Numerical Modelling

Semester Report

Antonio Peters

July 23, 2016

Contents

List of Figures

Chapter 1

Introduction and Preliminaries

This report will detail the work done for the Numerical Modelling mathematics Honours course. Whenever possible, decimal values will be calculated to double precision. All figures in this report will be generated in MATLAB, with the code written by the author, Antonio Peters. The code, as well as the LaTeX files for this report can be found at `https://github.com/caramel-koala/Numerical-Modelling`. The report will build up from analyzing the finite difference method, error approximation and convergence, and boundary conditions to understanding time evolutions and the Riemann problem. We will then look at Flux Conservative and Reconstruction Algorithms. We will finally look at Slope Limiter Methods and use these to simulate high resolution shock capturing in order to solve Burgers' equation.

Chapter 2

Finite Difference

There are several different variants of the finite difference methods (FDMs). This is dependent on the stencil, i.e. the number of points being used to determine a single value. The use of a larger stencil yields a lower error as can be seen in Figure 2.1, but also requires more computation time to generate a result.

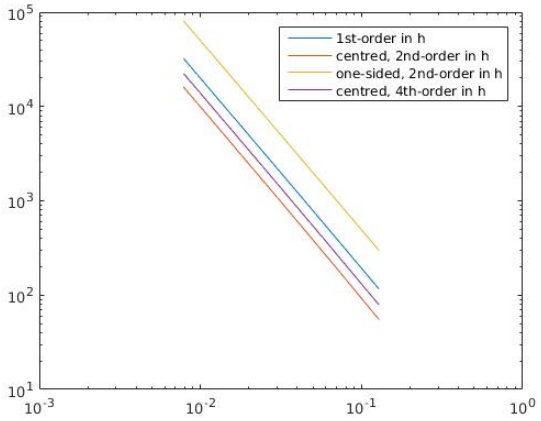


Figure 2.1: Comparison of Errors in FDM variants in log plot

2.1 First Derivatives

The FDM uses Taylor expansion to approximate the derivative of a function of finite points using the surrounding points such that

$$dv_i \approx \frac{df(x_i)}{dx} \tag{2.1}$$

The most basic of these is the first order FDM, given by Equation 2.2. It is a first order approximation as the error in the function is relative to the spacing in the first order, $\epsilon \approx h$.

$$dv_i = \frac{v_{i+1} - v_i}{h} \tag{2.2}$$

Equation 2.3 has a centered stencil and uses two points giving it a lower error, $\epsilon \approx h^2$.

$$dv_i = \frac{v_{i+1} - v_{i-1}}{2h} \tag{2.3}$$

Equation 2.4 uses a three point stencil, but due to the fact that the stencil is one-sided it is still of the second order, $\epsilon \approx h^2$.

$$dv_i = \frac{4v_{i+1} - v_{i+2} + 3v_i}{2h} \tag{2.4}$$

Equation 2.5 uses a four point, centered stencil to produce a fourth order approximation, $\epsilon \approx h^4$. As seen in Figure 2.1, this is the most accurate as it has the lowest error and the steepest gradient such that a smaller spacing gives a much higher reduction in error compared to that of the other stencils.

$$dv_i = \frac{-v_{i+2} + 8v_{i+1} - 8v_{i-1} + v_{i-2}}{12h} \tag{2.5}$$

2.2 Second Derivatives

Second derivatives are calculated in a similar manner to that of the first order and also have several variants depending on the stencil. It also uses the Taylor expansion to approximate the second derivative at a point such that

$$d^2v_i \approx \frac{df^2(x_i)}{dx^2} \tag{2.6}$$

The second ordered, centered stencil, seen in Equation 2.7, is second order in h, which again means the error is relative to the spacing or $\epsilon \approx h^2$.

$$d^2v_i = \frac{v_{i+1} - 2v_i + v_{i-1}}{h^2} \tag{2.7}$$

The fourth ordered, centered stencil, seen in Equation 2.8, is more accurate than the second ordered, with $\epsilon \approx h^4$, but due to how complex it is compared to the second order, it is more expensive to calculate.

$$d^2v_i = \frac{-v_{i+2} + 16v_{i+1} - 30v_i + 16v_{i-1} - v_{i-2}}{12h^2} \tag{2.8}$$

2.3 Ghost Points

The problem with FDMs, especially with those of a higher order, is their reliance on points on either side of the point being calculated. This is impossible closer to the function boundaries as the points needed to calculate the derivatives at these points are not within the bounds of the function. This can be overcome by the introduction of ghost points - points outside of the bounds of the function, but which are still usable for the calculation of derivatives. These points are usually generated by treating the function as periodic over its domain and extending it as such, or by setting the points to zero. Both cases could lead to large errors in the approximation of the functions derivative if the points do not align with the actual points of the function at those positions.

Chapter 3

Time Integration

In this chapter, the evolution of an equation will be analyzed by looking at the time evolutions of two Partial Differential Equations (PDEs), namely the Advection Equation (Equation 3.1) and the Wave Equation (Equation 3.2).

$$\begin{aligned} u_t + cu_x = 0, \qquad & c \in \mathbb{R} \\ u(x,0) = u_0(x) & \end{aligned} \tag{3.1}$$

$$\begin{aligned} u_{tt} - c^2 u_{xx} = 0, \qquad & c \in \mathbb{R} \\ u(x,0) = u_0(x), \qquad & u_t(x,0) = p_0(x) \end{aligned} \tag{3.2}$$

3.1 Characteristic Structure

Characteristics are a way of simplifying an equation by changing the coordinates in which the PDE is calculated. It also shows how the information of the PDE propagates with time.

3.1.1 Advection Equation

The Advection Equation is a first order PDE (meaning the highest order derivative is in the first order). This means we need only specify one characteristic, p. Where $x = x(p)$ and $t = t(p)$ therefore $u = u(x,t) = u(x(p),t(p)) = u(p)$.

By the method of characteristics we can determine that the characteristic equations for the Advection equation are

$$\frac{dx}{dp} = c, \qquad \frac{dt}{dp} = 1, \qquad \frac{du}{dp} = 0 \tag{3.3}$$

which are then integrated to give

$$x = cp + C_1, \qquad t = p + C_2, \qquad u = C_3 \tag{3.4}$$

By setting $p = 0$ at $t = 0$, C_2 is set to 0 and $t = p$ everywhere. Setting $x = p$ along $u_0(x)$ gives us

$$x = cp + q \qquad \text{and} \qquad u = u_0(q). \tag{3.5}$$

$x = cp + q$ and $t = p$ can be used to give us the characteristic coordinate

$$q = x - ct \tag{3.6}$$

which is a line with a slope of c in the $(x,t) - plane$. Substituting this into u_0 gives us

$$u = u_0(x - ct) \tag{3.7}$$

3.1.2 Wave Equation

The Wave Equation is a second order PDE, therefore it has two characteristics: p and q. By reducing the equation to its canonical form, $u_{pq} = 0$, we see that

$$p = x - ct \qquad \text{and} \qquad q = x + ct \tag{3.8}$$

Since $u_{pq} = u_{qp} = 0$ then it follows that $u_p = P(p)$. Let $F(p)$ be an anti-derivative of P. Therefore $\frac{dF}{dp}(p) = P(p)$ and therefore

$$\frac{\partial}{\partial p}(u - F(p)) = 0 \tag{3.9}$$

by integrating both sides, it then follows that

$$u - F(p) = G(q) \tag{3.10}$$

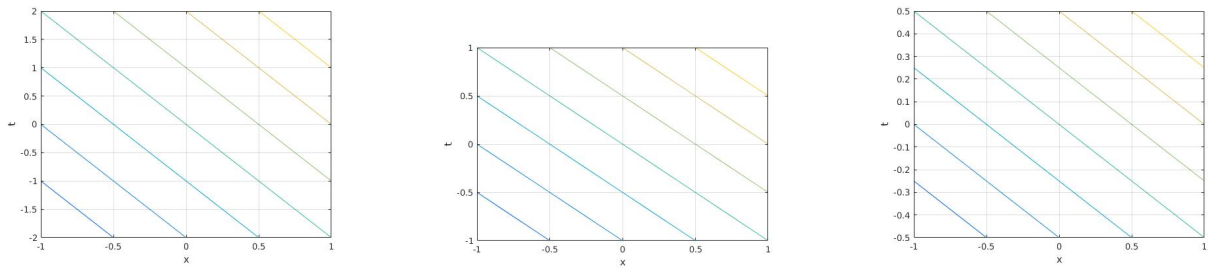
and therefore, by rearranging and substituting Equation 3.8, we get

$$u(x,t) = F(x - ct) + G(x + ct) \tag{3.11}$$

By analyzing this solution and comparing it to that of the Advection Equation we see that we in fact get two characteristic coordinates which are lines with slopes of $\pm c$.

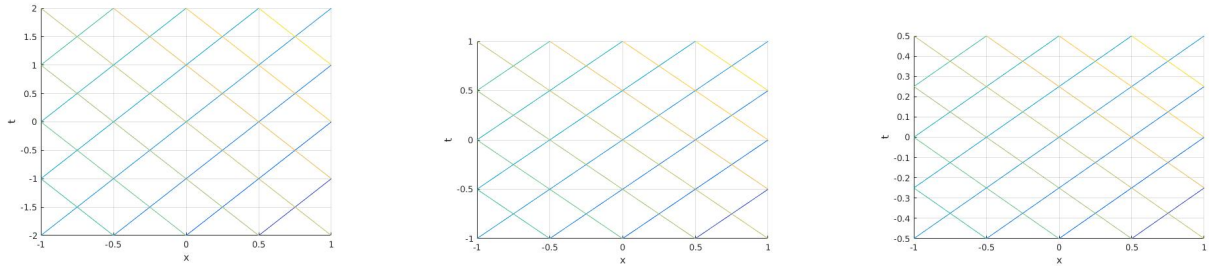
3.1.3 Spacing

Along with finding the characteristics comes the trouble of determining the spacing for the spacial and time axes when plotting. The spacial spacing will be denoted by h and time by k. If $h < k$ then the data needed to determine the value of the function at a given point in space and time will require data outside of the bounds of the stencil. Conversely if $h > k$ then data between two spacial points is needed to calculate the next value, but this can be interpolated at the cost of calculation time and may yield a higher accuracy. The standard is to set $h = k$ so that the stencil exactly fits the data needed to calculate the next point and every point being used need not be interpolated as it falls on an existing point in the spacing. This can be seen in Figure 3.1 for the Advection Equation and in Figure 3.2 for the Wave Equation.



