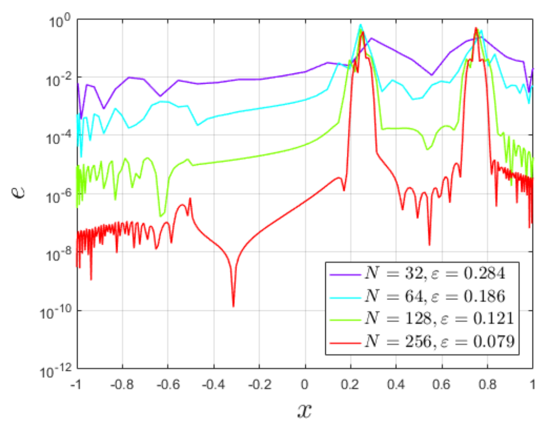


(b)

Figure 3.3: Time evolving numerical solution (a) and error (b) for $N = 256$, $m = 3$, $k = 8$, $\varepsilon = 0.079$



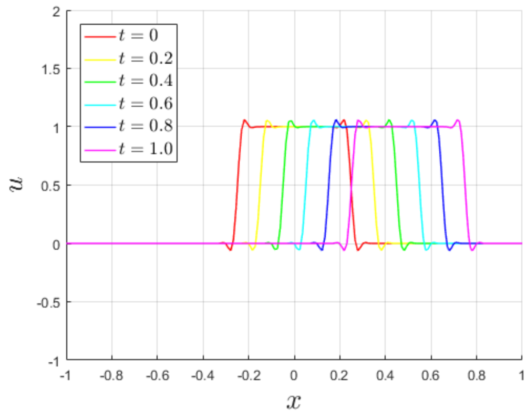
(a)



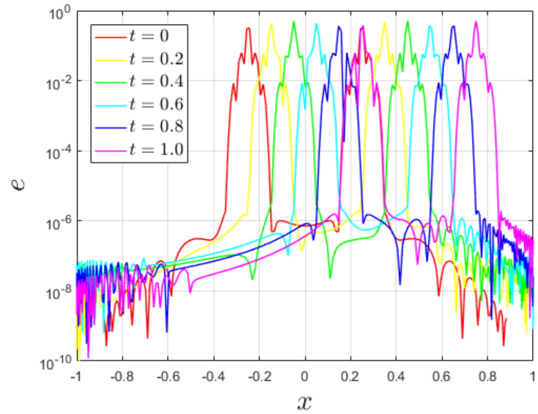
(b)

Figure 3.4: Numerical solution (a) and error (b) at $t = 1$, for the four different grids for $m = 3$, $k = 8$

3.2.3 Results, $m = 5$, $k = 8$

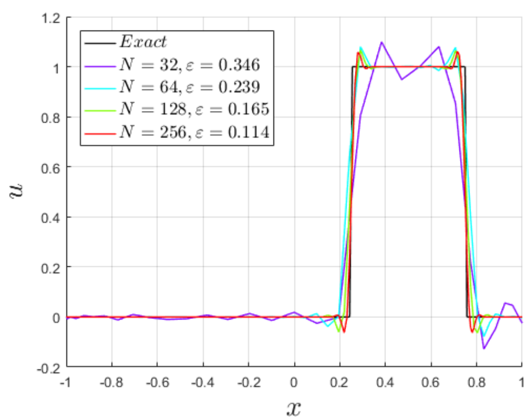


(a)

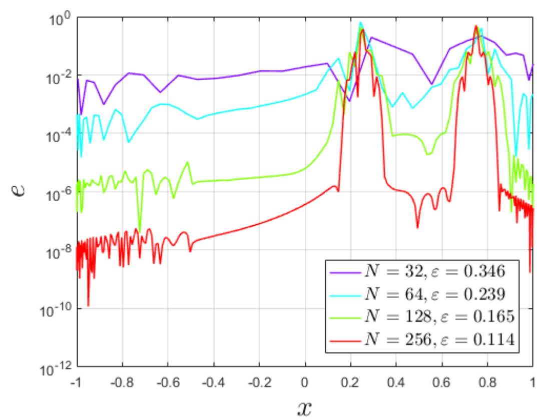


(b)

Figure 3.5: Time evolving numerical solution (a) and error (b) for $N = 256$, $m = 5$, $k = 8$, $\varepsilon = 0.114$



(a)



(b)

Figure 3.6: Numerical solution (a) and error (b) at $t = 1$, for the four different grids for $m = 5$, $k = 8$

3.2.4 L2 error norm convergence

In order to check the convergence outside the area with the discontinuity, the L_2 error is determined at $t = 1$ on the domain $-1 < x < -0.556$ for the three different cases. The L_2 error-norm (for the whole domain) is defined as:

$$L_2 = \sqrt{\frac{1}{N}\sum_{i=0}^{N}(u_{i,exact} - u_{i,numerical})^2} \tag{3.6}$$

The results are shown in figure 3.7. For reference, lines are plotted to show the order of convergence.

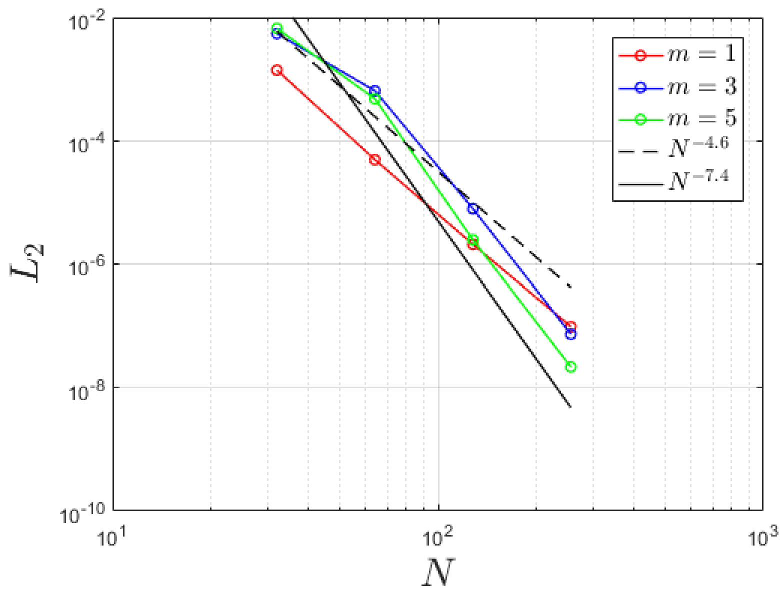


Figure 3.7: L_2 error norm convergence results for $-1 < x < -0.556$, top-hat initial condition

3.3 1D Advection Equation, sinus with discontinuity

The 1D advection equation with a sinus with discontinuity as initial condition on the domain $-1 < x < 1$ is defined as follows:

$$\begin{aligned}
\frac{\partial u}{\partial t} + \frac{\partial u}{\partial x} &= 0 \\
u(x,0) &= sin(\pi x) - 0.5 \qquad x \leq -0.25 \\
&= sin(\pi x) + 0.5 \qquad x > -0.25 \\
u(-1,t) &= sin(\pi(-1-t)) - 0.5
\end{aligned} \tag{3.7}$$

The analytical solution is given as:

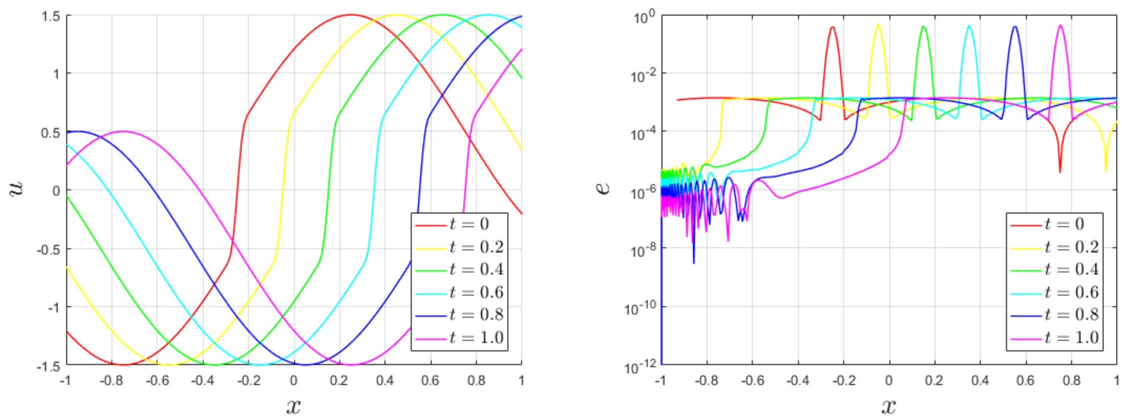
$$u = u_0(x - t) \tag{3.8}$$

The initial condition will be filtered using the filter matrix, 2.20:

$$\tilde{\boldsymbol{f}}_0 = \overline{\overline{F}}\boldsymbol{f}_0 \tag{3.9}$$

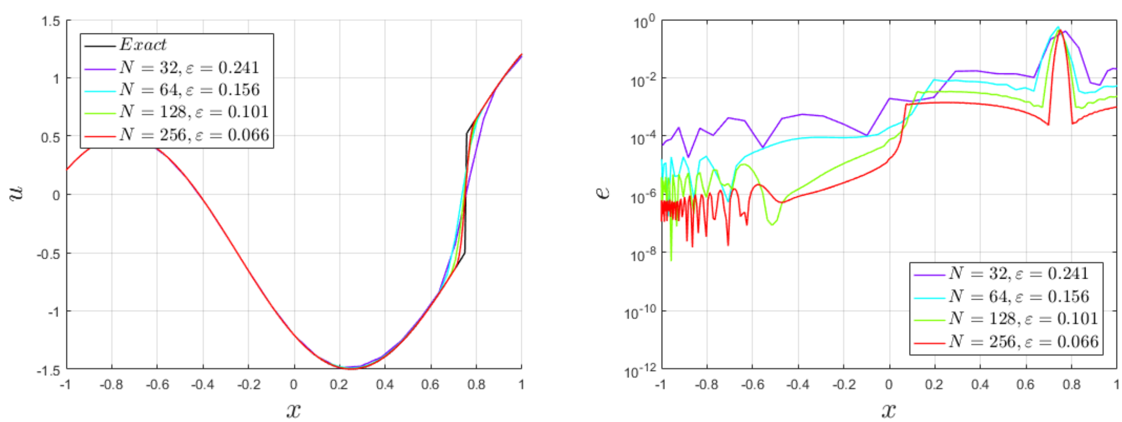
After this, the filtering isn't used again. Next, the partial differential equation will be solved using the discretizations described in 2.30 and 2.32.

3.3.1 Results, $m = 1$, $k = 5$



(a) (b)

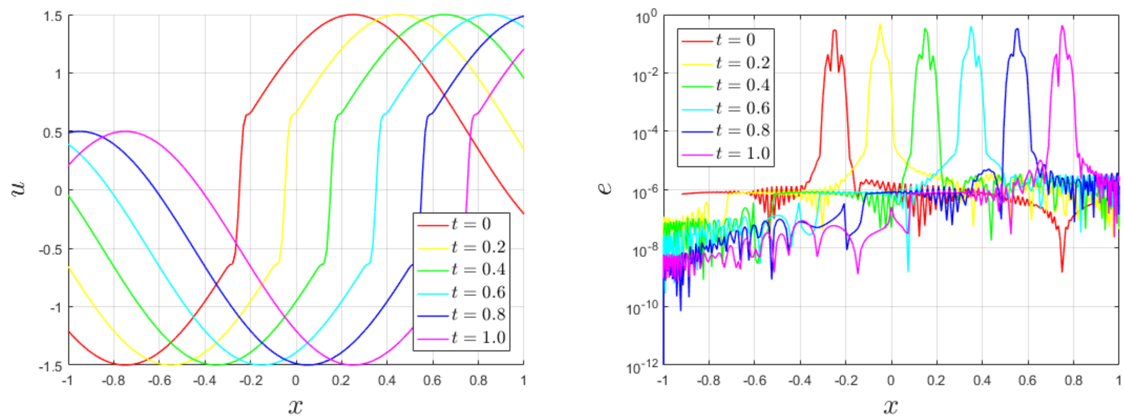
Figure 3.8: Time evolving numerical solution (a) and error (b) for $N = 256$, $m = 1$, $k = 5$, $\varepsilon = 0.066$



(a) (b)

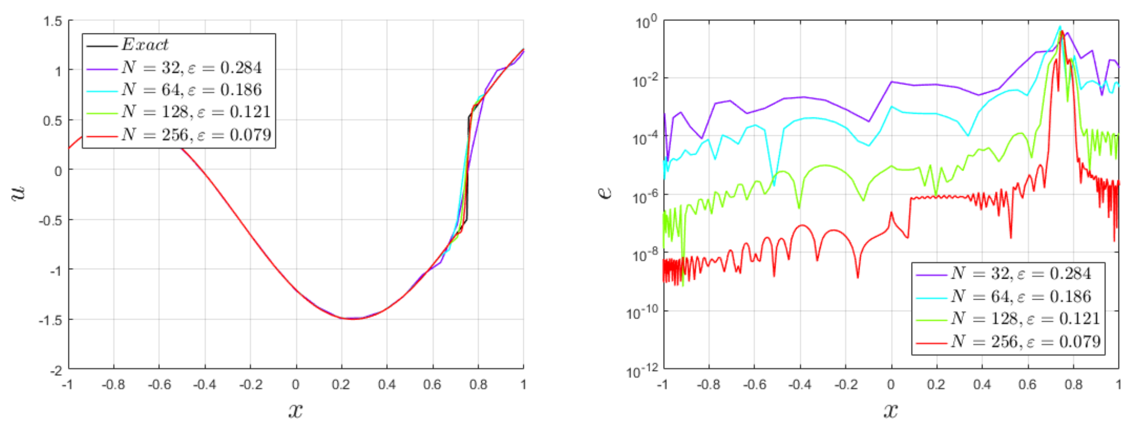
Figure 3.9: Numerical solution (a) and error (b) at $t = 1$, for the four different grids for $m = 1$, $k = 5$

3.3.2 Results, $m = 3$, $k = 8$



(a) (b)

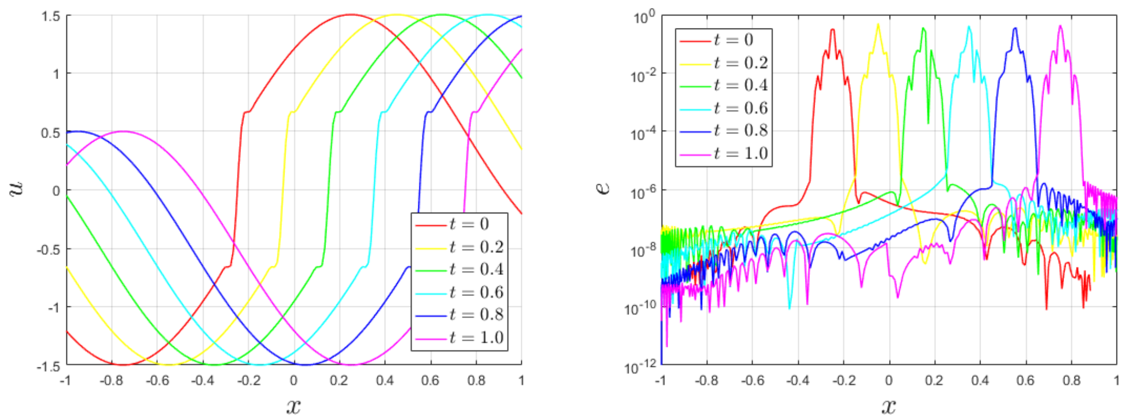
Figure 3.10: Time evolving numerical solution (a) and error (b) for $N = 256$, $m = 3$, $k = 8$, $\varepsilon = 0.079$



(a) (b)

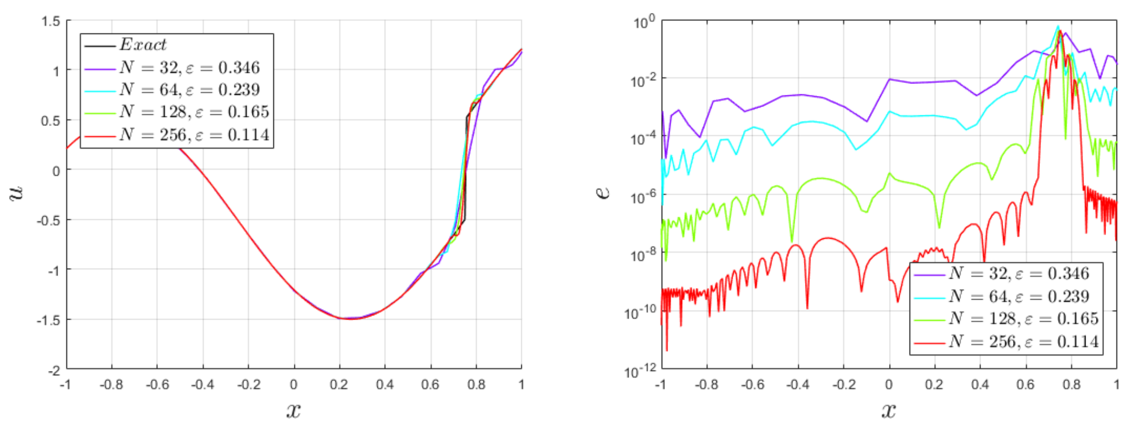
Figure 3.11: Numerical solution (a) and error (b) at $t = 1$, for the four different grids for $m = 3$, $k = 8$

