University of Reading

Department of Mathematics and Statistics

An Investigation of Conservative Moving-Mesh Methods for Conservation Laws

Niall Arthurs

Thesis submitted for the degree of
Doctor of Philosophy

September 2016

Abstract

In this thesis we consider a class of conservation based moving mesh methods applied to hyperbolic conservation laws. We mainly concentrate on the one dimensional case with the examples of the linear advection equation, inviscid Burgers' equation and the Buckley-Leverett equation. The moving mesh methods are generated using the conservation of mass as a method for determining the mesh velocity at the computational nodes. We use the notion of the reference space as a mathematical tool to analyse the moving mesh methods allowing us to show the accuracy, stability conditions and convergence. In addition we use the reference space as a technique for constructing new moving mesh methods which share the accuracy and stability properties of the fixed mesh scheme they are derived from. At the end of the thesis we use the knowledge gained from the scalar conservation laws to construct moving mesh methods for the isothermal equations.

Declaration

I confirm that this is my own work and the use of all materials from other sources have been properly and fully acknowledged

Signed.................................. Date............................

Niall Arthurs

Acknowledgements

Firstly, I would like to give a big thank you to my supervisors Dr Pete Sweby and Prof. Mike Baines. Without their guidance, encouragement and most of all incredible patience this thesis would have never been finished. I would also like to thank Dr. Tristan Pryer for sharing his wealth of knowledge and valuable time.

Many thanks go to Dr Alan Dawes of AWE for his regular support and many insights.

A very big thank you to my understanding fiancée, Xianrong 'Yoko' Wang, for her love, support and patience. It would have been impossible without her. Similarly, I would like to thank my family who have waited so long for this thesis to be finished.

University would have been a very different experience if not for all of the friends I have made. I wish them all the best in the future and thank them all for the great times.

Finally, I would like to acknowledge the financial support of both the EPSRC and AWE.

Contents

List of Figures

Chapter 1

Introduction

A class of Partial Differential Equations (PDEs) known as conservation laws frequently arises in physics whenever a conserved quantity is present. Generally the conservation laws which arise from physical phenomena are nonlinear and as a result it is not possible to find an analytic solution. As a result, numerical approximations are required.

Standard numerical techniques rely on static meshes to computationally solve conservation laws but this is often inefficient. The simplest technique of solving on a uniform static mesh performs poorly due to the need to have a large number of nodes to correctly resolve discontinuities. Adaptive static mesh techniques, known as h-refinement methods, improve this by only increasing the resolution in regions where necessary but suffer from the increased computational cost of calculating where the mesh needs to be refined and the fact that it may still take many nodes to achieve the required resolution.

It is clear that mesh adaptivity is important to generate a numerical method which accurately approximates the solutions to conservation laws without being too computationally expensive. However, as noted, h-refinement methods may be expensive while an improvement over non-adaptive meshes may still have a high computational overhead. Instead, more recent research has focused on moving mesh methods known as r-refinement methods. R-refinement methods are adaptive methods in which the computational mesh is moved in an attempt to automatically refine the mesh in an advantageous way without having to introduce more nodes.

One particular class of r-refinement methods of note for use with conservation laws is conservation based moving mesh methods. These methods work by considering a conserved quantity and using the local conservation of this quantity as a method for positioning or finding the velocity of the mesh nodes. As such they appear to be a natural fit for conservation laws which are also derived from conserved quantities. It is these conservation methods that will be the focus of this thesis.

Chapter 2 will provide a summary of prior knowledge required in the rest of the thesis as well as an overview of some of the work that has been done in the field. The chapter is split into three sections, the first focuses on the conservation laws themselves while the second and third both provide information regarding r-refinement methods.

In Chapter 3 the background work from Chapter 2 will be combined to derive the general class of schemes studied in later chapters. This derivation will be done step by step to show how standard Eulerian PDEs can be adapted for use with a Lagrangian conservation based moving mesh scheme. Particular care will be taken concerning the choices made to derive the schemes as well as potential issues such as boundary conditions. At the end of the chapter the general framework will be demonstrated by applying it to several test problems.

Chapter 4 introduces the notion of a ‘reference space’ as an analytical tool for analysing the class of conservative moving mesh methods considered. Using the reference space we will discuss some methods for finding the accuracy, stability and convergence of the conservation based moving mesh schemes. Finally numerical results are carried out to indicate the numerical order of convergence of a test scheme.

The main aim of Chapter 5 is to show how standard conservative Eulerian finite difference schemes can be adapted to produce new moving mesh schemes. The benefit of this is that the properties of the resulting moving mesh schemes are easily determined from the properties of the generating Eulerian scheme. The end of the chapter will compare some generated Lagrangian schemes with the Eulerian schemes used to derive them.

In Chapter 6 we will give a brief introduction to some of the problems faced when trying to extend the scheme to systems of hyperbolic conservation laws. The added issues will lead to a slightly modified method in which a more complex mesh equation is found. This modified scheme is applied to the isothermal Euler equations and the resulting mesh trajectories are shown. The end of the chapter will discuss possible improvements that could be made to improve the scheme for systems of equations as well as the work required to use the method for the full Euler equations of fluid dynamics.

The final chapter of the thesis will provide a summary of all of the work done. In addition there will be a discussion of potential future work regarding adapted schemes, systems of equations and higher dimensional problems.

The novel work done in this thesis appears in chapters 3-6. These original aspects are:

- In Chapter 3 we give a more in-depth discussion of how Eulerian boundary conditions are applied to Lagrangian schemes than appears in the literature.

- The notion of the transform to reference space from Chapter 4 is taken from the Moving Mesh Partial Differential Equation (MMPDE) methods but is applied as an analytical tool for the first time to find accuracy, stability and convergence.

- Chapter 5 discusses a novel approach to generating new moving mesh methods from existing fixed mesh conservative Eulerian methods.

- The attempt to solve the isothermal equations in Chapter 6 provides a moving mesh which does not tangle.

Chapter 2

Background

In this chapter we give an overview of the previous work done in both hyperbolic conservation laws and moving mesh methods. This background covers both related work which has already been done, as well as background knowledge required for application of the results found.

2.1 Hyperbolic Conservation Laws

Hyperbolic conservations law arise in many areas of physics, notably when conserved quantities are present in the system being modelled. In the Eulerian description, conservation laws can often be expressed as time-dependent systems of partial differential equations (PDEs) with a particularly simple structure.

In 1D the equations have the differential form

$$u(x,t)_t + f(u(x,t))_x = 0, \tag{2.1}$$

where $u : \mathbb{R} \times \mathbb{R} \to \mathbb{R}^m$ is an m-dimensional vector of conserved quantities, $f : \mathbb{R}^m \to \mathbb{R}^m$ is a known flux function which prescribes the rate of flow for each conserved variable and x and t are the spatial and time coordinates respectively [LeV92].

Note that in this thesis we are using the subscript notation to represent partial derivatives. Therefore u_x is equivalent to $\frac{\partial u}{\partial x}$ and $u_{tt} = \frac{\partial^2 u}{\partial t^2}$.

In this work the system of equations (2.1) is assumed to be hyperbolic. This implies that the Jacobian matrix of the flux function, $f'(u)$, has a complete set of m linearly independent real eigenvectors for each value of u.

To form a complete problem the PDE (2.1) must also be equipped with an initial condition $u^0(x)$ and suitable boundary conditions.

The requirement for boundary conditions depends on the domain of the problem. Importantly, it is well established that for a hyperbolic conservation law to be well-posed, boundary conditions must be applied only at boundaries where there is information entering the domain.

Definition 2.1.1. A boundary is defined to be an 'inflow boundary' if and only if

$$-\mathbf{n} \cdot f'(u)|_{\partial\Omega} > 0, \tag{2.2}$$

where $\partial\Omega$ is the boundary being considered and $\mathbf{n}$ is the normal unit vector which leaves the domain. Similarly, a boundary is defined to be an 'outflow boundary' if and only if

$$-\mathbf{n} \cdot f'(u)|_{\partial\Omega} < 0. \tag{2.3}$$

This definition of information in/out is arrived at via the characteristics of the solution, which is explained in more detail in section 2.1.2. Furthermore, it is noted that the case where $-\mathbf{n} \cdot f'(u)|_{\partial\Omega} = 0$ is not considered here as this is a special case. Further discussion of this special case can be found in Section 3.2.

In general, conservation laws which arise from physical phenomena have a nonlinear flux function and are therefore themselves nonlinear. It is not generally possible to derive an exact solution for these nonlinear systems and it is therefore necessary to construct and analyse numerical methods to find approximate solutions.

2.1.1 Derivation of Conservation Laws

While conservation laws are often written in the differential form (2.1), this is not how they arise from physical principles. The basic way for conservation laws to arise is in the form of a balance equation.

Balance equations arise when a conserved quantity is considered over a control volume, $[x_1, x_2]$. For example, consider the conservation of mass in a system where there is a flow. The mass in the control volume is given by the integral of a density, $u(x,t)$, over that volume, *i.e.*

$$\text{mass} = \int_{x_1}^{x_2} u(x,t)dx. \tag{2.4}$$

Since mass is conserved it follows that this integral can only change due to flow into or out of the control volume. Let $f(u(x,t))$ be a given flux function which describes the rate of flow of the mass, then the rate of change of mass is given by the balance law

$$\frac{d}{dt}\int_{x_1}^{x_2} u(x,t)dx = -\left[f(u(x,t))\right]_{x_1}^{x_2} \tag{2.5}$$

where $[\cdot]$ denotes the jump in the argument in the square brackets.

Note that while equation (2.5) was derived here for the scalar density function, the more general balance equation has the same form if the integrand, $u(x,t)$, is a vector of conserved quantities and the flux function, $f(u(x,t))$, is a function of the components of $u(x,t)$.

The differential form of the conservation law (2.1) introduced in the beginning of this chapter can be derived from the balance law (2.5) by first integrating over time. Integrating equation (2.5) over the time interval $[t_1, t_2]$ yields

$$\int_{t_1}^{t_2} \frac{d}{dt}\left(\int_{x_1}^{x_2} u dx\right) dt + \int_{t_1}^{t_2} [f(u)]_{x_1}^{x_2} dt = 0, \tag{2.6}$$

and using the fundamental theorem of calculus and rearranging gives

$$\int_{x_1}^{x_2} [u]_{t_1}^{t_2} dx + \int_{t_1}^{t_2} [f(u)]_{x_1}^{x_2} dt = 0. \tag{2.7}$$

Now assume that both $u(x,t)$ and $f(u(x,t))$ are differentiable functions. In this case it follows from the fundamental theorem of calculus that

$$[u(x,t)]_{t_1}^{t_2} = \int_{t_1}^{t_2} u(x,t)_t dt, \tag{2.8}$$

and

$$[f(u(x,t))]_{x_1}^{x_2} = \int_{x_1}^{x_2} f(u(x,t))_x dx. \tag{2.9}$$

Substituting equations (2.8) and (2.9) into equation (2.7) gives

$$\int_{t_1}^{t_2} \int_{x_1}^{x_2} \left(u(x,t)_t + f(u(x,t))_x\right) dx dt = 0, \tag{2.10}$$

where it is noted that the intervals $[x_1, x_2]$ and $[t_1, t_2]$ have been chosen arbitrarily. Hence it follows that the integrand of equation (2.10) must be identically zero leading to the equation

$$u(x,t)_t + f(u(x,t))_x = 0, \tag{2.11}$$

which we note is the differential form of the conservation law (2.1).

Remark 2.1.2. The differential form (2.1) is not the only differential form of the conservation laws, it is a special form referred to as the conservation form. Another key differential form that the reader should be aware of is the non-conservative form. This form takes a nonlinear conservation law and rewrites it as though it is linear by using the chain rule on the flux term to obtain an explicit $u(x,t)_x$. The non-conservative or 'quasilinear' form associated with the conservation law (2.1) is

$$u(x,t)_t + f'(u(x,t))u(x,t)_x = 0. \tag{2.12}$$

2.1.2 Mathematical Difficulties

Hyperbolic conservation laws have several mathematical difficulties which must be overcome in order for a 'correct' solution to be found. The main concerns are the differentiability of the solution and the existence of a unique solution.

Discontinuous Solutions

Discontinuous solutions are a major mathematical difficulty which arises when considering hyperbolic conservation laws. Since the problems are often stated in the differential form (2.1) it seems that the conservation law cannot hold for discontinuous solutions: however by contrast there is no assumption on the smoothness of

$u(x, t)$ in the original balance law (2.5).

The issue of discontinuous solutions arises because of the assumption of differentiability made in equations (2.8) and (2.9) that is required to derive the differential form. It is also not possible to disregard discontinuous solutions as they can arise in nonlinear conservation laws from smooth initial conditions and smooth boundary conditions.

To demonstrate how nonlinear flux functions can cause discontinuous solutions to form, the notion of characteristic solutions is introduced. Characteristic solutions involve finding contour lines of the solution in *(x,t)* space which are lines on which the solution $u(x, t)$ is constant. These characteristic lines can therefore be seen as carrying 'information' through time. As such each characteristic line has an associated trajectory equation, $x(t)$, which gives its position at a given time.

The velocity of the characteristic trajectories can be found by considering that the solution $u(x, t)$ is required to be constant along the characteristic lines. Hence,

$$\frac{d}{dt}u(x(t), t) = 0. \tag{2.13}$$

Applying the chain rule to the left hand side of equation (2.13) yields

$$u(x(t), t)_t + x'(t)u(x(t), t)_x = 0. \tag{2.14}$$

Comparing equation (2.14) with the quasilinear form of the conservation law (2.12) shows that there is a constant solution along the trajectory if

$$x'(t) = f'(u(x(t), t)). \tag{2.15}$$

Using the notion of characteristics we can consider how the structure of the flux function, $f(u(x,t))$, affects the solution $u(x,t)$. If $f(u(x,t))$ is linear then it is clear from equation (2.15) that the characteristic trajectories are independent of the value of the solution, $u(x(t),t)$, along that characteristic line. However, if the flux function is nonlinear then the characteristic trajectory is dependent on the u value carried by the characteristic line.

Consider the conservation law (Inviscid Burgers' Equation)

$$u_t + \left(\frac{1}{2}u^2\right)_x = 0, \tag{2.16}$$

for which $f(u) = \frac{1}{2}u^2$ and where we have left out the independent variables, x and t, for ease of reading. Suppose that (2.16) is coupled with the initial condition

$$u^0(x) = -x, \quad x \in [-1,1], \tag{2.17}$$

and the boundary conditions

$$u(-1,t) = 1 \quad \text{and} \quad u(1,t) = -1. \tag{2.18}$$

The resulting characteristic velocity is given by

$$x'(t) = f'(u) = u, \tag{2.19}$$

where u is a constant on the trajectories (characteristics). Figure 2.1 shows the trajectories of some of the characteristics in the x,t plane.

As can be seen in Figure 2.1, at time $t = 1$ the characteristic lines cross, meaning that the solution becomes multivalued for time, $t \geqslant 1$ and the differential equation

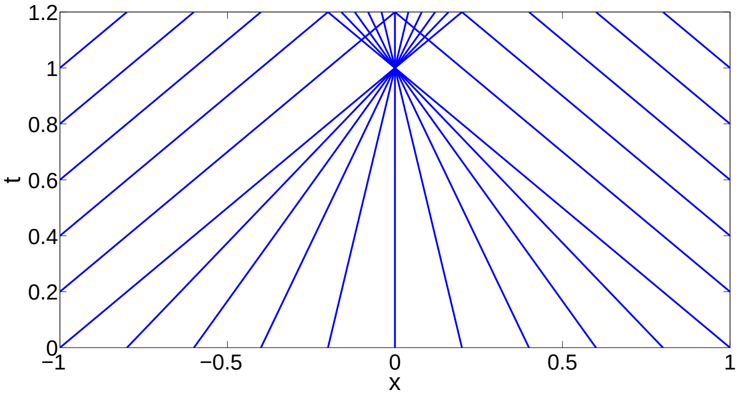


Figure 2.1: Characteristics crossing after time $t = 1$ causing the solution to become multivalued. This is not a physically valid solution to the problem (2.16)-(2.18).

(2.16) is invalid (the second part of this section will introduce the idea of entropy solutions and show that this multivalued solution is not physically relevant for the purposes of this example). However it is accepted that the actual solution to the problem (2.16)-(2.18) for time $t \geqslant 1$ is given by

$$u(x,t) = \begin{cases} 1 & x < 0 \\ -1 & x > 0 \end{cases}, \tag{2.20}$$

where a discontinuity is now present at $x = 0$. The reason why this is the accepted physically relevant solution will be presented in the next subsection. The integral form (2.5) is more useful is this situation.

The example (2.16)-(2.18) demonstrates that, for nonlinear conservation laws, discontinuities can occur even with smooth initial conditions and boundary conditions. The standard approach to solve this problem is to return to the integral form (2.5) of the conservation law and introduce a boundary across which $u(x,t)$ is discontinuous.

Much as the PDE (2.1) is not a unique differential form of the conservation law, the balance law (2.5) is not a unique integral form. Another key form that readers should know is the weak form. To find the weak form associated with the general 1D conservation law (2.1) first multiply the conservation law by a once differentiable compactly supported test function $\phi(x,t) \in C_0^1(\mathbb{R}\times\mathbb{R})$ and then integrate over space and time. This yields

$$\int_0^\infty \int_{-\infty}^\infty (\phi u_t + \phi f(u)_x)\, dxdt = 0. \tag{2.21}$$

Using integration by parts on equation (2.21) to move the derivatives from the solution variables to the test function yields

$$\int_0^\infty \int_{-\infty}^\infty (\phi_t u + \phi_x f(u))\, dxdt + \int_{-\infty}^\infty \phi(x,0)u(x,0)dx = 0, \tag{2.22}$$

where it is noted that the boundary terms have disappeared due to the compact support of the test function.

Uniqueness of the Solution

Recall the example (2.16)-(2.18) from the first part of this section. In Figure 2.1 it appears that the characteristics of the problem cross and therefore the solution becomes multivalued. However, we stated that the physically relevant solution for times $t \geqslant 1$ was (2.20). Figure 2.2 shows the x, t plane characteristic plot for this discontinuous solution.

The problem arises because after time $t = 1$ there is no longer a classical solution to the problem and we must instead turn to the weak form (2.22). The issue with this is that the weak form does not have a unique solution and therefore in order to find the physically relevant solution of the problem another condition is required.

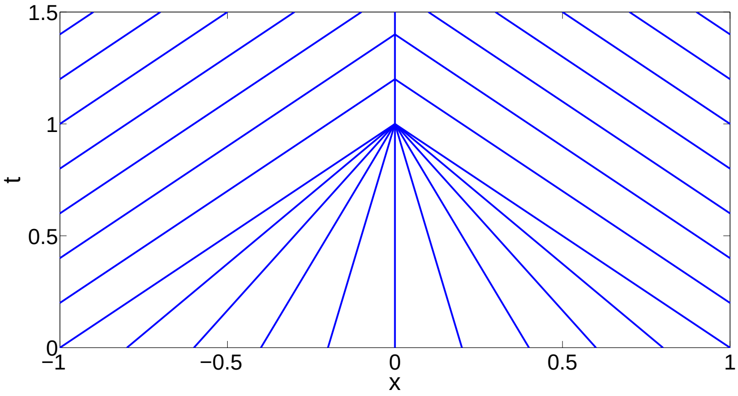


Figure 2.2: Characteristic plot with a discontinuity forming at time $t = 1$. This is the physically relevant characteristic solution to the problem (2.16)-(2.18).

The extra condition required to find the physically relevant solution can be derived from the notion of a 'vanishing viscosity solution'. This assumes that the conservation law is the limiting case of the viscous PDE

$$u_t + f(u)_x = \epsilon u_{xx}, \tag{2.23}$$

as $\epsilon \to 0$. The argument follows that since (2.23) has a classical solution for all $\epsilon > 0$, the physically relevant solution to the conservation law (2.22) should be the solution that is the limit of the classical solution of (2.23) as $\epsilon \to 0$. The solution that satisfies this restriction is called the 'entropy solution'.

In section 4.5.1 we look more in depth at vanishing viscosity solutions as a method for analysing numerical schemes. In this introductory chapter it is simply noted that this notion of a vanishing viscosity solution leads to a variety of entropy conditions which when applied alongside the weak form of the conservation law (2.22) leads to a unique solution. The most easily applied entropy condition for scalar conservation laws and general flux functions, $f(u)$, is due to Oleinik [Ole63].

Theorem 2.1.3 (Oleinik Entropy Condition)**.** $u(x, t)$ *is the entropy solution of the*

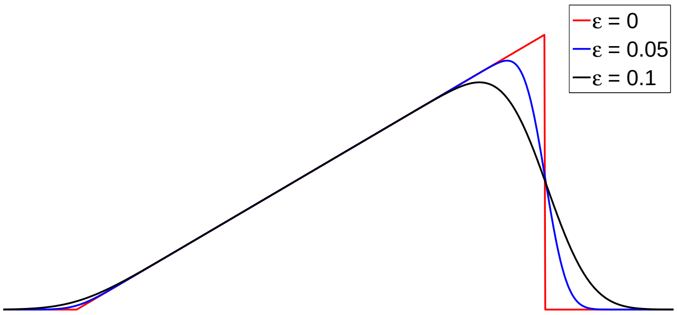


Figure 2.3: Sketch of a discontinuous solution where $\epsilon = 0$ (red) and two viscous solutions $\epsilon = 0.05$ (blue) and $\epsilon = 0.1$ (black).

weak form conservation law (2.22) if all discontinuities satisfy the condition that

$$\frac{f(u) - f(u_l)}{u - u_l} \geqslant s \geqslant \frac{f(u) - f(u_r)}{u - u_r} \tag{2.24}$$

for all u between u_l and u_r, where u_l is the limit of the solution as the discontinuity is approached from the left, u_r is the limit when approached from the right and s is the shock speed.

Applying the entropy condition (2.24) to the test problem (2.16)-(2.18) confirms that (2.20) is the physically relevant solution.

The shock speed noted in the entropy condition is found by considering the Rankine-Hugoniot jump condition. This is a relationship between the shock speed, s, and the states u_l and u_r derived from (2.5). The condition is given by

$$f(u_l) - f(u_r) = s\,(u_l - u_r)\,, \tag{2.25}$$

which simplifies to

$$s = \frac{f(u_l) - f(u_r)}{u_l - u_r} = \frac{[f(u)]}{[u]} \tag{2.26}$$

for scalar conservation laws.

The one issue that still needs to be considered is what happens if an entropy violating discontinuity is present in the initial condition. To demonstrate this consider the example conservation law (2.16) with the initial condition

$$u^0(x) = \begin{cases} -1 & x < 0 \\ 1 & x > 0 \end{cases} \tag{2.27}$$

and no boundary conditions due to the fact that both are 'outflow' boundaries.

The initial condition (2.27) is a solution to the weak form of the conservation law (2.16) for all time. Figure 2.4 shows the characteristic lines in the x, t plane for this solution.

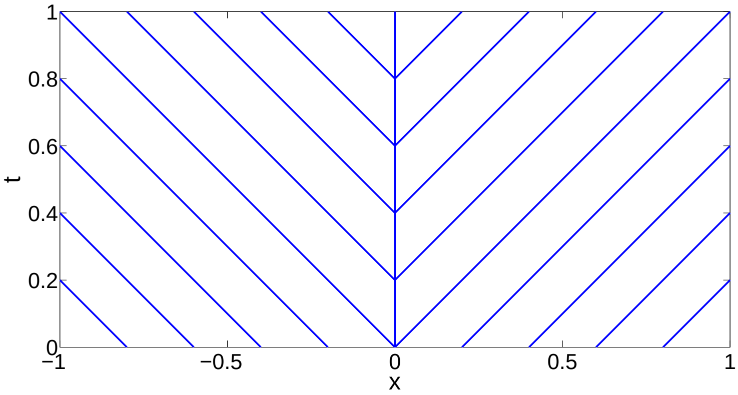


Figure 2.4: Characteristic plot to the time independent solution (2.27). This solution is entropy violating and therefore not a physically valid solution.

As can be seen in Figure 2.4 this solution has characteristics leaving the discontinuity. Applying the entropy condition (2.24) shows that this solution is not the entropy solution since the discontinuity is entropy violating.

The entropy solution to this entropy violating discontinuity in the initial condition is known as a 'rarefaction' fan. The entropy solution for the conservation law (2.16) with initial condition (2.27) is given by

$$u(x,t) = \begin{cases} -1 & x < -t \\ \frac{x}{t} & -t \leqslant x \leqslant t \\ 1 & x > t \end{cases}, \qquad (2.28)$$

and the characteristics for this solution are plotted in Figure 2.5.

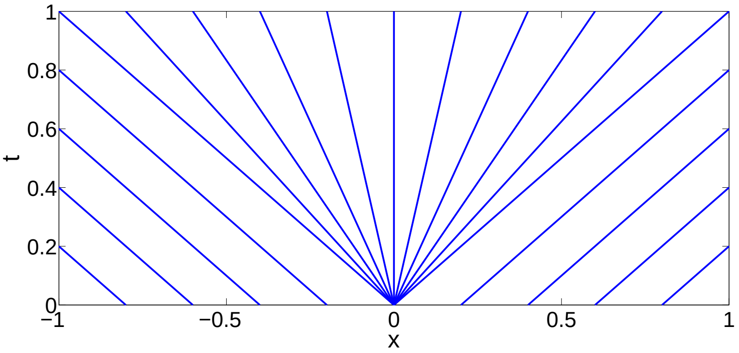


Figure 2.5: Characteristic plot to the rarefaction fan solution (2.28). This solution satisfies the entropy condition (2.24) and is therefore the physically valid solution.

It is clear that this solution (2.28) does not have an entropy violating discontinuity for any time $t \neq 0$, as is required to be the entropy solution.

2.1.3 Numerical Difficulties

In the previous section we considered some of the mathematical difficulties which arise when attempting to solve hyperbolic conservation laws. In this section we consider how these mathematical difficulties cause further numerical difficulties when we attempt to solve the conservation laws with numerical approximations.

Approximating Shocks

The main issue that arises is due to the fact that hyperbolic conservation laws can have discontinuous solutions.

As discussed in section 2.1.2, the differential form of the conservation law (2.1) does not hold at the discontinuity since the solution does not have a derivative at this point. Since many standard numerical approaches are based on the differential form of the conservation law it follows that these are equally poor at approximating the discontinuity.

Consider finite difference methods as an example. In Figure 2.6 we have plotted the results for the first order upwind method, the second order Lax-Wendroff scheme and the exact solution. More information concerning these schemes can be found in [LeV92].