(a) $h < k$ (b) $h = k$ (c) $h > k$

Figure 3.1: Advection characteristic



(a) $h < k$ (b) $h = k$ (c) $h > k$

Figure 3.2: Wave Equation Charactersitics

3.2 Discretization

Using the FDM discussed in Chapter 2 we will confirm the validity of three approximations of the advection equation - the Centered Euler, Leapfrog, and Upwind Euler methods; and the Leapfrog approximation of the wave equation.

3.2.1 Advection Equation

The advection equation can be approximated with the function v where

$$v_i^j = u(x_i, t_j) \tag{3.12}$$

by using the standard FDM we get the partial derivatives

$$\partial_x v_i^j = \frac{v_{i+1}^j - v_{i-1}^j}{2h} + O(h^2) \tag{3.13}$$

$$\partial_t v_i^j = \frac{v_i^{j+1} - v_i^{j-1}}{2k} + O(k^2) \tag{3.14}$$

Leapfrog

Substituting Equation 3.13 and 3.14 into the advection equation we get

$$\partial_t v_i^j + c\partial_x v_i^j = 0 \tag{3.15}$$

However, v_i^{j+1} is unknown. So we rearrange the equation to solve for v_i^{j+1} to get

$$\begin{aligned} v_i^{j+1} &= v_i^{j-1} - 2ck\partial_x v_i^j \\ &= v_i^{j-1} - c\frac{k}{h}(v_{i+1}^j - v_{i-1}^j) + O(h^2k^2) \end{aligned}$$

This is known as the "Leapfrog" method as it requires v_i^{j-1} in order to calculate v_i^{j+1}. This can be somewhat problematic with the initial data as $v_i^0 = u_0$ and v_i^2 onwards can be calculated this way but v_i^1 cannot be found by this method. Another problem concerning this method is the decoupling of odd and even points as all even points rely solely on the odd points of the previous time step and alternately for the odd points.

Centered Euler

The Centered Euler method seeks to circumvent this problem by setting

$$\partial_t v_i^j = \frac{v_i^{j+1} - v_i^j}{k} + O(k) \tag{3.16}$$

Substituting this into the advection equation and rearranging to solve for v_i^{j+1} gives

$$\begin{aligned} v_i^{j+1} &= v_i^j - ck\partial_x v_i^j \\ &= v_i^j - c\frac{k}{2h}(v_{i+1}^j - v_{i-1}^j) + O(kh^2) \end{aligned}$$

This method uses a one-sided time step to calculate the next step. This makes it unconditionally unstable and over time it varies wildly from the actual data. Due to the error being relatively low for the first step, it is generally used to initialize the Leapfrog method.

Upwind Euler

By setting

$$\begin{aligned} \partial_x v_i^j &= \frac{v_{i+1}^j - v_i^j}{h} + O(h) \\ \partial_t v_i^j &= \frac{v_i^{j+1} - v_i^j}{k} + O(k) \end{aligned}$$

we get

$$\begin{aligned} v_i^{j+1} &= v_i^j - ck\partial_x v_i^j \\ &= v_i^j - c\frac{k}{h}(v_{i+1}^j - v_i^j) + O(kh) \end{aligned}$$

This is arguably the best method to calculate the advection equation as it uses a biased stencil to propel the data from u_0 along the t and x axes, much like the advection equation itself.

3.2.2 Wave Equation

Using the second order, second derivative FDM on u with $v_i^j = u(x_i, t_j)$ we get

$$\begin{aligned} \partial_{xx} v_i^j &= \frac{v_{i+1}^j - 2v_i^j + v_{i-1}^j}{h^2} + O(h^2) \\ \partial_{tt} v_i^j &= \frac{v_i^{j+1} - 2v_i^j + v_i^{j-1}}{k^2} + O(k^2) \end{aligned}$$

Substituting this into the wave equation we get

$$\partial_{tt} v_i^j - c^2 \partial_{xx} v_i^j = 0 \tag{3.17}$$

As with the advection equation, v_i^{j+1} is unknown. re-arranging to solve for this, we get

$$v_i^{j+1} = 2v_i^j - v_i^{j+1} + c^2\frac{k^2}{h^2}(v_{i+1}^j - 2v_i^j + v_{i-1}^j) + O(h^2k^2) \tag{3.18}$$

This is a Leapfrog method approximation similar to that of the advection equation. It however does not have the decoupling of points as this stencil uses three points from the previous time step as opposed to the two used by the advection equation. This method also has the problem of v_i^1 being needed as initial data. This can usually be solved by calculating the row by hand and then feeding it into the system as initial data.

3.3 Implementation

The initial equation

$$u(x,0) = e^{-x^2} \tag{3.19}$$

was calculated on the interval $-10 < x < 10$ and $0 < t < 20$ with 100 and 1000 data points and compared to the exact solution. The L2-norm of the error was also analyzed. For each solution the ghost points were a periodic extension of the previous step's data. The results were as follows:

3.3.1 Centered Euler Advection Equation

The exact solution to the equation is

$$u(x,t) = e^{-(x-t)^2} \tag{3.20}$$

The wild variance of the Centered Euler method can be seen after only two seconds in Figure 3.3. The L2-norm of the error exponentially increases over time as seen in Figure 3.4. The method generates high frequency noise which is amplified over time and the previous step's noise feeds into the next step to be increased again. This is due to $k = h$ and how this step ratio sits exactly on the bounds of what can be seen as a stable causal step. If $k < h$, the stability of the method would increase greatly.

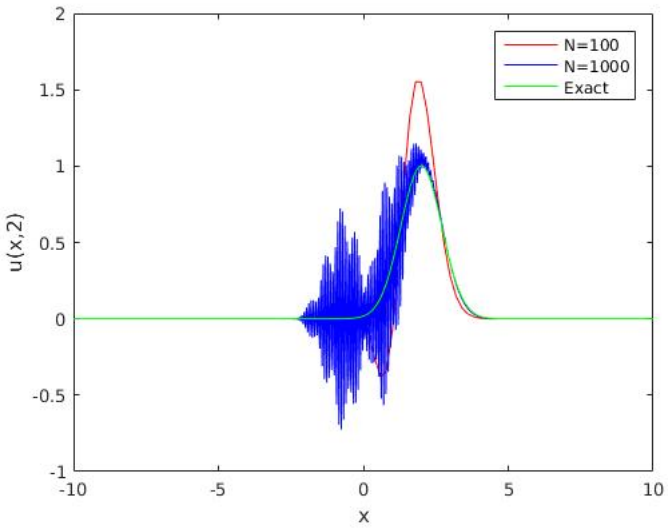


Figure 3.3: Advection Equation and Approximations after 2 seconds

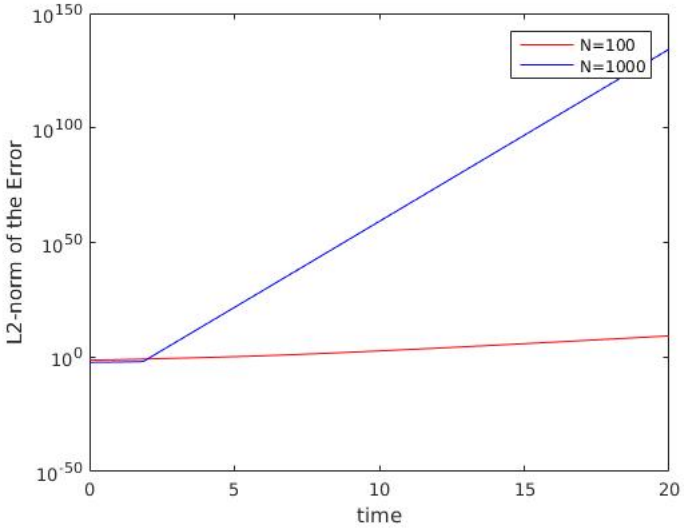


Figure 3.4: Change in L2-norm over time of Centered Euler Evolution

3.3.2 Leapfrog Advection Equation

The L2-norm of the error as seen in Figure 3.5 is due to the ghost points of the approximation being treated as a periodic extension of the function causing the function to be repeated every 20 time steps. The start of this can be seen in Figure 3.6, which shows the functions after nine time steps. The error seems, in the early steps, to be much smaller than expected: 10^{-15} instead of 10^{-4} and 10^{-8} for $N = 100$ and $N = 1000$ respectively. This is due to the ratio between the time and spacial spacing being $k = h$, and the fact that this perfectly projects the previous steps information to the next, leading to a more precise approximation.

