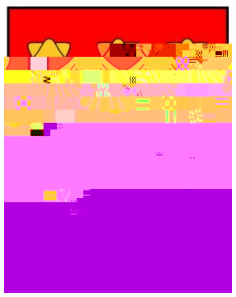


University of Reading

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 con rm 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 nished. 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 ancee, 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 nished.

University would have been a very di erent 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 nancial support of both the EPSRC and AWE.

Contents

1 Introduction 1

2 Background 5

3.2 Boundary Conditions 41
3.2.1 Fixed In ow Boundary 41
3.2.2 Boundary with Characteristic Velocity 42
3.2.3 Free Lagrangian Boundary Conditions 44
3.2.4 Limits on the Boundary Velocity 45
3.3 Partitioning the Domain 48
3.3.1 Standard Partition vs Overlapping Masses 49
3.3.2 Choice of Local Mass Constant 50
3.4 Discretising the Lagrangian Formulation 52
3.4.1 Mesh Movement 52
3.4.2 Quadrature Approximation 52
3.5 A General Mass Conservative Moving Mesh Method 54
3.6 Examples 59
3.6.1 Linear Advection Equation 59
3.6.2 Inviscid Burgers' Equation 63
3.6.3 Buckley-Leverett Equation 66

4 Analysis of Mass Conserving Moving Mesh Methods 69
4.1 Problems with Moving Meshes 70
4.2 Transformation to a Reference Space 74
4.3 Accuracy 79
4.3.1 Comparing the Standard Error and the True Error 80
4.3.2 Finding the Solution and Position Errors 82
4.4 Stability 84
4.4.1 Non-crossing Criterion 84
4.4.2 Stability Via Reference Space 87
4.5 Convergence 91
4.5.1 Vanishing Viscosity Solution 91

4.5.2 Regularisation in Reference Space 95
4.5.3 Regularised Numerical Scheme 96
4.5.4 Application to Inviscid Burgers' Equation 97
4.5.5 Rate of Convergence 98
4.5.6 Experimental Order of Convergence 102

5 Lagrangian Schemes Based on Existing Conservative Schemes 109
5.1 Existence of Schemes 110
5.2 Uniqueness of Schemes 112
5.3 An Example Scheme 113
5.4 Higher Order Schemes 116
5.5 Numerical Comparisons 116
5.5.1 Linear Advection 117
5.5.2 Inviscid Burgers' Equation 118

6 Systems of Equations 120
6.1 Problems that Arise with Systems of Equations 120
6.2 Isothermal Equations 121
6.2.1 The Lagrangian Formulation 122
6.2.2 A Lagrangian Numerical Scheme 123
6.2.3 Reference Space Transformation 125
6.2.4 Furihata's Method 128
6.2.5 An Alternative Lagrangian Scheme Based on Furihata's Method 132
6.2.6 Numerical Results 135

7 Summary and Further Work 138
7.1 Summary 138
7.2 Further Work 140

List of Figures

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). 12

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). 14

2.3 Sketch of a discontinuous solution where = 0 (red) and two viscous solutions = 0:05 (blue) and = 0:1 (black). 15

2.4 Characteristic plot to the time independent solution (2.27). This solution is entropy violating and therefore not a physically valid solution. 16

2.5 Characteristic plot to the rarefaction fan solution (2.28). This solu-

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 eld boundary conditions. 37

3.1 A moving domain in blue where the boundary velocity does not exceed the characteristic velocity, hence $f'(u) \dot{x}_t > 0$ and the left hand boundary is still an in ow boundary. 47

3.2 A moving domain in green where the boundary velocity exceeds the characteristic velocity, hence $f'(u) \dot{x}_t < 0$ and the left hand boundary is now an out ow boundary. 47

3.3 The standard partition in which subintervals of the domain do not overlap and consecutive intervals share a boundary. 49

3.4 The overlapping partition in which subinterval of the domain starts at the in ow boundary of the region. 50

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 $t = 1$. 62

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). 65

3.7 This graph shows the moving mesh appr . m 65

4.1 The True Error in the scheme broken down into solution error, U_j^n, and position error, X_j^n, components. 71
4.2 The Standard Finite Di erence Error (Blue) Compared with the True Error in the Moving Mesh Scheme (Red). 72
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. 105
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. 106
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. 107

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. 118
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. . 119

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). 136

6.2 The mesh trajectories of the conservation base moving mesh scheme

Chapter 1

Introduction

A class of Partial Di erential 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 nd 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 ine cient. 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-re nement methods, improve this by only increasing the resolution in regions where necessary but su er from the increased computational cost of calculating where the mesh needs to be re ned and the fa8 pl-21.66wshneeds aby

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-re nement methods. R-re nement methods are adaptive methods in which the computational mesh is moved in an attempt to automatically re ne the mesh in an advantageous way without having to introduce more nodes.