3.3.3 Results, $m = 5$, $k = 8$



(a) (b)

Figure 3.12: Time evolving numerical solution (a) and error (b) for $N = 256$, $m = 5$, $k = 8$, $\varepsilon = 0.114$



(a) (b)

Figure 3.13: Numerical solution (a) and error (b) at $t = 1$, for the four different grids for $m = 5$, $k = 8$

3.3.4 L2 error norm convergence

In order to check the convergence outside the area with the discontinuity, the L_2 error is determined at $t = 1$ on the domain $-1 < x < -0.556$ for the three different cases. The L_2 error-norm (for the whole domain) is defined as:

$$L_2 = \sqrt{\frac{1}{N}\sum_{i=0}^{N}(u_{i,exact} - u_{i,numerical})^2} \tag{3.10}$$

The results are shown in figure 3.14. For reference, lines are plotted to show the order of convergence.

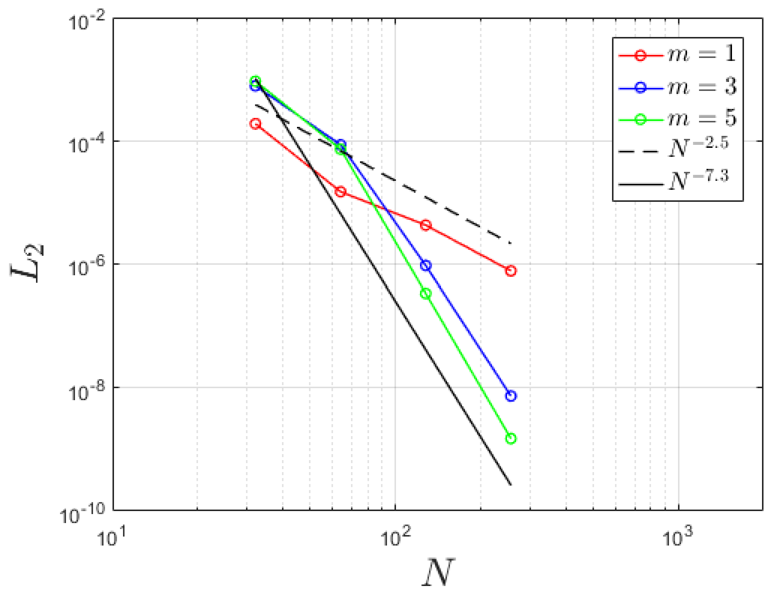


Figure 3.14: L_2 error norm convergence results for $-1 < x < -0.556$, sinus w. dicontinuity initial condition

3.4 CONCLUSIONS

In this section it is shown that high-order convergence away from the discontinuity can be obtained when filtering discontinuous initial conditions using the filter-matrix.

For the case of the top-hat initial condition the influence of m on the results is less clear. Higher values of m seem to show slightly better results, however give a slight over- and undershoot near the discontinuity and require a wider support width to give stable converging results.

In the case of the sinus with discontinuity, the influence of m is much more clear. Since the regularized delta function is a $(m+1)$th order accurate delta sequence, the error introduced on the smooth parts of the sinus are much lower for higher values of m. This can best be seen at $t = 0$, which is the filtered initial condition. Since the top-hat consists of straight lines, it does not show this behavior.

It can also be seen that the introduced error is advected and that a lower error enters the domain at the left boundary condition. This is clearest for the case $m = 1$.

CHAPTER 4

THEORETICAL FILTER-ERROR

4.1 SMOOTHING OF PIECEWISE FUNCTIONS

Given a function f, let $f_\varepsilon^{m,k}$ be the function defined by the convolution:

$$f_\varepsilon^{m,k} = (f * \delta_\varepsilon^{m,k})(x) = \int_{x-\varepsilon}^{x+\varepsilon} f(\tau)\delta_\varepsilon^{m,k}(x-\tau)d\tau \tag{4.1}$$

Then in [11], it is proven that $f^{m,k}$ converges pointwise to f as $\varepsilon \to 0$. Furthermore it is proven that:

$$f_\varepsilon^{m,k}(x) - f(x) = \mathcal{O}(\varepsilon^{m+1}) \quad \text{for} \quad a+\varepsilon < x < b-\varepsilon \tag{4.2}$$

We are interested if this convergence is retrieved when filtering a discontinuous signal using the filter-matrix previously defined, equation (2.20).

4.2 CONVOLUTION

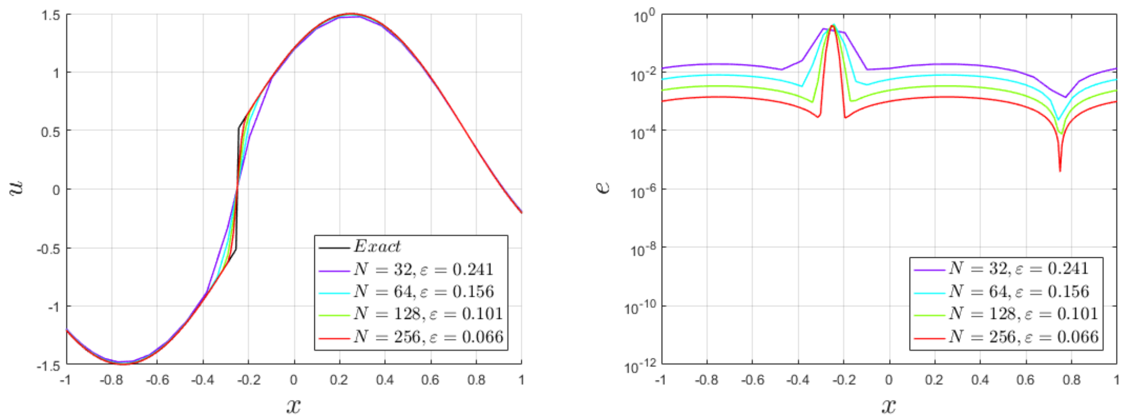
In the derivation of the filter-matrix, polynomial interpolation is used since the solution is only known on a finite amount of points. To exclude the effect of the polynomial interpolation, first the case of 'pure' convolution is considered for a sinus with a discontinuity. The signal is given by:

$$\begin{aligned} u(x) &= sin(\pi x) - 0.5 \qquad x \leq -0.25 \\ &= sin(\pi x) + 0.5 \qquad x > -0.25 \end{aligned} \tag{4.3}$$

The convolution applied to the signal u:

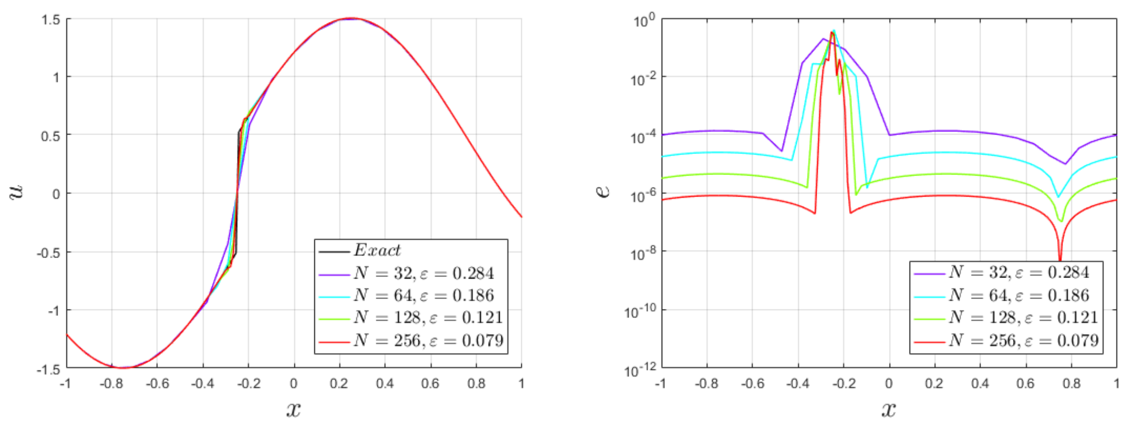
$$\tilde{u}(x) = \int_{x-\varepsilon}^{x+\varepsilon} u(\tau)\delta_\varepsilon^{m,k}(x-\tau)d\tau \tag{4.4}$$

The results are plotted for $m = 1/q = 2.1$, $m = 3/q = 2.4$ and $m = 5/q = 2.2$.