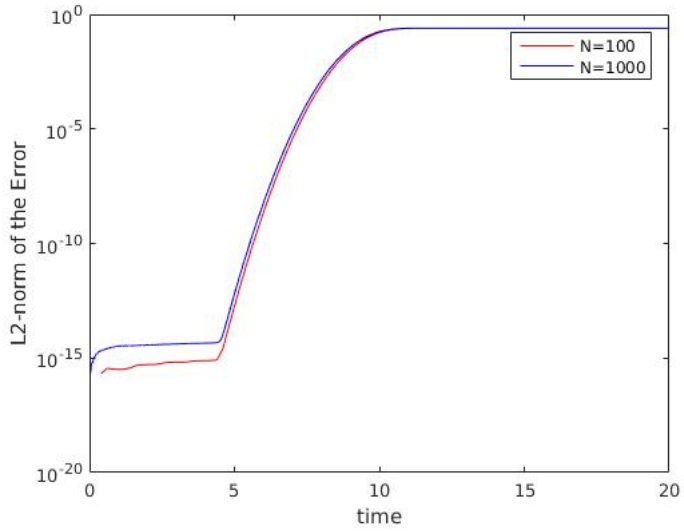


Figure 3.5: Change in L2-norm over time of Leapfrog Evolution

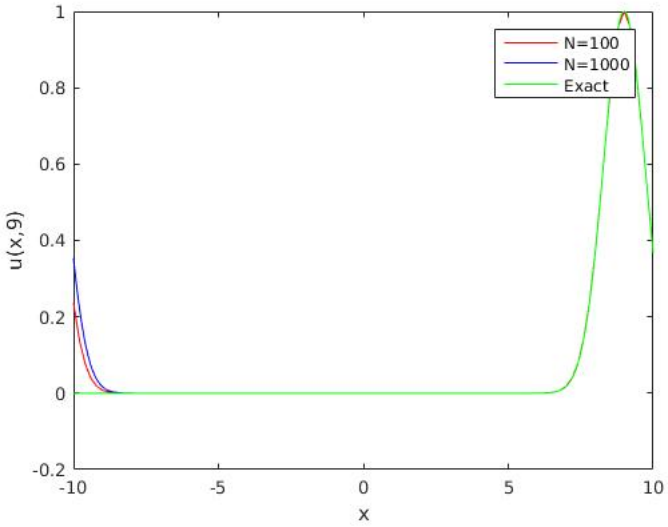


Figure 3.6: Advection Equation and Approximations after 9 seconds

3.3.3 Leapfrog Wave Equation

The d'Alembert solution to the equation is

$$u(x,t) = \frac{1}{2}(e^{-(x-t)^2} + e^{-(x+t)^2}) \tag{3.21}$$

Figure 3.10 shows how this approximation suffers from the same problem as the leapfrog advection approximation since it too has ghost points which see the function as a periodic extension and leads to the same problem of steps in the L2-norm of the error, shown in Figure 3.7. Adjusting the size of the time steps by making $k = 0.5h$ should lead to a minor reduction in the L2-norm of the error. However this is not the case as with $h = k$ the Leapfrog Wave Approximation has the same effect as that of the Leapfrog Advection Approximation where data is perfectly projected from one time step to the next leading it to be more accurate than that of a smaller time step as seen in Figure 3.8. With the large steps due to the periodic extension this is almost unnoticeable. If the time step is doubled to $k = 2h$ then the error increases exponentially over time after a relatively short period of time, as seen in Figure 3.9.

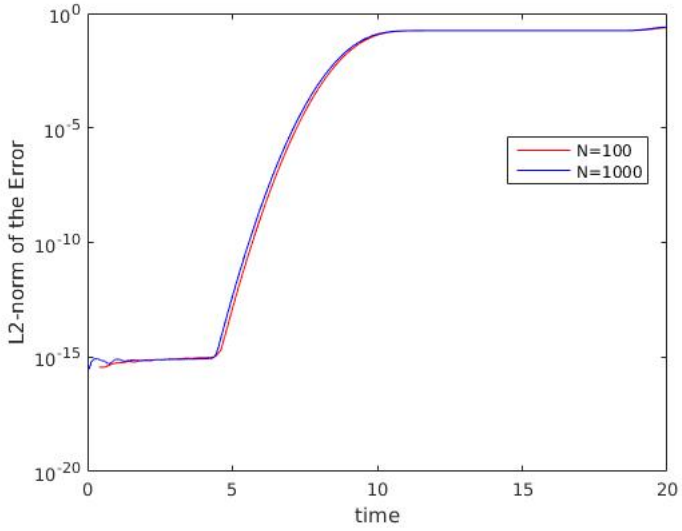


Figure 3.7: Change in L2-norm over time of Leapfrog Evolution with $k = h$

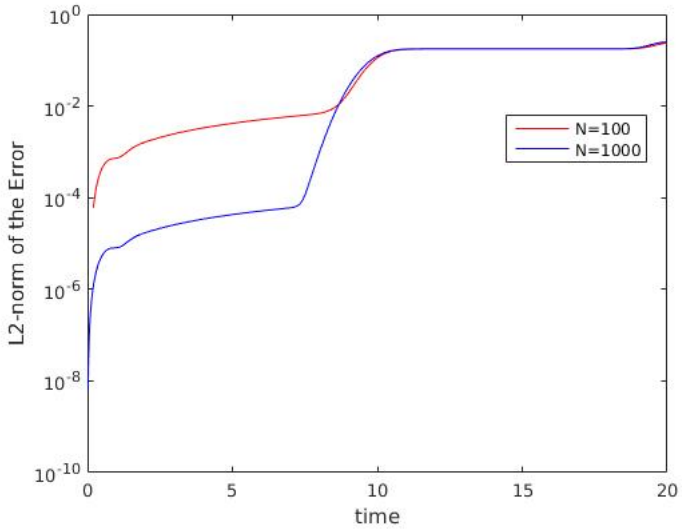


Figure 3.8: Change in L2-norm over time of Leapfrog Evolution with $k = 0.5h$

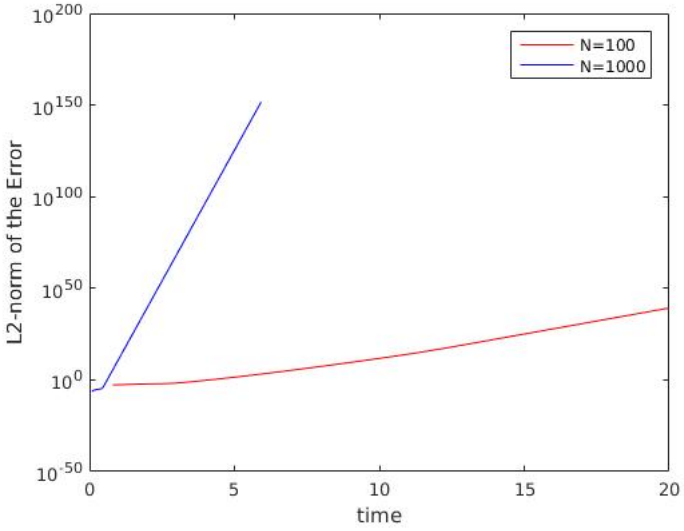


Figure 3.9: Change in L2-norm over time of Leapfrog Evolution with $k = 2h$

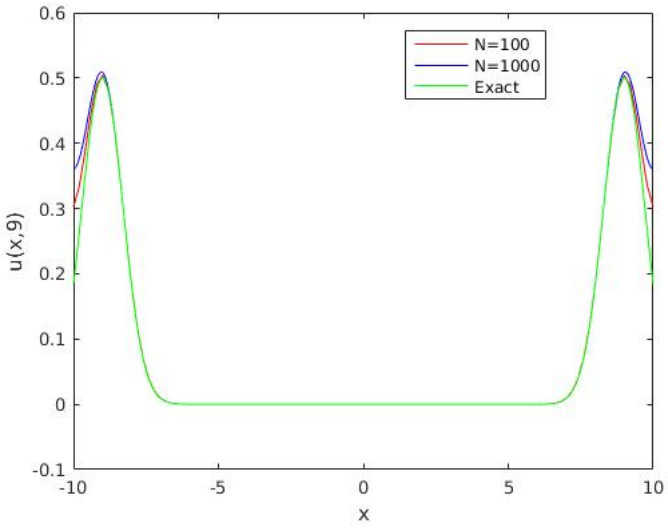


Figure 3.10: Wave Equation and Approximations after 9 seconds

By varying the spacial spacing size by setting $N = 2^i$ where $i = 6, \ldots, 12$ and fixing $k = 0.001$ we can determine the relationship between the spacing and the L2-norm of the error. Looking at this relationship at $t = 5$ we get Figure 3.11. As seen here, the logarithmic scaling of both axes produces a line graph with a slope of -2. This is due to the fact that the error of the leapfrog is proportional to h^2, which, when logarithmically scaled, goes as $2log(h)$ to produce the straight line.

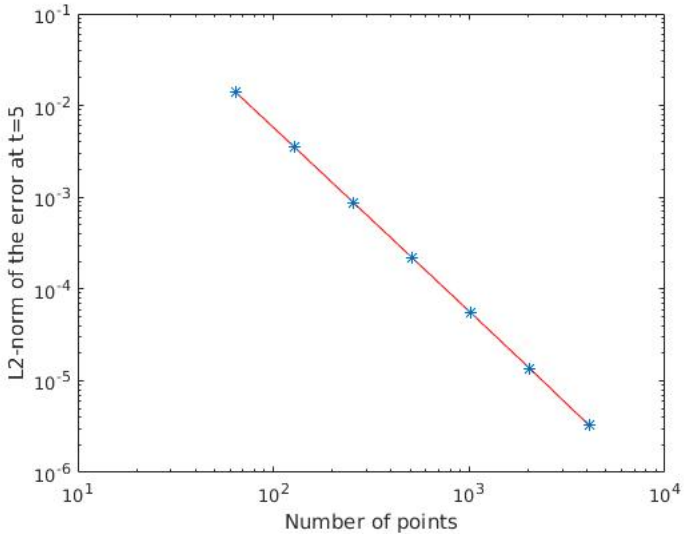


Figure 3.11: Change in the L2-norm of the error as the spacing varies

Chapter 4

Von Neumann Analysis

In this chapter we will discuss and analyze the stability of a discrete solution. We will do this by going over the Fourier transform and how it applies to the method. We will then discuss the amplification factor as well as its relation to dissipation and dispersion in a function. The analysis method will then be applied to the advection equation (Equation 4.1) and diffusion equation (Equation 4.2) and the results analyzed.

$$u_t + cu_x = 0, \qquad c \in \mathbb{R} \tag{4.1}$$

$$u_t - Ku_{xx} = 0, \qquad K \in \mathbb{R} \tag{4.2}$$

4.1 Fourier Transforms

Fourier Transforms allow us to shift an equation from one which is in the spatial domain (x) to one which is in the frequency domain (ω). The general pair of equations for this is given as:

$$\begin{aligned} \mathcal{F}[f] = \hat{f}(\omega) &= \int_{-\infty}^{\infty} e^{-i\omega x} f(x) dx \\ \mathcal{F}^{-1}[\hat{f}] = f(x) &= \int_{-\infty}^{\infty} \frac{1}{2\pi} e^{i\omega x} \hat{f}(x) dx \end{aligned}$$

The first equation is the Fourier Transform of $f(x)$ to the frequency domain and the second is its inverse back to the spatial domain.