One particular class of r-re nement 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 nding the velocity of the mesh nodes. As such they appear to be a natural t 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 eld. The chapter is split into three sections, the rst focuses on the conservation laws themselves while the second and third both provide information regarding r-re nement 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 Di erential Equation (MMPDE) methods but is applied as an analytical tool for the rst time to nd accuracy, stability and convergence.

Chapter 5 discusses a novel approach to generating new moving mesh methods from existing xed 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 di erential equations (PDEs) with a particularly simple structure.

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 $\frac{@u}{@x}$ and $u_{tt} = \frac{@^2u}{@t^2}$.

In this work the system of equations (2.1) is assumed to be hyperbolic. This implies that the Jacobian matrix of the ux 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

that the case where n $f'(u)|_{@} = 0$ 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 ux 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 nd approximate solutions.

2.1.1 Derivation of Conservation Laws

While conservation laws are often written in the di erential form g 0 G(

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 ux function, $f(u(x;t))$, is a function of the components of $u(x;t)$.

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

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

and using the fundamental theorem of calculus and rearranging gives

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

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 di erential form of the conservation law (2.1).

Remark 2.1.2. The di erential form (2.1) is not the only di erential form of the conservation laws, it is a special form referred to as the conservation form. Another key di erential 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 ux 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 Di culties

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

Discontinuous Solutions

Discontinuous solutions are a major mathematical di culty which arises when considering hyperbolic conservation laws. Since the problems are often stated in the di erential 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

Using the notion of characteristics we can consider how the structure of the ux function, $f(u(x;t))$, a ects 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 ux 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

$$u^0(x) = \quad x; \quad x \ 2 \ [\ \ 1;1]; \tag{2.17}$$

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, $t > 1$ and the di erential equation

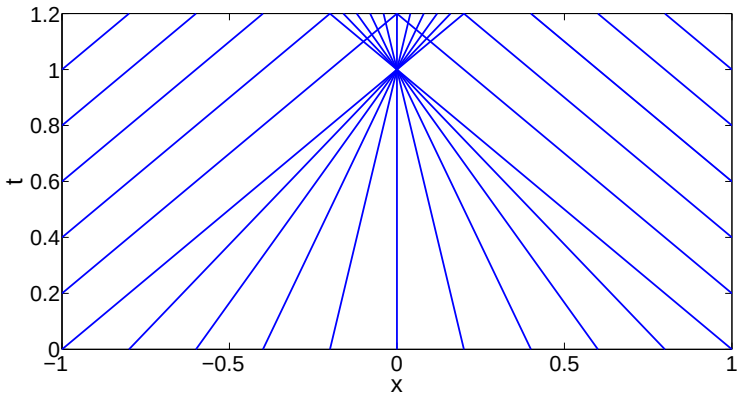


Figure 2.1: Characteristics crossing after time $t = 1$ causing the solution to become multivalued. This is not a physically valid solution to the problem (2.16)-(2.18).

(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 $t > 1$ 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, dis-

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 $(x;t)\ 2\ C_0^1(\mathrm{R}\quad \mathrm{R})$ and then integrate over space and time. This yields

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

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

$$\int_0^{\infty}\int_{-\infty}^{\infty} (\ {}_t u + \quad {}_x f(u))\,dxdt + \int_{-\infty}^{\infty} (x;0)u(x;0)dx = 0; \tag{2.22}$$

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 $t > 1$ 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.