The schemes shown in Figure 2.6 are representative of schemes of their respective orders. The first order upwind scheme shows that numerical diffusion leads to a smoothed out discontinuity and therefore poor accuracy in the surrounding area. This is very common behaviour in first order schemes and often leads to a need for higher accuracy. The second order Lax-Wendroff method captures the discontinuity very effectively but at the cost of introducing spurious oscillations. As with the

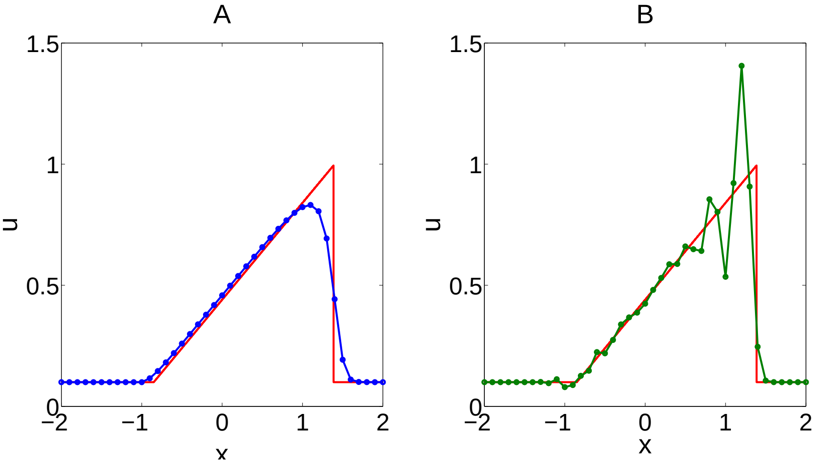


Figure 2.6: Finite difference approximations to the Inviscid Burgers' example problem introduced in Section 3.6 at the post-shock time regime $t = 1.5$. (A) shows the first order upwind scheme (blue) compared with the exact solution (red). (B) shows the second order Lax-Wendroff Scheme (green) compared with the exact solution (red).

numerical diffusion in the first order upwind case these oscillations are typical for higher order methods and can lead to instability in numerical solutions.

The approximation of discontinuities or 'shocks' is further complicated by the fact that numerical schemes may converge to an incorrect solution. In section 2.1.2 it was shown that there is not a unique solution to the weak form of the conservation law (2.22) unless an extra condition is applied. It is possible that unless care is taken numerical approximations of the conservation law (2.1) may converge to these physically invalid solutions.

There is a simple condition which is sufficient to guarantee that numerical solutions have the correct shock speeds. This requirement is simply that the numerical scheme must be able to be written in conservation form.

Definition 2.1.4 (Conservation Form)**.** A scheme is said to be in conservation form if it has the form

$$u_j^{n+1} = u_j^n - \frac{\Delta t}{\Delta x}\left(F(u_{j-p}^n, u_{j-p+1}^n, \cdots, u_{j+q}^n) - F(u_{j-p-1}^n, u_{j-p}^n, \cdots, u_{j+q-1}^n)\right) \quad (2.29)$$

where F is the numerical flux function with $p+q+1$ arguments and is consistent with the flux function $f(u)$ in the sense that $F(u, u, ..., u) = f(u)$. This form is a discrete equivalent to the balance law (2.5) as shown in [MM05].

The above definition however only ensures that discontinuities move with the correct speed and does not ensure that the scheme converges to the entropy solution. To ensure convergence to the entropy solution a monotone scheme or an e-scheme is required. These schemes are discussed in greater detail in [Tor99].

In this section we have discussed the issues that surround the attempt to accurately approximate discontinuities in the solution. The fact that higher order schemes, which better approximate these discontinuities, can lead to instability motivates discussion of the next numerical difficulty.

Scheme Stability

As mentioned in the previous section, first order schemes often end up being insufficiently accurate around discontinuities in the solution whereas higher order schemes tend to develop spurious oscillations which can lead to instability.

Since the issue of spurious oscillations arises frequently, many methods have been developed in an attempt to mitigate their effect. One such attempt is to define the notion of a monotonicity preserving scheme. These are schemes which do not allow new extrema in the solution to form and are therefore non-oscillatory. The formal definition of monotonicity preserving schemes is as follows.

Definition 2.1.5 (Monotonicity Preserving Scheme)**.** [Wes01] A numerical scheme is said to be monotonicity preserving if for every non-decreasing (non-increasing) initial condition $u^0(x)$ the numerical solution at all later instants $u_j^n, n \in \mathbb{N}$ is non-decreasing (non-increasing).

While definition 2.1.5 does help describe schemes that have the desired property of not introducing spurious oscillations, it is also not very useful for actually determining if a scheme is monotonicity preserving or not. To this end a variety of methods have been developed to test if a scheme is monotonicity preserving. In this section we only focus on the stricter condition that schemes are Total Variation Diminishing (TVD).

Definition 2.1.6 (Total Variation)**.** The Total Variation of a numerical solution at time $t = n\Delta t$ is given by

$$TV(u^n) = \sum_j \left| u_j^n - u_{j-1}^n \right|. \tag{2.30}$$

The total variation can easily be seen to increase if the solution is oscillatory and decrease if the solution becomes strictly increasing (decreasing). The fact that oscillations are not desired motivates the notion of a TVD scheme.

Definition 2.1.7. A scheme is defined to be Total Variation Diminishing (TVD) if

$$TV(u^{n+1}) \leqslant TV(u^n) \quad \forall n, \tag{2.31}$$

where the total variation is given by definition 2.1.6.

As in the case of monotonicity preserving schemes, the definition of TVD is not very helpful in determining if a specific scheme is TVD or not. The benefit of the TVD framework is a result by Harten [Har83] which gives certain conditions a scheme must meet to be TVD.

Theorem 2.1.8 (Harten's Theorem)**.** *If a numerical scheme can be written in the form*

$$u_j^{n+1} = u_j^n - C_{j-1/2}(u_j^n - u_{j-1}^n) + D_{j+1/2}(u_{j+1}^n - u_j^n)$$

where $C_{j-1/2} \geqslant 0$, $D_{j+1/2} \geqslant 0$ and $1 - C_{j-1/2} - D_{j-1/2} \geqslant 0$ then the scheme is Total Variation Diminishing (TVD).

Theorem 2.1.8 provides a method for easily determining if a scheme will be not oscillatory however, this does not solve all the problems with numerically solving conservation laws.

The main obstacle to having very accurate numerical approximations without oscillation problems comes from a theorem by Godunov [God59].

Theorem 2.1.9 (Godunov's Order Barrier Theorem)**.** *Linear numerical schemes for solving conservation laws which have the property of being monotonicity preserving, can be at most first order.*

Theorem 2.1.9 provides a simplification of the result obtained by Godunov, however it importantly highlights that any linear scheme found to be TVD via Harten's theorem 2.1.8, say, can at most be first order.

2.1.4 Further Reading

The previous parts of this section have focused on some of the important results in the field of hyperbolic conservation laws required in this work. This is by no means an exhaustive background in the field as many classical topics such as in-depth discussion of the Riemann problem and the CFL condition have been omitted.

Readers wishing to learn more about hyperbolic conservation laws and the numerical methods associated with them have plenty of resources to consider. Further information on the hyperbolic partial differential equations can be found in [Eva10], [Daf10] and [LeF02], while [LeV92], [Tor99] and [Wes01] provide a greater background in the numerical methods.

At the end of section 2.1.3 we discussed how Godunov's Order Barrier Theorem is a limit on traditional linear numerical methods for solving PDEs. As a result many of the more recent developments have been concerned with special schemes known as 'high-resolution schemes'. These schemes have the following properties:

- They are nonlinear.
- They have second order or higher spatial accuracy in smooth regions of the solution.
- Solutions obtained do not have spurious oscillations.
- High accuracy is obtained around discontinuities.
- They require a smaller number of computational nodes when compared with similar accuracy first order schemes.

There have been several notable high-resolution approaches which have been widely adopted and do not rely on a specific mesh.

The first involves the use of flux/slope limiters which take a weighted average of a high order scheme and a first order scheme, depending on how smooth the solution is locally. While flux and slope limiters arise from different approaches to solving the problems surrounding Godunov's order barrier theorem, they take a similar mathematical form. More in-depth discussion of flux/slope limiters can be found in [vL79], [Swe84] and [GL88].

Another type of high-resolution scheme which has been widely used comprises Essentially Non-Oscillatory (ENO) and Weighted Essentially Non-Oscillatory (WENO) schemes. These schemes work by allowing the size of the computational stencil to vary to control oscillations. Both types of scheme generate several candidate stencils in an attempt to minimise oscillations, ENO taking the least oscillatory while WENO takes a linear combination of the candidates. Further reading on these schemes can be found in [HEOC87] and [Shu09].

2.1.5 Example Conservation Laws

In this section we briefly introduce some conservation laws which will be used as examples in the rest of the thesis.

Note that in this section we will denote $u(x,t)$ by u for ease of reading.

Linear Advection Equation

The simplest conservation law that we consider is the Linear Advection Equation. This equation models fluid with a constant flow and is given in differential form by

$$u_t + au_x = 0, \tag{2.32}$$

where a is the constant fluid velocity.

The fact that the linear advection equation is linear makes it a useful first test problem for schemes since exact solutions are easy to calculate based on the method of characteristics discussed in section 2.1.2. It is trivial to see that

$$f'(u) = a \tag{2.33}$$

which makes the characteristic trajectories independent of the solution.

Inviscid Burgers' Equation

The next equation we consider is the Inviscid Burgers' equation. This equation is arguably the simplest nonlinear conservation law and arises in many areas of applied mathematics including fluid mechanics, nonlinear acoustics, gas dynamics and traffic flow.

The flux function for Inviscid Burgers' equation is $f(u) = \frac{1}{2}u^2$ which yields the differential form of the conservation law

$$u_t + \left(\frac{1}{2}u^2\right)_x = 0. \tag{2.34}$$

We use this conservation law to highlight the fact that smooth initial conditions and boundary conditions can lead to discontinuous solutions, as in Section 2.1.2, where we note that it has the associated quasilinear differential form

$$u_t + uu_x = 0. \tag{2.35}$$

This shows that the characteristic trajectories are given by $f'(u) = u$ and are therefore dependent on the value of the solution carried by the characteristic.

Inviscid Burgers' Equation has many known exact solutions making it a useful test case for numerical methods. Importantly, piecewise linear initial data and boundary conditions are easily incorporated via the method of characteristics. Furthermore, a smooth solution for a sine wave initial condition up to shock formation time is given in [GMP15].

Buckley-Leverett Equation

The final conservation law that we consider is the Buckley-Leverett Equation. The equation arises in two phase flow in porous media and is commonly used as a benchmark problem by the oil industry to model oil recovery via water-drive in 1D horizontal flow [VDPP07]. In this oil recovery example the solution u represents the saturation of water and therefore must lie between 0 and 1.

The flux function for the Buckley-Leverett Equation is given by

$$f(u) = \begin{cases} \dfrac{u^2}{u^2 + M(1-u)^2} & 0 \leqslant u \leqslant 1 \\ 0 & u < 0 \\ 1 & u > 1 \end{cases}, \tag{2.36}$$

where $M > 0$ is a fixed constant.

The equation is of particular interest since it is both used industrially and has a non-convex flux function. The flux function is of particular note since it means that initial discontinuities can split into both a shock and a rarefaction fan.

2.2 R-Refinement Methods

In the previous Sections we discussed hyperbolic conservation laws and some of the numerical methods used to attempt to solve them. One area that was not discussed was the notion of adaptive computational meshes as a method for attempting to solve them.

In this section we focus on relocation refinement (r-refinement) methods and give a brief history of some of the different ways in which they have been implemented.

Note that in this section we consider refinement methods for general PDEs instead of only focusing on conservation laws.

2.2.1 Motivation

The solutions of time dependent PDEs often have features which evolve significantly as time progresses. These feature include interfaces, shocks, singularities, change of phase, high vorticity and regions of complexity [BHR09]. Examples of such structures appear in a plethora of applications including fluid dynamics, conservation laws, free boundary problems, combustion, meteorology and mathematical biology. The evolution of these features often happens over short time scales in very fine regions of space and as such a computational mesh must be at least as fine to be able to capture this behaviour.

Using a uniform mesh to solve a problem with complex features is clearly not advisable since in order to resolve the fine grain features a small mesh spacing, Δx, is required but this is computationally inefficient away from such structures. Instead adaptive methods are applied which attempt to refine/coarsen the mesh as required. These adaptive methods generally fall into three categories, h-refinement, p-refinement and r-refinement.

The h-refinement methods are the most commonly applied type of adaptive mesh and is named after the widespread use of the notation $h = \Delta x$. Such methods usually start with a uniform mesh and locally coarsen or refine the mesh by removing or adding in mesh points, respectively. This is often achieved by considering some a posteriori estimate of the solution error and setting tolerances to indicate where nodes should be introduced or removed.

The p-refinement methods are only applicable to finite element methods (FEM) and stands for polynomial refinement. In p-refinement methods a finite element discretisation of the PDE is applied with local polynomials of some particular order. This order is then increased/decreased with regard to some a posteriori solution error. It is possible to combine h-refinement and p-refinement methods to generate hp-refinement methods which are explored in [AO97].

In r-refinement (relocation refinement) methods the computational mesh is allowed to move in the hope that the mesh refinement/coarsening will occur automatically without the need to add or remove computational nodes. The mesh movement is often dictated by some function of the solution in the hope that this will cause the computational nodes to gather in regions where a small spatial step is required and separate in regions where the solution is changing very little. These methods are not as widely used as either h-refinement or p-refinement methods but have been successfully applied to a variety of different applications including computational fluid mechanics [Tan05], convective heat transfer [CH01] and mathematical biology [LBLT13].

The main downside of r-refinement methods is that allowing the mesh to move introduces some problems which static meshes do not have namely, the mesh can tangle. Mesh tangling can occur in multiple ways but the main two are node crossing, where one computational node passes another, and mesh vorticity, where the mesh starts to spiral in on itself causing the connectivity of the mesh and the location of the nodes to be incompatible. Node crossing is often caused by a poor choice of time step while mesh vorticity is often caused by vorticity is the solution making it difficult to avoid.

The methods that will be studied in this thesis are r-refinement methods and as such this section will give a brief overview of other moving mesh schemes.

2.2.2 Useful Tools

In this section we introduce a few mathematical tools which are commonly used in r-refinement methods.

The first of these is the notion of a monitor function. As noted in the previous section, r-refinement methods often require some function of the solution to guide the mesh evolution. A monitor function, $m(u, u_x, u_{xx}, ...)$, is commonly used as part of the mesh evolution.

Some examples of monitor functions are the density monitor

$$m(u) = u \tag{2.37}$$

and the arc length monitor

$$m(u_x) = \sqrt{1 + (u_x)^2}. \tag{2.38}$$

Another important mathematical tool is the notion of an equidistribution principle. This is applied to a general monitor function and is important for many established results in moving mesh methods.

Definition 2.2.1 (Equidistribution Principle)**.** [dB73] The equidistribution principle states that for $0 \leqslant \chi \leqslant 1$,

$$\int_{a(t)}^{\hat{x}(\chi,t)} m(u, u_x, u_{xx}, ...)dx = \chi \int_{a(t)}^{b(t)} m(u, u_x, u_{xx}, ...)dx \tag{2.39}$$

where $a(t)$ is the left hand boundary of the domain, $b(t)$ is the right hand boundary and $\hat{x}(\chi, t)$ is a moving point in the interval $[a(t), b(t)]$.

The equidistribution principle shows that the variable χ can be seen as the fractions of the total monitor integral between $a(t)$ and the moving point $\hat{x}(\chi, t)$. This implies that the interval $[a(t), b(t)]$ can be divided up into a partition of subintervals all with the property that the monitor integral evaluated over them is equivalent.

Having introduced these useful mathematical tools we can now introduce some of the different types of r-refinement methods.

2.2.3 Types of Moving Mesh Methods

In section 2.2.1 we discussed how adaptive mesh refinement methods could be separated into three categories, h-refinement, p-refinement and r-refinement. In a similar way r-refinement methods may be further separated into two subcategories, location based methods and velocity based methods [CHR03].

In this section we will discuss how the two categories differ and give some examples of schemes in each category.

Location Based Methods

The premise of location based moving mesh methods is that the location (or density) of the computational nodes at a given time is found by solving a mesh equation. Typically this mesh equation takes the form of some nonlinear differential equation which is often hard to solve. The benefit of these methods is that they often lead to good global mesh properties such as avoiding the mesh tangling mentioned in the previous section and avoiding excessive skewness in computational cells. Three examples of location based methods are MMPDE-based methods, variational methods and optimal transport methods.

The most studied of the location based methods are the MMPDE methods with a large body of the work being done by Huang *et al.* [HRR94], [BHR96], [HR97] and [HR00]. An example of a variational method may be found in [CH01] and discussion on the optimal transport methods can be found in [BHR09].

Velocity Based Methods

The velocity based methods, also referred to as Lagrangian methods, rely on calculating the velocities of individual computational nodes and using this to update the node positions. These methods are often far simpler to implement than the location based methods but generally do not have the good global properties that are seen with the location based methods.

The main methods in this category are geometric conservation law (GCL) methods, moving mesh finite element methods, Arbitrary Lagrangian-Eulerian (ALE) methods and conservation methods. All of these methods are closely related to the concept of monitor conservation as a tool for finding mesh velocity.

In the GCL methods the mesh velocity is found by using the GCL as a minimisation problem in order to discretise. The term Geometric Conservation Laws was first coined in [TL79] and further work has been has been done by Cao, Huang and Russell [CHR02] and Baines, Hubbard and Jimack [BHJ11].

The conservation methods and the moving mesh finite element methods use very similar methods for finding the mesh however differ on how to recover the solution on this mesh. A large body of work has been done on the moving mesh finite element methods by Baines, Hubbard and Jimack [BHJ04], Baines [Bai94], [Bai98] and in the theses of Wells [Wel04] and Lee [Lee11]. Since the methods described in this thesis are conservation methods we leave further discussion of them until Section 2.3.

The final velocity based moving mesh scheme considered here is the Arbitrary Lagrangian-Eulerian (ALE) methods. These methods are again very similar to the other methods in this class, using monitor conservation to find the mesh velocity. These methods differ however as they allow for arbitrary velocities to be applied. Examples of ALE methods can be found in [KHDB03] and [SDP07].

2.3 Lagrangian Framework

In the previous section we introduced adaptive r-refinement methods and gave a brief overview of some of the different moving mesh methods that have been developed. At the end of the section we noted that the schemes that are studied in this thesis are a type of velocity based moving mesh scheme known as conservation methods.

In this section we detail how these methods arise, give an example of a particular method applied to a test problem and note further reading which may be of interest.

2.3.1 Derivation

The first step in solving a problem using these conservation methods is to first rewrite the PDE in the Lagrangian formulation. Start by choosing a monitor function, $m(u, u_x, u_{xx}, ...)$, and consider the associated monitor integral

$$M(u, u_x, u_{xx}, ... : \hat{x}_1(t), \hat{x}_2(t)) = \int_{\hat{x}_1(t)}^{\hat{x}_2(t)} m(u, u_x, u_{xx}, ...)dx, \qquad (2.40)$$

where $\hat{x}_1(t)$ and $\hat{x}_2(t)$ are moving coordinates. The moving coordinates are defined to be such that the monitor integral remains constant in time hence,

$$\frac{d}{dt}\int_{\hat{x}_1(t)}^{\hat{x}_2(t)} m(u, u_x, u_{xx}, ...)dx = 0. \qquad (2.41)$$

Since $\hat{x}_1(t)$ and $\hat{x}_2$ are only defined to move so that equation (2.41) holds only their velocity is prescribed. This implies that such coordinates can be arbitrary in the sense of initial starting position and hence, the region being considered could be the entire domain or an arbitrary time dependent subregion.

Leibniz integral rule [Fla73] can be applied to the left hand side of equation (2.41) in order to take the time derivative inside the integral,

$$\int_{\hat{x}_1(t)}^{\hat{x}_2(t)} m_t dx + [m\hat{x}(t)_t]_{\hat{x}_1(t)}^{\hat{x}_2(t)} = 0 \qquad (2.42)$$

where the dependent variables of the monitor function have been omitted for ease of reading.

Depending on the form of the monitor function either the Eulerian PDE (2.1) or the balance law (2.5) can be used to replace the m_t term in the integrand of equation (2.42). This is not trivial to do for a general monitor function but for a given monitor function equation (2.42) can be rewritten into a formula for $\hat{x}(t)_t$.

It is noted that the manipulation of equation (2.42), required to find the correct conservative velocity, includes having to divide by $m(u, u_x, u_{xx}, ...)$ and for this reason the monitor function should be non-zero.

Together the conservation of the monitor integral (2.40) and the conserving velocity $\hat{x}(t)_t$ form the Lagrangian formulation. To derive a specific scheme these two equations are then discretised.

Remark 2.3.1. Note that while the above discussion assumes that the monitor is conserved over the domain it is also possible to apply the method to problems in which the monitor function is not conserved. This is achieved by considering the monitor integral over a subregion relative to the monitor integral over the entire domain. In this way a subregion can be seen to conserve a fraction of the total monitor integral. This method is covered in [LBL15] for the density monitor function.

2.3.2 An Example

In the previous section we briefly discussed how to derive the Lagrangian formulation of a general monitor function. Since the conserving velocity is difficult to write down explicitly without making assumptions on the monitor integral we consider an example problem.

Consider the Porous Medium Equation given by

$$u(x,t)_t = (u(x,t)u(x,t)_x)_x \tag{2.43}$$

with appropriate initial conditions and boundary conditions [Aro].

Since mass is conserved for the PDE (2.43) we will consider the density monitor

$$m(u) = u, \tag{2.44}$$

which leads to the the conservation of mass

$$\frac{d}{dt}\int_{\hat{x}_1(t)}^{\hat{x}_2(t)} u dx = 0. \tag{2.45}$$

Applying Leibniz integral rule to equation (2.45) gives

$$\int_{\hat{x}_1(t)}^{\hat{x}_2(t)} u_t dx + [u\hat{x}_t]_{\hat{x}_1(t)}^{\hat{x}_2(t)} = 0 \tag{2.46}$$

and using the PDE (2.43) yields

$$\int_{\hat{x}_1(t)}^{\hat{x}_2(t)} (uu_x)_x dx + [u\hat{x}_t]_{\hat{x}_1(t)}^{\hat{x}_2(t)} = 0. \tag{2.47}$$

Finally applying the fundamental theorem of calculus [CJ12] leads to

$$[uu_x + u\hat{x}_t]_{\hat{x}_1(t)}^{\hat{x}_2(t)} = 0, \tag{2.48}$$

which has a solution if

$$\hat{x}_t = -u_x. \tag{2.49}$$

Together the mass conservation (2.45) and the conservative velocity (2.49) form the mass conservative Lagrangian formulation for the porous medium equation (2.43), then all that remains is to discretise both of these equations.

Approximating the conservation of mass (2.45) by the trapezium rule yields

$$A_{j-1/2} = \frac{1}{2}(\hat{x}_j^n - \hat{x}_{j-1}^n)(u_j^n + u_{j-1}^n), \tag{2.50}$$

where $A_{j-1/2}$ is the local mass constant.

Using a discretisation of the velocity (2.49) and inserting it into the Forward Euler methods gives

$$\hat{x}_j^{n+1} = \hat{x}_j^n - \Delta t \frac{u_j^n - u_{j-1}^n}{\hat{x}_j^n - \hat{x}_{j-1}^n}. \tag{2.51}$$

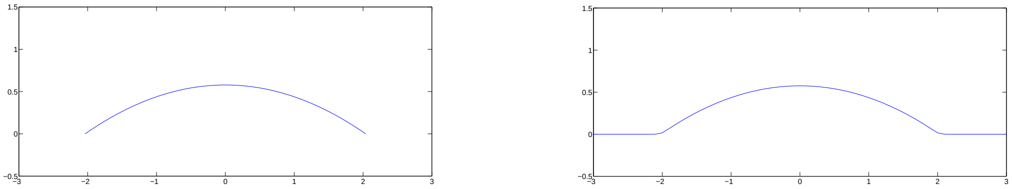
In order to test the scheme we consider the initial condition,

$$u(x,0) = \begin{cases} (1-x^2)^{\frac{1}{2}} & |x| \leqslant 1 \\ 0 & \text{otherwise} \end{cases}, \tag{2.52}$$

with far field boundary conditions.

Note that by ‘far field’ boundary conditions we mean that the boundaries are far away from the behaviour we are interested in and the solution is essentially constant near these boundaries.

Figure 2.7 shows the results of the moving mesh scheme and the results from the Eulerian Crank-Nicolson scheme [CN47] when run with the same number of nodes and the same timestep for comparison.



(a) Moving Mesh Scheme (2.50)-(2.51)

(b) Crank-Nicolson Scheme

Figure 2.7: Numerical comparison of the moving mesh scheme and the Eulerian Crank-Nicolson scheme when both applied to the Porous Medium Equation with initial condition (2.52) and far field boundary conditions.

The key note from this example is that the Eulerian scheme required us to have a larger initial domain to accommodate for the fact that the solution diffuses outward. In contrast the moving mesh scheme could move with the boundary of the diffusion and therefore did not waste nodes outside of this region.

2.3.3 Further Reading

In this section we have given an overview of the velocity based moving mesh method by paying particular attention to Eulerian PDE's of the form (2.1).

Lagrangian methods are all closely related and often use very similar mesh movement techniques but vary in how the solution is recovered on this mesh. The conservative methods which are studied in this thesis have been applied to diffusion problems [Bir14], general conservation laws [SMR01], hydrocodes [Col13], the Sod shock tube problem [DD87], ice sheets [BBNP16] and many more.

Less conventional conservative methods have been developed in recent years. In [BL14] the conservative method is extended to cope with non-conservative problems and in [CS07] the conservative mesh is used with a WENO/ENO type scheme used to recover the solution of the mesh.

Chapter 3

Mass Conserving Moving Mesh Methods for Conservation Laws

In this section we set out the general class of schemes that we consider in this thesis. This is done by demonstrating how to derive the mass based Lagrangian formulation for a general conservation law and providing a generalised numerical approximation to the resulting equations.

Recall from section 2.1 that the scalar hyperbolic conservation law with solution $u(x,t)$ is given by

$$u_t + f(u)_x = 0, \quad x \in (a(t), b(t)), \quad t \in \mathbb{R}^+, \tag{3.1}$$

$$u(x,0) = u^0(x), \quad x \in (a(0), b(0)), \tag{3.2}$$

$$u(a,t) = u^a(t), \quad t \in \mathbb{R}^+. \tag{3.3}$$

We assume that $a(t)$ is the inflow boundary, as defined in section 2.1, leading to a boundary condition only being required here. This assumption does not lead to a loss of generality and the results of this section follow similarly under the assumption that $b(t)$ is the inflow boundary.

3.1 Lagrangian Formulation

Following the same procedure as in Section 2.3 requires choosing a monitor function. Since a mass conserving scheme is desired an obvious choice of monitor function is the density function,

$$m(u) = u(x,t). \tag{3.4}$$

As in section 2.3 this imposes a restriction on the problems that can be solved with a conservation-based moving mesh method. Namely, only problems where

$$u(x,t) > 0, \quad \forall x, t, \tag{3.5}$$

may be considered.