When applied to derivatives, Fourier Transforms exhibit a useful property whereby the derivatives of the function in the spatial domain become complex polynomial scalings of the function in the frequency domain.

$$\begin{aligned} \mathcal{F}[\partial_x f] &= -i\omega x \hat{f} \\ \mathcal{F}[\partial_{xx}^2 f] &= -\omega^2 x \hat{f} \end{aligned}$$

This property gives us some insight into the nature of a function and its evolution. This can be seen by the analysis of the advection equation shown below. The magnitude of the Fourier Transform of u, $\hat{u}$, is constant

by the magnitude of the complex scalar C since $||e^{ic\omega t}|| = 1$. This indicates that it is stable as time evolves.

$$\begin{aligned}
\mathcal{F}[u_t + cu_x] &= \mathcal{F}[0] \\
\hat{u}_t - i\omega c\hat{u} &= 0 \\
\frac{\hat{u}_t}{\hat{u}} &= i\omega c \\
\hat{u} &= Ce^{ic\omega t} \\
||\hat{u}|| &= ||Ce^{ic\omega t}|| \\
&= ||C|| \cdot ||e^{ic\omega t}|| \\
&= ||C||
\end{aligned}$$

Applying the same method to the diffusion equation shows that the function is dissipative over time due to its negative exponential relationship with ω. This means that the function loses energy over time.

$$\begin{aligned}
\mathcal{F}[u_t - ku_{xx}] &= \mathcal{F}[0] \\
\hat{u}_t + \omega^2 k\hat{u} &= 0 \\
\frac{\hat{u}_t}{\hat{u}} &= i\omega c \\
\hat{u} &= Ce^{-\omega^2 kt} \\
||\hat{u}|| &= ||Ce^{-\omega^2 kt}|| \\
&= ||C|| \cdot ||e^{-\omega^2 kt}|| \\
&= ||C||
\end{aligned}$$

4.2 The Amplification Factor

We can seek a similar analysis method as the one shown in Section 4.1 for difference methods. This is done by analyzing the stability of a difference scheme approximation with a single frequency. We start by assuming the following discrete solution form

$$v_j^n = \hat{v}^n(\omega)e^{i\omega x} = \hat{v}^n(\omega)e^{i\omega hj} \tag{4.3}$$

Now, inserting this into the difference scheme, we can attempt to (and will) find a recurrence relation for $\hat{v}^n$ such that

$$\hat{v}^{n+1} = \hat{Q}\hat{v}^n \tag{4.4}$$

This allows us to simplistically analyze the evolution of a function as the nth step can be calculated as

$$\hat{v}^n = \hat{Q}\hat{v}^{n-1} = \hat{Q}^2\hat{v}^{n-2} = \cdots = \hat{Q}^n\hat{v}^0 \tag{4.5}$$

The value of $\hat{Q}$ for a specific frequency therefore determines how stable the difference scheme is.

$\hat{Q} > 1$	$\Rightarrow ||\hat{v}^n|| = ||\hat{v}^0||$	$\Rightarrow$ Solution is unstable
$\hat{Q} < 1$	$\Rightarrow ||\hat{v}^n|| < ||\hat{v}^0||$	$\Rightarrow$ Solution is stable, but dissipative
$\hat{Q} = 1$	$\Rightarrow ||\hat{v}^n|| > ||\hat{v}^0||$	$\Rightarrow$ Solution is stable

4.3 Examples

4.3.1 Centered Euler

$$v_j^{n+1} = v_j^n - c\frac{k}{2h}(v_{j+1}^n - v_{j-1}^n) + O(kh^2)$$

Substituting Equation 4.3

$$\begin{aligned}
\hat{v}^{n+1}e^{i\omega hj} &= \hat{v}^n e^{i\omega hj} - c\frac{k}{2h}(\hat{v}^n e^{i\omega h(j+1)} - \hat{v}^n e^{i\omega h(j-1)}) \\
\hat{v}^{n+1}e^{i\omega hj} &= \hat{v}^n e^{i\omega hj} - c\frac{k}{2h}e^{i\omega hj}\hat{v}^n(e^{i\omega h} - e^{-i\omega h}) \\
\hat{v}^{n+1} &= \hat{v}^n - ic\frac{k}{h}\hat{v}^n \sin\omega h \\
&= \hat{v}^n(1 - ic\frac{k}{h}\sin\omega h)
\end{aligned}$$

Now substituting Equation 4.4, we get

$$\begin{aligned}
\hat{Q}\hat{v}^n &= hatv^n(1 - ic\frac{k}{h}\sin\omega h) \\
\hat{Q} &= 1 - ic\frac{k}{h}\sin\omega h
\end{aligned}$$

Now $||\hat{Q}|| = 1$ where $||1 - ic\frac{k}{h}\sin\omega h|| = 1$. Therefore $ic\frac{k}{h}\sin\omega h$ must be zero for it to be stable, meaning $\sin\omega h$ must be zero which only occurs at $\omega = \frac{m\pi}{h}$ where $m \in \mathbb{Z}$. Since, in practice, the function is influenced by many small noisy frequencies, the solution will be unstable as $\hat{Q} > 1$ for these and their influence will compound over time due to this.

4.3.2 Upwind Euler

$$v_j^{n+1} = v_j^n - c\frac{k}{h}(v_{j+1}^n - v_j^n) + O(kh)$$

Substituting Equation 4.3

$$\begin{aligned}
\hat{v}^{n+1}e^{i\omega hj} &= \hat{v}^n e^{i\omega hj} - c\frac{k}{h}(\hat{v}^n e^{i\omega h(j+1)} - \hat{v}^n e^{i\omega hj}) \\
\hat{v}^{n+1}e^{i\omega hj} &= \hat{v}^n e^{i\omega hj} - c\frac{k}{h}\hat{v}^n e^{i\omega hj}(e^{i\omega h} - 1) \\
\hat{v}^{n+1}e^{i\omega hj} &= \hat{v}^n e^{i\omega hj}(1 - c\frac{k}{h}(e^{i\omega h} - 1)) \\
\hat{v}^{n+1} &= \hat{v}^n(1 - c\frac{k}{h}(e^{i\omega h} - 1))
\end{aligned}$$

Now substituting Equation 4.4, we get

$$\hat{Q}\hat{v}^n = \hat{v}^n(1 - c\frac{k}{h}(e^{i\omega h} - 1))$$
$$\hat{Q} = 1 - c\frac{k}{h}(e^{i\omega h} - 1)$$

Let $c\frac{k}{h} = \alpha$

$$\begin{aligned}\hat{Q} &= 1 - \alpha(e^{i\omega h} - 1)\\ &= 1 - \alpha(\cos\omega h + i\sin\omega h - 1)\\ &= 1 - \alpha(\cos\omega h - 1) - \alpha i\sin\omega h\end{aligned}$$

Now, calculating the magnitude of $\hat{Q}$, gives us:

$$\begin{aligned}\hat{Q}^2 &= (1 - \alpha(\cos\omega h - 1) - \alpha i\sin\omega h)^2\\ &= (1 - \alpha(\cos\omega h - 1))^2 - (\alpha i\sin\omega h)^2\\ &= 1 - 2\alpha(\cos\omega h - 1) + \alpha^2(\cos\omega h - 1)^2 + \alpha^2\sin^2\omega h\\ &= 1 - 2\alpha(\cos\omega h - 1) + \alpha^2\cos^2\omega h - 2\alpha^2\cos\omega h + \alpha^2 + \alpha^2\sin^2\omega h\\ &= 1 - 2\alpha(\cos\omega h - 1) + 2\alpha^2 - 2\alpha^2\cos\omega h\\ &= 1 - 2\alpha(\cos\omega h - 1 + \alpha - \alpha\cos\omega h)\\ &= 1 - 2\alpha(1 - \alpha)(\cos\omega h - 1)\end{aligned}$$

Therefore, for $||\hat{Q}|| \leq 1$, $1 - \alpha = 0$, and so:

$$\begin{aligned}1 - \alpha &\leq 0\\ 1 &\leq \alpha\\ 1 &\leq c\frac{k}{h}\\ \frac{h}{k} &\leq c\end{aligned}$$

Therefore the Upwind Euler method is stable when Courant's stability condition is met.