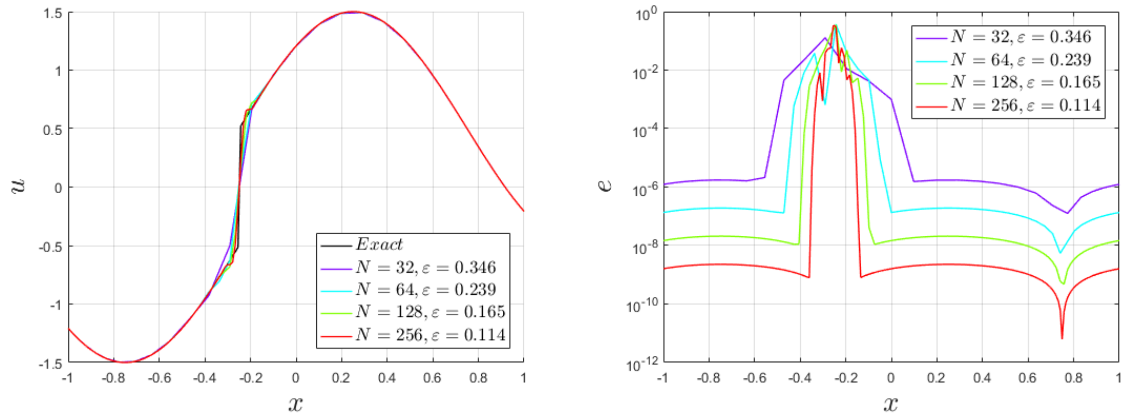
(a) (b)

Figure 4.1: Filtered signal (a) and error (b) for the four different grids, $m = 1$, $k = 5$, $q = 2.1$



(a) (b)

Figure 4.2: Filtered signal (a) and error (b) for the four different grids, $m = 3$, $k = 8$, $q = 2.4$



(a) (b)

Figure 4.3: Filtered signal (a) and error (b) for the four different grids, $m = 5$, $k = 8$, $q = 2.2$

The L_2 error results are shown below, for convenience the lines for the theoretical error convergence and lines to indicate the slope are also plotted.

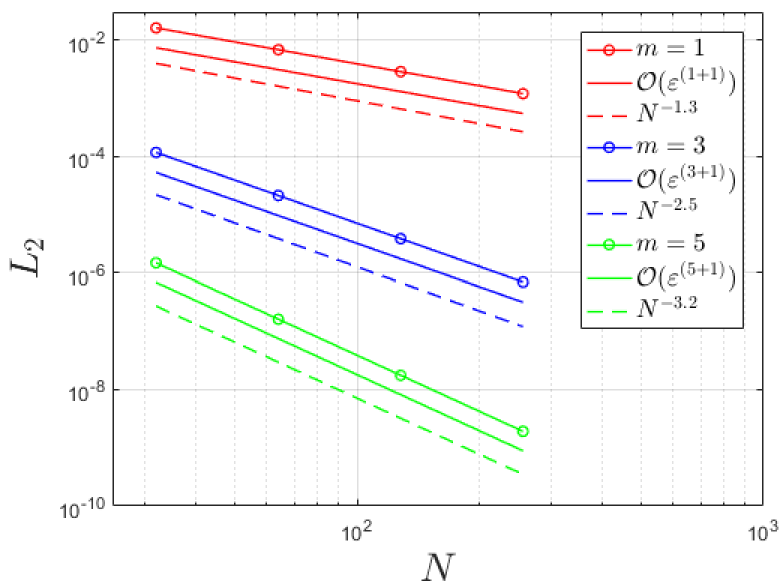


Figure 4.4: L_2 error norm convergence results for $0.098 < x < 0.3827$, sinus w. discontinuity

As can be seen from the plot, the error converges exactly according to the theory as would be expected.

4.3 Polynomial interpolation

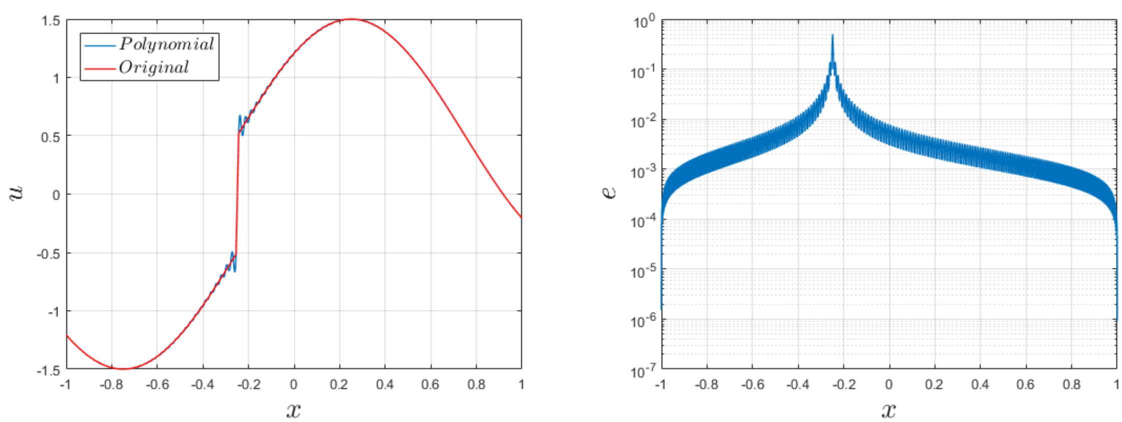
In the following the polynomial interpolation of the sinus with discontinuity is plotted. Since the filtermatrix basically applies the convolution to the polynomial interpolation of a discrete signal, it is interesting to see the result of this operation. For convenience the expressions for the polynomial interpolation are repeated, [8]:

$$f(x) = \sum_{i=0}^{N} f(x_i) l_i(x) \tag{4.5}$$

With

$$l_i(x) = \Pi_{j=0, j\neq i}^{N} \frac{x - x_j}{x_i - x_j} \qquad (0 \leq i \leq N) \tag{4.6}$$

The results are shown below.



(a) (b)

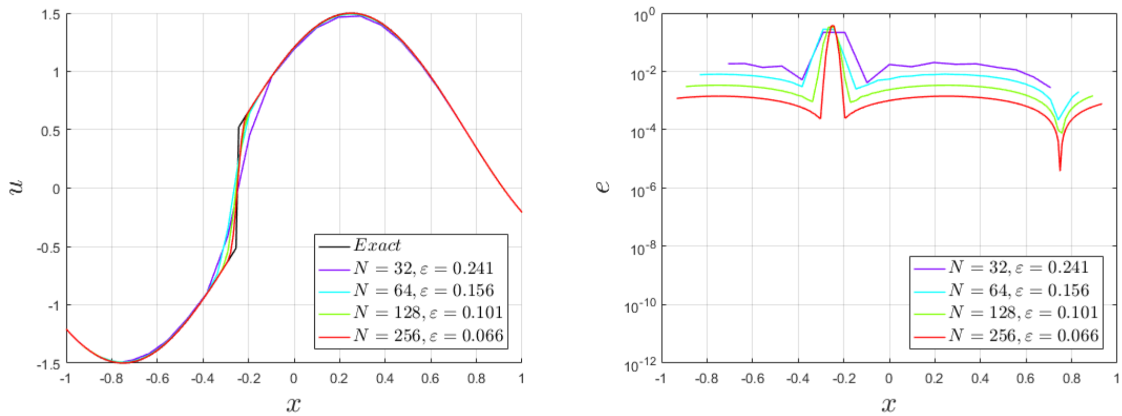
Figure 4.5: Polynomial interpolation (a) and error (b) for $N = 256$

4.4 FILTERMATRIX

In the following the error convergence of the (discrete) sinus with discontinuity that is filtered using the filtermatrix, equation (2.20), is considered. The filtered signal simply follows from:

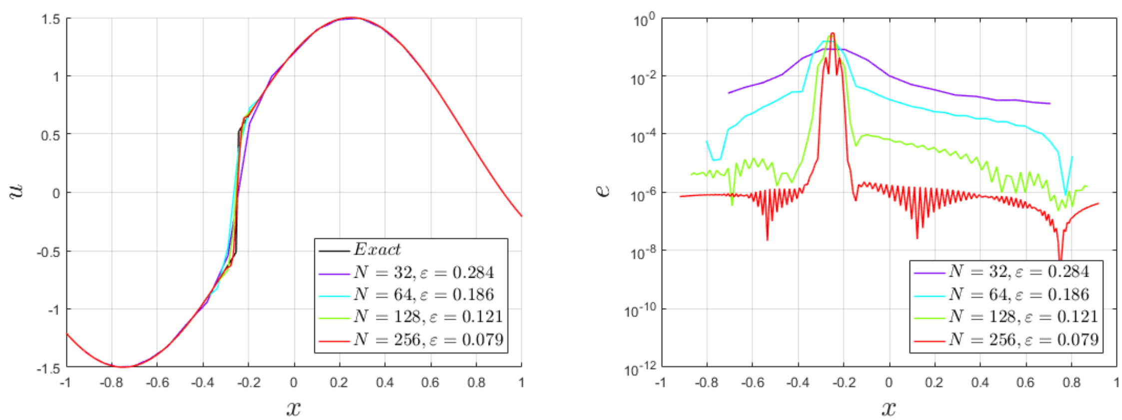
$$\tilde{\boldsymbol{u}} = \overline{\overline{F}}\boldsymbol{u} \tag{4.7}$$

Next, the results are shown for $m = 1/q = 2.1$, $m = 3/q = 2.4$ and $m = 5/q = 2.2$.



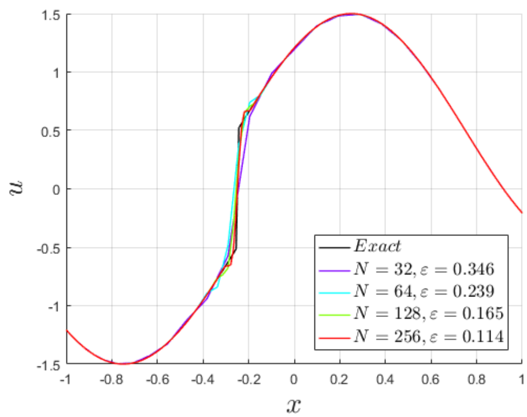
(a) (b)

Figure 4.6: Filtered signal (a) and error (b) for the four different grids, $m = 1$, $k = 5$, $q = 2.1$

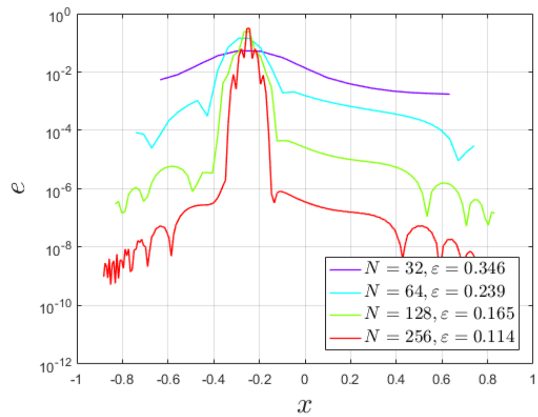


(a) (b)

Figure 4.7: Filtered signal (a) and error (b) for the four different grids, $m = 3$, $k = 8$, $q = 2.4$



(a)



(b)

Figure 4.8: Filtered signal (a) and error (b) for the four different grids, $m = 5$, $k = 8$, $q = 2.2$

The L_2 error results are shown below.

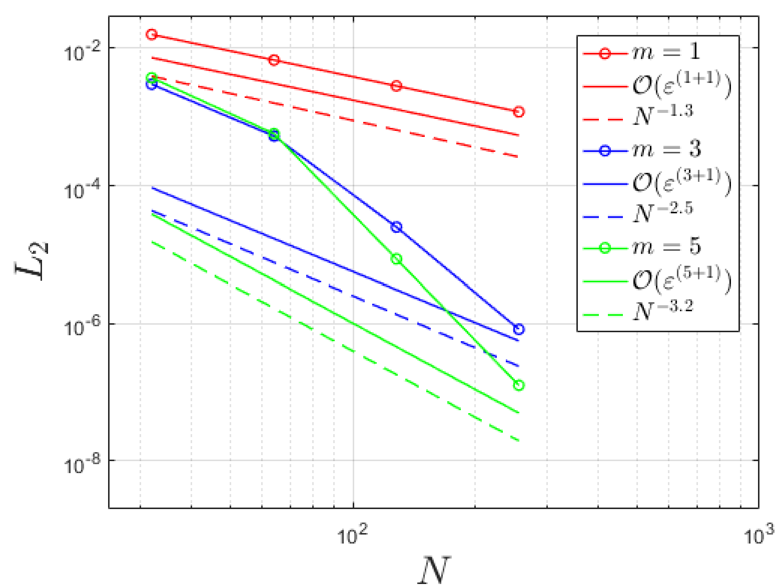


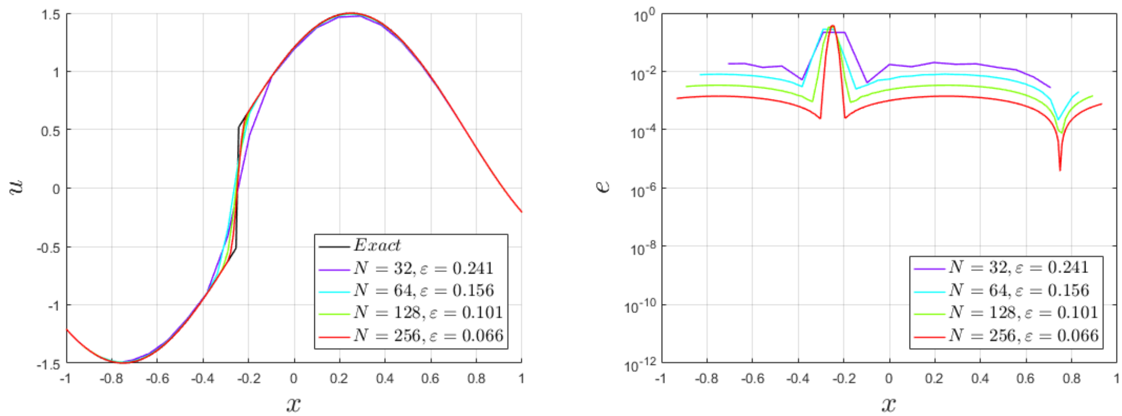
Figure 4.9: L_2 error norm convergence results for $0.098 < x < 0.3827$, sinus w. dicontinuity

From the plot it can be seen that the results are not in accordance with theory. Since the results of section 4.2 were in accordance with theory, this must have to do with the polynomial interpolation. Looking at figure 4.5 gives rise to the idea that this has to do with the oscillatory behavior of the interpolation and that a wider support width ε may give better results.

4.5 FILTERMATRIX, WIDER SUPPORT

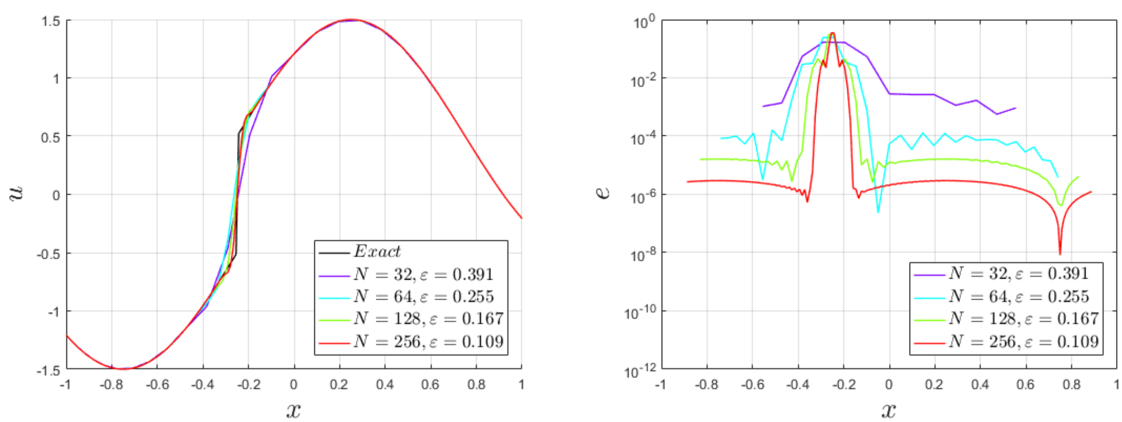
In the following the error convergence of the (discrete) sinus with discontinuity that is filtered using the filtermatrix, equation (2.20), is again considered. However, the support widths are increased.