The choice of monitor function (3.4) leads to the monitor integral

$$M(u, \hat{x}_1(t), \hat{x}_2(t)) = \int_{\hat{x}_1(t)}^{\hat{x}_2(t)} u(x,t)dx, \tag{3.6}$$

which is required to be constant in time for $\hat{x}_1(t)$ and $\hat{x}_2(t)$ moving with appropriate velocity. Hence,

$$\frac{d}{dt}\int_{\hat{x}_1(t)}^{\hat{x}_2(t)} u(x,t)dx = 0. \tag{3.7}$$

It follows that by applying Leibniz integral rule to $\frac{dM}{dt}$ that,

$$\frac{d}{dt}\int_{\hat{x}_1(t)}^{\hat{x}_2(t)} u(x,t)dx = \int_{\hat{x}_1(t)}^{\hat{x}_2(t)} u_t dx + \left[u(x,t)\hat{x}_t\right]_{\hat{x}_1(t)}^{\hat{x}_2(t)}, \tag{3.8}$$

and appealing to the conservation law (3.1) further yields

$$\frac{d}{dt}\int_{\hat{x}_1(t)}^{\hat{x}_2(t)} u(x,t)dx = -\int_{\hat{x}_1(t)}^{\hat{x}_2(t)} f(u)_x dx + \left[u(x,t)\hat{x}_t\right]_{\hat{x}_1(t)}^{\hat{x}_2(t)}. \tag{3.9}$$

Application of the fundamental theorem of calculus gives

$$\frac{d}{dt}\int_{\hat{x}_1(t)}^{\hat{x}_2(t)} u(x,t)dx = \left[u(x,t)\hat{x}_t - f(u)\right]_{\hat{x}_1(t)}^{\hat{x}_2(t)}. \tag{3.10}$$

Note that equation (3.10) is a generalised balance law which we call the 'Lagrangian balance law'. This balance law has a different flux function associated with it that we call the 'net flux',

$$\text{Net Flux} = f(u) - u\hat{x}_t. \tag{3.11}$$

It is clear from equation (3.10) and the time independence requirement (3.7) that

$$\left[u(x,t)\hat{x}_t - f(u)\right]_{\hat{x}_1(t)}^{\hat{x}_2(t)} = 0. \tag{3.12}$$

Let $\hat{x}_1 = a(t)$ be the position of the inflow boundary and $\hat{x}_2 = \hat{x}, \hat{x} \in (a(t), b(t))$ be an arbitrary moving coordinate, with velocity $\hat{x}_t$. It follows from equation (3.12) that

$$u^a(t)\ \hat{x}_t\big|_a - f(u^a) = u(\hat{x},t)\hat{x}_t - f(u). \tag{3.13}$$

Rearranging for $\hat{x}_t$ yields

$$\hat{x}_t = \frac{f(u) + u^a(t)\ \hat{x}_t|_a - f(u^a)}{u(\hat{x},t)}, \tag{3.14}$$

where the division by $u(\hat{x},t)$ is justified by the restriction (3.5).

Together equations (3.7) and (3.14) form the general Lagrangian formulation

$$\frac{d}{dt}\int_{\hat{x}_1(t)}^{\hat{x}_2(t)} u(x,t)dx = 0, \tag{3.15}$$

$$\hat{x}_t = \frac{f(u) + u^a(t)\ \hat{x}_t|_a - f(u^a)}{u(\hat{x},t)}. \tag{3.16}$$

3.2 Boundary Conditions

In the previous section we derived the general Lagrangian formulation (3.15)-(3.16) of the general conservation law (3.1). We note that the general form of the moving coordinate velocity (3.16) requires both knowledge of the solution, $u(x,t)$, and the position, $a(t)$, of the inflow boundary.

In this section we consider some example boundary conditions that can be prescribed for various problems, namely, a fixed inflow boundary for comparison with fixed mesh schemes, a boundary moving with characteristic velocity as described in section 2.1.2 and, the 'natural' Lagrangian boundary condition.

3.2.1 Fixed Inflow Boundary

First consider a standard conservation problem for a fixed mesh scheme to solve. This will take the form of equations (3.1)-(3.3) where $a(t)$ and $b(t)$ are constant in time.

Since the inflow boundary $a(t)$ is fixed it is clear that $\hat{x}_t|_a = 0$ and hence the velocity (3.16) simplifies to become

$$\hat{x}_t = \frac{f(u) - f(u^a)}{u(\hat{x}, t)}. \tag{3.17}$$

Note further that the standard Eulerian problem formulation fixes both the inflow boundary, $a(t)$, and the outflow boundary, $b(t)$. This defines a fixed 'volume' and as a consequence cannot guarantee global mass conservation for a given initial condition.

In the Lagrangian formulation global mass conservation is required and therefore we cannot fix the 'volume'. This implies that since the inflow boundary, $a(t)$, is the only given boundary condition and fixed we must allow $b(t)$ to remain free to move as prescribed by the mass conservation. This is prescribed by

$$\hat{x}_t|_b = \frac{f(u^b) - f(u^a)}{u^b(t)}. \tag{3.18}$$

Therefore, in the Lagrangian formulation the net flux across all such coordinates must be equal at any given time, t. This means that the net flux of the inflow boundary condition determines the net flux for every other $\hat{x}$ which moves with a consistent mass conserving velocity. In the case of the fixed inflow boundary, equation (3.12) tells us that the net flux for any coordinate must be equivalent to $f(u^a)$.

3.2.2 Boundary with Characteristic Velocity

The next case to consider is the special case where the inflow boundary is prescribed to move with the characteristic velocity. This means that the inflow boundary follows the characteristic starting at the same point exactly.

Recall from section 2.1.2 that on characteristic lines the solution $u(x,t)$ remains constant and the characteristic velocity can be found by

$$x'(t) = f'(u). \tag{3.19}$$

Substituting the given boundary data into equation (3.19) yields

$$\hat{x}_t\big|_a = f'(u^a), \tag{3.20}$$

which may be further substituted into equation (3.14) to give the general coordinate velocity

$$\hat{x}_t = \frac{f(u) - \omega}{u(\hat{x},t)}, \tag{3.21}$$

where $\omega = f(u^a) - u^a(t)f'(u^a)$ is prescribed by the given boundary condition and remains constant in time since u^a is constant on the characteristic line.

As in the case of a fixed inflow boundary condition, the outflow boundary position is determined by mass conservation. In this case equation (3.21) yields the outflow boundary velocity to be

$$\hat{x}_t\big|_b = \frac{f(u^b) - \omega}{u^b(t)}. \tag{3.22}$$

Note that for this given boundary condition the associated net flux across any coordinate moving with the 'correct' velocity (3.21) is ω.

3.2.3 Free Lagrangian Boundary Conditions

The final boundary condition we consider is called the 'free Lagrangian' boundary condition. In this case the boundary condition is allowed to arise naturally from the Lagrangian formulation of the problem through a zero net flux.

Recall equation (3.12),

$$[u(x,t)\hat{x}_t - f(u)]_{\hat{x}_1(t)}^{\hat{x}_2(t)} = 0. \tag{3.23}$$

This was previously used to find the correct velocity of an arbitrary coordinate to give mass conservation between two such coordinates. This was done by setting the net flux at each coordinate to be equivalent to the net flux at the given inflow boundary.

The 'free Lagrangian' velocity is found by setting all net fluxes, including at the inflow boundary, equal to 0. In this way the net flux sets the boundary condition, in contrast with previous examples where the boundary condition set the net flux.

Since it is required that $u^a(t)\ \hat{x}_t|_a - f(u^a) = 0$, it follows directly from equation (3.14) that the general coordinate velocity in this case is simply given by

$$\hat{x}_t = \frac{f(u)}{u(\hat{x},t)}, \tag{3.24}$$

where it is noted that there is no dependence on the inflow boundary.

Inserting our outflow boundary into equation (3.24) gives the outflow boundary velocity to be

$$\hat{x}_t|_b = \frac{f(u^b)}{u^b(t)}. \tag{3.25}$$

Note that for the rest of this thesis we will only consider 'free Lagrangian' boundary conditions. This does not lead to a loss of generality of any of the results contained and there are several example problems with other boundary conditions to demonstrate this.

3.2.4 Limits on the Boundary Velocity

Having given three examples of boundary conditions, the question remains as to whether there are any limits on how the boundary conditions may be prescribed.

Section 2.1 discussed when and where boundary conditions should be applied to standard Eulerian hyperbolic conservation laws, namely they should only be applied to inflow boundaries. To determine if a boundary is an inflow or outflow, consider the expression

$$-\mathbf{n} \cdot f'(u)|_{\partial\Omega} \tag{3.26}$$

where $\partial\Omega$ is the point on the boundary you are considering and $\mathbf{n}$ is the normal unit vector which leaves the domain. Note that in our 1D case $\mathbf{n}$ is simply -1 if we are considering the left hand boundary and 1 if we are considering the right hand boundary.

The expression (3.26) can be used to determine if a boundary is inflow or outflow by calculating whether it is positive or negative. If (3.26) is positive then the boundary is an inflow and if (3.26) is negative then it is an outflow.

Remark 3.2.1. If the expression (3.26) is equal to 0 then we are in the special case where the boundary lies exactly on a characteristic line. In this case the fact that $u(x,t)$ must be constant along characteristics implies that the boundary condition is determined by the initial condition. Further note that this does not make the problem solely an IVP since the other boundary could still be an inflow boundary.

It is natural to ask if a similar function can be found for the Lagrangian formulation of the problem, *i.e.* is it possible to find a function to replace $f'(u)$ which changes sign for the two types of boundary?

We propose that a suitable function is

$$f'(u) - \hat{x}_t \tag{3.27}$$

where $\hat{x}_t$ is the calculated boundary velocity. This leads to the Lagrangian inflow/outflow expression being given by

$$-\mathbf{n} \cdot \left(f'(u) - \hat{x}_t\right)\big|_{x=a(t)} . \tag{3.28}$$

This function follows naturally when considering the Eulerian fixed boundary case since $f'(u)$ informs us of the velocity of the characteristic trajectories and since the boundary is not moving this is sufficient to determine whether the characteristic lines are entering or leaving the domain. In the Lagrangian moving boundary case however it is not sufficient and we therefore compare the chosen boundary velocity with the characteristic velocity to determine which is moving faster.

Figures 3.1 and 3.2 highlight the two cases where (3.27) is positive and negative leading to an inflow and an outflow boundary respectively.

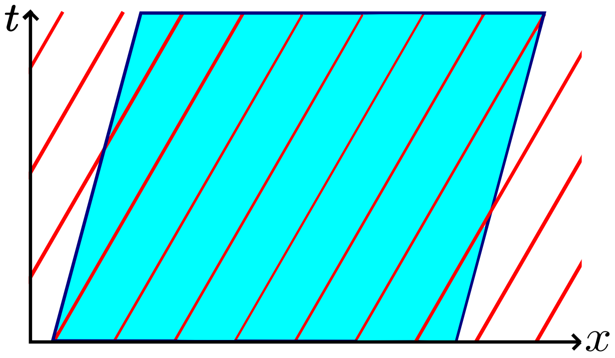


Figure 3.1: A moving domain in blue where the boundary velocity does not exceed the characteristic velocity, hence $f'(u) - \hat{x}_t > 0$ and the left hand boundary is still an inflow boundary.

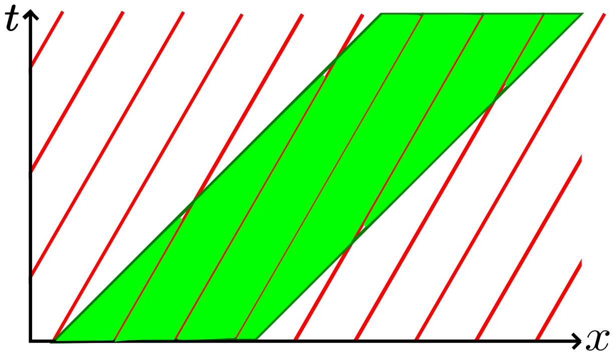


Figure 3.2: A moving domain in green where the boundary velocity exceeds the characteristic velocity, hence $f'(u) - \hat{x}_t < 0$ and the left hand boundary is now an outflow boundary.

Note that as in remark 3.2.1 the case where (3.28) is equal to zero implies that the boundary is moving along a characteristic line and is therefore neither an inflow nor an outflow boundary. This is the characteristic velocity defined earlier in this section.

Remark 3.2.2. It is important to note that given our definition of the Lagrangian inflow/outflow condition (3.28) it may not be possible to apply the 'free Lagrangian' boundary condition to the boundary which would be considered inflow in an Eulerian sense. This is due to the fact that the behaviour of

$$f'(u) - \frac{f(u)}{u} \tag{3.29}$$

changes dramatically depending on the given flux function $f(u)$.

In conclusion we can determine whether a boundary requires any solution information by evaluating the expression (3.28).

3.3 Partitioning the Domain

Now that we have the Lagrangian formulation of the problem in section 3.1 and the boundary conditions in section 3.2, we must now decide on how we will partition the domain of the problem numerically.

Since the Lagrangian formulation is an integral form of the problem it is easy to consider the partition of the domain before actually discretising the Lagrangian formulation. The two choices which must be considered are:

- The structure of the partition, how do the subintervals cover the domain of the problem? Is there any overlapping?
- The initial interval distribution, do we start with a uniform partition or weight it based on the initial condition?

In this section we consider two alternative options for each of these choices. Note however that neither of these sections covers the options available exhaustively and instead focuses on the more promising alternatives.

3.3.1 Standard Partition vs Overlapping Masses

The first choice is what form the partition of the domain will take. It is clear that whatever partition we choose must cover the entire domain, however we may also choose to have overlapping intervals.

The obvious option here is to simply split the domain into a standard partition where each subsequent subinterval of the domain shares a boundary with the previous subinterval and there is no overlapping. In this case the local mass conservation is given by

$$\int_{\hat{x}_j(t)}^{\hat{x}_{j+1}(t)} u dx = A_{j+1/2}, \qquad \forall j < J, \quad j \in \mathbb{N}, \tag{3.30}$$

where $A_{j+1/2}$ is the local mass constant and J is the number of nodes in the partition. This partitioning of the domain is illustrated in Figure 3.3.

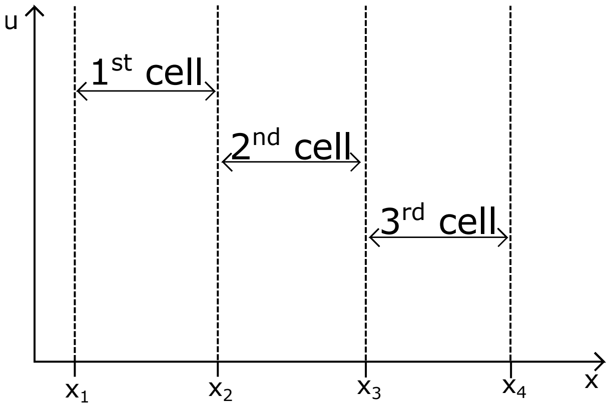


Figure 3.3: The standard partition in which subintervals of the domain do not overlap and consecutive intervals share a boundary.

The other clear option is to instead have overlapping intervals which all relate back to the inflow boundary. In this case the local mass conservation is given by

$$\int_{\hat{x}_1(t)}^{\hat{x}_j(t)} u dx = A_{(j+1)/2}, \qquad \forall j \leqslant J, \quad j \in \mathbb{N}, \tag{3.31}$$

where $A_{(j+1)/2}$ is the mass constant and J is the number of nodes in the partition. Note that in this description the local mass constant A_J is the total mass over the entire domain. Figure 3.4 illustrates this overlapping partition for completeness.

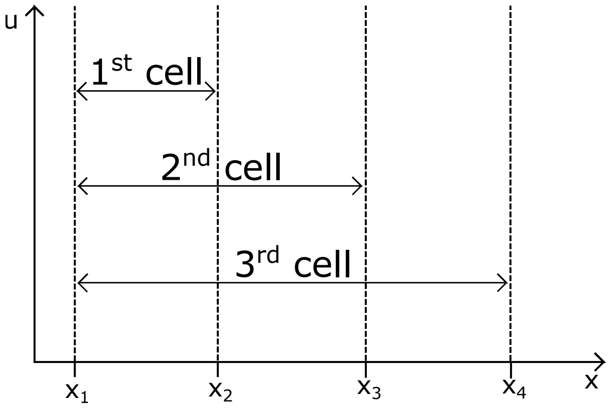


Figure 3.4: The overlapping partition in which subinterval of the domain starts at the inflow boundary of the region.

Now that we have introduced the two main ways of partitioning the domain we leave the choice of which to use until section 3.4 where the consequences of the choice can be more clearly seen.

3.3.2 Choice of Local Mass Constant

The second choice that must be made in regards to partitioning the domain is how to choose the length of the subintervals.

The obvious choice here is to start with a uniform length of intervals. Explicitly, this is defining the intervals such that

$$\hat{x}_{j+1}(0) - \hat{x}_j(0) = l, \qquad \forall j < J, \quad j \in \mathbb{N}, \tag{3.32}$$

where J is the number of nodes in the partition and $l = \frac{\hat{x}_J - \hat{x}_1}{J}$.

While this may be the most obvious choice, we will later find that this leads to some awkwardness. Hence we suggest an alternative based on the equidistribution principle given in section 2.2.2. Here we restrict the local mass constants to be equal so that

$$A_j = A \quad \forall j < J, \quad j \in \mathbb{N}, \tag{3.33}$$

where

$$A = \frac{1}{J-1} \int_{\hat{x}_1(0)}^{\hat{x}_J(0)} u(x,t)dx. \tag{3.34}$$

Note here that this does not imply that A is time dependent as the integral is the mass integral and the boundary values are chosen to be mass conserving.

While both options given in this section are valid, in practice the equidistributed starting partition often leads to better results later on.

3.4 Discretising the Lagrangian Formulation

In this section we focus on how to discretise the Lagrangian formulation. Unlike the approximation of the Eulerian formulation of the problem which only requires the PDE to be discretised, the Lagrangian formulation requires that the two equations (3.15), (3.16) are discretised. These are the local conservation and the partition or mesh velocity.

3.4.1 Mesh Movement

The key idea of the Lagrangian moving mesh method is that the domain is partitioned and the boundaries move such that the local mass in the partitions is conserved. This may be discretised by considering these boundaries as nodes on a computational mesh, leading to a numerical approximation in which the local mass is conserved in each cell.

To compute the mesh movement the conservative velocity (3.16) must be discretised. To approximate this we have choose a general Runge-Kutta approximation

$$\hat{x}_j^{n+1} = \hat{x}_j^n + \Delta t \sum_{i=1}^{s} b_i k_i \tag{3.35}$$

where $k_i = \hat{x}(\hat{x}_j^n + \Delta t \sum_{l=1}^{s} a_{il} k_l, t^n + c_i \Delta t)_t$ and a_{il}, b_i and c_i are the coefficients which define the individual scheme [MM05].

3.4.2 Quadrature Approximation

The discretisation of the local conservation involves making a choice of quadrature approximation to discretise equation (3.15) as well as considering the choice of how the domain is partitioned, which was discussed in the previous section. A general form of quadrature approximation is given by

$$\int_{x_p}^{x_q} u(x)dx \approx (x_q - x_p)\sum_{i=p}^{q} d_i u(x_i), \tag{3.36}$$

where d_i is a set of normalised weights such that $\sum_{i=p}^{q} d_i = 1$.

Consider the overlapping partition given by equation (3.31). In this partition the size of each interval increases the further away from the inflow boundary the moving coordinate we are considering lies. As discussed in section 3.3 the J^{th} interval is the entire domain and it is clear that under such a partition the quadrature must be very high order to accurately represent the mass over this large interval.

Consider instead the standard partition given by equation (3.30). In this case the intervals are much closer in relative size and importantly the size of the interval depends on the local solution $u(x,t)$, instead of how far the outflow boundary of the interval is from the inflow boundary condition. In this case we can take a simple quadrature approximation to the local mass conservation.

The main difference between the two choices of partition are in how we introduce an error. In the standard partition the error is introduced by the fact that the error in each small interval leads to an inaccurate velocity of the local outflow boundary. This in turn is defined to be the inflow boundary of the next interval and hence the error is accumulated as you move further from the inflow boundary condition. The overlapping partition introduces a similar error but this is instead due to the large approximation on the solution $u(x,t)$ as the intervals become larger.

In this thesis we choose to only concern ourselves with the standard partition. This is due to the fact that the error is more easily quantifiable and hence easier to work with. Since we are only considering intervals with solutions known at the nodes we simplify the general quadrature (3.36) to

$$\int_{x_j}^{x_{j-1}} u(x)dx \approx (x_j - x_{j-1})(d_j u(x_j) + d_{j-1}u(x_{j-1})), \tag{3.37}$$

where d_i are weights such that $d_j + d_{j-1} = 1$.

3.5 A General Mass Conservative Moving Mesh Method

In this section we put everything we have developed in the previous section together and finally arrive at a general moving mesh method for solving a general conservation law.

In the rest of the general discussion in this thesis we will assume that we are required to find a strictly positive numerical solution to the following scalar Eulerian conservation law,

$$u_t + f(u)_x = 0, \quad x \in (a, b), \quad t \in \mathbb{R}^+, \tag{3.38}$$

$$u(x, 0) = u^0(x), \quad x \in (a, b), \tag{3.39}$$

$$u(a, t) = \alpha, \quad t \in \mathbb{R}^+, \tag{3.40}$$

where $\alpha > 0$ is a constant and $u(x, t) > 0$.

Following the steps demonstrated in section 3.1 it can be easily shown that the resulting Lagrangian formulation to this Eulerian conservation law (3.38) is given by a mass conservation equation,

$$\frac{d}{dt}\int_{\hat{a}(t)}^{\hat{b}(t)} u dx = 0, \tag{3.41}$$

together with a velocity which defines the motion of the nodes to enforce conservation of mass,

$$\hat{x}_t = \frac{f(u) - \omega}{u}, \tag{3.42}$$

where ω is the net flux which is to be determined by the boundary condition.

Section 3.2 considered what boundary conditions can be given in the Lagrangian formulation and the question of what boundary condition should be applied for this problem is not simple. While the initial problem (3.38)-(3.40) called for solving the problem on a fixed interval (a, b) the Lagrangian mass conservation relies on a moving interval $(\hat{a}(t), \hat{b}(t))$.

As discussed in Section 3.2, it is possible to fix $\hat{a}(t) = a$ and find a resulting net flux ω however this still does not lead to a solution on a fixed interval since in general the resulting outflow boundary will not be stationary, hence $\hat{b}(t) \neq b$. This can be shown to be unavoidable in non-trivial cases where $u(x, t) \neq$ a constant, by considering $u(x, t)$ as a density and making an analogy to the basic physics relationship between mass, volume and density, given by Mass = Volume × Density. In the Eulerian problem (3.38)-(3.40) we are fixing the 'volume' by restricting the domain to be the fixed interval (a, b) while in the Lagrangian framework we are requiring that the 'Mass' is unchanging in time. Clearly the only way that both the 'Mass' and the 'Volume' can remain invariant in time is if the 'Density' is also invariant, $u(x, t) =$ a constant. Therefore for non-trivial problems to be solved we must allow $\hat{b}(t)$ to move.

Remark 3.5.1. It is noted that allowing one of the boundaries to move when the problem specifies a fixed domain may cause alarm for readers who are used to solving such problems using Eulerian methods. An example that may arise is what happens if the PDE (3.38) does not hold beyond b, while the Lagrangian solution allows $\hat{b}(t) > b$ for some t?

This issue can be alleviated to some degree by considering that real world problems that lead to constraints on the physical domain rely on systems of conservation laws as opposed to a single scalar conservation law. Indeed a scalar conservation law is often not suited to having interfaces where behaviour changes, for example a wall or a change in material, and should instead be considered as a restriction of an infinite domain problem where we have truncated the domain for computational purposes rather than physical restraints. In section 6 this is supported by considering a problem of a fixed physical domain and showing that the Lagrangian formulation allows the same fixed domain.

Having demonstrated that the outflow boundary can move, now consider if the inflow boundary is actually restricted to being stationary. Indeed the restriction (3.40) on the boundary condition being constant implies that $u(x,t)$ is constant in any region where the characteristics trace back to the boundary condition. Hence we could choose to apply any boundary that starts at a and has a velocity such that

$$\hat{a}(t) \leqslant f'(\alpha)t + a, \quad \forall t. \tag{3.43}$$

This is because any characteristic line which starts at the boundary carries the same constant value of u. It follows that we may choose any path for the moving boundary to follow in this region without change the boundary condition of the solution.

As long as the prescribed boundary velocity satisfies the condition (3.43) then this boundary will always lie within the region where $u(x,t) = \alpha$ and this is therefore the solution boundary condition which applies for such boundaries.

Assume for ease of notation that the 'free Lagrangian' boundary condition (3.24) satisfies the condition (3.43), i.e. $f(\alpha) \leqslant \alpha f'(\alpha)$. In this case we can simplify the Lagrangian formulation since the net flux is zero. Under this assumption the complete Lagrangian formulation can be written as a local conservation of mass

$$\frac{d}{dt}\int_{\hat{x}_1(t)}^{\hat{x}_2(t)} u dx = 0, \tag{3.44}$$

giving a simplified velocity which defines the motion of the subinterval boundaries to allow for this conservation,

$$\hat{x}_t = \frac{f(u)}{u}, \tag{3.45}$$

with the initial condition

$$u(x,0) = u^0(x), \quad x \in (\hat{a}(0), \hat{b}(0)), \tag{3.46}$$

and, the boundary conditions

$$\hat{a}_t = \frac{f(\alpha)}{\alpha} \quad \text{and} \quad u(\hat{a}(t),t) = \alpha. \tag{3.47}$$

Note that $\hat{x}_1(t)$ and $\hat{x}_2(t)$ are arbitrary coordinates in the interval $(\hat{a}(t), \hat{b}(t))$ that move with the mass conserving velocity (3.45).

The discussion in section 3.4 provides the final step required to derive the general mass conservative moving mesh method to solve the original conservation law (3.38)-(3.40), namely the discretization of equations (3.44) and (3.45).

Using the quadrature approximation (3.37) to equation (3.44) and a general Runge-Kutta scheme for the mesh motion leads to the general moving mesh scheme

$$\hat{x}_j^{n+1} = \hat{x}_j^n + \Delta t \sum_{i=1}^{s} b_i k_i \tag{3.48}$$

and

$$(\hat{x}_j^{n+1} - \hat{x}_{j-1}^{n+1})(d_j u_j^{n+1} + d_{j-1} u_{j-1}^{n+1}) = A_{j-1/2}, \tag{3.49}$$

where $k_i = \hat{x}_t(\hat{x}_j^n + \Delta t \sum_{l=1}^{s} a_{il} k_l, t^n + c_i \Delta t)$ and a_{il}, b_i, c_i and d_i are the coefficients which define the scheme.

The numerical solution is found by applying the approximations (3.48) and (3.49) in the following algorithm:

1. Choose the initial partition of the initial domain and use the initial condition (3.46) to evaluate the solution values, u_j^0 on this initial mesh.

2. Use the calculated u_j^0 to calculate the local mass constants, $A_{j-1/2}$, using the quadrature approximation (3.49).

3. Use an appropriate timestep, Δt, in the timestepping scheme (3.48) to update the node positions, $\hat{x}_j^{n+1}$, the new mesh.

4. On the new mesh use the general quadrature (3.49) to recover the new solution values, u_j^{n+1}, on the nodes.

5. Repeat steps 3 and 4 until the desired termination time is reached.

3.6 Examples

In this section we demonstrate how the framework developed in this section can be applied to the example conservation laws introduced in section 2.1.5.

In the first example we look at the derivation of the scheme step by step to demonstrate how such a scheme can be constructed from scratch, while in the other two examples we use the general forms found earlier in the chapter as shortcuts to deriving the scheme.

3.6.1 Linear Advection Equation

As noted in section 2.1.5 the simplest scalar conservation law is the linear advection equation where $f(u) = au$ with a constant. This leads to the linear conservation law

$$u_t + au_x = 0, \tag{3.50}$$

with a given initial condition $u^0(x)$ and a free Lagrangian boundary condition.