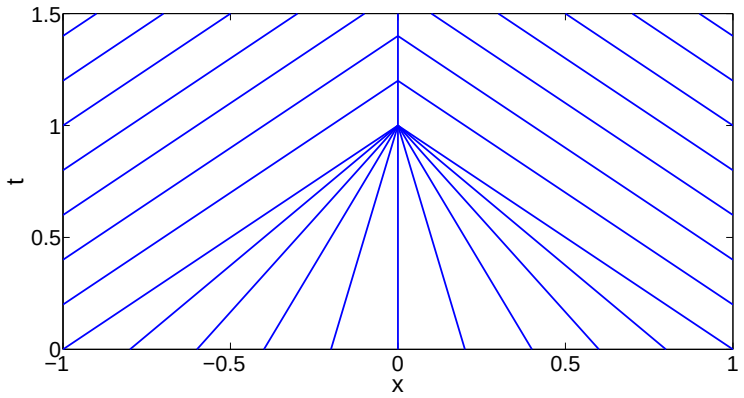


Figure 2.2: Characteristic plot with a discontinuity forming at time $t = 1$. This is the physically relevant characteristic solution to the problem (2.16)-(2.18).

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

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

as $\varepsilon \to 0$. The argument follows that since (2.23) has a classical solution for all $\varepsilon > 0$, 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 $\varepsilon \to 0$. 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 $= 0$ (red) and two viscous solutions $= 0:05$ (blue) and $= 0:1$ (black).

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

$$\frac{f(u) \quad f(u_l)}{u \quad u_l} > s > \frac{f(u) \quad f(u_r)}{u \quad u_r} \tag{2.24}$$

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) con rms 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

which simplifies to

$$s = f(u_l$$

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.

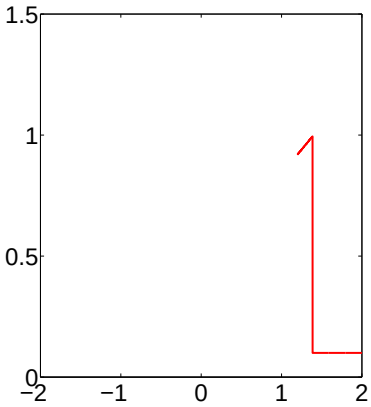
Approximating Shocks

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


1.5
1
0.5
0
−2
−1
0
1
2

if it has the form

$$u_j^{n+1} = u_j^n - \frac{t}{x} F(u_{j-p}^n; u_{j-p+1}^n; \quad ; u_{j+q}^n) - F(u_{j-p-1}^n; u_{j-p}^n; \quad ; u_{j+q-1}^n) \quad (2.29)$$

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.

De nition 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 $u_j^n; n\ 2$ N is non-decreasing (non-increasing).

While de nition 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).

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

$$TV(u^n) = \sum_j \quad u_j^n \quad u_{j-1}^n \ : \tag{2.30}$$

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.

De nition 2.1.7. A scheme is de ned to be Total Variation Diminishing (TVD) if

$$TV(u^{n+1}) \ 6 \ TV(u^n) \quad 8n; \tag{2.31}$$

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*

$$u^n$$

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

solution is locally. While ux 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 ux/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.

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

$$f$$

2.2 R-Re nement 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 re nement (r-re nement) methods and give a brief history of some of the di erent ways in which they have been implemented.

Note that in this section we consider re nement 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 uid dynamics, conservation laws, free boundary problems, combustion, meteorology and mathematical biology. The evolution of these features often happens over short time scales in very ne regions of space and as such a computational mesh must be at least as ne 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 ne grain features a small mesh spacing, x, is required but this is computationally ine cient away from such structures. Instead adaptive methods are applied which attempt to re ne/coarsen the mesh as

required. These adaptive methods generally fall into three categories, h-re nement, p-re nement and r-re nement.

The h-re nement methods are the most commonly applied type of adaptive mesh and is named after the widespread use of the notation $h = x$. Such methods usually start with a uniform mesh and locally coarsen or re ne 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-re nement methods are only applicable to nite element methods (FEM) and stands for polynomial re nement. In p-re nement methods a nite 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-re nement and p-re nement methods to generate hp-re nement methods which are explored in [AO97].

In r-re nement (relocation re nement) methods the computational mesh is allowed to move in the hope that the mesh re nement/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-re nement or p-re nement methods but have been successfully applied to a variety of di erent applications including computational uid mechanics [Tan05], convective heat transfer [CH01] and mathematical biology [LBLT13].

The main downside of r-re nement 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 di cult to avoid.