Below, the results are shown for $m = 1/q = 2.1$, $m = 3/q = 3.3$ and $m = 5/q = 3.4$.



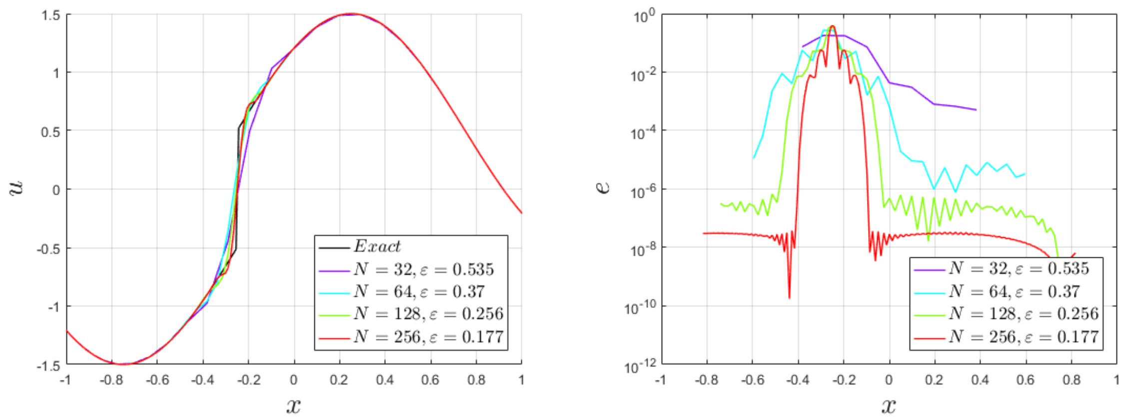
(a) (b)

Figure 4.10: Filtered signal (a) and error (b) for the four different grids, $m = 1$, $k = 5$, $q = 2.1$



(a) (b)

Figure 4.11: Filtered signal (a) and error (b) for the four different grids, $m = 3$, $k = 8$, $q = 3.3$



(a) (b)

Figure 4.12: Filtered signal (a) and error (b) for the four different grids, $m = 5$, $k = 8$, $q = 3.4$

The L_2 error results are shown below.

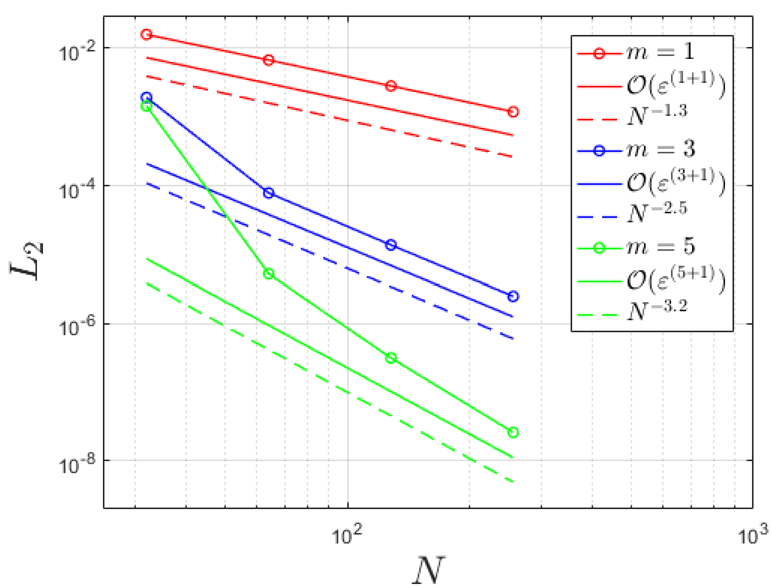


Figure 4.13: L_2 error norm convergence results for $0.098 < x < 0.3827$, sinus w. discontinuity

From this plot it follows that the error will eventually converge according to the theoretical error, provided that the support width ε is sufficiently wide.

4.6 1D ADVECTION EQUATION

In the 1D advection equation, the initial condition is simply advected with the advection speed, which also follows from the exact solution:

$$u(x,t) = u_0(x-t) \tag{4.8}$$

In case the initial condition is filtered, as was done in chapter 3, the filtered initial condition will be advected and so will be the error introduced by the filtering. Therefore, intuitively, we expect to retrieve the theoretical filter error convergence, $\mathcal{O}(\varepsilon^{m+1})$, in the solution of the 1D advection equation as well. In the following this will be proven mathematically.

If we define the linear operator L as:

$$L = \frac{\partial}{\partial t} + \frac{\partial}{\partial x} \tag{4.9}$$

Then, the 1D advection equation for the unknown u can be written as:

$$L(u(x,t)) = 0 \tag{4.10}$$

Convolution of this expression with the regularized delta function, $\delta_\varepsilon^{m,k}$, yields:

$$\int_{\tau-\varepsilon}^{\tau+\varepsilon} L(u(\tau,t))\delta_\varepsilon^{m,k}(\tau - x)d\tau = 0 \tag{4.11}$$

Since L is a linear operator, Leibniz's rule can be used to take L out of the integral:

$$L\left(\int_{\tau-\varepsilon}^{\tau+\varepsilon} u(\tau,t)\delta_\varepsilon^{m,k}(\tau - x)d\tau\right) = 0 \tag{4.12}$$

The term inside brackets is the filtered version of u, which we will write as $\tilde{u}$:

$$L(\tilde{u}(x,t)) = 0 \tag{4.13}$$

Combining equation (4.10) and equation (4.13) gives:

$$L(u(x,t)) - L(\tilde{u}(x,t)) = 0 \tag{4.14}$$

$$L(u(x,t) - \tilde{u}(x,t)) = 0 \tag{4.15}$$

$$L(error_{filter}) = 0 \tag{4.16}$$

So that the filter error indeed behaves similar to the the solution in the sense that it is also advected. Therefore the error of the 1D advection equation in which the initial condition is filtered will also converge as $\mathcal{O}(\varepsilon^{m+1})$.

The foregoing is verified with an example. Consider the problem from section 3.3:

$$\frac{\partial u}{\partial t} + \frac{\partial u}{\partial x} = 0 \tag{4.17}$$

$$u(x,0) = sin(\pi x) - 0.5 \qquad x \leq 0 \tag{4.18}$$

$$= sin(\pi x) + 0.5 \qquad x > 0 \tag{4.19}$$

$$u(-1,t) = sin(\pi(-1-t)) - 0.5 \tag{4.20}$$

The initial condition will be filtered using the filtermatrix, equation (2.20). However when using the filtermatrix, the areas near the boundaries of the domain aren't filtered. Furthermore the boundary condition isn't filtered either so that a zero-error enters the domain. This makes it impossible to retrieve the theoretical convergence.

To resolve this, the areas close to the boundaries will be filtered as if the initial condition outside the boundaries were known. This is off coarse not the case in a practical problem, so this is not possible in general. In these areas the filtered initial condition is given as:

$$\tilde{u}(x,0) = \int_{x-\varepsilon}^{x+\varepsilon} [sin(\pi(\tau - 0)) - 0.5]\delta_{\varepsilon}^{m,k}(x-\tau)d\tau \tag{4.21}$$

$$\text{for} \quad -1 < x < -1+\varepsilon \quad \text{and} \quad 1 < x < 1-\varepsilon \tag{4.22}$$

The problem with the boundary condition is resolved by applying the convolution operation to the boundary condition, yielding a 'filtered' boundary condition:

$$\tilde{u}(-1,t) = \int_{-1-\varepsilon}^{-1+\varepsilon} [sin(\pi(\tau - t)) - 0.5]\delta_{\varepsilon}^{m,k}(-1-\tau)d\tau \tag{4.23}$$