As the scheme is desired to be mass conserving, the rate of change of the mass in a moving interval over time should be zero, hence

$$\frac{d}{dt}\int_{\hat{x}_1(t)}^{\hat{x}_2(t)} udx = 0. \tag{3.51}$$

Applying Leibniz integral rule to equation (3.51) gives

$$\int_{\hat{x}_1(t)}^{\hat{x}_2(t)} u_t dx + [u\hat{x}_t]_{\hat{x}_1(t)}^{\hat{x}_2(t)} = 0, \tag{3.52}$$

and appealing to the conservation law (3.50) yields

$$-\int_{\hat{x}_1(t)}^{\hat{x}_2(t)} au_x dx + [u\hat{x}_t]_{\hat{x}_1(t)}^{\hat{x}_2(t)} = 0. \tag{3.53}$$

The fundamental theorem of calculus can be used to show that

$$[u\hat{x}_t - au]_{\hat{x}_1(t)}^{\hat{x}_2(t)} = 0. \tag{3.54}$$

Under the given boundary condition we know that $u\hat{x}_t - au = 0$ when evaluated at the inflow boundary. Hence, it follows that the mass conservative velocity for an arbitrary moving coordinate is given by

$$\hat{x}_t = a. \tag{3.55}$$

Having chosen mass conservation and found the associated conservative velocity, the actual discretisation can be chosen. In regard to the choice of partition from section 3.3 we consider only the standard partition and choose an initial node distribution based on equidistribution.

Selecting a simple one-sided quadrature as the discretisation of the local mass conservation (3.51) leads to

$$u_j^n \left(\hat{x}_j^n - \hat{x}_{j-1}^n\right) = A, \tag{3.56}$$

where A is a single constant for all cells due to equidistribution.

To approximate the mesh velocity (3.55) we choose to use the forward Euler explicit method, yielding

$$\hat{x}_j^{n+1} = \hat{x}_j^n + a\Delta t. \tag{3.57}$$

Together equations (3.56) and (3.57) form the mass conservative moving mesh method for solving the linear advection equation (3.50). The scheme is applied by using equation (3.57) to update the mesh and then using the quadrature (3.56) to recover the solution at each timestep.

Remark 3.6.1. Note that the linear advection equation (3.50) is a special case for our class of numerical schemes. This is due to the fact that $f(u)$ is linear and as a result $\hat{x}_t = \frac{f(u)}{u}$ is a constant for all nodes in the mesh. Considering the numerical method (3.56)-(3.57) derived, it is clear that since the nodes $\hat{x}_j$ and $\hat{x}_{j-1}$ have the same velocity for all time, the quadrature is not actually required as the solution value, u_j^n, remains constant. As a result the scheme is easily verifiable as exact in time with the only error occurring in the original discretisation of the initial condition.

To demonstrate the scheme we consider a single wave with u constant everywhere else. The test problem is given by defining the constant

$$a = 1, \tag{3.58}$$

the initial condition

$$u^0(x) = \begin{cases} (x^2-1)^2 + 0.5 & -1 \leqslant x \leqslant 1 \\ 0.5 & \text{otherwise} \end{cases}, \tag{3.59}$$

the inflow boundary trajectory

$$\hat{x}_0(t) = t - 2, \tag{3.60}$$

and an initial domain, $[-2, 2]$.

Note that we are not required to prescribe a solution boundary condition as for the linear advection equation the 'free Lagrangian' boundary condition coincides with the characteristic boundary condition since $f'(u) = \frac{f(u)}{u}$. This implies that the problem is actually an IVP as opposed to a IBVP.

The numerical scheme (3.56)-(3.57) is run with 41 computational nodes over an initial domain of $[-2, 2]$ with a timestep of $\Delta t = 0.1$. Figure 3.5 shows the numerical solution up to time $t = 1$.

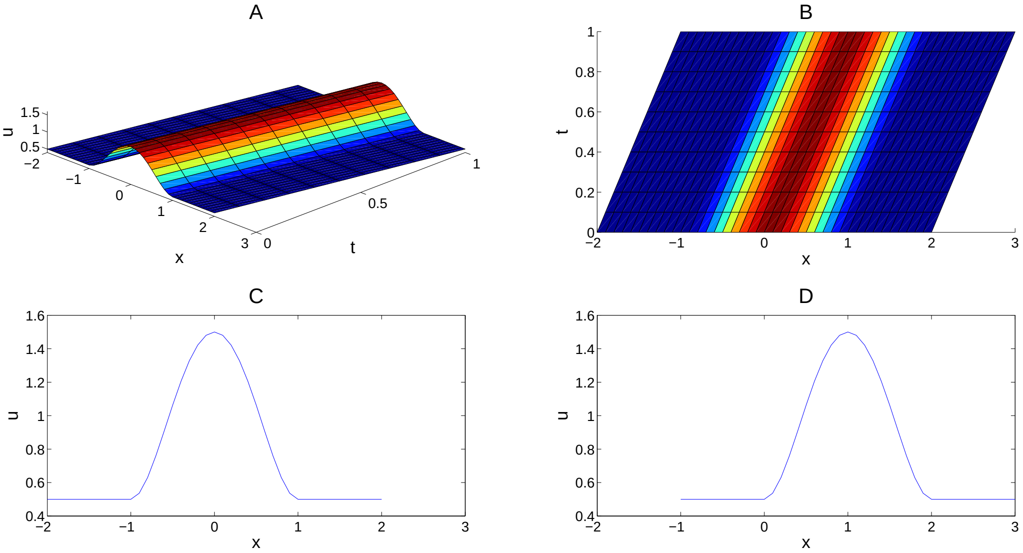


Figure 3.5: These graphs show various aspects of the numerical solution to the linear advection equation obtained from the scheme (3.56)-(3.57). The graphs show (A) the solution as a surface in *(x,t,u)* space. (B) computational node trajectories in the *(x,t)* plane. (C) the solution at time $t = 0$. (D) the solution at time $t = 1$.

It is clear from Figure 3.5 that the notes made in remark 3.6.1 hold since the solution does not diffuse or blow up. The numerical solution moves with the speed expected and the only error is incurred in the discretisation of the initial condition.

3.6.2 Inviscid Burgers' Equation

The simplest nonlinear conservation law is the Inviscid Burgers' equation. In this equation $f(u) = \frac{1}{2}u^2$ leading to the PDE

$$u_t + \left(\frac{1}{2}u^2\right)_x = 0, \tag{3.61}$$

with a given initial condition $u^0(x)$ and a free Lagrangian boundary condition.

Since Lagrangian mass conservation is desired, the general mass conservative formulation (3.44)-(3.45) is applied. This leads to the Lagrangian formulation for Inviscid Burgers' equation being the mass conservation equation,

$$\int_{\hat{x}_1(t)}^{\hat{x}_2(t)} u dx = \text{constant}, \tag{3.62}$$

and the conservative velocity

$$\hat{x}_t = \frac{f(u)}{u} = \frac{1}{2}u. \tag{3.63}$$

As in the linear advection example we choose to apply a one-sided quadrature approximation to the local conservation law (3.62),

$$u_j^n \left(\hat{x}_j^n - \hat{x}_{j-1}^n\right) = A, \tag{3.64}$$

where A is a single constant for all cells due to equidistribution. To approximate the mesh movement (3.63) we again use the explicit forward Euler method, yielding

$$\hat{x}_j^{n+1} = \hat{x}_j^n + \frac{\Delta t}{2} u_j^n. \tag{3.65}$$

As a test problem for this numerical scheme we give the piecewise linear initial condition

$$u^0(x) = \begin{cases} 1.1 + x & -1 < x \leqslant 0 \\ 1.1 - x & 0 < x < 1 \\ 0.1 & \text{otherwise} \end{cases}, \tag{3.66}$$

the inflow boundary condition

$$u(\hat{x}_0(t), t) = 0.1, \tag{3.67}$$

the inflow boundary velocity

$$\frac{d\hat{x}_0(t)}{dt} = \frac{1}{2} u(\hat{x}_0(t), t) = 0.05 \tag{3.68}$$

with the initial domain, $[-2, 2]$.

As noted in section 2.1.5, since the initial and boundary conditions are piecewise linear an exact solution for this problem can be calculated via the method of characteristics. In this case the exact solution has two distinct time regimes, pre-shock formation and post-shock formation. The shock can easily be shown to occur at time $t = 1$. The pre-shock solution for $t < 1$ is then given as

$$u(x,t) = \begin{cases} \dfrac{x+1.1}{t+1} & 0.1t - 1 < x \leqslant 1.1t \\ \dfrac{x-1.1}{t-1} & 1.1t < x < 0.1t + 1 \\ 0.1 & \text{otherwise} \end{cases}, \tag{3.69}$$

and the post-shock solution for $t \geqslant 1$ is

$$u(x,t) = \begin{cases} \dfrac{x+1.1}{t+1} & 0.1t - 1 < x \leqslant 0.1t + \sqrt{2t+2} - 1 \\ 0.1 & \text{otherwise} \end{cases}. \tag{3.70}$$

The numerical method (3.64)-(3.65) is run with 41 computational nodes over the initial domain $[-2, 2]$ with a timestep of $\Delta t = 0.05$. Figure 3.6A shows the solution at time $t = 0.9$ before a shock has formed and Figure 3.6B shows the solution at time $t = 1.5$ after the shock has formed and propagated.

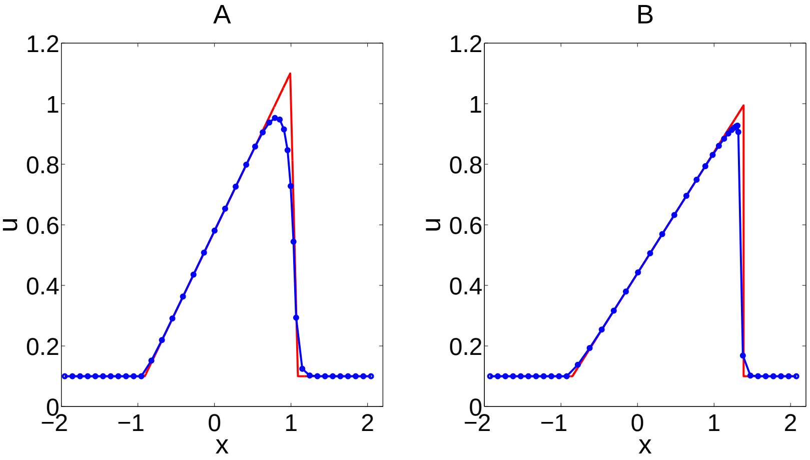


Figure 3.6: These graphs show the comparison between the numerical solution (Blue) to Inviscid Burgers Equation obtained from the scheme (3.64)-(3.65) and the exact solution (Red). Comparisons are taken at the pre-shock time regime $t = 0.9$ (A) and the post shock time regime $t = 1.5$ (B).

Figure 3.6A shows that unlike the linear example the Inviscid Burgers' scheme is prone to numerical diffusion. The interesting result is the post shock time regime in Figure 3.6B, where the numerical scheme is correctly approximating the shock speed.

3.6.3 Buckley-Leverett Equation

The final conservation law we consider from Section 2.1.5 is the Buckley-Leverett equation. This equation is given by

$$u_t + f(u)_x = 0, \tag{3.71}$$

with a given initial condition $u^0(x)$ and a free Lagrangian boundary condition, where

$$f(u) = \begin{cases} \dfrac{u^2}{u^2 + M(1-u)^2} & 0 \leqslant u \leqslant 1 \\ 0 & u < 0 \\ 1 & u > 1 \end{cases}, \tag{3.72}$$

and $M > 0$ is a given constant.

Using the general Lagrangian velocity formula for free Lagrangian boundary conditions (3.45) gives

$$\hat{x}_t = \frac{f(u)}{u} = \begin{cases} \dfrac{u}{u^2 + M(1-u)^2} & 0 < u \leqslant 1 \\ \frac{1}{u} & u > 1 \end{cases}, \tag{3.73}$$

where we note that the $u \leqslant 0$ cases have been omitted since these problems do not fall into the class of problem solvable by the conservation based moving mesh method.

We use the same approximations to the local conservation (3.44) and the mesh movement (3.73) as in the previous two examples, these are a one-sided quadrature and the explicit forward Euler method, respectively.

We test the scheme with 41 computational nodes and a timestep of $\Delta t = 0.0001$ over the region $[-2, 0]$. We choose $M = 0.5$, the discontinuous initial condition

$$u(x,0) = \begin{cases} 1 & x < -1 \\ 0.11 & \text{otherwise} \end{cases}, \tag{3.74}$$

the boundary condition

$$u(\hat{x}_0(t), t) = 1, \tag{3.75}$$

and the boundary velocity

$$\frac{d\hat{x}_0(t)}{dt} = \frac{u(\hat{x}_0(t), t)}{u(\hat{x}_0(t), t)^2} = 1. \tag{3.76}$$

In Figure 3.7 the solution is shown at times $t = 0$ (black), $t = 0.2$ (red), $t = 0.4$ (blue) and $t = 0.6$ (green).

We note that in Figure 3.7 the initial discontinuity splits as would be expected from a non-convex flux function. In addition we also note that the discontinuity which is initially poorly approximated steepens as the mesh refines around it.

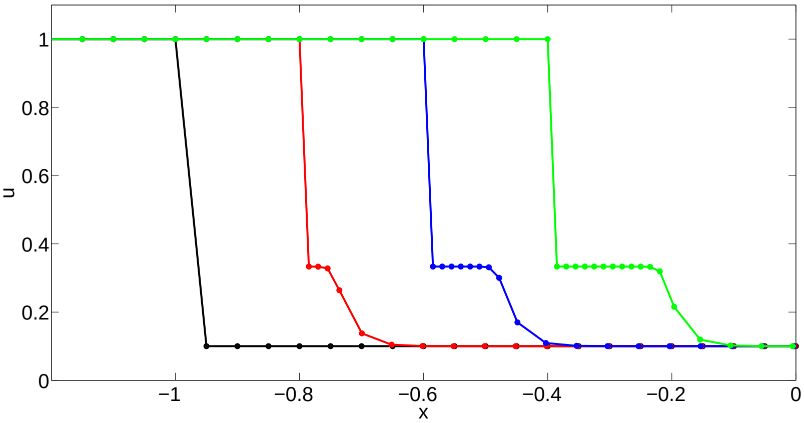


Figure 3.7: This graph shows the moving mesh approximation of the Buckley-Leverett equation with initial condition (3.74) and boundary conditions (3.75) and (3.76). The numerical approximation is shown at times $t = 0$ (black), $t = 0.2$ (red), $t = 0.4$ (blue) and $t = 0.6$ (green).

Chapter 4

Analysis of Mass Conserving Moving Mesh Methods

In the previous chapter we demonstrated how to derive a mass conservative moving mesh scheme for a given conservation law. As noted in Section 2.3, this class of methods has been widely applied to nonlinear diffusion problems to yield effective numerical results, for example in [BHJ11]. However, analysis of the schemes is often omitted since special issues arise when considering the moving mesh for which the schemes are nonlinear.

In this chapter we will discuss the main issues that arise when considering moving mesh methods as opposed to standard Eulerian fixed mesh methods before introducing a transformation to a fixed reference space in which the analysis of the schemes is feasible. Finally we will use this new space to show how the accuracy, stability and convergence of such schemes can be obtained.

4.1 Problems with Moving Meshes

The analysis of fixed grid finite difference and finite volume schemes for conservation laws is well established and is based on estimating a global error in the solution at each node of the mesh [GMP15]. This global error is defined to be

$$E_j^n = |u(x_j, t^n) - u_j^n| \tag{4.1}$$

at node j, where u_j^n is the numerical solution at the node and $u(x_j, t^n)$ is the exact solution at this point at time t^n.

A similar approach may be taken with moving mesh methods by defining the standard global truncation error to be

$$S_j^n = |u(\hat{x}_j^n, t^n) - u_j^n| \tag{4.2}$$

where $u(\hat{x}_j^n, t^n)$ is the exact solution at the position of the moving node and u_j^n is the corresponding numerical solution.

This gives the error that users of the moving mesh method are most interested in, since it is easily comparable with standard fixed grid method errors. However, the definition (4.2) of global error neglects to take into account the displacement of the mesh, and hence we introduce the notion of the true global error. This is defined to be

$$T_j^n = \sqrt{\left(u(\hat{x}(\chi_j, t^n), t^n) - u_j^n\right)^2 + \left(\hat{x}(\chi_j, t^n) - \hat{x}_j^n\right)^2} \tag{4.3}$$

where χ_j is a spatial variable label that defines a specific trajectory, $\hat{x}(\chi_j, t^n)$ which gives the point in space where the node would be if the time-stepping scheme were exact, and $u(\hat{x}(\chi_j, t^n), t^n)$ is the exact solution at that point.

The true error can be broken into two component errors, namely the error in the solution, u_j^n, and the error in the position, $\hat{x}_j^n$. This decomposition of the error is shown graphically in Figure 4.1.

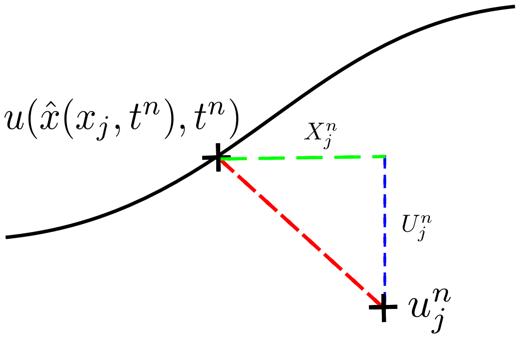


Figure 4.1: The True Error in the scheme broken down into solution error, U_j^n, and position error, X_j^n, components.

It is clear from Figure 4.1 that by defining the solution error to be

$$U_j^n = |u(\hat{x}(\chi_j, t^n), t^n) - u_j^n|, \tag{4.4}$$

and the position error to be

$$X_j^n = |\hat{x}(\chi_j, t^n) - \hat{x}_j^n| \tag{4.5}$$

and, using the orthogonality of the errors leads to our definition of the true error (4.3),

$$T_j^n = \sqrt{\left(U_j^n\right)^2 + \left(X_j^n\right)^2}. \tag{4.6}$$

Both the standard error and the true error are useful for different purposes and it is therefore important that a connection is made between the errors. Figure 4.2 demonstrates an example of *(i)* a single node and *(ii)* the two points on the exact solution which are used to define the two global errors.

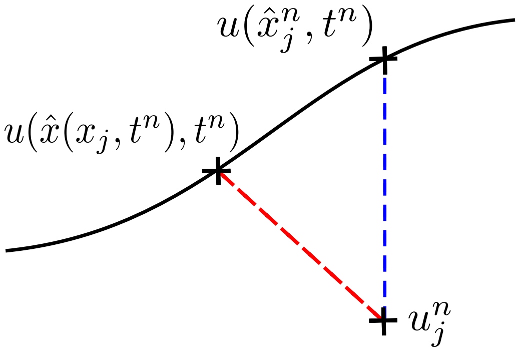


Figure 4.2: The Standard Finite Difference Error (Blue) Compared with the True Error in the Moving Mesh Scheme (Red).

As previously mentioned, the standard error is the most intuitive for anyone using the moving mesh scheme since the point $\hat{x}(\chi_j, t^n)$ is not generally considered when applying this type of scheme. However, since the moving mesh scheme is a discretisation of the continuous moving frame formulation (3.44)-(3.45) it is the true error which is required to converge. Theorem 4.1.1 shows that, under certain conditions, convergence of the true error causes convergence of the standard error.

Theorem 4.1.1. *Given a moving mesh formulation (3.44)-(3.45) of a PDE (2.1) and a corresponding moving mesh numerical approximation (3.48)-(3.49): if the numerical approximation is such that the true error tends to zero and the exact solution $u(x,t)$ is continuous at the point $\hat{x}(\chi_j, t^n)$ then the standard error also tends to zero.*

Proof. We start by considering the true error as $\hat{x}_j^n \to \hat{x}(\chi_j, t^n)$ and as $u_j^n \to u(\hat{x}(\chi_j, t^n), t^n)$. In particular, substituting equations (4.4) and (4.5) into equation

(4.3) gives

$$\sqrt{\left(U_j^n\right)^2 + \left(X_j^n\right)^2} \to 0, \tag{4.7}$$

which implies that both component errors must also be approaching 0. Hence as the true error tends to 0,

$$U_j^n \to 0 \quad \text{and} \quad X_j^n \to 0. \tag{4.8}$$

If we now consider the definition of the standard error (4.2) we may apply the inequality,

$$|u(\hat{x}_j^n, t^n) - u_j^n| \leqslant |u(\hat{x}_j^n, t^n) - u(\hat{x}(\chi_j, t^n), t^n)| + |u(\hat{x}(\chi_j, t^n), t^n) - u_j^n|, \tag{4.9}$$

and by using the definition of the solution error (4.4) we can simplify (4.9) to yield

$$|u(\hat{x}_j^n, t^n) - u_j^n| \leqslant |u(\hat{x}_j^n, t^n) - u(\hat{x}(\chi_j, t^n), t^n)| + U_j^n. \tag{4.10}$$

It follows from the definition of the position error (4.5) that as $X_j^n \to 0$

$$|\hat{x}(\chi_j, t^n) - \hat{x}_j^n| \to 0, \tag{4.11}$$

and furthermore due to the assumption that the exact solution is continuous at $\hat{x}(\chi_j, t^n)$ it follows from the Mean Value Theorem that

$$|u(\hat{x}_j^n, t^n) - u(\hat{x}(\chi_j, t^n), t^n)| = \left| \left.\frac{\partial u}{\partial x}\right|_\theta \right| |\hat{x}(\chi_j, t^n) - \hat{x}_j^n| \to 0, \tag{4.12}$$

where θ is a position between $\hat{x}(\chi_j, t^n)$ and $\hat{x}_j^n$.

Combining equations (4.8), (4.10) and (4.12) yields the required result. □

Theorem 4.1.1 shows that at all points besides discontinuities in the solution, convergence of the true error implies convergence of the standard error. Hence for the rest of this chapter we concern ourselves only with the true error of the numerical scheme.

4.2 Transformation to a Reference Space

In this section we introduce a useful tool for analysing our moving mesh schemes. The idea is to transform both the conservation law and the corresponding numerical scheme into a space in which the scheme is applied over a fixed grid. This allows the use of well developed methods of finding accuracy, stability and convergence.

The transformation used is based on a mapping given in [BHR96]. However, in that paper the authors use the transformation as an actual tool for numerically solving problems whereas here it is simply used as an analytical tool to obtain a reference space.

We call the space in which our problem is posed 'physical space' and the space into which we transform the 'reference space'.

The reference space is defined by the following properties:

1. Any point $\hat{x}$ moving with the required velocity for conservation of the monitor function in physical space is stationary in the reference space.
2. The physical domain of our scheme maps to $[0, 1]$ in the reference space.

3. The integral of the monitor function between an arbitrary point moving with the correct speed, $\hat{x}$, and an anchor point in physical space is linearly proportional to the distance between them in the reference space.

We first transform the moving co-ordinate, $\hat{x}$, from physical space to a fixed spatial co-ordinate, ξ, in a reference space. In particular we let

$$\hat{x} = \hat{x}(\xi, \tau),\ t = \tau \ \text{ and }\ u(\hat{x}(\xi, \tau), \tau) = \tilde{u}(\xi, \tau). \tag{4.13}$$

It follows by the chain rule that

$$\begin{aligned}
\begin{pmatrix} \dfrac{\partial}{\partial \tau} \\ \dfrac{\partial}{\partial \xi} \end{pmatrix} &= \begin{pmatrix} 1 & \dfrac{\partial \hat{x}}{\partial \tau} \\ 0 & \dfrac{\partial \hat{x}}{\partial \xi} \end{pmatrix} \begin{pmatrix} \dfrac{\partial}{\partial t} \\ \dfrac{\partial}{\partial \hat{x}} \end{pmatrix} \\
\Rightarrow \begin{pmatrix} \dfrac{\partial}{\partial t} \\ \dfrac{\partial}{\partial \hat{x}} \end{pmatrix} &= \frac{1}{\hat{x}_\xi} \begin{pmatrix} \dfrac{\partial \hat{x}}{\partial \xi} & -\dfrac{\partial \hat{x}}{\partial \tau} \\ 0 & 1 \end{pmatrix} \begin{pmatrix} \dfrac{\partial}{\partial \tau} \\ \dfrac{\partial}{\partial \xi} \end{pmatrix}.
\end{aligned} \tag{4.14}$$

Applying this transformation to the conservation law (2.1) we see that

$$\begin{aligned}
u_t + f_x(u) &= 0 \\
\Rightarrow \frac{1}{\hat{x}_\xi} \left(\hat{x}_\xi \tilde{u}_\tau - \hat{x}_\tau \tilde{u}_\xi + f(\tilde{u})_\xi \right) &= 0 \\
\Rightarrow \tilde{u}_\tau + \frac{1}{\hat{x}_\xi} \left(f(\tilde{u})_\xi - \hat{x}_\tau \tilde{u}_\xi \right) &= 0.
\end{aligned} \tag{4.15}$$

We now note that since $\hat{x}_\tau = \frac{f(u)}{u}$ under local mass conservation from (3.45), equation (4.15) becomes

$$\begin{aligned} \tilde{u}_\tau + \frac{\tilde{u} f(\tilde{u})_\xi - f(\tilde{u}) \tilde{u}_\xi}{\tilde{u} \hat{x}_\xi} &= 0 \\ \Rightarrow \tilde{u}_\tau + \frac{\tilde{u}}{\hat{x}_\xi} \left(\frac{f(\tilde{u})}{\tilde{u}} \right)_\xi &= 0. \end{aligned} \tag{4.16}$$

All that remains to obtain a PDE for u is to eliminate $\hat{x}_\xi$. In order to achieve this we define

$$K(\xi) = \int_{\xi_0}^{\xi} \tilde{u}(\hat{\xi}, \tau) \hat{x}_{\hat{\xi}} d\hat{\xi}, \tag{4.17}$$

where ξ_0 is the transform of the point $\hat{x}_0$. We note that applying the inverse transform (4.13) yields

$$\int_{\xi_0}^{\xi} \tilde{u}(\hat{\xi}, \tau) \hat{x}_{\hat{\xi}} d\hat{\xi} = \int_{\hat{x}_0}^{\hat{x}} u(x, t) dx, \tag{4.18}$$

which is defined to be constant in time, hence the definition that $K(\xi)$ is independent of τ is valid.

Differentiating (4.17) with respect to ξ we see that

$$K_\xi = \tilde{u} \hat{x}_\xi. \tag{4.19}$$

Substituting equation (4.19) into equation (4.16), we obtain the PDE

$$\tilde{u}_\tau + \frac{\tilde{u}^2}{K_\xi} \left(\frac{f}{\tilde{u}} \right)_\xi = 0. \tag{4.20}$$

We note that (4.20) is not a classical conservation law due to the factor $\frac{u^2}{K_\xi}$. However, in order to exploit the equation it would be beneficial if we could reformulate it as a classical conservation law, as follows.

The first step in reformulating equation (4.20) is to rewrite the equation such that the dependent variable $\tilde{u}$ does not appear outside of the derivative in the second term. This can be achieved by making certain polynomial assumptions on f. However, for general f we can define a new dependent variable $w = \frac{1}{\tilde{u}}$.

Using this new variable the transformed PDE (4.20) becomes

$$\begin{aligned} \left(\frac{1}{w}\right)_\tau + \frac{1}{w^2 K_\xi}\left(wf\left(\frac{1}{w}\right)\right)_\xi &= 0 \\ \Rightarrow w_\tau - \frac{1}{K_\xi}\left(wf\left(\frac{1}{w}\right)\right)_\xi &= 0. \end{aligned} \tag{4.21}$$

All that remains in order for the transformed PDE to be a classical conservation law is to show that K_ξ is a constant in space. In order to show this we begin by noting that we have made no assumptions on ξ other than that it is time independent and recall the equidistribution principle from Section 2.2.2.

Definition 4.2.1 (Equidistribution Principle)**.** The equidistribution principle states that for $0 \leqslant \chi \leqslant 1$

$$\int_{\hat{x}_0}^{\hat{x}(\chi,t)} u dx = \chi \int_{\hat{x}_0}^{\hat{x}_F} u dx. \tag{4.22}$$

Note here that the LHS of the definition (4.22) defines a proportion of the integral over the whole region $(\hat{x}_0, \hat{x}_F)$ in physical space which has a constant value due to our requirement of mass conservation.