4.3.3 Leapfrog Advection Equation

$$v_j^{n+1} = v_j^{n-1} - c\frac{k}{h}(v_{j+1}^n - v_{j-1}^n)$$

Substituting Equation 4.3

$$\begin{aligned}\hat{v}^{n+1}e^{i\omega hj} &= \hat{v}^{n-1}e^{i\omega hj} - c\frac{k}{h}(\hat{v}^n e^{i\omega h(j+1)} - \hat{v}^n e^{i\omega h(j-1)})\\ &= \hat{v}^{n-1}e^{i\omega hj} - c\frac{k}{h}\hat{v}^n e^{i\omega hj}(e^{i\omega h} - e^{-i\omega h})\\ \hat{v}^{n+1} &= \hat{v}^{n-1} - 2ic\frac{k}{h}\hat{v}^n\sin\omega h\end{aligned}$$

Now substituting Equation 4.4 and remembering Equation 4.5 we get

$$\begin{aligned}
\hat{Q}^2\hat{v}^{n-1} &= \hat{v}^{n-1} - 2ic\frac{k}{h}\hat{Q}\hat{v}^{n-1}\sin\omega h \\
\hat{Q}^2 &= 1 - 2ic\frac{k}{h}\hat{Q}\sin\omega h \\
0 &= \hat{Q}^2 + 2ic\frac{k}{h}\sin(\omega h)\hat{Q} - 1 \\
\hat{Q} &= \frac{-2ic\frac{k}{h}\sin(\omega h) \pm \sqrt{(2ic\frac{k}{h}\sin(\omega h))^2 - 4(1)(-1)}}{2(1)} \\
&= -ic\frac{k}{h}\sin(\omega h) \pm \frac{\sqrt{4 - 4c^2\frac{k^2}{h^2}\sin^2(\omega h)}}{2} \\
&= -ic\frac{k}{h}\sin(\omega h) \pm \sqrt{1 - c^2\frac{k^2}{h^2}\sin^2(\omega h)}
\end{aligned}$$

Now, calculating the magnitude of $\hat{Q}$, gives us:

$$\begin{aligned}
\hat{Q}^2 &= \left(- ic\frac{k}{h}\sin(\omega h) \pm \sqrt{1 - c^2\frac{k^2}{h^2}\sin^2(\omega h)}\right)^2 \\
&= (-ic\frac{k}{h}\sin(\omega h))^2 + \left(\sqrt{1 - c^2\frac{k^2}{h^2}\sin^2(\omega h)}\right)^2 \\
&= c^2\frac{k^2}{h^2}\sin^2(\omega h) + 1 - c^2\frac{k^2}{h^2}\sin^2(\omega h) \\
&= 1
\end{aligned}$$

This shows that the Leapfrog is stable provided the square rooted term is real. This is only true if:

$$\begin{aligned}
c^2\frac{k^2}{h^2}\sin^2(\omega h) &< 1 \\
c^2\frac{k^2}{h^2} &< 1 \\
c\frac{k}{h} &< 1 \\
c &< \frac{h}{k}
\end{aligned}$$

Therefore the Leapfrog's stability also depends on whether or not Courant's stability condition is met.

4.3.4 Diffusion Equation

Using Equation 2.2 for the time derivative and Equation 2.7 for the second spatial derivative, we can transform the diffusion equation and solve for v_j^{n+1}. For the sake of removing confusion K will be replaced

with C.

$$\begin{aligned} u_t - Cu_{xx} &= 0 \\ \frac{v_j^{n+1} - v_j^n}{k} - C\big(\frac{v_{j+1}^n - 2v_j^n + v_{j-1}^n}{h^2}\big) &= 0 \\ v_j^{n+1} - v_j^n &= kC\big(\frac{v_{j+1}^n - 2v_j^n + v_{j-1}^n}{h^2}\big) \\ v_j^{n+1} &= v_j^n + C\frac{k}{h^2}(v_{j+1}^n - 2v_j^n + v_{j-1}^n) \end{aligned}$$

Substituting Equation 4.3:

$$\begin{aligned} \hat{v}^{n+1}e^{i\omega hj} &= \hat{v}^n e^{i\omega hj} + C\frac{k}{h^2}(\hat{v}^n e^{i\omega h(j+1)} - 2\hat{v}^n e^{i\omega hj} + \hat{v}^n e^{i\omega h(j-1)}) \\ &= \hat{v}^n e^{i\omega hj} + C\frac{k}{h^2}\hat{v}^n e^{i\omega hj}(e^{i\omega h} - 2 + e^{-i\omega h}) \\ \hat{v}^{n+1} &= \hat{v}^n + C\frac{k}{h^2}\hat{v}^n(2\cos\omega h - 2) \\ &= (1 + 2C\frac{k}{h^2}(\cos\omega h - 1))\hat{v}^n \end{aligned}$$

Now substituting Equation 4.4 we get

$$\hat{Q} = 1 + 2C\frac{k}{h^2}(\cos\omega h - 1)$$

For $\hat{Q}$ to be stable and since $\cos\omega h - 1$ varies between 0 and -2

$$\begin{aligned} 2C\frac{k}{h^2} &< 1 \\ \frac{k}{h^2} &< \frac{1}{2C} \end{aligned}$$

This shows that in order for the method to be stable for each doubling of the number of spatial points, the number of time points must be quadrupled.

Chapter 5

Time Evolutions

In this chapter we will be looking at the conservative form of an equation. An equation is said to be in it's conservative form if it can be written as:

$$u_t + F(x,u)_x = 0 \tag{5.1}$$

The advection and diffusion equations, as well as Burger's equation, can therefore be written in conservative form.

5.1 Lax-Friedrichs

The Lax-Friedrichs (LF) time evolutions scheme for conservative equation is given by:

$$v_j^{n+1} \simeq \frac{1}{2}(v_{j+1}^n + v_{j-1}^n) - \frac{k}{2h}(F_{j+1}^n - F_{j-1}^n) + O(k,h^2)$$

Focusing on the first term of the LF method, we see

$$\frac{1}{2}(v_{j+1}^n + v_{j-1}^n) = v_j^n + \frac{1}{2}(v_{j+1}^n - 2v_j^n + v_{j-1}^n)$$

If F_j^n is then set to cv_j^n, we get the centered Euler form of the advection equation with this second derivative component. This component then warps the advection equation to

$$u_t + cu_x = \frac{h^2}{2k}u_{xx}$$

This gives the centered Euler a dissipative term, $\epsilon = \frac{h^2}{2k}$. Setting $\frac{h}{k}$ to be a constant, ζ, we get

$$\epsilon = \frac{1}{2}\zeta h$$

Therefore as $h \to 0$, $\epsilon \to 0$, and the exact form of the advection equation is formed. This is known as artificial dissipation and is used to stabilize an unstable scheme. It is also known as the Kreiss-Oliger dissipation.

5.2 Lax-Wendroff

The Lax-Wendroff (LW) scheme uses the LF scheme to first obtain a "half step", which it then uses to perform a leapfrog-like step to achieve the next time step. The half steps are given as:

$$v_{j-1/2}^{n+1/2} \simeq \frac{1}{2}(v_j^n + v_{j-1}^n) - \frac{k}{2h}(F_j^n - F_{j-1}^n)$$
$$v_{j+1/2}^{n+1/2} \simeq \frac{1}{2}(v_{j+1}^n + v_j^n) - \frac{k}{2h}(F_{j+1}^n - F_j^n)$$

And the leapfrog step is given as:

$$v_j^{n+1} \simeq v_j^n - \frac{k}{h}(F_{j+1/2}^{n+1/2} - F_{j-1/2}^{n+1/2})$$

The addition of this half step improves on the LF scheme from $O(h^2, k)$ to $O(h^2, k^2)$.

For the advection equation specifically, we can derive the LW scheme with F set to cu to obtain

$$v_{j-1/2}^{n+1/2} \simeq \frac{1}{2}(v_j^n + v_{j-1}^n) - \frac{ck}{2h}(v_j^n - v_{j-1}^n)$$
$$v_{j+1/2}^{n+1/2} \simeq \frac{1}{2}(v_{j+1}^n + v_j^n) - \frac{ck}{2h}(v_{j+1}^n - v_j^n)$$

as the half steps. Therefore the full scheme is:

$$v_j^{n+1} \simeq v_j^n - \frac{ck}{h}\big(\frac{1}{2}(v_j^n + v_{j-1}^n) - \frac{ck}{2h}(v_j^n - v_{j-1}^n) - \frac{1}{2}(v_{j+1}^n + v_j^n) - \frac{ck}{2h}(v_{j+1}^n - v_j^n)\big)$$
$$v_j^{n+1} \simeq v_j^n - \frac{ck}{2h}(v_{j+1}^n - v_{j-1}^n) + \frac{c^2k^2}{2h^2}(v_{j+1}^n - 2v_j^n + v_{j-1}^n)$$

We can see that this is similar to the LF scheme for the advection equation, but with $\epsilon = c^2k^2/2$ as part of the dissipative term. Unlike the LF scheme however, the dissipative term is not dependent on h which would result in a fixed ζ, but rather, due to this, it may be subject to amplitude dissipation. We can check this by using Von Neumann analysis on the scheme.