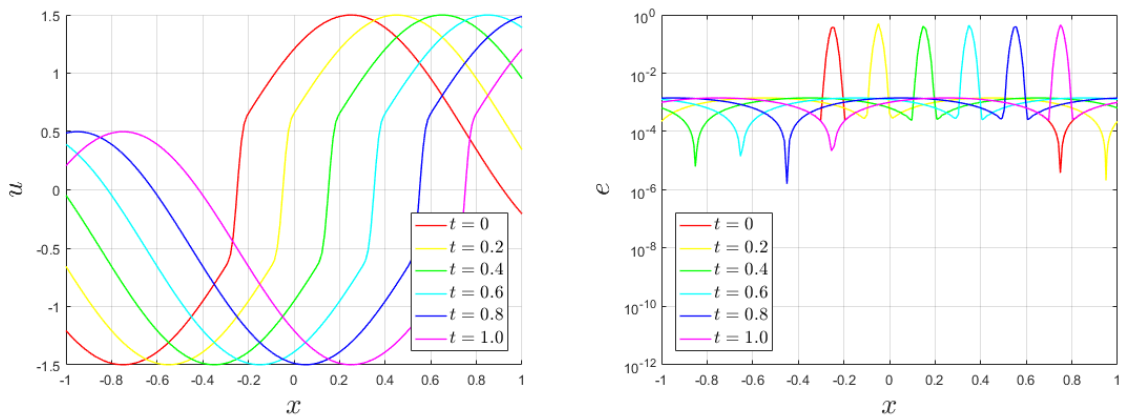
These integrals can again be solved using Clenshaw-Curtis quadrature as was done with the filtermatrix.

Using the aforementioned artificial 'tricks', the 1D advection equation with a discontinuous sinus as initial condition is solved again to see whether the theoretical filter error is retrieved. The support widths are chosen the same as the ones in section 4.5. For convenience the variables used are summarized below.

		$N = 32$	$N = 64$	$N = 128$	$N = 256$
$m = 1, k = 5$	$q = 2.1$	$\varepsilon = 0.241$	$\varepsilon = 0.156$	$\varepsilon = 0.101$	$\varepsilon = 0.066$
$m = 3, k = 8$	$q = 3.3$	$\varepsilon = 0.391$	$\varepsilon = 0.255$	$\varepsilon = 0.167$	$\varepsilon = 0.109$
$m = 5, k = 8$	$q = 3.4$	$\varepsilon = 0.535$	$\varepsilon = 0.370$	$\varepsilon = 0.256$	$\varepsilon = 0.177$
Timestep	$\Delta t = 0.00001$				

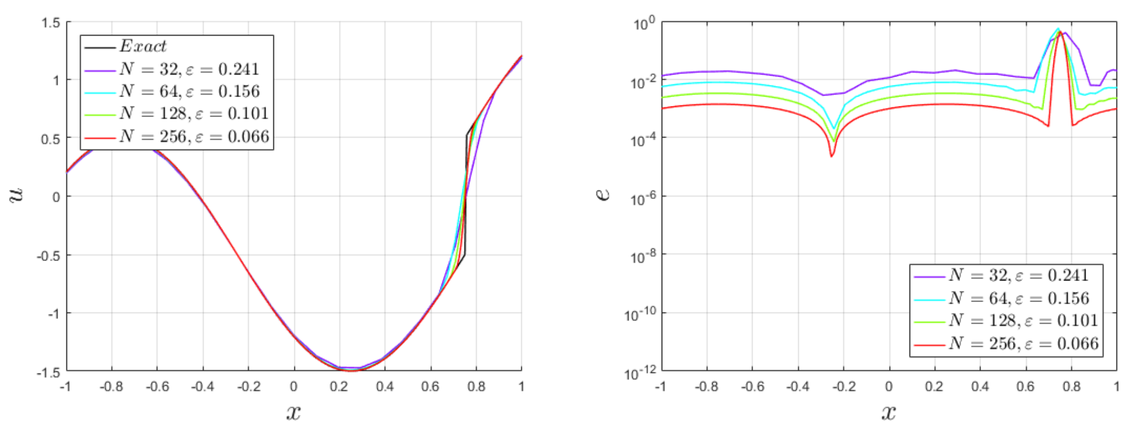
Table 4.1: Variables used in numerical experiments

4.6.1 Results, $m = 1$, $k = 5$



(a) (b)

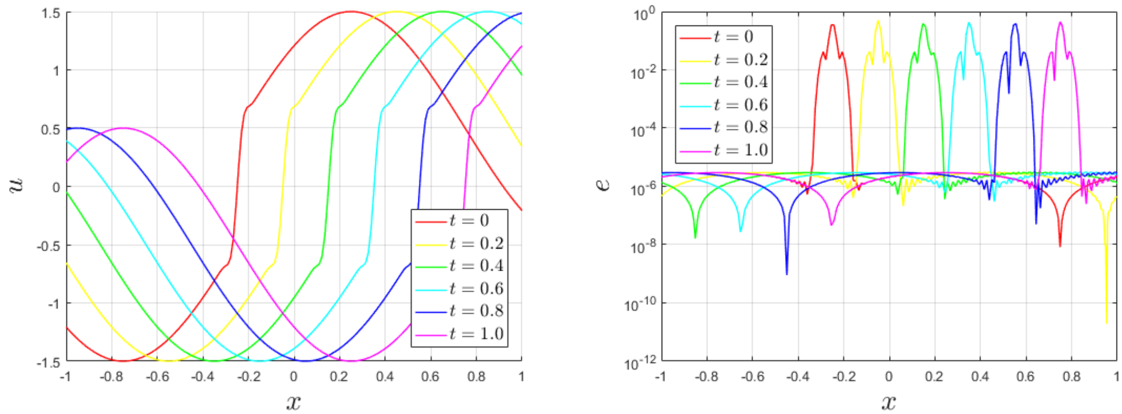
Figure 4.14: Time evolving numerical solution (a) and error (b) for, $N = 256$, $m = 1$, $k = 5$, $q = 2.1$ and $\varepsilon = 0.066$



(a) (b)

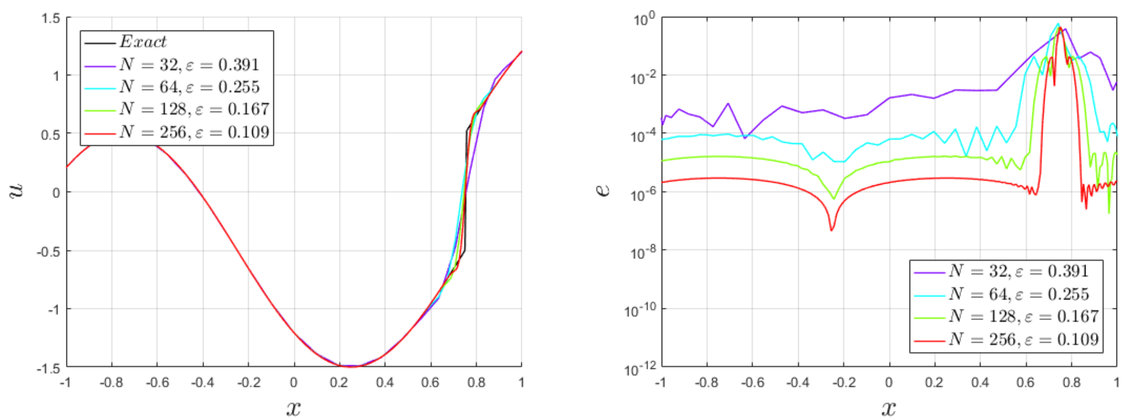
Figure 4.15: Numerical solution (a) and error (b) for $t = 1$ for the four different grids, $m = 1$, $k = 5$, $q = 2.1$

4.6.2 Results, $m = 3$, $k = 8$



(a) (b)

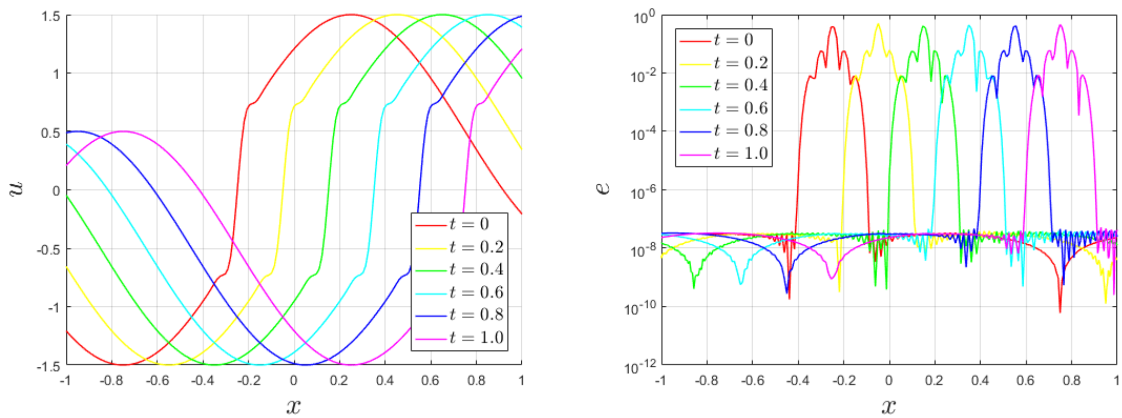
Figure 4.16: Time evolving numerical solution (a) and error (b) for, $N = 256$, $m = 3$, $k = 8$, $q = 3.3$ and $\varepsilon = 0.109$