If we let the reference space coordinate ξ be χ as defined by the equidistribution principle and transform equation (4.22) to the fixed reference space then

$$\int_{\xi_0}^{\xi} \frac{\hat{x}_\xi}{w} d\hat{\xi} = \xi \int_{\xi_0}^{\xi_F} \frac{\hat{x}_\xi}{w} d\hat{\xi}. \tag{4.23}$$

We notice that the LHS of equation (4.23) is by definition $K(\xi)$ from (4.17) hence

$$K(\xi) = \xi \int_{\xi_0}^{\xi_F} \frac{\hat{x}_{\hat{\xi}}}{w} d\hat{\xi}, \tag{4.24}$$

is linear in ξ and therefore K_ξ is constant as required.

We now define the transformed flux function, $\tilde{f}(w)$, to be given by

$$\tilde{f}(w) := -\frac{w}{K_\xi} f\left(\frac{1}{w}\right). \tag{4.25}$$

Summarising, if we choose the mapping (4.13) to ξ which is given by the equidistribution principle, then equation (4.21) is a conservation law with the transformed flux function (4.25). This conservation law is given by

$$w_\tau + \tilde{f}(w)_\xi = 0. \tag{4.26}$$

The transform of the numerical scheme (3.48)-(3.49) to the reference space is simple to achieve. Using Equations (3.48)-(3.49), we eliminate any $\hat{x}$ terms: indeed, for the example scheme (3.64)-(3.65) by subtracting (3.65) at node $j-1$ from (3.65) at node j,

$$\hat{x}_j^{n+1} - \hat{x}_{j-1}^{n+1} - (\hat{x}_j^n - \hat{x}_{j-1}^n) = \Delta t \left(\frac{f_j^n}{u_j^n} - \frac{f_{j-1}^n}{u_{j-1}^n}\right). \tag{4.27}$$

Using the local conservation principle (3.64) to eliminate the $\hat{x}$ terms gives

$$\frac{c}{u_j^{n+1}} - \frac{c}{u_j^n} = \Delta t \left(\frac{f_j^n}{u_j^n} - \frac{f_{j-1}^n}{u_{j-1}^n} \right), \tag{4.28}$$

where c is the local mass constant.

We note that Equation (4.28) can be seen as the upwind moving mesh approximation to Equation (4.25).

Finally using $w_j^n = \frac{1}{u_j^n}$, $\Delta t = \Delta\tau$ and rearranging gives

$$\frac{w_j^{n+1} - w_j^n}{\Delta\tau} - \frac{1}{c}\left(w_j^n f_j^n - w_{j-1}^n f_{j-1}^n \right) = 0. \tag{4.29}$$

This shows that the transformation to reference space is relatively simple for both the PDE and the numerical method and gives a conservation law for the PDE and a conservative form numerical method with associated numerical flux function. In the following sections we make use of this transformation as a tool to obtain the accuracy and stability conditions and show convergence.

4.3 Accuracy

In the previous section we introduced the transformation to the equidistribution reference space as a mathematical tool for helping with analysis of our moving mesh schemes. In this section we use the transformation to show how the accuracy of a scheme can be analysed.

Before the order of the schemes can be analysed however we must first consider the order of both the true error (4.3) and the standard error (4.2) which were introduced at the start of this chapter.

4.3.1 Comparing the Standard Error and the True Error

At the beginning of this chapter we briefly discussed the issues with calculating the error for our conservation based moving mesh schemes. Importantly, the actual error that should be decreasing is not very useful for end users of the scheme. Theorem 4.1.1 shows that the true error (4.3) and the standard error (4.2) converge to zero together away from shocks, however this is insufficient for considering the order of each scheme.

As shown in the proof of Theorem 4.1.1 both the standard and the true error may be written in terms of the component errors

$$S_j^n = u(\hat{x}(\chi_j, t^n), t^n) - u_j^n \quad \text{and} \quad X_j^n = \hat{x}(\chi_j, t^n) - \hat{x}_j^n. \tag{4.30}$$

The true error is given by

$$T_j^n = \sqrt{(U_j^n)^2 + (X_j^n)^2} \tag{4.31}$$

and the standard error is bounded by

$$S_j^n \leqslant \left|\frac{\partial u}{\partial x}\right|\Bigg|_{\theta} X_j^n + U_j^n. \tag{4.32}$$

Now assume that the orders of the component errors U_j^n and X_j^n are known. Then we may write each as

$$U_j^n = \mathcal{O}(\Delta x^k) + \mathcal{O}(\Delta t^m) \tag{4.33}$$

and

$$X_j^n = \mathcal{O}(\Delta x^l) + \mathcal{O}(\Delta t^n) \tag{4.34}$$

for some k, l, m and n.

It follows from inserting equations (4.33) and (4.34) into equation (4.31) that the true error is then

$$T_j^n = \sqrt{(\mathcal{O}(\Delta x^k) + \mathcal{O}(\Delta t^m))^2 + (\mathcal{O}(\Delta x^l) + \mathcal{O}(\Delta t^n))^2}, \tag{4.35}$$

which simplifies under the assumption that Δx and Δt are changed proportionally and disregarding higher order terms to yield

$$T_j^n = \mathcal{O}(\Delta x^{\min(k,l)}) + \mathcal{O}(\Delta t^{\min(m,n)}). \tag{4.36}$$

Equation (4.36) shows that the true error is a combination of the lowest space and time orders in the component errors.

Applying the same process to the inequality (4.32) gives

$$S_j^n \leqslant \left|\frac{\partial u}{\partial x}\right|\Bigg|_\theta (\mathcal{O}(\Delta x^l) + \mathcal{O}(\Delta t^n)) + \mathcal{O}(\Delta x^k) + \mathcal{O}(\Delta t^m) \tag{4.37}$$

which trivially reduces to

$$S_j^n \leqslant \mathcal{O}(\Delta x^{\min(k,l)}) + \mathcal{O}(\Delta t^{\min(m,n)}). \tag{4.38}$$

It follows from the inequality (4.38) that in the worst case scenario the standard error is of the same order as the true error.

As a result of the equation (4.36) and the inequality (4.38) we can now find the order of the true error and the worst case scenario order for the standard error if we know the orders of the solution error U_j^n and the position error X_j^n.

4.3.2 Finding the Solution and Position Errors

Having reduced the question of accuracy down to having to find the order of both the solution error U_j^n and the position error X_j^n we now use the reference space transform from Section 4.2 to find these orders.

It is noted that in the reference space the mesh is static and hence there is no position error in the reference space. As a result there is only a single error in the reference space which is related to the solution error by

$$U_j^n = \frac{1}{w(\xi_j, \tau^n) w_j^n} (w(\xi_j, \tau^n) - w_j^n). \tag{4.39}$$

Since the reference space PDE is known and the transformed numerical scheme is on a static mesh the order of the transformed numerical scheme is easily calculated. Assuming that the reference space scheme is p^{th} order in space and q^{th} order in time gives the solution error as

$$U_j^n = \mathcal{O}(\Delta \xi^p) + \mathcal{O}(\Delta \tau^q), \tag{4.40}$$

which can be changed to physical space parameters to give

$$U_j^n = \mathcal{O}(\Delta A^p) + \mathcal{O}(\Delta t^q), \tag{4.41}$$

where A is the local mass constant.

Note that unlike static mesh methods the spatial order is given in terms of the local mass constant A instead of the spatial step Δx. This is due to the fact that the scheme replaces the notion of fixed 'volumes' with fixed masses and hence the quantity Δx is not useful since we cannot say what size it has in general. This is still consistent with regular definitions of order since both a reduction in spatial step Δx for Eulerian methods and a reduction in the local mass constant A in our scheme are results of increasing the number of nodes in the mesh.

Now all that remains is to find the position error. Recall from Section 4.2 that

$$\hat{x}_\xi = K_\xi w, \tag{4.42}$$

where K_ξ is a constant as defined in equation (4.19).

Since equation (4.42) shows that $\hat{x}_\xi$ and w are related by a constant this implies that the errors in both are accurate to the same order. Hence under the same assumption made on the order of the reference space scheme before it follows that the numerical approximation of $\hat{x}_\xi$ is accurate to the p^{th} order in space and the q^{th} order in time.

We now need to find the order of the error in $\hat{x}$ from the error in $\hat{x}_\xi$. It is an established result that if an approximation is n^{th} order then the error in the derivative is $n - 1^{\text{th}}$ order. This holds only over a single interval however and summing over a number of intervals inversely proportional to the spatial step reduces the order by one. Hence, it follows that the order does not change since these two effects cancel out and thus

$$X_j^n = \mathcal{O}(\Delta A^p) + \mathcal{O}(\Delta t^q). \tag{4.43}$$

Combining the component errors (4.41) and (4.43) with the results (4.36) and (4.38) show that

$$T_j^n = \mathcal{O}(\Delta x^p) + \mathcal{O}(\Delta t^q), \tag{4.44}$$

and

$$S_j^n \leqslant \mathcal{O}(\Delta x^p) + \mathcal{O}(\Delta t^q). \tag{4.45}$$

In conclusion the error in the moving mesh Lagrangian scheme is of equal order in both space and time to the transformed reference space PDE.

4.4 Stability

In this section we consider a stability framework for the schemes derived using the Lagrangian formulation of the problem.

We start by considering the non-crossing criterion which prevents mesh tangling in physical space before moving on to using the transform given in Section 4.2 to find a true stability condition.

4.4.1 Non-crossing Criterion

As mentioned at the start of this chapter, a large issue that arises concerning moving mesh methods is mesh tangling. Mesh tangling occurs when the order of the nodes changes due to a poor discretisation of the problem. In this section we will demonstrate a general non-crossing criterion and show how this is necessary but not sufficient for stability of the solution.

Start by assuming that at the n^{th} timestep the computational mesh is untangled. Then consider that for nodes j and $j-1$ to remain ordered after a further time step, the inequality

$$\hat{x}_j^{n+1} > \hat{x}_{j-1}^{n+1}. \tag{4.46}$$

must hold for non-tangling.

Using a general timestepping scheme (3.48), where we simplify the velocity term to v_j^n for ease of reading, yields the inequality

$$\hat{x}_j^n + \Delta t v_j^{n+1/2} > \hat{x}_{j-1}^n + \Delta t v_{j-1}^{n+1/2} \tag{4.47}$$

where $v_j^{n+1/2}$ is a general approximation of the j^{th} node velocity and may be fully explicit, fully implicit or a combination of both.

The inequality (4.47) may be rearranged to find the restriction on Δt to ensure that (4.46) holds. This restriction is given by

$$\Delta t < \frac{\hat{x}_j^n - \hat{x}_{j-1}^n}{v_{j-1}^{n+1/2} - v_j^{n+1/2}}, \tag{4.48}$$

since $v_{j-1}^{n+1/2} > v_j^{n+1/2}$ for crossing to occur and hence $v_{j-1}^{n+1/2} - v_j^{n+1/2}$ is positive.

Note that the inequality (4.48) only accounts for the possibility that the $j-1^{\text{th}}$ node will cross the j^{th} node during this particular timestep. Hence this leads to the requirement that the timestep for nodes j and $j-1$ between timesteps n and $n+1$

is given by

$$\Delta t_{j-1/2}^{n+1/2} \leqslant \frac{\hat{x}_j^n - \hat{x}_{j-1}^n}{v_{j-1}^{n+1/2} - v_j^{n+1/2}} \qquad \text{if} \quad v_{j-1}^{n+1/2} > v_j^{n+1/2}, \tag{4.49}$$

and no requirement otherwise.

Definition (4.49) allows $\Delta t^{n+1/2}$ to be calculated for the entire mesh. Explicitly this is

$$\Delta t^{n+1/2} = \min_j \Delta t_{j-1/2}^{n+1/2}. \tag{4.50}$$

The timestep definition (4.50) ensures that mesh tangling will not occur, however this does not imply stability of the scheme and is generally not a practical condition to apply.

The problem with this definition of the timestep is that we cannot guarantee that the timestep will not approach zero, stopping the method from proceeding further. To demonstrate this consider a problem in which the coordinate $\hat{x}_j$ remains stationary and the coordinate $\hat{x}_{j-1}$ is moving towards $\hat{x}_j$ with a constant speed. In this case the local timestepping restriction can be calculated to give

$$\Delta t_{j-1/2}^{n+1/2} < \frac{\hat{x}_j^n - \hat{x}_{j-1}^n}{v_{j-1}^{n+1/2}}. \tag{4.51}$$

The inequality (4.51) shows that since $\hat{x}_j^n$ and $v_{j-1}^{n+1/2}$ are constant and $\hat{x}_{j-1}^n$ is approaching $\hat{x}_j^n$ the timestep must go to zero. This exact situation can occur around a shock in a solution and is therefore a very real problem.

The second problem with using (4.50) as the adaptive timestep choice is that this is not a guarantee of stability. To illustrate this consider the simple quadrature approximation to the local mass conservation (3.44),

$$u_j^n \left(\hat{x}_j^n - \hat{x}_{j-1}^n\right) = A_{j-1/2}, \tag{4.52}$$

where $A_{j-1/2}$ is the local mass constant.

The choice of adaptive timestep (4.50) ensures that $\hat{x}_j^n - \hat{x}_{j-1}^n > 0$, however it places no limit on how small this can become. It follows directly from the quadrature choice (4.52) that since $\hat{x}_j^n - \hat{x}_{j-1}^n$ can become arbitrarily small then u_j^n can grow arbitrarily large.

The fact that the non-crossing criterion does not ensure stability of the solution, u, as well as not guaranteeing a positive timestep leads to the conclusion that while this is a necessary condition for non-tangling of the mesh, it is not sufficient for ensuring a stable numerical approximation of u.

In the next few sections we will develop a better time step restriction and show that this also ensures that the mesh does not tangle, as well as providing stability of the numerical solution.

4.4.2 Stability Via Reference Space

We now consider a method for finding the stability of the numerical solution, u_j^n by considering the reference space PDE and associated numerical scheme from Section 4.2. The premise here is that we can take the stability condition of the transformed scheme and show that applying the inverse transform to the condition does not cause it to change.

Total Variation Diminishing

Since we have been able to transform both the PDE and the numerical scheme to a space with a fixed spatial coordinate, we may now appeal to well established results for the stability of fixed grid schemes. In particular, since we have a conservation law and a scheme in conservative form, we consider TVD stability analysis [Har83].

We recall from section 2.1.3 that a scheme may be shown to be TVD by applying Harten's Theorem.

Theorem 4.4.1 (Harten's Theorem)**.** *If a scheme can be written in the form*

$$w_j^{n+1} = w_j^n - C_{j-1/2}(w_j^n - w_{j-1}^n) + D_{j+1/2}(w_{j+1}^n - w_j^n)$$

where $C_{j-1/2} \geqslant 0$*,* $D_{j+1/2} \geqslant 0$ *and* $1 - C_{j-1/2} - D_{j-1/2} \geqslant 0$ *then the scheme is TVD.*

If we are able to show that our transformed scheme for w is TVD via Harten's theorem then since $u = \frac{1}{w}$ it only remains to be shown that the scheme for u is also stable.

TVD Property and the Transform

In section 4.2 we defined the transformation of variables,

$$\hat{x} = \hat{x}(\xi, \tau), \quad t = \tau \quad \text{where} \quad u(\hat{x}(\xi, \tau), \tau) = \frac{1}{w(\xi, \tau)}, \tag{4.53}$$

and demonstrated how this transform may be applied to our numerical scheme.

In this section we seek to show that this TVD property remains valid through this transform and may therefore be applied to the original numerical scheme.

In order to show this we first show that by choosing our timestep in such a way that our scheme meets the criteria for Harten's theorem, the transform does not affect the order of nodes (i.e. there is no mesh tangling). Secondly, having shown this, we consider whether the change of variables from u to w leads to the introduction of new extrema or an increase in the current extrema.

In this section we consider the general mass conserving moving mesh numerical scheme (3.48)-(3.49) since all of the results hold of any such scheme.

Lemma 4.4.2. *If a moving mesh numerical scheme of the form (3.48)-(3.49) has a corresponding transformed scheme (4.29) that has been shown to be TVD for $\Delta t \leqslant T$, then*

$$\hat{x}_j^{n+1} < \hat{x}_{j+1}^{n+1} \quad \forall j. \tag{4.54}$$

Proof. Assume that the timestep has met the criteria that $\Delta t \leqslant T$ and that in the moving mesh numerical scheme (3.48)-(3.49) there is at least one J such that $\hat{x}_J^{n+1} \geqslant \hat{x}_{J+1}^{n+1}$ (i.e. the mesh has tangled).

Since the transformed numerical scheme (4.29) is obtained by the simple elimination of $\hat{x}$ terms in the system (3.48)-(3.49) it is clear that if $u_j^n = \frac{1}{w_j^n}$ then $u_j^{n+1} = \frac{1}{w_j^{n+1}}$.

Furthermore by equation (3.49) it is clear that since the d_j's and $A_{j-1/2}$ are all positive that $u_{\tilde{j}}^{n+1} \leqslant 0$ for some $\tilde{j}$ since $\hat{x}_{J+1}^{n+1} - \hat{x}_J^{n+1} \leqslant 0$. Then the corresponding $w_{\tilde{j}}^{n+1}$ must also be negative, but this is a contradiction with the strictly positive initial data and the fact that the transformed scheme is TVD. □

Lemma 4.4.3. *If a moving mesh numerical scheme of the form (3.48)-(3.49) has a corresponding transformed scheme (4.29) that has been shown to be TVD for $\Delta t \leqslant T$,* *then the original moving mesh scheme is also TVD for $\Delta t \leqslant T$.*

Proof. We have already shown in the previous Lemma that for this choice of Δt the moving mesh scheme is not prone to mesh tangling. Hence all that remains to be shown is that the change of variable from u_j^n to w_j^n does not lead to the introduction of new extrema or the increase of current extrema.

Again, since the transformed numerical scheme (4.29) is obtained by simple elimination of $\hat{x}$ terms in the system (3.48)-(3.49) it is clear that if $u_j^n = \frac{1}{w_j^n}$ then $u_j^{n+1} = \frac{1}{w_j^{n+1}}$. It follows that since the numerical solution obtained from the transformed scheme is bounded, the moving mesh numerical solution must also be bounded.

Consider the approximation to w_ξ,

$$\begin{aligned} \frac{w_j - w_{j-1}}{\xi_j - \xi_{j-1}} &= \frac{1}{\xi_j - \xi_{j-1}}\left(\frac{1}{u_j} - \frac{1}{u_{j-1}}\right) \\ &= -\frac{1}{u_j u_{j-1}} \frac{u_j - u_{j-1}}{\xi_j - \xi_{j-1}}, \end{aligned} \tag{4.55}$$

where we note again that u_j and u_{j-1} are bounded and positive.

Since we have already shown that mesh tangling does not occur for $\Delta t \leqslant T$ it follows that since $u > 0$

$$\frac{u_j - u_{j-1}}{\xi_j - \xi_{j-1}} = C_j \frac{u_j - u_{j-1}}{\hat{x}_j - \hat{x}_{j-1}}, \tag{4.56}$$

where C_j is a positive constant associated with each cell. Hence,

$$\operatorname{sgn}\left(\frac{w_j - w_{j-1}}{\xi_j - \xi_{j-1}}\right) = -\operatorname{sgn}\left(\frac{u_j - u_{j-1}}{\hat{x}_j - \hat{x}_{j-1}}\right), \tag{4.57}$$

and it follows that since the transformed scheme is TVD we cannot have spurious oscillations occurring in our solution u_j^n. □

4.5 Convergence

In the previous section we showed how a numerical scheme (3.48)-(3.49) which approximates the moving frame formulation (3.44)-(3.45) may be transformed into a reference space in order to determine conditions under which the scheme is total variation diminishing.

In this section we will continue to use the reference space as a tool to show convergence of an altered transformed scheme. We first introduce the notion of a vanishing viscosity solution.

4.5.1 Vanishing Viscosity Solution

In general there are infinitely many solutions to the weak form of the PDE (2.22). We therefore seek the physically relevant solution and motivate this by introducing the viscous regularisation through the problem

$$u_t^\epsilon + f(u^\epsilon)_x = \epsilon u_{xx}^\epsilon, \quad x \in (a,b), \quad t \in \mathbb{R}^+, \tag{4.58}$$

$$u^\epsilon(x,0) = u_0^\epsilon(x), \quad x \in (a,b), \tag{4.59}$$

$$u^\epsilon(a,t) = u^\epsilon(b,t), \quad t \in \mathbb{R}^+, \tag{4.60}$$

where $\epsilon > 0$ and a and b are constant.

This regularised PDE (4.58) admits a unique weak solution for positive ϵ. Cockburn [Coc03] suggests that the physically relevant solution of (3.1) may be found by taking the limit of these weak solutions of (4.58) as $\epsilon \downarrow 0$.

This requires that we show that the limiting solution satisfies the weak form of the conservation law (2.22) and in order to show this we introduce the notion of entropy.

Definition 4.5.1 (Entropy and Entropy Flux)**.** Two smooth functions $\eta(u)$ and $q(u)$ form an entropy/entropy flux pair of the conservation law (3.1) provided that $\eta(u)$ is convex and

$$q'(u) = \eta'(u) f'(u). \tag{4.61}$$

Remark 4.5.2. Since we only consider scalar conservation laws, any convex function of u is a valid entropy, $\eta(u)$, with a corresponding entropy flux, namely

$$q(u) = \int \eta'(\tilde{u}) f'(\tilde{u}) d\tilde{u}. \tag{4.62}$$

It follows from the requirement (4.61) that for smooth solutions to the conservation law (3.1) the entropy also satisfies a scalar conservation law since,

$$\begin{aligned} \eta(u)_t + q(u)_x &= \eta'(u) u_t + q'(u) u_x \\ &= \eta'(u) u_t + \eta'(u) f'(u) u_x \\ &= \eta'(u)(u_t + f(u)_x) \\ &= 0. \end{aligned} \tag{4.63}$$

However as we have previously stated, general solutions to the conservation law (3.1) are not smooth. Hence we suggest replacing (4.63) by the inequality

$$\eta(u)_t + q(u)_x \leqslant 0, \tag{4.64}$$

which leads to the definition of an entropy solution to the conservation law (3.1).

Definition 4.5.3 (Entropy Solution)**.** A function u is said to be an entropy solution of the conservation law (3.1), with associated entropy/entropy flux pair (η, q), if it satisfies the weak form of the the PDE,

$$\int_0^\infty \int_a^b (\psi_t u + \psi_x f(u)) dx dt + \int_a^b \psi_0 u_0 dx = 0, \tag{4.65}$$

and the entropy inequality,

$$\int_0^\infty \int_a^b (\phi_t \eta(u) + \phi_x q(u)) dx dt + \int_a^b \phi_0 \eta(u_0) dx \geqslant 0, \tag{4.66}$$

where ψ and ϕ are periodic Lipschitz continuous test functions and $\phi > 0$. Note that the subscript 0's here denote the initial condition of the function, i.e. when $t = 0$.

Remark 4.5.4. We may also consider the regularised solution, u^ϵ, as an entropy solution of the regularised PDE (4.58) provided that it satisfies the weak form of (4.58) and the entropy equality

$$\int_0^\infty \int_a^b (\phi_t \eta(u^\epsilon) + \phi_x q(u^\epsilon)) dx dt + \int_a^b \phi_0 \eta(u_0^\epsilon) dx = 0. \tag{4.67}$$

Theorem 4.5.5. *If u^ϵ is a smooth solution of the regularised PDE (4.58) and there exists a function u such that*

$$u^\epsilon \to u \quad \text{almost everywhere as} \quad \epsilon \downarrow 0, \tag{4.68}$$

then u is an entropy solution of the conservation law (3.1).

Proof. In order to show that u is an entropy solution of the conservation law (3.1) we must show that it satisfies both the weak form of the conservation law (4.65) and the entropy inequality (4.66).

Multiplying the regularised PDE (4.58) by the test function ψ and integrating over $(a, b) \times [0, \infty)$, it follows that by integration by parts

$$\int_0^\infty \int_a^b (\psi_t u^\epsilon + \psi_x f(u^\epsilon) + \psi \epsilon u^\epsilon_{xx}) dxdt + \int_a^b \psi_0 u^\epsilon_0 dx = 0. \tag{4.69}$$

It is clear that if we take the limit of (4.69) as $\epsilon \downarrow 0$ then the weak form (4.65) follows directly. Hence all that remains to be shown is that the entropy inequality is satisfied.

Since u^ϵ is a smooth solution to the regularised PDE (4.58) similar manipulations to (4.63) may be used to show the entropy satisfies

$$\eta(u^\epsilon)_t + q(u^\epsilon)_x = \epsilon \eta'(u^\epsilon) u^\epsilon_{xx}. \tag{4.70}$$

We may now rewrite the right hand side of (4.70) to obtain

$$\begin{aligned} \eta(u^\epsilon)_t + q(u^\epsilon)_x &= \epsilon \eta(u^\epsilon)_{xx} - \epsilon \eta''(u^\epsilon)(u^\epsilon_x)^2 \\ &\leqslant \epsilon \eta(u^\epsilon)_{xx}, \end{aligned} \tag{4.71}$$

since $\eta''(u^\epsilon)$, ϵ and $(u^\epsilon)^2$ are all non-negative.

Multiplying (4.71) by the non-negative test function ϕ and integrating over $(a, b) \times [0, \infty)$ gives

$$\int_0^\infty \int_a^b \phi(\eta(u^\epsilon)_t + q(u^\epsilon)_x) dxdt \leqslant \epsilon \int_0^\infty \int_a^b \eta(u^\epsilon)_{xx} dxdt. \tag{4.72}$$

Taking the limit as $\epsilon \downarrow 0$ shows that u satisfies

$$\int_0^\infty \int_a^b \phi(\eta(u^\epsilon)_t + q(u^\epsilon)_x) dxdt \leqslant 0, \tag{4.73}$$

and application of integration by parts yields (4.66) and therefore completes the proof. □

4.5.2 Regularisation in Reference Space

In previous work we have shown that we may obtain stability results for the class of moving mesh schemes that we are studying by making a transformation from physical space into a reference space.

Transforming the conservation law (3.1) to the reference space yields the transformed PDE

$$w_\tau - \frac{1}{K_\xi}\left(wf\left(\frac{1}{w}\right)\right)_\xi = 0, \tag{4.74}$$

which we have previously noted is also a conservation law since K_ξ is a constant. Hence we may consider the vanishing viscosity solution from Chapter 2 in the reference space to give the regularised transformed PDE

$$w^\epsilon_\tau - \frac{1}{K_\xi}\left(w^\epsilon f\left(\frac{1}{w^\epsilon}\right)\right)\xi = \frac{\epsilon}{K_\xi^2}w^\epsilon_{\xi\xi}. \tag{4.75}$$

To simplify notation we introduce the transformed flux function and the transformed viscosity coefficient.

Definition 4.5.6. We define the transformed viscosity coefficient, δ, to be given by

$$\delta := \frac{\epsilon}{K_\xi^2}. \tag{4.76}$$

Inserting definitions (4.25) and (4.76) into the regularised transformed PDE (4.75) gives

$$w^{\epsilon}_{\tau} + \tilde{f}(w^{\epsilon})_{\xi} = \delta w^{\epsilon}_{\xi\xi}. \tag{4.77}$$

Similarly the transformed conservation law (4.74) is now given by

$$w^{\epsilon}_{\tau} + \tilde{f}(w^{\epsilon})_{\xi} = 0. \tag{4.78}$$

4.5.3 Regularised Numerical Scheme

We cannot continue to apply the results to the general scheme and hence we consider the example scheme used in Section 3.6.