Start by setting

$$v_j^n = \hat{v}^n(\omega)e^{i\omega hj}$$

substituting this into the LW scheme advection equation, we get:

$$\hat{v}^{n+1}e^{i\omega hj} = \hat{v}^n e^{i\omega hj} - \frac{ck}{2h}(\hat{v}^n e^{i\omega h(j+1)} - \hat{v}^n e^{i\omega h(j-1)}) + \frac{c^2k^2}{2h^2}(\hat{v}^n e^{i\omega h(j+1)} - 2\hat{v}^n e^{i\omega hj} + \hat{v}^n e^{i\omega h(j-1)})$$
$$\hat{Q}\hat{v}^n e^{i\omega hj} = \hat{v}^n e^{i\omega hj}\big(1 - \frac{ck}{2h}(e^{i\omega h} + e^{-i\omega h}) + \frac{c^2k^2}{2h^2}(e^{i\omega h} - 2 + e^{-i\omega h})\big)$$
$$\hat{Q} = 1 - \frac{ck}{2h}(e^{i\omega h} - e^{-i\omega h}) + \frac{c^2k^2}{2h^2}(e^{i\omega h} - 2 + e^{-i\omega h})$$
$$\hat{Q} = 1 - i\frac{ck}{h}\sin\omega h - \frac{c^2k^2}{h^2}(1 - \cos\omega h)$$

Setting $\frac{ck}{h} = \alpha$ and solving for $||\hat{Q}||^2$, we get:

$$
\begin{aligned}
||\hat{Q}||^2 &= (1 - \alpha^2(1 - \cos\omega h))^2 + (\alpha \sin\omega h)^2 \\
||\hat{Q}||^2 &= 1 - 2\alpha^2(1 - \cos\omega h) + \alpha^4(1 - \cos\omega h)^2 + \alpha^2 \sin^2\omega h \\
||\hat{Q}||^2 &= 1 + \alpha^2\big(2(1 - \cos\omega h) + \alpha^2(1 - \cos\omega h)^2 + \sin^2\omega h\big) \\
||\hat{Q}||^2 &= 1 + \alpha^2(2 - 2\cos\omega h + \alpha^2(1 - \cos\omega h)^2 + 1 - \cos^2\omega h) \\
||\hat{Q}||^2 &= 1 + \alpha^2\big(\alpha^2(1 - \cos\omega h)^2 - (\cos^2\omega h - 2\cos\omega h + 1)\big) \\
||\hat{Q}||^2 &= 1 - \alpha^2(1 - \alpha^2)(1 - \cos\omega h)^2
\end{aligned}
$$

Therefore

$$
\begin{aligned}
||\hat{Q}||^2 &\leq 1 \text{ so long as} \\
1 - \alpha^2 &\leq 0 \\
1 &\leq \alpha \\
1 &\leq \frac{ck}{h} \\
c &\leq \frac{h}{k}
\end{aligned}
$$

Which means that it is stable so long as Courant's stability condition is met.

5.3 Implementing and Comparing the Lax-Wendroff

Implementing this scheme with the advection equation of a Gaussian pulse with $c = 1$ and $k = h$ and evolving the system over 4 units of time. Figure 5.1 shows that the pulse is almost perfectly propagated by this method. The L2-norm of the error can be seen in Figure 5.2 showing that the method is very stable.

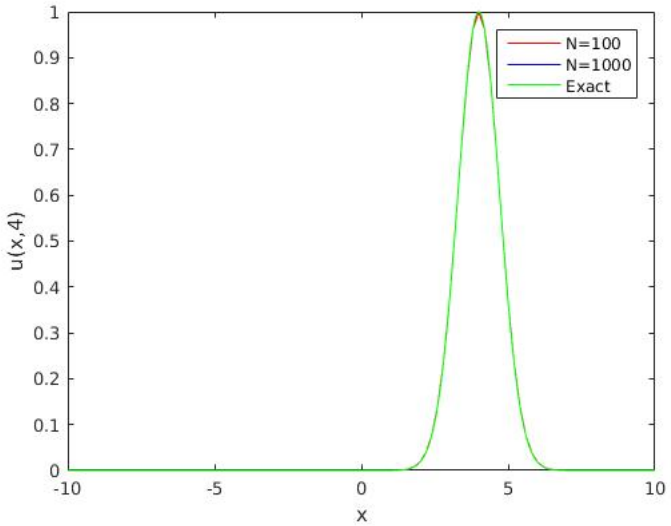


Figure 5.1: Lax-Wendroff Approximation after 4 seconds

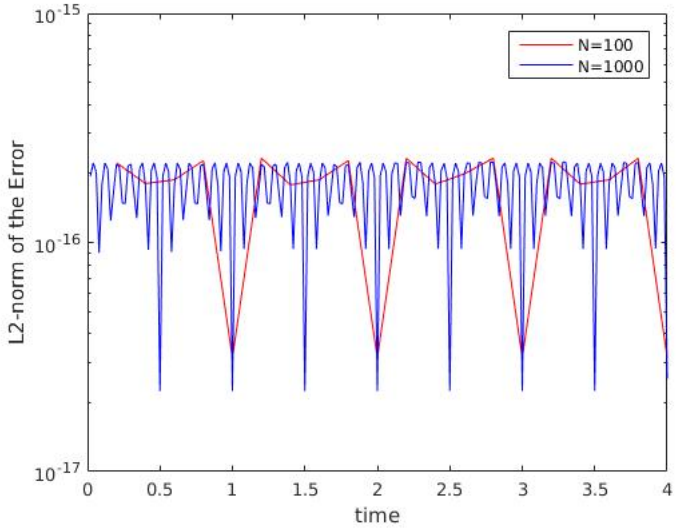


Figure 5.2: L2-norm of the Error of Lax-Wendroff Approximation over 4 seconds

Comparing the Lax-Wendroff to the Centered Leapfrog and Upwind Euler, we find that it is not as accurate at evolving the function as the Upwind Euler, but it compares favorably to the Centered Leapfrog as shown by Figure 5.3 and with the different step sizes analyzed in Figure 5.4 where it can clearly be seen that the largest error lies on the step.

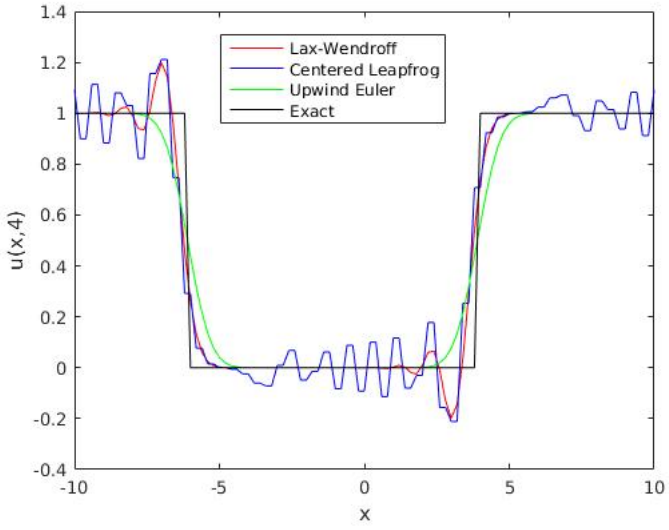
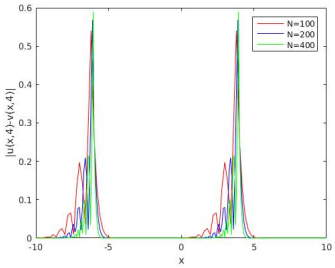
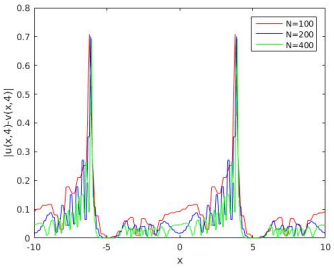


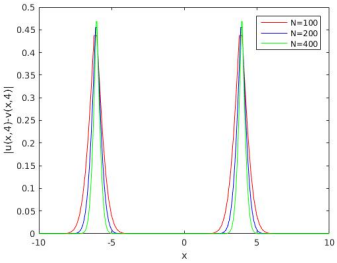
Figure 5.3: Comparing Lax-Wendroff, Centered Leapfrog and Upwind Euler.



(a) Lax-Wendroff



(b) Centered Leapfrog



(c) Upwind Euler

Figure 5.4: Comparison of methods at different step sizes.

Chapter 6

Flux Conservative Methods

In this chapter we will analyze the different types of discrete evolutionary methods. We will then more deeply analyze the Flux Conservative and REA schemes and see how they are similar in their solutions, although very different in how they approach the problem.

6.1 Comparison of Methods

Discrete evolutionary methods can be grouped into three main classes each different in the way in which the approach the problem of numerically approximating the solution of an equation. All of these methods work by initially turning the continuous initial solution into a discrete set of points, but differ in how they use these points to obtain a solution. Therefore, with all of these solutions, using a smaller spacing with a finer mesh allows for greater accuracy.

6.1.1 Finite Differencing

The Finite Difference Method (FDM) solves a differential equation by approximating the solution by solving reordered Taylor polynomials. Depending on how these polynomials are used and to what order the Taylor series is expanded, many different solution forms can be obtained. These solutions are more accurate with a higher order expansion, but at the cost of being more expensive to calculate.

6.1.2 Flux Conservation

Flux Conservative Methods (FCM) seek to evolve an equation by using the conservation laws to preserve a conservative property of the system, namely $u(x,t)$. By comparing $u(x,t)$ to, for example, the physical form of the density of a gas in a sealed pipe we can assume that the total amount of gas in the pipe is constant. The amount of gas between two points, x_1 and x_2, described by $M(t) = \int_{x_1}^{x_2} u(x,t)dx$, will change over time as

$$\begin{aligned}\frac{d}{dt}M(t) &= \frac{d}{dt}\int_{x_1}^{x_2} u(x,t)dx \\ &= F(x_1,t) - F(x_2,t)\end{aligned}$$

where $F(x,t)$ describes the flux through a point. This describes the fact that "gas" enters the region at x_1 and exits at x_2 and therefore $\frac{d}{dt}M(t)$ is conservative. This can then be extended to form what is know as conservative form as seen in Equation 5.1. Unlike the FDM, which uses the differential form of an equation to evolve it, FCM uses the integral form, $\frac{d}{dt}M(t)$, to evolve a solution. The schemes differ in the way in which the flux is calculated, with a more complex flux approximation yielding a higher accuracy.

6.1.3 Reconstruction Methods

Also referred to as Reconstruct-Evolve-Average (REA) algorithms, these work by assuming the function is made up of a set of piecewise constant values. The system is evolved by first fitting the average value to a cell and assuming the value is constant across the cell (Reconstruct). The piecewise function is then evolved over one time step, with values of the cell being carried into one another (Evolve). The new values are then averaged by the "amount" of the function which has left the cell and the amount which has entered (Average). These methods can be improved by replacing the piecewise constant with sloping equations or with polynomials to allow for better accuracy with the function being smoother at the cell boundaries.