(a) (b)

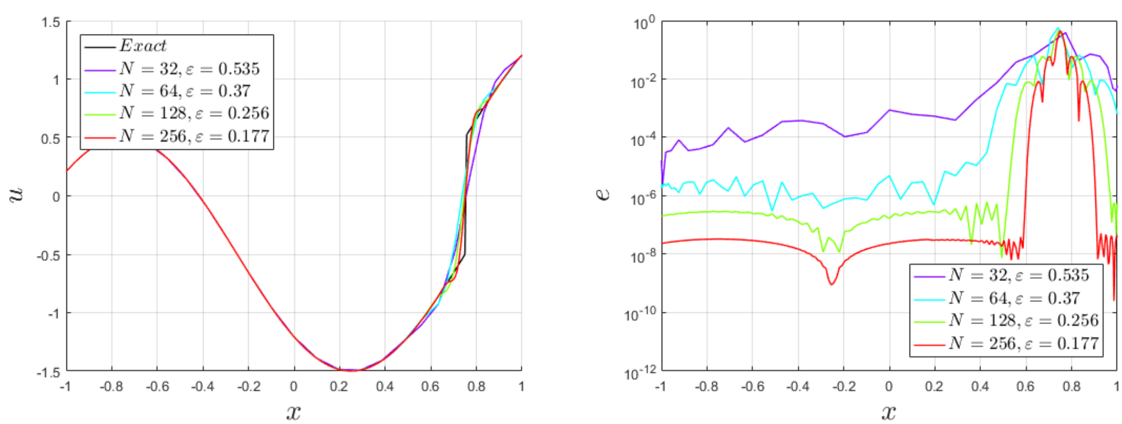
Figure 4.17: Numerical solution (a) and error (b) for $t = 1$ for the four different grids, $m = 3$, $k = 8$, $q = 3.3$

4.6.3 Results, $m = 5$, $k = 8$



(a) (b)

Figure 4.18: Time evolving numerical solution (a) and error (b) for, $N = 256$, $m = 5$, $k = 8$, $q = 3.4$ and $\varepsilon = 0.177$



(a) (b)

Figure 4.19: Numerical solution (a) and error (b) for $t = 1$ for the four different grids, $m = 5$, $k = 8$, $q = 3.4$

4.6.4 L2 error norm convergence

In order to check the convergence outside the area with the discontinuity, the L_2 error is determined at $t = 1$ on the domain $-1 < x < 0.2903$ for the three different cases.

The L_2 error results are shown below.

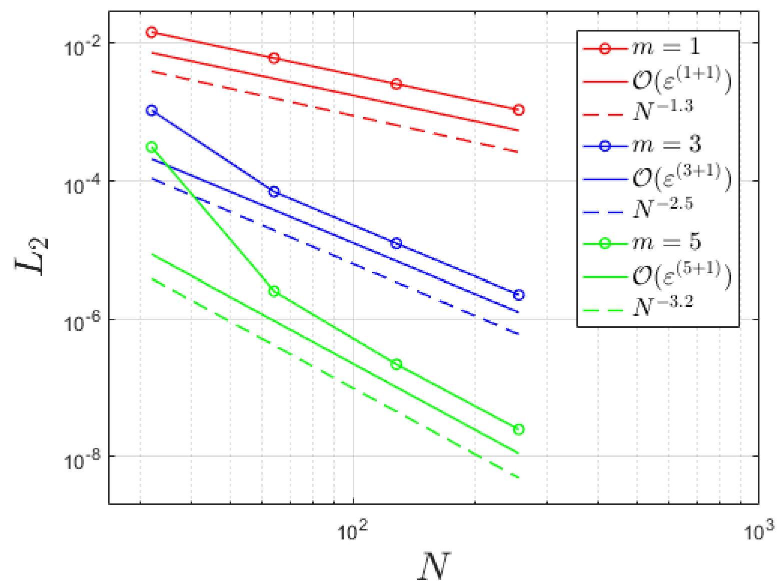


Figure 4.20: L_2 error norm convergence results for $-1 < x < 0.2903$, sinus w. discontinuity

From the previous plot it follows that the theoretical error convergence $\mathcal{O}(\varepsilon^{m+1})$ is indeed retrieved as would be expected.

4.7 Conclusions

In section 4.2 it is shown that the error of a signal that is convoluted with the regularized delta function, converges according to the theoretical filter error.

Then in section 4.3, section 4.4 and section 4.5 it is shown that this is also the case when the polynomial interpolation of the signal is convoluted with the regularized delta function, i.e. using the filter matrix with sufficiently wide support widths to suppress the influence of the polynomial interpolation.

Finally, in section 4.6 it is proven that the 1D advection equation should also converge according to the theoretical filter error. This is confirmed in an experiment with filtered boundary conditions and wide support widths.

CHAPTER 5
CONCLUSIONS AND FUTURE WORK

In this present research, the high-order Dirac-delta function presented in [11] is used to smoothen discontinuous initial conditions for the one-dimensional advection equation that is solved using the spectral collocation method. The filtering is based on convolution of the polynomial interpolation of the initial condition with the regularized Dirac-delta function. This operation is written as a matrix vector multiplication using a so-called filter-matrix.

The one-dimensional advection equation is solved for two different filtered initial conditions using different variables for the Dirac-delta function. High-order convergence is found outside the regularization zone. Higher values of m give higher convergence especially for the case of the sinus with discontinuity, however they require a wider support width to yield stable converging results.

Finally, the error is compared with the theoretical error. It is shown that in the case of a sufficiently wide support width and filtered boundary conditions, the theoretical value is retrieved.

For future work it is suggested that the application of filtering based on the convolution with the high-order Dirac-delta function on non-linear equations, for instance the Burgers' equation and the Euler equations of gasdynamics is investigated. Furthermore the extension to higher dimensions should be considered as well.

BIBLIOGRAPHY

[1] *Clenshaw curtis quadrature*. Wikipedia, https://en.wikipedia.org/wiki/Clenshaw-Curtis quadrature, accessed March 2016.

[2] *Richtmyer-meshkov instability*. Wikipedia, https://en.wikipedia.org/wiki/Richtmyer-Meshkov instability, accessed September 2016.

[3] *High order regularization of dirac-delta sources in two space dimensions*. Internship report, Wouter de Vries March 2015.

[4] G. JACOBS AND W.-S. DON, *A high-order weno-z finite difference scheme based particle-source-in-cell method for computation of particle-laden flows with shocks*, J. Comput. Phys., 228 (2009), pp. 1365–1379.

[5] S. G. JAN HESTHAVEN AND D. GOTTLIEB, *Spectral Methods for Time-Dependent Problems*, Cambridge University Press, Cambridge, United Kingdom.

[6] J. JUNG, *A note on the spectral collocation approximation of some differential equations with singular source terms*, J. Sci. Comput., 39 (2009), pp. 49–66.

[7] J. JUNG. AND W.-S. DON, *Collocation methods for hyperbolic partial differential equations with singular sources*, Adv. Appl. Math. Mech., 1 (2009), pp. 769–780.

[8] D. KINCAID AND W. CHENEY, *Numerical Analysis, Mathematics of Scientific Computing*, Brooks/Cole Publishing Company, Pacific Grove, California, 1991.

[9] R. J. LEVEQUE, *Finite-Volume Methods for Hyperbolic Problems*, Cambridge University Press, Cambridge, United Kingdom, 2004.

[10] C. SEGAL, *The scramjet engine; processes and characteristics*, University of Cambridge, Cambridge, 2009.

[11] J. P. S. SOLANO, *Regularization of Singular Sources for PSIC Computations of Particle-Laden Flows with Shocks*, PhD thesis, San Diego State University, San Diego, CA, 2015.

[12] L. N. TREFETHEN, *Spectral Methods in Matlab*, 2000.