Now we seek to use this theory to obtain a moving mesh numerical scheme to approximate the conservation law (3.1).

We first make a standard discretisation of the regularised transformed PDE (4.77) to give the reference space scheme

$$\frac{w^{n+1}_j - w^n_j}{\Delta\tau} - \frac{1}{K_\xi \Delta\xi}(w^n_j \tilde{f}^n_j - w^n_{j-1}\tilde{f}^n_{j-1}) = \frac{\epsilon}{K^2_\xi \Delta\xi^2}(w^n_{j+1} - 2w^n_j + w^n_{j-1}). \tag{4.79}$$

Rearranging and using the fact that $c = K_\xi \Delta\xi$ yields,

$$c(w^{n+1}_j - w^n_j) = \Delta\tau(w^n_j \tilde{f}^n_j - w^n_{j-1}\tilde{f}^n_{j-1}) + \frac{\epsilon\Delta\tau}{c}((w^n_{j+1} - w^n_j) - (w^n_j - w^n_{j-1})). \tag{4.80}$$

Setting $cw_j^n = \hat{x}_j^n - \hat{x}_{j-1}^n$ to be consistent with the scheme we started with, (3.64)-(3.65), gives

$$\begin{aligned}\hat{x}_j^{n+1} - \hat{x}_{j-1}^{n+1} - (\hat{x}_j^n - \hat{x}_{j-1}^n) =& \Delta\tau(w_j^n \tilde{f}_j^n - w_{j-1}^n \tilde{f}_{j-1}^n) \\ &+ \frac{\epsilon\Delta\tau}{c}((w_{j+1}^n - w_j^n) - (w_j^n - w_{j-1}^n)),\end{aligned} \tag{4.81}$$

which we note is a Forward Euler approximation at j, minus the same approximation at $j-1$, hence given an anchor point at $j=0$, say, gives the associated moving mesh scheme to be

$$\begin{cases} \hat{x}_j^{n+1} = \hat{x}_j^n + \Delta t \left(\dfrac{\tilde{f}_j^n}{u_j^n} + \dfrac{\epsilon}{u_{j+1}^n} \dfrac{\frac{1}{u_{j+1}^n} - \frac{1}{u_j^n}}{\hat{x}_{j+1}^n - \hat{x}_j^n} \right) \\ u_j^n(\hat{x}_j^n - \hat{x}_{j-1}^n) = c \end{cases}. \tag{4.82}$$

4.5.4 Application to Inviscid Burgers' Equation

In order to obtain results on the performance of the scheme we need to apply it to a test problem. We choose to use the Inviscid Burgers' equation where $f(u) = \frac{1}{2}u^2$.

Applying a Taylor expansion and Taylor's remainder theorem to the reference space scheme (4.79) gives the truncation error

$$\begin{aligned} T_n =& \frac{1}{2}\Delta\tau w_{\tau\tau}^{\epsilon}(\xi,\zeta) + \frac{\Delta\xi}{4K_\xi}\left(\frac{1}{w^{\epsilon}(\chi_1,\tau)}\right)_{\xi\xi} \\ &- \frac{\epsilon\Delta\xi^2}{24K_\xi^2}\left(w^{\epsilon}(\chi_1,\tau)_{\xi\xi\xi\xi} + w^{\epsilon}(\chi_2,\tau)_{\xi\xi\xi\xi}\right), \end{aligned} \tag{4.83}$$

where ζ, χ_1 and χ_2 are intermediate values.

We may also rearrange the reference space scheme (4.79) into the form required for Harten's theorem. This gives the required coefficients to be

$$C_{j-1/2} = \frac{\Delta\tau}{c}\left(\frac{1}{2w_j^n w_{j-1}^n} + \frac{\epsilon}{c}\right) \quad \text{and} \quad D_{j+1/2} = \frac{\epsilon\Delta\tau}{c^2} \tag{4.84}$$

which may both be shown to be non-negative since all of the variables are known to be positive. All that remains is to show under what conditions

$$1 - C_{j-1/2} - D_{j-1/2} \geqslant 0. \tag{4.85}$$

Inserting $C_{j-1/2}$ and $D_{j-1/2}$ into this inequality gives the timestep restriction,

$$\Delta t \leqslant \frac{2c^2}{4\epsilon + cu_j^n u_{j-1}^n}. \tag{4.86}$$

4.5.5 Rate of Convergence

In the previous section we were able to show that we may choose the timestep of the scheme such that the scheme is TVD. However, this does not guarantee convergence to the correct solution.

We now introduce some important concepts before showing how convergence can be obtained.

Definition 4.5.7. The $L^\infty([0,T], L^p(\Omega))$ Bochner norm is defined as

$$||u||_{L^\infty([0,T],L^p(\Omega))} = \text{ess sup}_{t\in[0,T]}||u(t)||_{L^p(\Omega)} \tag{4.87}$$

where ess sup is the essential supremum which is the supremum over all but finitely many points. Furthermore $p \geqslant 1$ and Ω is the spatial domain.

To simplify notation we denote the Bochner norm by

$$||u||_{L^\infty(L^p)} \tag{4.88}$$

where there is no confusion in doing so.

In this section we aim to obtain a bound on the error of the regularised scheme (4.82) of the form

$$||u - u_h^\epsilon||_{L^\infty(L^1)} \leqslant E(\epsilon) \tag{4.89}$$

where u_h^ϵ is the regularised moving mesh numerical solution, u is the entropy solution to the conservation law (3.1), and $E(\epsilon)$ is some function also to be determined.

Theorem 4.5.8. *Let u be the entropy solution to the conservation law (3.1) with an initial condition, $u^0(x)$, and periodic boundary conditions and u_h^ϵ be the numerical approximation (4.82). Further assume that w^ϵ is bounded and sufficiently smooth so that $w_\xi^\epsilon < C_1$, $w_{\xi\xi}^\epsilon < C_2$, $w_{\xi\xi\xi\xi}^\epsilon < C_3$ and $w_{\tau\tau}^\epsilon < C_4$. Then the error between u and $u_h^\epsilon$is given by*

$$||w - w_h^\epsilon||_{L^\infty(L^1)} \leqslant C_5|u_0|_{TV}\sqrt{t\delta} + \frac{\Delta\tau C_4}{2} + \frac{\Delta\xi}{4K_\xi}\big(2\{C_1^2||u_0^3||_{L^\infty} - C_2||u_0^2||_{L^\infty}\big) - \frac{\epsilon\Delta\xi^2}{12K_\xi^2}C_3 \tag{4.90}$$

where $|u_0|_{TV}$ denotes the total variation of the initial condition.

In order to prove this result we must first refer to some other theorems.

Theorem 4.5.9. *If u is an entropy solution of the conservation law (3.1) and u^ϵ is* *a smooth solution of the regularised PDE then*

$$||u - u^{\epsilon}||_{L^{\infty}([0,T),L^{1}(a(t),b(t)))} \leqslant C_5|u_0|_{TV}\sqrt{t\epsilon}. \tag{4.91}$$

where C_5 is a constant independent of ϵ.

We refer the reader to Theorem 6.1 of [Fur01] for the proof of this result.

Lemma 4.5.10. *Let w be the entropy solution to the transformed conservation law (4.78), w_h^{ϵ} be the regularised transformed numerical solution given by (4.79), u be the entropy solution of the conservation law (3.1), and u_h^{ϵ} be the moving mesh numerical solution given by (4.82) then*

$$||u - u_h^{\epsilon}||_{L^{1}(a(t),b(t))} \leqslant ||u_0||_{L^{\infty}} K_{\xi} ||w - w_h^{\epsilon}||_{L^{\infty}(L^{1}(0,1))}. \tag{4.92}$$

Proof. Starting with Definition 4.5.7 applied to $w - w_h^{\epsilon}$ gives

$$||w - w_h^{\epsilon}||_{L^{\infty}(L^{1}(0,1))} = \text{ess sup}_{t\in[0,T]} \int_0^1 |w - w_h^{\epsilon}| d\xi \tag{4.93}$$

Transforming from reference space to physical space gives

$$||w - w_h^{\epsilon}||_{L^{\infty}(L^{1}(0,1))}{=}\text{ess sup}_{t\in[0,T]} \int_{a(t)}^{b(t)} \left|\frac{1}{u} - \frac{1}{u_h^{\epsilon}}\right| \frac{1}{\hat{x}_{\xi}} d\hat{x}. \tag{4.94}$$

Using $\hat{x}_{\xi}u = K_{\xi}$ and rearranging yields

$$||w - w_h^{\epsilon}||_{L^{\infty}(L^{1}(0,1))}{=}\text{ess sup}_{t\in[0,T]} \int_{a(t)}^{b(t)} \frac{|u - u_h^{\epsilon}|}{|u||u_h^{\epsilon}|} \frac{u}{K_{\xi}} d\hat{x}, \tag{4.95}$$

and since both u and u^ϵ are positive and bounded by the initial condition $u_0(x)$,

$$||w - w_h^\epsilon||_{L^\infty(L^1(0,1))} \geqslant \frac{1}{||u_0||_{L^\infty} K_\xi} \text{ess sup}_{t\in[0,T]} \int_{a(t)}^{b(t)} |u - u_h^\epsilon| d\hat{x}. \tag{4.96}$$

Rearranging and using the definition of the $L^\infty(L^1)$ norm yields the result. □

Proof of Theorem 4.5.8. We start by considering the error between the entropy solution of the transformed conservation law and the transformed numerical solution and using the triangle inequality

$$||w - w_h^\epsilon||_{L^\infty(L^1)} \leqslant ||w - w^\epsilon||_{L^\infty(L^1)} + ||w^\epsilon - w_h^\epsilon||_{L^\infty(L^1)}, \tag{4.97}$$

where w^ϵ is the solution to the regularised transformed PDE.

Since w is an entropy solution to a conservation law and w^ϵ is the vanishing viscosity regularisation we may apply Theorem 4.5.9 to obtain a bound on $||w - w^\epsilon||_{L^\infty(L^1)}$. Hence

$$||w - w_h^\epsilon||_{L^\infty(L^1)} \leqslant C_5 |w_0|_{TV} \sqrt{t\delta} + ||w^\epsilon - w_h^\epsilon||_{L^\infty(L^1)}. \tag{4.98}$$

The final term on the right hand side of (4.98) is the error between the solution of the regularised transformed PDE and the transformed numerical solution. Hence we refer to the truncation error given by (4.83), so that

$$||w - w_h^\epsilon||_{L^\infty(L^1)} \leqslant C_5 |w_0|_{TV} \sqrt{t\delta} + \frac{\Delta\tau}{2} w_{\tau\tau}^\epsilon + \frac{\Delta\xi}{4K_\xi} \left(\frac{1}{w^\epsilon}\right)_{\xi\xi} - \frac{\epsilon\Delta\xi^2}{12K_\xi^2} w_{\xi\xi\xi\xi}^\epsilon. \tag{4.99}$$

Directly differentiating $\left(\frac{1}{w^\epsilon}\right)_{\xi\xi}$ gives

$$\left(\frac{1}{w^\epsilon}\right)_{\xi\xi} = \frac{2(w_\xi^\epsilon)^2}{(w^\epsilon)^3} - \frac{w_{\xi\xi}^\epsilon}{(w^\epsilon)^2} \tag{4.100}$$

which can be inserted into (4.99) to give

$$||w - w_h^\epsilon||_{L^\infty(L^1)} \leqslant C_5|w_0|_{TV}\sqrt{t\delta} + \frac{\Delta\tau}{2}w^\epsilon_{\tau\tau} + \frac{\Delta\xi}{4K_\xi}\left(\frac{2(w^\epsilon_\xi)^2}{(w^\epsilon)^3} - \frac{w^\epsilon_{\xi\xi}}{(w^\epsilon)^2}\right) - \frac{\epsilon\Delta\xi^2}{12K_\xi^2}w^\epsilon_{\xi\xi\xi\xi}. \tag{4.101}$$

Using the assumptions of bounds on the derivatives of w^ϵ in equation (4.101) yields

$$||w - w_h^\epsilon||_{L^\infty(L^1)} \leqslant C_5|w_0|_{TV}\sqrt{t\delta} + \frac{\Delta\tau C_4}{2} + \frac{\Delta\xi}{4K_\xi}\left(\frac{2C_1^2}{(w^\epsilon)^3} - \frac{C_2}{(w^\epsilon)^2}\right) - \frac{\epsilon\Delta\xi^2}{12K_\xi^2}C_3. \tag{4.102}$$

We can now bound above by using the fact that $u = \frac{1}{w}$ and that u is bounded by the initial solution gives

$$||w - w_h^\epsilon||_{L^\infty(L^1)} \leqslant C_5|u_0|_{TV}\sqrt{t\delta} + \frac{\Delta\tau C_4}{2} + \frac{\Delta\xi}{4K_\xi}\big(2\{C_1^2||u_0^3||_{L^\infty} - C_2||u_0^2||_{L^\infty}\big) - \frac{\epsilon\Delta\xi^2}{12K_\xi^2}C_3 \tag{4.103}$$

Finally applying Theorem 2.15 yields the result. □

4.5.6 Experimental Order of Convergence

In the final section of this chapter we look at some numerical results to determine the Experimental Order of Convergence (EOC) for some of our test schemes. The equations we will apply this to are the Linear Advection Equation (2.32) and Inviscid Burgers' Equation (2.34).

We run the schemes multiple times doubling the number of intervals N each time and calculating the error, $e_N = |u(\hat{x}_j^n, t^n) - u_j^n|$, in the Bochner norms $L^\infty(L^1)$, $L^\infty(L^2)$ and $L^\infty(L^\infty)$. We then calculate the experimental order of convergence (EOC) using

$$EOC(N) = \frac{\log\left(\frac{||e_{N/2}||}{||e_N||}\right)}{\log(2)} \tag{4.104}$$

where the norms correspond to the Bochner norm the error is measured in.

The experimental order of convergence is a useful notion for showing that the results of the numerical schemes are consistent with the theoretical rates of convergence. To this end what we actually seek is what happens to the EOC as $N \to \infty$.

The EOC works by taking successive mesh refinements and calculating the error in the coarser mesh divided by the error in the finer mesh. In order to make the results easier to see on a graph we take the natural logarithm of this fraction and divide by another natural logarithm to normalise. In this thesis we double the computational nodes in each successive test. This is why Equation (4.104) has the division by log(2), if we instead had chosen to triple the number of nodes for each comparison we would instead choose log(3).

Note that in this Section we will show graphs with the Bochner norm errors for each time the code is run. The lighter colours, starting with yellow, represent the fewest number of computational intervals while the darker colours, ending in black, have the highest number of intervals.

Linear Advection Equation

We start by testing the scheme on the linear advection equation. This is a very simple problem, however it is important since it will highlight several key points concerning how the scheme performs. The equation is given by

$$u_t + u_x = 0, \tag{4.105}$$

where we take the boundary condition to be that of the free Lagrangian boundary. In this case the Lagrangian velocity is simply $\hat{x}_t = 1$.

We use the numerical method derived in the example in Section 3.6.1. Explicitly this is

$$\begin{cases} \hat{x}_j^{n+1} = \hat{x}_j^n + \Delta t \\ (\hat{x}_j^n - \hat{x}_{j-1}^n) u_j^n = A \end{cases} . \tag{4.106}$$

The first initial data we apply this scheme to is

$$u(x,t) = \frac{1}{\exp(5x^2)} + 0.1, \quad x \in [-2, 2]. \tag{4.107}$$

Note that we have added 0.1 to the initial condition due to our requirement that $u(x,t) > 0$.

In Figure 4.3 we see that the EOC in all three norms is 1 and the errors do not increase in time. This is as we would expect since Forward Euler is exact for linear problems and the only error is in the data representation of the numerical solution.

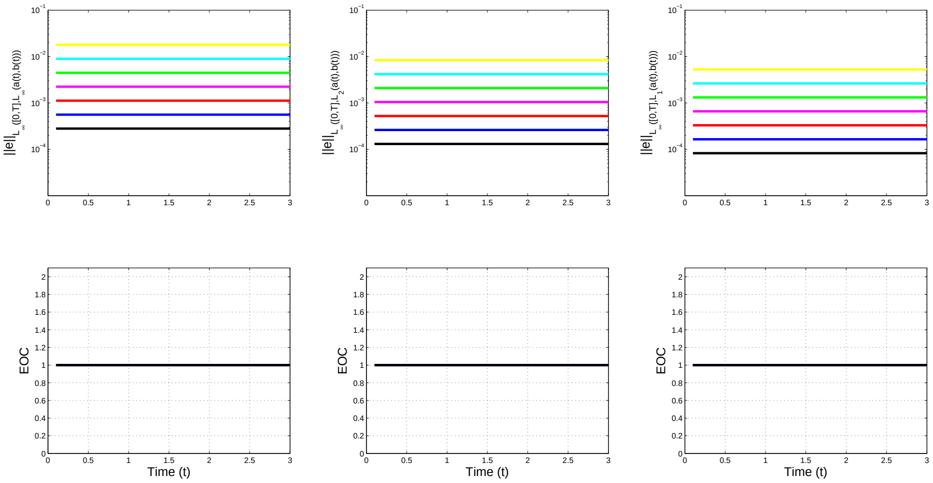


Figure 4.3: Global Errors and associated EOC for the numerical scheme (4.106) applied the linear advection equation with initial data (4.107). The $L^\infty(L^\infty)$ error is on the left, the $L^\infty(L^2)$ error is in the middle and the $L^\infty(L^1)$ is on the right.

A more challenging problem is to see how the scheme copes when there is a discontinuity in the initial data. We therefore propose the initial data

$$u(x,t) = \begin{cases} 0.15 & x \leqslant \pi - 3 \\ 0.05 & x > \pi - 3 \end{cases}, \tag{4.108}$$

this ensures that the discontinuity will not have a node placed on it in the initial node placement which is important since otherwise the scheme is exact for piecewise constant initial data.

In Figure 4.4, it may be initially worrying that we do not see convergence in the $L^\infty(L^\infty)$ norm, however this is to be expected since this error is caused at the discontinuity. Increasing the nodes in the scheme reduces the error in $L^\infty(L^1)$ and $L^\infty(L^2)$ however it does this by reducing the distance of nodes from the discontinuity, the $L^\infty(L^\infty)$ error however remains equal to the jump in the discontinuity since a

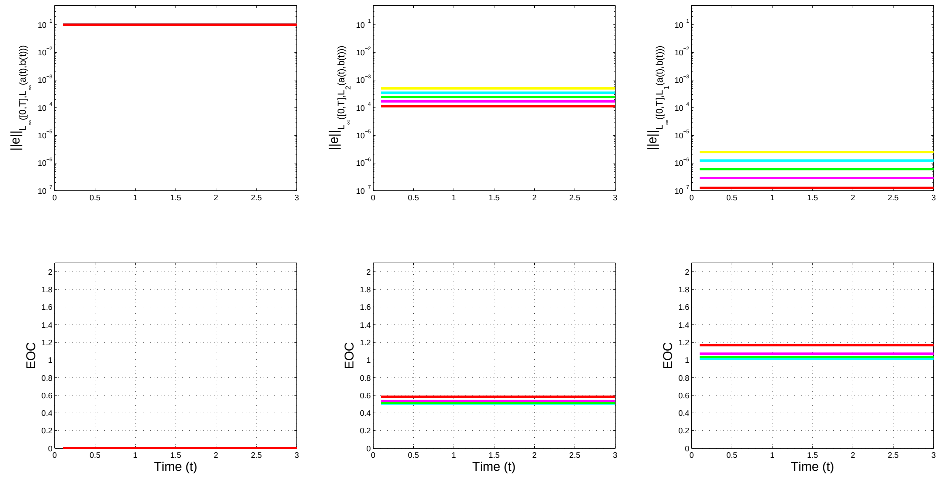


Figure 4.4: Global Errors and associated EOC for the numerical scheme (4.106) applied the linear advection equation with initial data (4.108). The $L^\infty(L^\infty)$ error is on the left, the $L^\infty(L^2)$ error is in the middle and the $L^\infty(L^1)$ is on the right.

node never exists on the discontinuity.

Inviscid Burgers Equation

We now recall Inviscid Burgers' Equation from Section 2.1.5. This conservation law is given by

$$u_t + \left(\frac{1}{2}u^2\right)_x = 0. \tag{4.109}$$

As with the linear advection equation we use the example scheme from Section 3.6.2. The scheme is given by

$$\begin{cases} \widehat{x}_j^{n+1} = \widehat{x}_j^n + \dfrac{\Delta t}{2} U_j^n \\ (\widehat{x}_j^n - \widehat{x}_{j-1}^n) u_j^n = A \end{cases}. \tag{4.110}$$

It is noted that unlike the linear advection case, this time the scheme is not exact in time hence we expect the error to increase in time.

We use the initial data

$$u(x,t) = \begin{cases} x + 0.1 & x \in [0,1] \\ 2.1 - x & x \in (1,2] \\ 0.1 & \text{otherwise} \end{cases}, \tag{4.111}$$

which is initially piecewise linear and forms a shock at time $t = 1$.

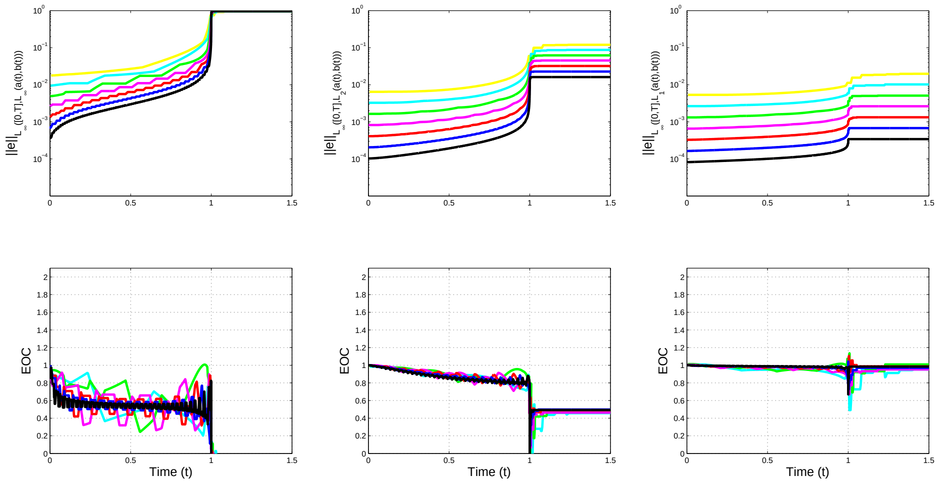


Figure 4.5: Global Errors and associated EOC for the numerical scheme (4.110) applied the linear advection equation with initial data (4.111). The $L^\infty(L^\infty)$ error is on the left, the $L^\infty(L^2)$ error is in the middle and the $L^\infty(L^1)$ is on the right.

In this test case we appear to converge in all norms pre-shock the $L^\infty(L^1)$ error has a steady convergence rate of 1 in this region, the $L^\infty(L^2)$ error starts at a convergence rate of 1 but decreases to a rate of 0.8 as shock time is approached and the $L^\infty(L^\infty)$ error converges at a rate of about 0.5. The noise in the $L^\infty(L^1)$ error is likely due to the derivative discontinuity in the solution passing the nodes with the error peak at the centre of each cell.

In the post-shock time regime we no longer converge in $L^\infty(L^\infty)$. This may be due to similar issues with the discontinuity in the previous example, namely there is not a node on the discontinuity and therefore the error in this norm cannot be less than the jump in the shock. The $L^\infty(L^2)$ error converges in the post-shock time regime with a rate of 0.5 and the $L^\infty(L^1)$ error appears to continue to converge with a rate of 1, this needs studying further to determine why.

In conclusion the schemes do converge in the $L^\infty(L^1)$ norm as expected. Issues occur in the $L^\infty(L^\infty)$ norm due to errors around the discontinuity which is not unexpected for conservation laws.

Chapter 5

Lagrangian Schemes Based on Existing Conservative Schemes

In the previous chapter we used a transformation of the conservation law and the moving mesh scheme to a reference space in order to find stability conditions for the scheme and prove convergence. The proof of convergence relied on regularising the numerical scheme in the reference space and working backwards to see how this changed the original moving mesh scheme.

In this section we build upon the idea that we can take schemes applied to the transformed conservation law and work backwards to a moving mesh mass conservative scheme. Hence instead of applying a scheme to the original conservation law

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

we instead apply standard finite difference schemes to the transformed PDE

$$w_\tau + \tilde{f}(w)_\xi = 0, \tag{5.2}$$

where $\tilde{f}(w) = -\frac{w}{K_\xi} f\left(\frac{1}{w}\right)$.

The benefit of this approach is that if we are able to find a method for taking established finite difference schemes and deriving moving mesh schemes from them, then our work from Sections 4.3 - 4.5 proves that the resulting scheme will have the same stability conditions and be convergent.

5.1 Existence of Schemes

Since the transformed PDE (5.2) is a conservation law and generally nonlinear, we used established theory for finite difference schemes which says that the scheme we apply should be able to be written in conservation form.

Definition 5.1.1 (Conservation Form)**.** If a scheme can be written in the form

$$u_j^{n+1} = u_j^n - \frac{\Delta t}{\Delta x}\left(F(u_{j-p}^n, u_{j-p+1}^n, \cdots, u_{j+q}^n) - F(u_{j-p-1}^n, u_{j-p}^n, \cdots, u_{j+q-1}^n)\right) \tag{5.3}$$

for some p and q which are positive integers, then the scheme is a conservative method. F is the numerical flux function consistent with the flux function $f(u)$. This means that if we insert the exact solution into the numerical flux function then, $F(u, ..., u) = f(u)$.

Schemes which can be written in conservation form have been shown to be mass conservative, which we require since we wish to derive a mass conservative scheme.