6.2 Advection by Flux Conservative Schemes

Implementing the advection equation using an FCM we can see that the optimal flux boundaries can be given by the midpoint between two discrete points. Setting these boundaries as the boundaries of a cell, we can set the flux as the average value of the cell. For some x_j this is then given as $v_j = [x_{j-1/2}, x_{j+1/2}]$ with $x_{j+1/2} = x_j + h/2$. We also see that the flux at a given point is proportional to the value of the function in that cell at that point in time, v_j^n, and the speed at which the flux propagates, c. Therefore the flux at the cell v_j at time t^n can be written as

$$F_{j+1/2}^n = cv_j^n \tag{6.1}$$

The adjustment of $1/2$ is due to the fact that the center of a flux cell lies in between two points.

The value of the cell can be seen as the average value of the function between its boundaries, $v_j^n = \frac{1}{h}\int_{x_{j-1/2}}^{x_{j+1/2}} u(x,t^n)dx$, and similarly, $v_j^{n+1} = \frac{1}{h}\int_{x_{j-1/2}}^{x_{j+1/2}} u(x,t^{n+1})dx$. The flux, or change, in the value for the cell over its domain from t^n to t^{n+1} can therefore be seen as $hv_j^{n+1} - hv_j^n$ which is equivalent to the difference in the flux values at the boundaries over the time step, $kF_{j-1/2}^n - kF_{j+1/2}^n$. The flux can therefore be seen as the collection of instantaneous fluxes:

$$F_{j-1/2}^n = \int_{t^{n+1}}^{t^n} f(x_{i-1/2}, t)dt$$

Now we have

$$hv_j^{n+1} - hv_j^n = kF_{j-1/2}^n - kF_{j+1/2}^n \tag{6.2}$$

$$v_j^{n+1} - v_j^n = \frac{k}{h}(cv_{j-1}^n - cv_j^n) \tag{6.3}$$

$$v_j^{n+1} = v_j^n - \frac{ck}{h}(v_j^n - v_{j-1}^n) \tag{6.4}$$

which is the Upwind Euler method. It should be noted that the FCM relies on the value from one cell being projected to its neighbors, therefore $ck \leq h$, meaning that Courant's stability condition must hold as a restriction for these methods.

6.3 Lax-Friedrich as a Flux Conservative Scheme

By observing Equation 6.4, we can see that the general form of a Flux Conservative method is

$$v_j^{n+1} = v_j^n - \frac{k}{h}(F_{j-1/2}^n - F_{j+1/2}^n) \tag{6.5}$$

We now seek to find a flux function, $F^n_{j-1/2}$, such that the Lax-Friedrich method can be recast to a flux conservative form. Starting with the Lax-Friedrich form for $u_t + f(u)_x = 0$

$$
\begin{aligned}
v_j^{n+1} &= \frac{1}{2}(v_{j+1}^n + v_{j-1}^n) - \frac{k}{2h}(f(v_{j+1}^n) - f(v_{j-1}^n)) \\
v_j^{n+1} &= v_j^n + \frac{1}{2}(v_{j+1}^n - 2v_j^n + v_{j-1}^n) - \frac{k}{2h}(f(v_{j+1}^n) + f(v_j^n) - f(v_j^n) - f(v_{j-1}^n)) \\
v_j^{n+1} &= v_j^n + \frac{1}{2}(v_{j+1}^n - v_j^n + v_{j-1}^n - v_j^n) - \frac{k}{2h}(f(v_{j+1}^n)) + f(v_j^n) - f(v_j^n) - f(v_{j-1}^n)) \\
v_j^{n+1} &= v_j^n - \frac{k}{h}(\frac{1}{2}(f(v_{j+1}^n) + f(v_j^n) - f(v_j^n) - f(v_{j-1}^n)) - \frac{h}{2k}(v_{j+1}^n - v_j^n + v_{j-1}^n - v_j^n)) \\
v_j^{n+1} &= v_j^n - \frac{k}{h}\left(\frac{1}{2}\big(f(v_{j+1}^n) + f(v_j^n)\big) - \frac{1}{2}\big(f(v_j^n) + f(v_{j-1}^n)\big) - \frac{h}{2k}\big(v_{j+1}^n - v_j^n\big) + \frac{h}{2k}\big(v_j^n - v_{j-1}^n\big)\right) \\
v_j^{n+1} &= v_j^n - \frac{k}{h}\left(\left(\frac{1}{2}\big(f(v_{j+1}^n) + f(v_j^n)\big) - \frac{h}{2k}\big(v_{j+1}^n - v_j^n\big)\right) - \left(\frac{1}{2}\big(f(v_j^n) + f(v_{j-1}^n)\big) - \frac{h}{2k}\big(v_j^n - v_{j-1}^n\big)\right)\right)
\end{aligned}
$$

Now we can see that our flux function is

$$F^n_{j-1/2} = \frac{1}{2}\big(f(v_j^n) + f(v_{j-1}^n)\big) - \frac{h}{2k}\big(v_j^n - v_{j-1}^n\big) \tag{6.6}$$

6.4 Godunov's Method

The Piecewise Constant Method, also known as Godunov's Method, is a Reconstructive Method. Similarly, the constant value assigned to the cell is the average of the cell whose boundaries are again defined as $[x_{j-1/2}, x_{j+1/2}]$. The value of the cell is again defined as $v_j^n = \frac{1}{h}\int_{x_{j-1/2}}^{x_{j+1/2}} u(x, t^n)dx$. With the advection equation specifically this value is shifted across at the rate of ck with a cell width of h. This propagated value is again then averaged as the value of v_j^{n+1}. For the advection equation, this can be seen as

$$
\begin{aligned}
v_j^{n+1} &= \frac{1}{h}(ckv_{j-1}^n + (h - ck)v_j^n) \\
&= \frac{1}{h}(ckv_{j-1}^n + hv_j^n - ckv_j^n) \\
&= v_j^n + \frac{ck}{h}(v_{j-1}^n - v_j^n) \\
&= v_j^n - \frac{ck}{h}(v_j^n - v_{j-1}^n)
\end{aligned}
$$

which again is exactly equivalent to the Upwind Euler Method.

This therefore shows that while having different approaches they all yield equivalent results.

Chapter 7

Slope Limiter Methods

7.1 Method Recap

In Chapter 5, we looked at the Upwind Euler, Centered Leapfrog and Lax-Wendroff methods. Setting aside the Centered Leapfrog as it is the most inaccurate for evolving a step function, we now look at the Upwind, Lax-Wendroff and Lax-Friedrich methods instead. By looking at Figure 7.1 we can see that the Lax-Wendroff performs the best, although marginally, against the other two methods on the Gaussian pulse but the worst at the heaviside step function. The Upwind Euler, however, performs best on the heaviside step function, but none of these methods are able to adequately evolve the heaviside step without some form of sloping or erratic behaviour at the shock points. Therefore we need a method which adequately evolves both smooth, differentiable functions and functions with sharp points or steps These are known as High-Resolution Methods or Slope/Flux Limiter Methods.

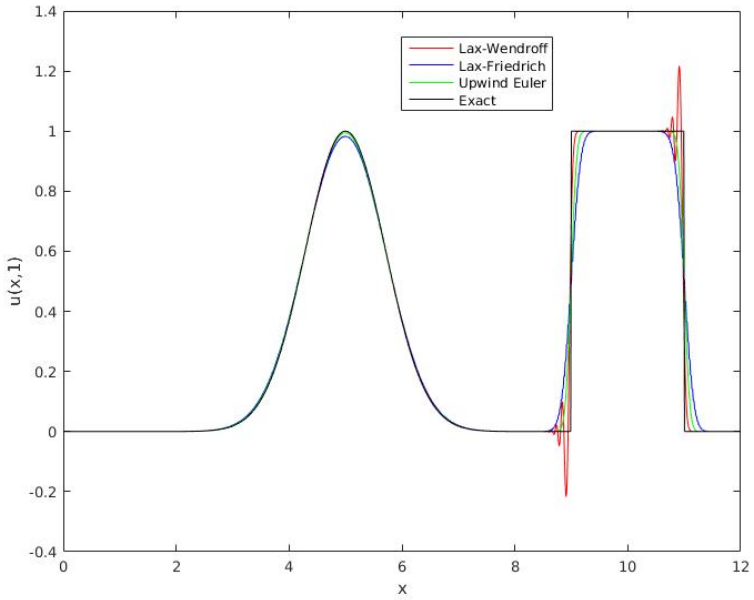


Figure 7.1: Comparison of Lax-Wendroff, Lax-Friedrich and Upwind Euler methods

7.2 Slope Limiter Methods

Slope Limiter Methods (SLMs) are a special form of Piecewise-Linear functions which in turn are a more complex form of a Reconstruction Method (Section 6.1.3). All SLMs take the form of

$$v_j^{n+1} = v_j^n + \frac{k}{h}(f(v_{j+1}^n) - f(v_{j-1}^n)) + \frac{1}{2}(h - ck)(\sigma_j^n - \sigma_{j-1}^n) \tag{7.1}$$

They differ in how they choose σ_j^n to be used to calculate the slope of the cell. There are three different potential forms of σ, namely

$$\begin{aligned}
\sigma_{us} &= \frac{v_j - v_{j-1}}{h} \text{ Upwind Slope} \\
\sigma_{ds} &= \frac{v_{j+1} - v_j}{h} \text{ Downwind Slope} \\
\sigma_{cs} &= \frac{v_{j+1} - v_{j-1}}{2h} \text{ Centered Slope}
\end{aligned}$$

7.2.1 Minmod

Minmod is the simplest of these methods, using the minmod function as follows

$$\begin{aligned}
minmod(a, b) &= a && if|a| < |b| \\
&= b && if|b| < |a| \\
&= 0 && ifab < 0
\end{aligned}$$

to determine the value of σ_j^n. It does this by setting the value as

$$\sigma_j^n = \text{ minmod}(\sigma_{us}, \sigma_{ds})$$

7.2.2 Superbee

The Superbee works by finding a maxmod (similar to a minmod) of two values of which the minimum has been found.

$$\begin{gathered}
\sigma_j^n = \text{ maxmod}(\sigma^1, \sigma^2) \\
\text{where:} \quad \sigma^1 = \text{ minmod}(\sigma_{ds}, 2\sigma_{us}) \quad \text{and} \quad \sigma^2 = \text{ minmod}(2\sigma_{ds}, \sigma_{us})
\end{gathered}$$

It biases the weighings in such a way that even if one of the gradients is much larger than the other, the resulting gradient isn't too biased towards the smaller gradient.