Theorem 5.1.2. *If a finite difference scheme can be written in conservation form then it admits at least one conservative moving mesh scheme for the conservation law (5.1) when applied to the reference space conservation law (5.2).*

Proof. Applying the general conservation form (5.3) to the transformed conservation law (5.2) gives

$$
\begin{aligned}
w_j^{n+1} = w_j^n - \frac{\Delta\tau}{\Delta\xi}\Big(&\tilde{F}(w_{j-p}^n, w_{j-p+1}^n, \cdots, w_{j+q}^n)\\
&-\tilde{F}(w_{j-p-1}^n, w_{j-p}^n, \cdots, w_{j+q-1}^n)\Big). \qquad (5.4)
\end{aligned}
$$

Now rewriting $\tilde{G} = K_\xi \tilde{F}$ yields,

$$
\begin{aligned}
w_j^{n+1} = w_j^n - \frac{\Delta\tau}{\Delta\xi K_\xi}\Big(&\tilde{G}(w_{j-p}^n, w_{j-p+1}^n, \cdots, w_{j+q}^n)\\
&-\tilde{G}(w_{j-p-1}^n, w_{j-p}^n, \cdots, w_{j+q-1}^n)\Big). \qquad (5.5)
\end{aligned}
$$

Defining the local mass constant to be

$$
A = \Delta\xi K_\xi, \qquad (5.6)
$$

and inserting into (5.5) gives

$$
\begin{aligned}
A(w_j^{n+1} - w_j^n) = \Delta\tau\Big(&\tilde{G}(w_{j-p}^n, w_{j-p+1}^n, \cdots, w_{j+q}^n)\\
&-\tilde{G}(w_{j-p-1}^n, w_{j-p}^n, \cdots, w_{j+q-1}^n)\Big). \qquad (5.7)
\end{aligned}
$$

Assume now that local mass conservation is approximated by the upwind form,

$$
Aw_j^n = \hat{x}_j^n - \hat{x}_{j-1}^n, \qquad (5.8)
$$

which we note is in the form of the general quadrature (3.49). Equation (5.8) may then be substituted into (5.7) to give,

$$\begin{aligned}\hat{x}_j^{n+1} - \hat{x}_{j-1}^{n+1} - \hat{x}_j^n + \hat{x}_{j-1}^n = \Delta\tau \Big(&\tilde{G}(w_{j-p}^n, w_{j-p+1}^n, \cdots, w_{j+q}^n) \\ &-\tilde{G}(w_{j-p-1}^n, w_{j-p}^n, \cdots, w_{j+q-1}^n)\Big) . \qquad (5.9)\end{aligned}$$

Taking equation (5.9) at $j = 1$ gives

$$\hat{x}_1^{n+1} - \hat{x}_1^n = \Delta t\tilde{G}(w_{1-p}^n, w_{2-p}^n, \cdots, w_{1+q}^n) \qquad (5.10)$$

since $j = 0$ is the given boundary so $\hat{x}_0^{n+1} - \hat{x}_0^n = \Delta t\tilde{G}(w_{-p}^n, w_{1-p}^n, \cdots, w_q^n)$ by definition as this is the discrete form of the boundary velocity. By induction this holds for all subsequent intervals and when combined with the fact that $\Delta\tau = \Delta t$ gives the numerical mesh movement approximation to be

$$\hat{x}_j^{n+1} = \hat{x}_j^n + \Delta t\tilde{G}(w_{j-p}^n, w_{j-p+1}^n, \cdots, w_{j+q}^n), \qquad (5.11)$$

which we note is in the general Runge-Kutta form (3.48).

Together equations (5.8) and (5.11) form a valid mass conservative moving mesh scheme. □

5.2 Uniqueness of Schemes

Theorem 5.1.2 shows that for any standard Eulerian finite difference scheme that can be written in conservation form it is possible to derive a corresponding moving mesh numerical scheme. However, there is not a unique moving mesh scheme for each given Eulerian scheme.

This non-uniqueness of moving mesh schemes can be seen if we consider the assumption (5.8) made in arriving at the moving mesh scheme. An equally valid approximation to local mass conservation would be

$$Aw_j^n = \hat{x}_{j+1}^n - \hat{x}_j^n. \tag{5.12}$$

Following the same steps as the proof of theorem 5.1.2 yields the alternative timestepping scheme

$$\hat{x}_j^{n+1} = \hat{x}_j^n + \Delta t\tilde{G}(w_{j-1-p}^n, w_{j-p}^n, \cdots, w_{j+q-1}^n). \tag{5.13}$$

It is clear that any quadrature which allows us to eliminate w terms on the LHS of equation (5.7) will produce a distinct moving mesh scheme for the original PDE. Hence the scheme is only unique up to the choice of quadrature approximation to the monitor function and it remains to be shown if there is a 'best' choice of approximation to produce a moving mesh scheme.

5.3 An Example Scheme

To illustrate the derivation of the moving mesh formulation we consider how we apply the well known first order upwind approximation,

$$u_j^{n+1} = u_j^n + \frac{\Delta t}{\Delta x}\left(F(u_j^n) - F(u_{j-1}^n)\right), \tag{5.14}$$

to the Inviscid Burgers equation,

$$u_t + \left(\frac{u^2}{2}\right)_x = 0. \tag{5.15}$$

Using the general forms (5.1) and (5.2) it can be easily shown that in the transformed reference space the PDE associated with Inviscid Burgers equation is,

$$w_\tau - \frac{1}{2K_\xi}\left(\frac{1}{w}\right)_\xi = 0. \tag{5.16}$$

Applying the first order upwind approximation (5.14) to the transformed PDE (5.16) and noting that $K_\xi \Delta\xi = A$ gives the reference space scheme,

$$w_j^{n+1} = w_j^n - \frac{\Delta\tau}{2A}\left(\frac{1}{w_j^n} - \frac{1}{w_{j-1}^n}\right), \tag{5.17}$$

which we note is the transformed scheme from Example 3.6.3 in Section 3.6. Indeed, taking the same approximation as in the example, namely

$$Aw_j^n = \hat{x}_j^n - \hat{x}_{j-1}^n, \tag{5.18}$$

yields

$$\hat{x}_j^{n+1} - \hat{x}_{j-1}^{n+1} - \hat{x}_j^n + \hat{x}_{j-1}^n = \frac{\Delta\tau}{2}\left(\frac{1}{w_j^n} - \frac{1}{w_{j-1}^n}\right). \tag{5.19}$$

The anchor point, the fact that $\Delta\tau = \Delta t$ and $u_j^n = \frac{1}{w_j^n}$ gives the timestepping scheme to be

$$\hat{x}_j^{n+1} = \hat{x}_j^n + \frac{\Delta t}{2} u_j^n. \tag{5.20}$$

The overall moving mesh scheme to solve the Inviscid Burgers equation (5.15) is then

$$\hat{x}_j^{n+1} = \hat{x}_j^n + \frac{\Delta t}{2} u_j^n, \tag{5.21}$$

$$A = u_j^n \left(\hat{x}_j^n - \hat{x}_{j-1}^n \right), \tag{5.22}$$

which is the scheme we started with in the example in section 3.6. However as noted in section 5.2 this is not the only moving mesh scheme that can be derived from starting with first order upwind as a method.

Instead of approximating the local mass integral using (5.18) we instead use the approximation (5.12) which leads to the timestepping scheme

$$\hat{x}_j^{n+1} = \hat{x}_j^n + \frac{\Delta t}{2} u_{j-1}^n. \tag{5.23}$$

The alternative scheme is then given by

$$\hat{x}_j^{n+1} = \hat{x}_j^n + \frac{\Delta t}{2} u_{j-1}^n, \tag{5.24}$$

$$A = u_j^n \left(\hat{x}_{j+1}^n - \hat{x}_j^n \right). \tag{5.25}$$

This leads us to the obvious question of which scheme is actually better for solving the original Inviscid Burgers equation. As both are derived from the same reference space numerical scheme, it is clear from the results of Section 4.4 that they both have the same stability condition and accuracy.

5.4 Higher Order Schemes

Having established that we may use existing fixed mesh schemes as a basis for generating moving mesh schemes, it remains to be discussed what effect changing the order of the underlying scheme has on the resulting moving mesh scheme.

It is clear that since higher order schemes can be written in conservation form then they also produce moving mesh schemes when applied to the transformed PDE. However this does not guarantee that the resulting scheme will be of the same order or even an increased order. Fortunately the work of Section 4.3 applies and therefore the schemes generated are of the same order as the Eulerian scheme used to generate them.

5.5 Numerical Comparisons

In the rest of this section we have developed the idea of using established fixed grid numerical methods as a way of generating moving mesh methods. It remains to give a demonstration of why we would choose to do this extra work in deriving a scheme.

In this section we compare the results of directly applying the fixed grid schemes with the results of the resulting moving mesh schemes as numerical motivation for this extra work.

The schemes we will consider are the first order upwind scheme

$$u_j^{n+1} = u_j^n - \frac{\Delta t}{\Delta x}(F_j^n - F_{j-1}^n), \tag{5.26}$$

and the second order upwind scheme

$$u_j^{n+1} = u_j^n - \frac{\Delta t}{2\Delta x}(3F_j^n - 4F_{j-1}^n + F_{j-2}^n). \tag{5.27}$$

We compare these Eulerian schemes with Lagrangian schemes derived from them in two different test problems. In each problem we used 41 computational nodes, a time step of $\Delta t = 0.001$ and run to an end time of $t = 1.1$. In addition we start both schemes on the same uniform mesh for the best comparison.

5.5.1 Linear Advection

The first problem we consider is the linear advection equation introduced in Section 2.1.5. This is augmented with the initial condition

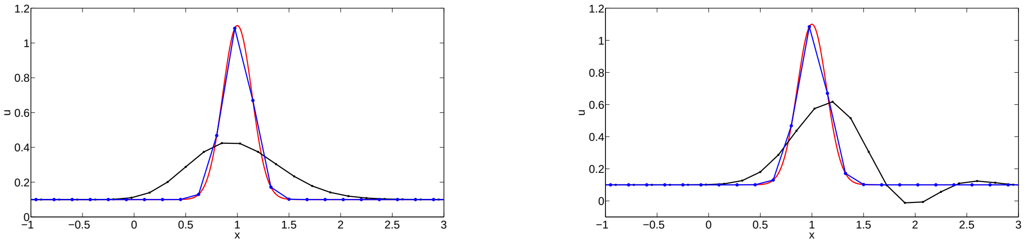
$$u(x,t) = \frac{1}{\exp(5x^2)} + 0.1, \quad x \in [-2,2], \tag{5.28}$$

and the boundary condition

$$u(-2,t) = 0.1. \tag{5.29}$$

In Figure 5.1(a) we plot the solutions from the first order upwind scheme and in Figure 5.1(b) we plot the solutions from the second order upwind scheme, both at time $T = 1$.

It is clear from Figure 5.1 that the moving mesh schemes far outperform the Eulerian schemes from which they are derived. This is not unexpected however since the linear advection equation is a special case for the moving mesh schemes where there is no error accumulation in time.



(a) First Order Upwind

(b) Second Order Upwind

Figure 5.1: Numerical comparison of the moving mesh schemes (blue) and the Eulerian schemes (black) which they are derived from when applied to the linear advection equation. The plotted solution is at time, $T = 1$. The exact solution is plotted in red.

5.5.2 Inviscid Burgers' Equation

The next test problem we consider is Inviscid Burgers' Equation which also was first introduced in Section 2.1.5. We give the initial condition

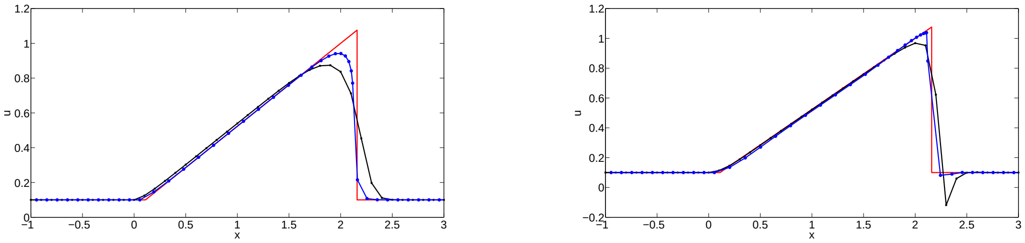
$$u(x,0) = \begin{cases} x + 0.1 & 0 < x \leqslant 1 \\ 2.1 - x & 1 < x \leqslant 2 \\ 0.1 & \text{otherwise} \end{cases}, \tag{5.30}$$

and the boundary condition

$$u(-1,t) = 0.1. \tag{5.31}$$

In Figure 5.2(a) we plot the solutions from the first order upwind scheme and in Figure 5.2(b) we plot the solutions from the second order upwind scheme.

As was the case with the linear advection equation, the Lagrangian schemes both out performed the Eulerian schemes they were based on. This provides some evidence that the extra work to apply the schemes in this manner is worthwhile.



(a) First Order Upwind

(b) Second Order Upwind

Figure 5.2: Numerical comparison of the moving mesh schemes (blue) and the Eulerian schemes (black) which they are derived from when applied to Inviscid Burgers' Equation. The exact solution is plotted in red.

Chapter 6

Systems of Equations

In this chapter we attempt to apply the conservation based moving mesh methods to a system of hyperbolic conservation laws to see if the insights we have obtained in the scalar case can help.

We start by briefly discussing some of the issues that arise when attempting to solve systems of equations before looking at the isothermal Euler equations as a test problem.

6.1 Problems that Arise with Systems of Equations

In this section we briefly cover some of the issues that occur when attempting to solve systems of equations with our conservation based moving mesh methods.

The first major difference is that we are now considering more than one conservation law and as a result we must choose which of the conserved quantities will be used as a monitor function to find the mesh velocity. Special care has to be taken here to ensure that the conserved quantity cannot be zero in the domain or the scheme will break down. Another item to note is that we cannot take the conservation law in vector form and conserve $\mathbf{u}$ since this would lead to a vector of positions for each node, although it may be possible to consider some functional of the components of $\mathbf{u}$.

The next major problem that arises is that the reference space PDEs are much more difficult to solve and are indeed on par with the original physical space conservation laws. To overcome this we change direction slightly for systems and instead consider a $\hat{x}$ equation in the reference space.

Now that we have briefly discussed some of the issues with systems of equations we now move on to attempt to solve the isothermal Euler equations.

6.2 Isothermal Equations

In this section we consider the 1D Isothermal equations given in Eulerian coordinates by

$$\rho_t + (\rho v)_x = 0, \tag{6.1}$$

$$(\rho v)_t + (\rho v^2 + P)_x = 0, \tag{6.2}$$

where $\rho > 0$ is density, v is the fluid velocity, $P = a^2\rho$ is the pressure and a is the wave speed.

We consider the system for general initial conditions $\rho(x, 0) = \rho_0(x)$ and $v(x, 0) = v_0(x)$, and periodic boundary conditions.

6.2.1 The Lagrangian Formulation

As in the case of scalar conservation laws we apply Leibniz integral rule to find the Lagrangian formulation of the problem. Application to the conservation of mass equation (6.1) gives

$$\begin{aligned} \frac{d}{dt}\int_{\hat{x}_1(t)}^{\hat{x}_2(t)} \rho dx &= \int_{\hat{x}_1(t)}^{\hat{x}_2(t)} \rho_t dx + [\rho \hat{x}_t]_{\hat{x}_1(t)}^{\hat{x}_2(t)} \\ &= -\int_{\hat{x}_1(t)}^{\hat{x}_2(t)} (\rho v)_x dx + [\rho \hat{x}_t]_{\hat{x}_1(t)}^{\hat{x}_2(t)} \\ &= [\rho \hat{x}_t - \rho v]_{\hat{x}_1(t)}^{\hat{x}_2(t)}, \end{aligned} \tag{6.3}$$

and similar application to the momentum equation (6.2) gives

$$\begin{aligned} \frac{d}{dt}\int_{\hat{x}_1(t)}^{\hat{x}_2(t)} \rho v dx &= \int_{\hat{x}_1(t)}^{\hat{x}_2(t)} (\rho v)_t dx + [\rho v \hat{x}_t]_{\hat{x}_1(t)}^{\hat{x}_2(t)} \\ &= -\int_{\hat{x}_1(t)}^{\hat{x}_2(t)} (\rho v^2 + P)_x dx + [\rho v \hat{x}_t]_{\hat{x}_1(t)}^{\hat{x}_2(t)} \\ &= [\rho v \hat{x}_t - \rho v^2 - P]_{\hat{x}_1(t)}^{\hat{x}_2(t)}. \end{aligned} \tag{6.4}$$

All that remains is to decide on the monitor function which will be used. The two obvious choices for monitor functions are the density, ρ, and the momentum, ρv, however we note that the monitor function must be one-signed and hence momentum is not suitable since v may be zero or negative.

Taking density, ρ, as the monitor function implies that the left hand side of the balance equation (6.3) is identically zero for all $\hat{x}_1$, $\hat{x}_2$. Hence

$$[\rho(\hat{x}_t - v)]_{\hat{x}_1(t)}^{\hat{x}_2(t)} = 0 \tag{6.5}$$

for all $\hat{x}_1$, $\hat{x}_2$ which, since ρ is one-signed, implies that $\hat{x}_t = v$. The Lagrangian formulation given by the density monitor is therefore

$$\hat{x}_t = v, \tag{6.6}$$

$$\int_{\hat{x}_1(t)}^{\hat{x}_2(t)} \rho dx = A(\hat{x}_1(t), \hat{x}_2(t)), \tag{6.7}$$

$$\frac{d}{dt}\int_{\hat{x}_1(t)}^{\hat{x}_2(t)} \rho v dx = -[P]_{\hat{x}_1(t)}^{\hat{x}_2(t)} \tag{6.8}$$

where $A(\hat{x}_1(t), \hat{x}_2(t))$ is constant in time.

Our aim is to solve the system (6.6)-(6.8) for $\hat{x}(t) = \hat{x}_2(t)$ (given an anchor point $\hat{x}_1(t)$) and then recover the solutions ρ and v at these positions.

6.2.2 A Lagrangian Numerical Scheme

Having found the Lagrangian formulation (6.6)-(6.8) based on the density monitor in the previous section, we now discretise this set of equations by following a similar approach to the one applied to scalar conservation laws. The first step is to use the Mean Value Theorem (MVT) to re-write (6.7) as

$$\int_{\hat{x}_1(t)}^{\hat{x}_2(t)} \rho(x,t)dx = \rho(\zeta_1,t)(\hat{x}_2(t) - \hat{x}_1(t)) = A(\hat{x}_1(t), \hat{x}_2(t)), \tag{6.9}$$

for some $\zeta_1 \in (\hat{x}_1, \hat{x}_2)$, and the integral (6.8) as

$$\int_{\hat{x}_1(t)}^{\hat{x}_2(t)} \rho(x,t)v(x,t)dx = \rho(\zeta_2,t)v(\zeta_2,t)(\hat{x}_2(t) - \hat{x}_1(t)), \tag{6.10}$$

for some $\zeta_2 \in (\hat{x}_1, \hat{x}_2)$ where A is constant in time and we have used (6.9) to simplify (6.10).

Introduce a set of discrete points $\{\hat{x}_j(t)\}$ at time t and let $\hat{x}_1(t) = \hat{x}_j(t)$ and $\hat{x}_2(t) = \hat{x}_{j+1}(t)$ define an individual cell of the discrete scheme. Assuming that ρ and v are constant within a cell $(\hat{x}_j, \hat{x}_{j+1})$ leads to the spatial discretisations of equations (6.9) and (6.10), which are

$$\rho_j(\hat{x}_{j+1}(t) - \hat{x}_j(t)) = A_{j+1/2} \tag{6.11}$$

and

$$\rho_j v_j(\hat{x}_{j+1}(t) - \hat{x}_j(t)) = B_{j+1/2}(t), \tag{6.12}$$

where $B_{j+1/2}(t)$ is the semi-discrete approximation to the momentum integral $B(t) = \int_{\hat{x}_j(t)}^{\hat{x}_{j+1}(t)} \rho(t)v(t)dx$.

We note that in the above, all constant approximations of ρ and v in a cell are chosen to be biased by taking the value at the left hand side of the cell. Further we note that equations (6.11) and (6.12) lead to a simple relationship between $A_{j+1/2}$ and $B_{j+1/2}(t)$, namely

$$A_{j+1/2}v_j(t) = B_{j+1/2}(t). \tag{6.13}$$

Together equations (6.6), (6.8), (6.11) and (6.12) form a semi-discrete numerical scheme in which the spatial co-ordinate $\hat{x}(t)$ is discretised and time remains continuous. To obtain a fully discrete formulation it remains to discretise the time evolution in equations (6.6) and (6.8).

Application of the forward Euler method to equation (6.6) yields

$$\hat{x}_j^{n+1} = \hat{x}_j^n + \Delta t v_j^n, \tag{6.14}$$

while application of the backward Euler method to equation (6.8) gives

$$B_{j+1/2}^{n+1} = B_{j+1/2}^{n} - a^2\Delta t(\rho_{j+1}^{n+1} - \rho_j^{n+1}). \tag{6.15}$$

Finally we use equation (6.13) to replace equation (6.15) by

$$v_j^{n+1} = v_j^n - \frac{a^2\Delta t}{A_{j+1/2}}(\rho_{j+1}^{n+1} - \rho_j^{n+1}). \tag{6.16}$$

The final form of the discrete scheme is therefore

$$\hat{x}_j^{n+1} = \hat{x}_j^n + \Delta t v_j^n, \tag{6.17}$$

$$\rho_j^{n+1} = \frac{A_{j+1/2}}{\hat{x}_{j+1}^{n+1} - \hat{x}_j^{n+1}}, \tag{6.18}$$

$$v_j^{n+1} = v_j^n - \frac{a^2\Delta t}{A_{j+1/2}}(\rho_{j+1}^{n+1} - \rho_j^{n+1}). \tag{6.19}$$

It is clear that this scheme is consistent with the PDEs (6.6)-(6.8), as for scalar conservation laws. We first update the mesh using equation (6.17) and then use equations (6.18) and (6.19) to recover the variables ρ and v at the new timestep.

We now attempt to follow the method for finding a stability criterion that was laid out in the scalar case and hence we apply the reference space transformation to equations (6.1) and (6.2).

6.2.3 Reference Space Transformation

In this section we derive the reference space transformations of equations (6.1) and (6.2). As is the case for scalar conservation laws the reference space is defined such that the interval $(\hat{x}_0(t), \hat{x}_N(t))$ maps to $(0, 1)$ for all time, t. Points moving with speed v in physical space are stationary in the reference space.

The transformation of the independent variables is given by $\hat{x}(\xi,\tau) \to \xi$, $t \to \tau$, following from Equation (4.13), leading to the transformations

$$\partial_t = \partial_\tau - \frac{\hat{x}_\tau}{\hat{x}_\xi}\partial_\xi, \tag{6.20}$$

and

$$\partial_x = \frac{1}{\hat{x}_\xi}\partial_\xi, \tag{6.21}$$

in the derivatives, following from the similar scalar case in Equation (4.14).

Applying (6.20) and (6.21) to equation (6.1) yields

$$\rho_\tau - \frac{v}{\hat{x}_\xi}\rho_\xi + \frac{1}{\hat{x}_\xi}(\rho v)_\xi = 0, \tag{6.22}$$

where $\hat{x}_\tau$ is substituted using Equation (6.6). Equation (6.22) then simplifies to give

$$\rho_\tau + \frac{\rho}{\hat{x}_\xi}v_\xi = 0. \tag{6.23}$$

Similarly, application of equations (6.20) and (6.21) to equation (6.2) gives

$$(\rho v)_\tau - \frac{v}{\hat{x}_\xi}(\rho v)_\xi + \frac{1}{\hat{x}_\xi}(\rho v^2 + \rho a^2)_\xi = 0. \tag{6.24}$$

Expanding out and using equation (6.23) to eliminate ρ_τ terms then yields

$$v_\tau + \frac{a^2}{\hat{x}_\xi}\frac{\rho_\xi}{\rho} = 0. \tag{6.25}$$

Since we have mass conservation in physical space for any $\hat{x}_1(t)$, $\hat{x}_2(t)$ we consider the transformed mass integral between the left hand boundary and an arbitrary point $\tilde{\xi} \in (0,1]$ giving

$$A(\tilde{\xi}) = \int_0^{\tilde{\xi}} \rho \hat{x}_\xi d\xi. \tag{6.26}$$

We may now differentiate (6.26) to give

$$A_\xi = \rho \hat{x}_\xi, \tag{6.27}$$

where we note that A_ξ is a constant. This follows from the scalar case and involves comparing the right hand side of (6.26) with the equidistribution principle for ρ.

Inserting (6.27) into equations (6.23) and (6.25) gives the reference space isothermal equations,

$$\rho_\tau + \frac{\rho^2}{A_\xi} v_\xi = 0, \tag{6.28}$$

$$v_\tau + \frac{a^2}{A_\xi} \rho_\xi = 0. \tag{6.29}$$

In the scalar case, having found the reference space transformations of the original PDEs we applied Harten's Theorem to show that the transformed scheme was TVD under certain timestep restrictions. We cannot use the same method here since it does not apply for systems of equations and we must therefore consider a new notion of stability.

Instead of concerning ourselves with the stability of the system (6.28)-(6.29) we instead look at the mesh stability. The motivation for this is found by considering equations (6.6) and (6.27) which show relationships between ρ, v and the derivatives of our mesh variable $\hat{x}$, namely

$$\hat{x}_\tau = v \quad \text{and} \quad \hat{x}_\xi = \frac{A_\xi}{\rho}.$$

Differentiating (6.6) with respect to τ and using (6.29) and (6.27) eliminates ρ and v and gives

$$\hat{x}_{\tau\tau} = v_\tau = -\frac{a^2}{A_\xi}\rho_\xi = -a^2\left(\frac{1}{\hat{x}_\xi}\right)_\xi \tag{6.30}$$

which may be rearranged to give the mesh PDE

$$\hat{x}_{\tau\tau} = a^2\frac{\hat{x}_{\xi\xi}}{\hat{x}_\xi^2}. \tag{6.31}$$

Since equation (6.31) is a non-linear wave equation we could use existing theory concerning finite difference schemes for second order equations in an attempt to show that the transformed scheme derived from equations (6.17)-(6.19) is stable under some condition, however this is not a common form of nonlinear wave equation and standard results such as assuming $\frac{a^2}{\hat{x}_\xi}$ is constant over a single timestep do not seem to work, as attempts to use this method lead to large instability regardless of timestep.

In the next section we work around this restriction by introducing a new numerical method based on moving the mesh using equation (6.31) instead of equation (6.6).

6.2.4 Furihata's Method

In this section we introduce a class of finite difference methods for solving non-linear wave equations that conserve an energy integral. This method is proposed by D. Furihata in “Finite-difference schemes for nonlinear wave equation that inherit energy conservation property" [Fur01] and presented in a general Eulerian $u(x,t)$ framework.

The family of nonlinear wave equations that are considered in [Fur01] take the form

$$\frac{\partial^2 u}{\partial t^2} = -\frac{\delta G}{\delta u}, \tag{6.32}$$

where $G = G(u, u_x)$ is a function of both u and u_x and $\frac{\delta G}{\delta u} = \frac{\partial G}{\partial u} - \frac{\partial}{\partial x}\left(\frac{\partial G}{\partial u_x}\right)$ is the variational derivative of G with respect to u. Furthermore $x \in [0, L], L < \infty$ is the one-dimensional spatial variable and t is the time variable.

Definition 6.2.1. Given $u(x,t)$ and a function G of u, u_x the energy integral is defined to be

$$I = \int_0^L \left(\frac{1}{2}u_t^2 + G\right) dx. \tag{6.33}$$

Theorem 6.2.2. *[Fur01] If the boundary conditions satisfy*

$$[G_{u_x} u_t]_0^L = 0, \tag{6.34}$$

then the energy integral, I, is conserved in time, i.e.

$$\frac{d}{dt}\int_0^L \left(\frac{1}{2}u_t^2 + G\right) dx = 0. \tag{6.35}$$

Proof. Applying Leibniz integral rule to the left hand side of equation (6.35),

$$\frac{d}{dt}\int_0^L \left(\frac{1}{2}u_t^2 + G\right) dx = \int_0^L \left(u_{tt}u_t + G_u u_t + G_{u_x} u_{xt}\right) dx. \tag{6.36}$$

Now use integration by parts on the final term of the integrand and apply the definition of the variational derivative,

$$\begin{aligned}\frac{d}{dt}\int_0^L \left(\frac{1}{2}u_t^2 + G\right) dx &= \int_0^L u_t \left(u_{tt} + G_u - \frac{\partial}{\partial x}G_{u_x}\right) dx + \left[G_{u_x}u_t\right]_0^L \\ &= \int_0^L u_t \left(u_{tt} + \frac{\delta G}{\delta u}\right) dx + \left[G_{u_x}u_t\right]_0^L . \qquad (6.37)\end{aligned}$$

Using the general form of the PDE (6.32) shows that the integrand is identically zero, hence

$$\frac{d}{dt}\int_0^L \left(\frac{1}{2}u_t^2 + G\right) dx = \left[G_{u_x}u_t\right]_0^L . \qquad (6.38)$$

Using the assumption on the boundary conditions (6.34) yields the required result.

□

In order to derive a numerical scheme using Furihata's method we first make an approximation to G which we call G_d. In general G has the form

$$G(u, u_x) = \sum_{l=0}^{\tilde{m}} f_l(u)g_l(u_x), \qquad (6.39)$$

where $f_l(u)$ are functions of u and $g_l(u_x)$ are functions of u_x. Note that this is not a restrictive assumption for our purposes since the functions f_l and g_l can be constants.

To discretise G we therefore take some $m \geqslant \tilde{m}$ and construct an approximation of the form

$$G_d(u_j) = \sum_{l=0}^{m} f_l(u_j)g_l^+(u_j)g_l^-(u_j), \qquad (6.40)$$

where $f_l(u_j)$ are functions of u_j, $g_l^+(u_j)$ are functions of u_j which approximate u_x using an upwind difference and $g_l^-(u_j)$ are functions of u_j which approximate u_x using a downwind difference.

Having found a consistent approximation G_d we now use it to calculate a discrete equivalent to the variational derivative $\frac{\delta G}{\delta u}$ which we denote $\frac{\delta G_d}{\delta(u_j,v_j)}$. This is achieved by considering the following property of the variational derivative, namely

$$\int_0^L G(u)dx - \int_0^L G(v)dx \approx \int_0^L \frac{\delta G}{\delta u}(u-v)dx + \left[G_{u_x}(u-v)\right]_0^L. \tag{6.41}$$

Before we can consider a discrete equivalent to (6.41) we first consider a discrete equivalent of integration by parts.

Theorem 6.2.3 (Summation by Parts)**.** *The summation by parts formula,*

$$\sum_{k=0}^{N} f_k \delta^+ g_k \Delta x = -\sum_{k=0}^{N} \delta^- f_k g_k \Delta x + \textit{boundary terms} \tag{6.42}$$

is a consistent discrete equivalent to integration by parts, where $\delta^+ f_j = \frac{f_{j+1}-f_j}{\Delta x}$ *and* $\delta^- f_j = \frac{f_j - f_{j-1}}{\Delta x}$

The proof of the above theorem can be found in chapter 3 of Furihata's book [FM11]. We note that we do not consider the boundary terms here since they are not important to deriving the numerical method.

Furihata shows that a discrete equivalent to equation (6.41) is given by

$$\sum_{j=0}^{N} G_d(u_j)\Delta x - \sum_{j=0}^{N} G_d(v_j)\Delta x = \sum_{j=0}^{N} \frac{\delta G_d}{\delta(u_j,v_j)}(u_j - v_j)\Delta x + \text{boundary terms.} \tag{6.43}$$

Equation (6.43) can easily be verified by applying the summation by parts formula to the left hand side of (6.43).

Equation (6.43) suggests that if we insert our approximation G_d into the left hand side of (6.43) and apply the summation by parts formula then we should arrive at the right hand side with our equivalent approximation, $\frac{\delta G_d}{\delta(u_j,v_j)}$, of $\frac{\delta G}{\delta u}$.

Theorem 6.2.4. *The explicit finite difference scheme,*

$$\frac{u_j^{n+1} - u_j^n - u_j^{n-1} + u_j^{n-2}}{2\Delta t} = -\frac{\delta G_d}{\delta(u_j^n, u_j^{n-1})}, \tag{6.44}$$

is consistent and conserves the discrete energy integral.

The proof of this theorem can be found in Furihata's paper [Fur01] as theorem 4.

In this section we have given an overview of how the Furihata method is applied to a nonlinear wave equation. In the next section we will show how this can be used as part of a Lagrangian scheme to solve the isothermal equations.

6.2.5 An Alternative Lagrangian Scheme Based on Furihata's Method

In section 6.2.3 we found the mesh PDE to be

$$\hat{x}_{\tau\tau} = a^2 \frac{\hat{x}_{\xi\xi}}{\hat{x}_\xi^2}. \tag{6.45}$$

which is a nonlinear wave equation. In section 6.2.4 we gave an overview of Furihata's method for developing an energy conserving finite difference method for solving such nonlinear wave equations. In this section we will put these two items together to find a stable scheme for the mesh movement and use this in a new Lagrangian scheme for solving the isothermal equations.

The main difference between the Lagrangian scheme derived in section 6.2.2 and the one we will derive here is the equations that we are approximating. In Section 6.2.2 our approximations were of the system of equations (6.6)-(6.8): now we choose to approximate equations (6.31), (6.27) and (6.6) which we restate for clarity,

$$\hat{x}_{\tau\tau} = a^2 \frac{\hat{x}_{\xi\xi}}{\hat{x}_\xi^2}, \tag{6.46}$$

$$\hat{x}_\xi = \frac{A_\xi}{\rho}, \tag{6.47}$$

$$\hat{x}_\tau = v. \tag{6.48}$$

The method we propose uses the Furihata scheme on (6.46) to update the mesh and once the desired time has been reached, approximations to (6.47) and (6.48) to recover the density and velocity respectively.

To find the Furihata scheme we must first relate equation (6.46) with the general nonlinear wave equation (6.32). It can be easily verified that equation (6.46) is indeed of the desired form with

$$G(\hat{x}, \hat{x}_\xi) = -a^2 \ln(\hat{x}_\xi). \tag{6.49}$$

We approximate the function G using (6.40) in the form

$$G_d(\hat{x}_j) = -\frac{a^2}{2}\left(\ln(\delta^+ \hat{x}_j) + \ln(\delta^- \hat{x}_j)\right). \tag{6.50}$$

Inserting (6.50) into the left hand side of equation (6.43) gives

$$\sum_{j=0}^{N}(G_d(\hat{x}_j^n) - G_d(\hat{x}_j^{n-1}))\Delta\xi = -\frac{a^2}{2}\sum_{j=0}^{N}(\ln(\delta^+\hat{x}_j^n) + \ln(\delta^-\hat{x}_j^n) \\ - \ln(\delta^+\hat{x}_j^{n-1}) - \ln(\delta^-\hat{x}_j^{n-1}))\Delta\xi, \qquad (6.51)$$

and applying the summation by parts formula gives

$$\sum_{j=0}^{N}(G_d(\hat{x}_j^n) - G_d(\hat{x}_j^{n-1}))\Delta\xi = \frac{a^2}{2}\sum_{j=0}^{N}\left(\delta^-\left(\frac{\ln(\delta^+\hat{x}_j^n) - \ln(\delta^+\hat{x}_j^{n-1})}{\delta^+\hat{x}_j^n - \delta^+\hat{x}_j^{n-1}}\right)\right. \\ \left. +\delta^+\left(\frac{\ln(\delta^-\hat{x}_j^n) - \ln(\delta^-\hat{x}_j^{n-1})}{\delta^-\hat{x}_j^n - \delta^-\hat{x}_j^{n-1}}\right)\right)(\hat{x}_j^n - \hat{x}_j^{n-1})\Delta\xi \\ +\text{boundary terms.} \qquad (6.52)$$

Comparison with the right hand side of equation (6.43) then shows that

$$\frac{\delta G_d}{\delta(\hat{x}_j^n, \hat{x}_j^{n-1})} = a^2\left(\frac{\ln(\Delta\hat{x}_{j+1/2}^n) - \ln(\Delta\hat{x}_{j+1/2}^{n-1})}{\Delta\hat{x}_{j+1/2}^n - \Delta\hat{x}_{j+1/2}^{n-1}} - \frac{\ln(\Delta\hat{x}_{j-1/2}^n) - \ln(\Delta\hat{x}_{j-1/2}^{n-1})}{\Delta\hat{x}_{j-1/2}^n - \Delta\hat{x}_{j-1/2}^{n-1}}\right), \qquad (6.53)$$

where $\Delta\hat{x}_{j+1/2} = \hat{x}_{j+1} - \hat{x}_j$.

It would now seem that we could simply insert (6.53) into the numerical scheme (6.44) and be finished. However, there is a computational issue with this form of the discrete variational derivative since as the denominator of either fraction → 0 the fraction is unbounded.

In order to get around this issue we rewrite (6.53) as

$$\frac{\delta G_d}{\delta(\hat{x}_j^n, \hat{x}_j^{n-1})} = a^2 \left(\frac{\ln(1+\alpha)}{\alpha \Delta \hat{x}_{j-1/2}^n} - \frac{\ln(1+\beta)}{\beta \Delta \hat{x}_{j+1/2}^n} \right) \tag{6.54}$$

where $\alpha = \frac{\Delta \hat{x}_{j-1/2}^n - \Delta \hat{x}_{j-1/2}^{n-1}}{\Delta \hat{x}_{j-1/2}^n}$ and $\beta = \frac{\Delta \hat{x}_{j+1/2}^n - \Delta \hat{x}_{j+1/2}^{n-1}}{\Delta \hat{x}_{j+1/2}^n}$.

Assuming that we are taking sufficiently small timesteps such that α and β are small then we may approximate (6.54) using the Taylor expansion of $\ln(1+\epsilon)$. Hence

$$\begin{aligned} \frac{\delta G_d}{\delta(\hat{x}_j^n, \hat{x}_j^{n-1})} &\approx a^2 \left(\frac{1}{\Delta \hat{x}_{j-1/2}^n} \left(1 - \frac{\alpha}{2}\right) - \frac{1}{\Delta \hat{x}_{j+1/2}^n} \left(1 - \frac{\beta}{2}\right) \right) \\ &= a^2 \left(\frac{3}{2} \left(\frac{1}{\Delta \hat{x}_{j-1/2}^n} - \frac{1}{\Delta \hat{x}_{j+1/2}^n} \right) - \frac{1}{2} \left(\frac{\Delta \hat{x}_{j-1/2}^{n-1}}{(\Delta \hat{x}_{j-1/2}^n)^2} - \frac{\Delta \hat{x}_{j+1/2}^{n-1}}{(\Delta \hat{x}_{j+1/2}^n)^2} \right) \right). \end{aligned} \tag{6.55}$$

We may now insert (6.55) into (6.44) to obtain our scheme for the mesh movement. It remains to be shown however if our Taylor series approximation affects the conservation of the energy integral.

All that remains is to approximate equations (6.47) and (6.48) in order to recover the desired variables from the mesh. We do this using a centred difference and a backward difference respectively.

6.2.6 Numerical Results

In this section we apply the scheme derived in the previous section to a test problem. The chosen test problem is a 'lump' of mass in the centre with zero velocity everywhere. The initial condition is given by

$$\rho(x,0) = \begin{cases} (1-x^2)^2 + 0.1 & |x| \leqslant 1 \\ 0.1 & \text{otherwise} \end{cases}, \tag{6.56}$$

and

$$v(x,0) = 0. \tag{6.57}$$

It is also augmented by the boundary conditions

$$v(-4,t) = 0 \quad \text{and} \quad v(4,t) = 0. \tag{6.58}$$

The scheme is run until time $t = 1.5$ with 41 computational nodes and a time step of $\Delta t = 0.001$. The solutions for density and velocity are plotted in Figure 6.1.

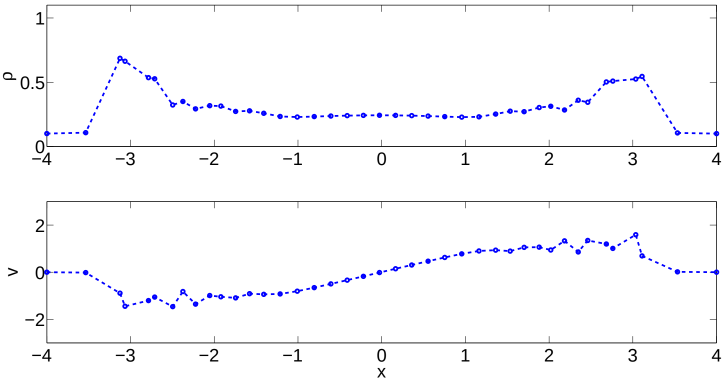


Figure 6.1: The conservation base moving mesh scheme derived in Section (6.2.5) applied to the isothermal equations with initial data (6.56)-(6.57) and boundary conditions (6.58).

It is clear from Figure 6.1 that the scheme is oscillatory in both solutions and the mesh. To further show this we plot the trajectories of the mesh in Figure 6.2.

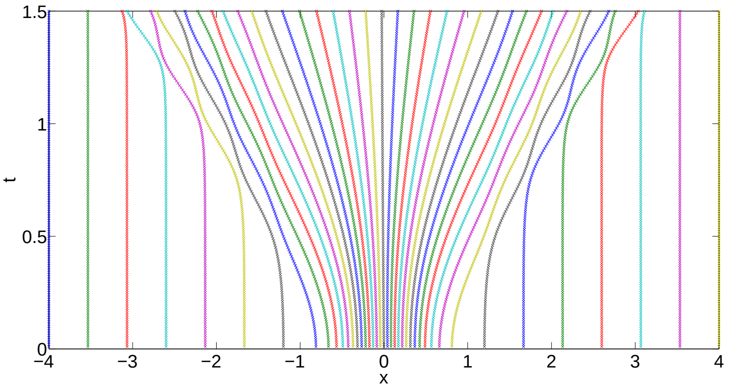


Figure 6.2: The mesh trajectories of the conservation base moving mesh scheme derive in Section (6.2.5) applied to the isothermal equations with initial data (6.56)-(6.57) and boundary conditions (6.58).

Figure 6.2 shows that our attempts to find a scheme in which the mesh does not tangle were successful however the mesh still oscillates near solution discontinuities which causes the solution oscillations. These oscillations appear to occur regardless of timestep leading to the conclusion that the Furihata scheme may not be sufficient for the mesh PDE. Further work is required to see if another Eulerian solver in reference space can lead to a non-oscillatory mesh.

Chapter 7

Summary and Further Work

In this chapter we summarise the work done in this thesis and suggest some further research which may be carried out. The novel contributions of the thesis are also noted.

7.1 Summary

Chapter 1 introduced the work of the thesis, giving an overview of the work that would be carried out and the original work done.

In Chapter 2 we discussed the background knowledge required for the work in the rest of the thesis. In addition we also briefly noted some of the recent developments in the surrounding areas of research. The chapter was broken into three sections which focused on hyperbolic conservation laws, relocation refinement (r-refinement) methods and the conservation based Lagrangian moving mesh methods. The hyperbolic conservation laws section introduced some example problems that we would use later while the final two sections helped introduce the methods we studied.

The main aim of Chapter 3 was to take the two main areas studied in the background chapter and use them to develop a general conservation based moving mesh method for solving hyperbolic conservation equations. The main discussion in this chapter focused around the choices made to derive the numerical method as well as some of the issues surrounding how to reformulate fixed frame Eulerian conservation problems as Lagrangian moving frame problems.

Chapter 4 was based on using a transformation from physical space to a reference space as a mathematical tool for analysing the numerical methods introduced in Chapter 3. This chapter started with a description of the difficulties that arise in attempting to analyse moving mesh numerical methods before introducing the transformation that would be used for our analysis. The final three sections of the chapter studied a method for determining the accuracy, stability and convergence of the conservation based moving mesh methods. At the end of the convergence section we did numerical experiments to verify our findings.

Based on a notion introduced while attempting to show convergence of the numerical methods, Chapter 5 introduced a new approach to deriving Lagrangian moving mesh methods by using well established fixed mesh schemes in reference space and performing the inverse transform on the method. In this chapter we showed that it is always possible to do this if the numerical scheme could be written in conservative form and that the Lagrangian schemes derived are only unique down to a choice of quadrature to approximate the local mass constant. Based on the work in Chapter 4, we conclude that the derived Lagrangian schemes have the same accuracy and stability conditions as the scheme from which they were generated and we end the chapter by comparing the Lagrangian schemes with the schemes from which they were generated.

Finally, in Chapter 6 we attempted to use the work from scalar conservation laws to generate a scheme for systems of hyperbolic conservation laws. We discussed several issues that arise when considering systems of equations before attempting to solve the isothermal Euler equations. At the end of the chapter we successfully generated a scheme which was not prone to mesh tangling, however this scheme is still oscillatory.

The original work in this thesis appears in Chapters 3-6. This consists of:

- In Chapter 3 we gave a more in-depth discussion of how Eulerian boundary conditions are applied to Lagrangian schemes than appears in the literature.
- The notion of the transform to reference space from Chapter 4 was taken from the MMPDE methods but was applied as an analytical tool for the first time to find accuracy, stability and convergence.
- Chapter 5 discussed a novel approach to generating new moving mesh methods from existing conservative Eulerian methods.
- The attempt to solve the isothermal Euler equations in Chapter 6 provided a moving mesh which does not tangle.

7.2 Further Work

There are many ways in which the work in this thesis could be continued. The most obvious of these are the extension to multi-dimensional problems and a more general approach to systems of equations, however both of these are non-trivial.

In the case of extension to higher dimensional problems, the method itself is very difficult to even implement. Reconstruction of the mesh in particular can be difficult and care needs to be taken since vorticity can add new ways of tangling which cannot be stopped in a trivial manner. While the Lagrangian formulation can still be derived by using Reynolds transport theorem, it can be difficult to find the mesh velocity without making assumptions on the flow.

Extension to systems of equations is also difficult since the mesh PDE derived will have the same order in time as the number of equations in the system. In our isothermal Euler equations example this leads to problems since the two equations led to a PDE with an $\hat{x}_{\tau\tau}$ term. This gets even worse if we consider the full Euler equations where a $\hat{x}_{\tau\tau\tau}$ term appears in the mesh PDE.

There is also a lot of scope for doing similar analysis with other monitor functions. It is not clear that the mass monitor is the best in all circumstances and analysis of general monitor functions would be beneficial.

There are many other small ways in which the work could be extended. In Chapter 3 we discussed that other ways of partitioning the domain were possible and it is not clear that the standard partition chosen is indeed the best option, hence further comparisons are required. In Chapter 5, we introduced a method for deriving new moving mesh schemes from existing conservative schemes, this could be studied further to see if there are benefits to applying more advanced fixed grid methods here such as flux/slope limiters. Finally, a large improvement of the mass conserving methods would be to combine it with h-refinement to add nodes in front of shocks.

Bibliography

[AO97] Mark Ainsworth and J Tinsley Oden, *A posteriori error estimation in finite element analysis*, Computer Methods in Applied Mechanics and Engineering **142** (1997), no. 1, 1–88.

[Aro] DG Aronson, *The porous medium equaion. in nonlinear diffusion problems*, Lecture notes in mathematics, no. 1224.

[Bai94] MJ Baines, *Moving finite elements*, Oxford University Press, Inc., 1994.

[Bai98] ———, *Grid adaptation via node movement*, Applied Numerical Mathematics **26** (1998), no. 1, 77–96.

[BBNP16] B Bonan, MJ Baines, NK Nichols, and D Partridge, *A moving-point approach to model shallow ice sheets: a study case with radially symmetrical ice sheets*, The Cryosphere **10** (2016), no. 1, 1–14.

[BHJ04] MJ Baines, ME Hubbard, and PK Jimack, *A moving mesh finite element algorithm for the adaptive solution of time-dependent partial differential equations with moving boundaries*, Journal of Computational and Applied Mathematics **54** (2004), pp. 450 – 469.

[BHJ11] ———, *Velocity-based moving mesh methods for nonlinear partial differential equations*, Communications in Computational Physics **10** (2011), no. 03, 509–576.

[BHR96] Chris Budd, Weizhang Huang, and Robert Russell, *Moving mesh methods for problems with blow-up*, SIAM Journal on Scientific Computing **17** (1996), pp. 305 – 327.

[BHR09] ———, *Adaptivity with moving grids*, Acta Numerica (2009), pp. 1 – 131.

[Bir14] Nicholas Bird, *A moving-mesh method for high order nonlinear diffusion*, Ph.D. thesis, University of Reading, 2014.

[BL14] MJ Baines and TE Lee, *A large time-step implicit moving mesh scheme for moving boundary problems*, Numerical Methods for Partial Differential Equations **30** (2014), no. 1, 321–338.

[CH01] Hector D Ceniceros and Thomas Y Hou, *An efficient dynamically adaptive mesh for potentially singular solutions*, Journal of Computational Physics **172** (2001), no. 2, 609–639.

[CHR02] Weiming Cao, Weizhang Huang, and Robert Russell, *A moving mesh method based on the geometric conservation law*, SIAM Journal on Scientific Computing **24** (2002), pp. 188 – 142.

[CHR03] ———, *Approaches for generating moving adaptive meshes: location versus velocity*, Applied Numerical Mathematics **47** (2003), no. 2, 121–138.

[CJ12] Richard Courant and Fritz John, *Introduction to calculus and analysis i*, Springer Science & Business Media, 2012.

[CN47] John Crank and Phyllis Nicolson, *A practical method for numerical evaluation of solutions of partial differential equations of the heat-conduction* *type*, Mathematical Proceedings of the Cambridge Philosophical Society, vol. 43, Cambridge Univ Press, 1947, pp. 50–67.

[Coc03] Bernardo Cockburn, *Continuous dependence and error estimation for viscosity methods*, Acta Numerica **12** (2003), pp. 127 – 180.

[Col13] Sarah Cole, *Truncation error estimates for mesh refinement in lagrangian hydrocodes*, Ph.D. thesis, University of Reading, 2013.

[CS07] Juan Cheng and Chi-Wang Shu, *A high order eno conservative lagrangian type scheme for the compressible euler equations*, Journal of Computational Physics **227** (2007), no. 2, 1567–1596.

[Daf10] Constantine Dafermos, *Hyperbolic conservation laws in continuum physics*, Springer, 2010, ISBN: 00727830, 9783642040474.

[dB73] Carl de Boor, *Good approximation by splines with variable knots*, Spline functions and approximation theory, Springer, 1973, pp. 57–72.

[DD87] Ernst Dorfi and L. Drury, *Simple adaptive grids for 1d initial value problems*, Journal of Computational Physics **69** (1987), pp. 175 – 195.

[Eva10] Lawrence Evans, *Partial differential equations*, AMS, 2010, ISBN: 0821849743, 9780821849743.

[Fla73] Harley Flanders, *Differentiation under the integral sign*, The American Mathematical Monthly **80** (1973), no. 6, 615–627.

[FM11] Daisuke Furihata and Takayasu Matsuo, *Discrete variational derivative method: A structure-preserving numerical method for partial differential equations*, Chapman & Hall/CRC Press, 2011, ISBN: 1420094467, 9781420094466.

[Fur01] Daisuke Furihata, *Finite-difference schemes for nonlinear wave equation that inherit energy conservation property*, Journal of Computational and Applied Mathematics **134** (2001), pp. 37 – 57.

[GL88] Jonathan B Goodman and Randall J LeVeque, *A geometric approach to high resolution tvd schemes*, SIAM journal on numerical analysis **25** (1988), no. 2, 268–284.

[GMP15] Jan Giesselmann, Charalambos Makridakis, and Tristan Pryer, *A posteriori analysis of discontinuous galerkin schemes for systems of hyperbolic conservation laws*, SIAM Journal on Numerical Analysis **53** (2015), no. 3, 1280–1303.

[God59] S.K. Godunov, *A difference method for numerical calculation of discontinuous solutions of the equations of hydrodynamics*, Mat. Sb. (N.S.) **47(89)** (1959), no. 3, 271–306.

[Har83] Ami Harten, *High resolution schemes for hyperbolic conservation laws*, Journal of Computational Physics **49** (1983), no. 3, 357 – 393.

[HEOC87] Ami Harten, Bjorn Engquist, Stanley Osher, and Sukumar R Chakravarthy, *Uniformly high order accurate essentially non-oscillatory schemes, iii*, Upwind and High-Resolution Schemes, Springer, 1987, pp. 218–290.

[HR97] Weizhang Huang and Robert Russell, *Analysis of moving mesh partial differential equations with spatial smoothing*, SIAM Journal on Numerical Analysis **34** (1997), pp. 1106 – 1126.

[HR00] ———, *Adaptive mesh movement the mmpde approach and its applications*, Journal of Computational and Applied Mathematics **128** (2000), pp. 383 – 398.

[HRR94] Weizhang Huang, Yuhe Ren, and Robert Russell, *Moving mesh partial differential equations (mmpdes) based on the equidistribution principle*, SIAM Journal on Numerical Analysis **31** (1994), pp. 709 – 730.

[KHDB03] E Kuhl, S Hulshoff, and R De Borst, *An arbitrary lagrangian eulerian finite-element approach for fluid–structure interaction phenomena*, International journal for numerical methods in engineering **57** (2003), no. 1, 117–142.

[LBL15] TE Lee, MJ Baines, and Steve Langdon, *A finite difference moving mesh method based on conservation for moving boundary problems*, Journal of Computational and Applied Mathematics **288** (2015), pp. 1 – 17.

[LBLT13] TE Lee, MJ Baines, Steve Langdon, and Marcus J Tindall, *A moving mesh approach for modelling avascular tumour growth*, Applied Numerical Mathematics **72** (2013), 99–114.

[Lee11] TE Lee, *Modelling time-dependent partial differential equations using a moving mesh approach based on conservation*, Ph.D. thesis, University of Reading, 2011.

[LeF02] Philippe LeFloch, *Hyperbolic systems of conservation laws*, Birkhuser Basel, 2002, ISBN: 3034881517, 9783034881517.

[LeV92] Randall J. LeVeque, *Numerical methods for conservation laws*, Birkhuser Basel, 1992, ISBN: 3764327235, 9783764327231.

[MM05] K.W. Morton and D.F. Mayers, *Numerical solution of partial differential equations*, 2 ed., Cambridge University Press, 2005.

[Ole63] O. Oleinik, *Discontinuous solutions of non-linear differential equations*, American Mathematical Society Translations - Series 2 **26** (1963), pp. 100 – 177.

[SDP07] PH Saksono, WG Dettmer, and D Peric, *An adaptive remeshing strategy for flows with moving boundaries and fluid-structure interaction*, International Journal for Numerical Methods in Engineering **71** (2007), no. 9, 1009–1050.

[Shu09] Chi-Wang Shu, *High order weighted essentially nonoscillatory schemes for convection dominated problems*, SIAM review **51** (2009), no. 1, 82–126.

[SMR01] John Stockie, John Mackenzie, and Robert Russell, *A moving mesh method for one-dimensional hyperbolic conservation laws*, SIAM Journal on Scientific Computing **22** (2001), pp. 1791 – 1813.

[Swe84] Peter K Sweby, *High resolution schemes using flux limiters for hyperbolic conservation laws*, SIAM journal on numerical analysis **21** (1984), no. 5, 995–1011.

[Tan05] Tao Tang, *Moving mesh methods for computational fluid dynamics*, Contemporary mathematics **383** (2005), 141–174.

[TL79] PD Thomas and CK Lombard, *Geometric conservation law and its application to flow computations on moving grids*, AIAA journal **17** (1979), no. 10, 1030–1037.

[Tor99] Eleuterio Toro, *Riemann solvers and numerical methods for fluid dynamics*, Springer, 1999, ISBN: 3540659668, 9783540659662.

[VDPP07] CJ Van Duijn, Lambertus A Peletier, and Iuliu Sorin Pop, *A new class of entropy solutions of the buckley-leverett equation*, SIAM Journal on Mathematical Analysis **39** (2007), no. 2, 507–536.

[vL79] Bram van Leer, *Towards the ultimate conservative difference scheme. v. a second-order sequel to godunov's method*, Journal of Computational Physics **32** (1979), no. 1, 101 – 136.

[Wel04] Ben Wells, *A moving mesh finite element method for the numerical solution of partial differential equations and systems*, Ph.D. thesis, University of Reading, 2004.

[Wes01] Pieter Wesseling, *Principles of computational fluid dynamics*, 1 ed., Springer Series in Computational Mathematics 29, Springer-Verlag Berlin Heidelberg, 2001.