7.2.3 MC Mod

The MC Mod method uses the minmod of all three gradient measures to determine the next value. It biases the value of the central slope as it is seen as the average of the three values.

$$\sigma_j^n = \text{ minmod}(\sigma_{cs}, 2\sigma_{us}, 2\sigma_{ds})$$

7.2.4 Analysis

By using these three methods on the same function as Figure 7.1, we can see in Figure 7.2 that these methods provide a high accuracy for both the Gaussian and the heaviside step functions and have a steeper step than that of the Upwind Euler.

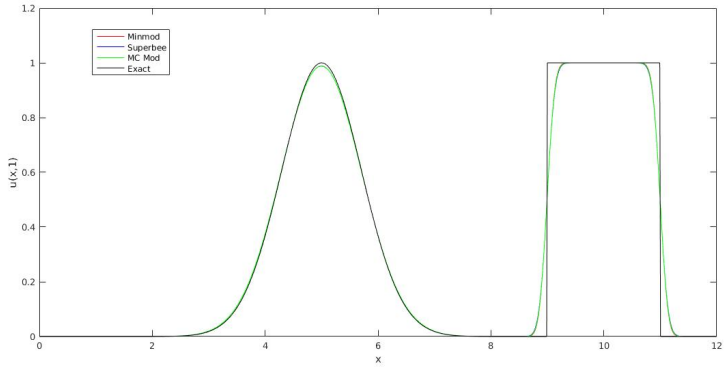


Figure 7.2: Comparison of Minmod, Superbee and MC Mod methods

Chapter 8

The Inviscid Burgers' Equation

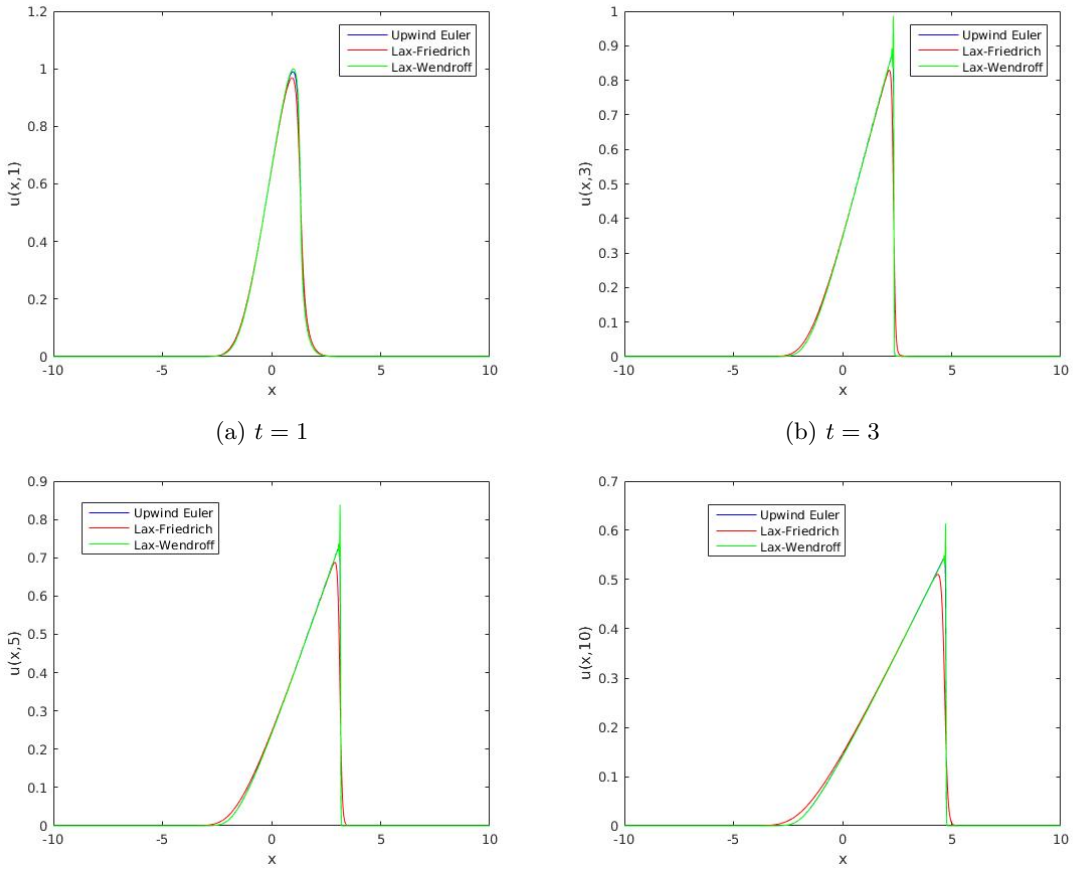
The Inviscid Burgers' Equation is defined as a conservative, first order, quasilinear hyperbolic equation with the general solution of

$$u_t + uu_x = 0 \tag{8.1}$$

From this we can analyze the characteristics to see that the characteristics intercept at $t = 1$. It also means that an exact solution to the equation is difficult to obtain analytically. By numerically approximating the solution over time using the Upwind Euler, Lax-Friedrich and Lax-Wendroff methods discussed in previous chapters as well as the Minmod, Superbee and MC Mod methods discussed in Chapter 7, we can see how the solution evolves and how these methods fall short in the approximation.

Evolving a Gaussian pulse over time we can see how the methods approximate a solution at $t = 1, 3, 5$ and 10. Figure 8.1 shows the solution being approximated using the Upwind Euler, Lax-Friedrich and Lax-Wendroff methods. We can see that, while all of these solutions lose energy over time, the Lax-Friedrich does so especially. The Lax-Wendroff develops a non-smooth spike near the end of the pulse showing it is especially lacking approximation as well.

Figure 8.2 uses Minmod, Superbee and MC Mod and has a much smoother solution. While it is also subject to energy loss with the peak of the wave being lowered, it is less than that of the methods used in Figure 8.1.



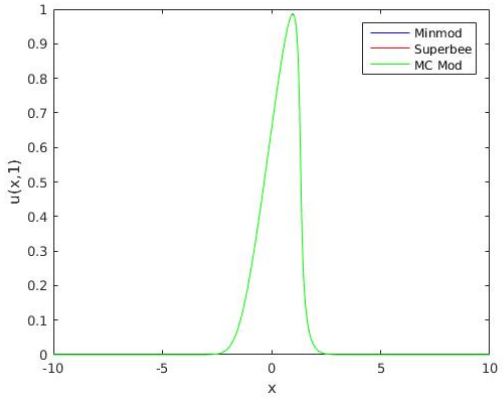
(a) $t = 1$

(b) $t = 3$

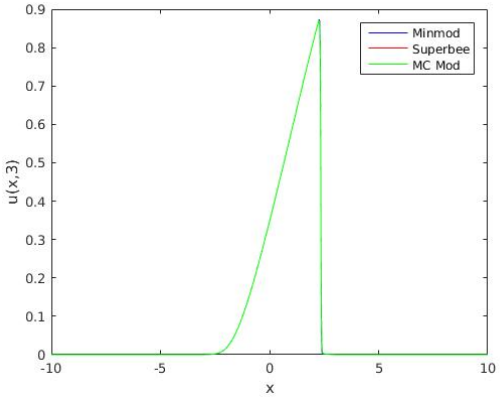
(c) $t = 5$

(d) $t = 10$

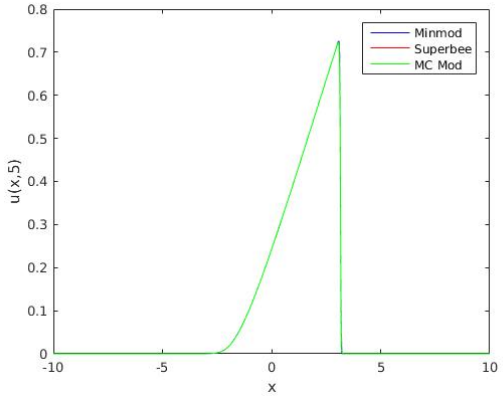
Figure 8.1: Evolution of Burgers' Equation using Upwind Euler, Lax-Friedrich and Lax-Wendroff



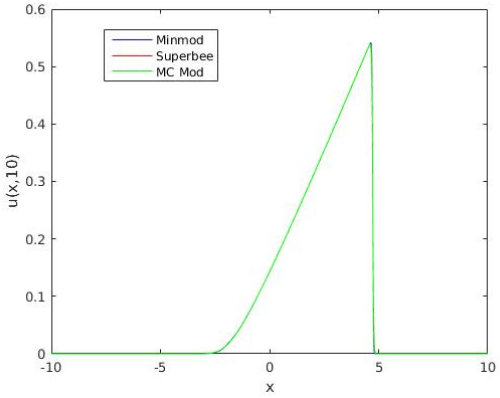
(a) $t = 1$



(b) $t = 3$



(c) $t = 5$



(d) $t = 10$

Figure 8.2: Evolution of Burgers' Equation using Minmod, Superbee and MC Mod