The methods that will be studied in this thesis are r-re nement 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-re nement methods.

The rst of these is the notion of a monitor function. As noted in the previous section, r-re nement 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 $0 \leq \xi \leq 1$,

$$\int_{a(t)}^{x(\xi,t)} m(u; u_x; u_{xx}; :::)dx = \int_{a(t)}^{b(t)} m(u; u_x; u_{xx}; :::)dx \tag{2.39}$$

where $a(t)$ is the left hand boundary of the domain, $b(t$

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

$$M(u; u_x; u_{xx}; ::: : \xi_1(t); \xi_2(t)) = \int_{\xi_1(t)}^{\xi_2(t)} m(u; u_x; u_{xx}; :::)dx; \tag{2.40}$$

where $\xi_1(t)$ and $\xi_2(t)$ are moving coordinates. The moving coordinates are defined to be such that the monitor integral remains constant in time hence,

$$\frac{d}{dt} \int_{\xi_1(t)}^{\xi_2(t)} m(u; u_x; u_{xx}; :::)dx = 0: \tag{2.41}$$

Since $\xi_1(t)$ and ξ_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,

$$\int_{\xi_1(t)}^{\xi_2(t)} m_t dx + [m\dot{\xi}(t)_t]_{\xi_1(t)^\wedge}^{\xi_2(t)}$$

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 $\dot{x}(t)_t$.

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 $\dot{x}(t)_t$ 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

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

and using the PDE (2.43) yields

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

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

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

which has a solution if

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

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

$$A_{j-1=2} = \frac{1}{2}(x_j^n - x_{j-1}^n)(u_j^n + u_{j-1}^n); \tag{2.50}$$

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

$$x_j^{n+1} = x_j^n - t\frac{u_j^n - u_{j-1}^n}{x_j^n - x_{j-1}^n}: \tag{2.51}$$

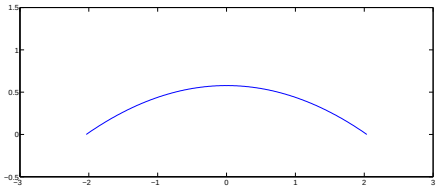
In order to test the scheme we consider the initial condition,

$$u(x;0) = \begin{cases} (1 - x^2)^{\frac{1}{2}} & jxj < 1 \\ 0 & \text{otherwise} \end{cases}; \tag{2.52}$$

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)

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

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

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

$$u(a;t) = u^a(t); \quad t \in \mathrm{R}^+: \tag{3.3}$$

We assume that $a(t)$ is the in ow boundary, as de ned 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 in ow 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

$$u(x;t) > 0; \quad 8x;t; \tag{3.5}$$

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,

$$\frac{d}{dt}\int_{x_1(t)}^{x_2(t)} u(x;t)dx = \int_{x_1(t)}^{x_2(t)} u_t dx + [u(x;t)x_t]_{x_1(t)}^{x_2(t)}; \tag{3.8}$$

and appealing to the conservation law (3.1) further yields

$$\frac{d}{dt}\int_{x_1(t)}^{x_2(t)} u(x;t)dx = \int_{x_1(t)}^{x_2(t)} f(u)_x dx + [u(x;t)x_t]_{x_1(t)}^{x_2(t)}: \tag{3.9}$$

Application of the fundamental theorem of calculus gives

$$\frac{d}{dt}\int_{x_1(t)}^{x_2(t)} u(x;t)dx = [u(x;t)x_t \quad f(u)]_{x_1(t)}^{x_2(t)}: \tag{3.10}$$

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

$$\text{Net Flux} = f(u) \quad ux_t: \tag{3.11}$$

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

$$[u(x;t)x_t \quad f(u)]_{x_1(t)}^{x_2(t)} = 0: \tag{3.12}$$

Let $x_1 = a(t)$ be the position of the in ow boundary and $x_2 = x; x\ 2\ (a(t); b(t))$ be an arbitrary moving coordinate, with velocity x_t. It follows from equation (3.12) that

$$u^a(t)\ x_t j_a \quad f(u^a) = u(x;t)x_t \quad f(u): \tag{3.13}$$

time.

Since the in ow boundary $a(t)$ is xed it is clear that $\dot{x}_t j_a = 0$ and hence the velocity (3.16) simpli es to become

$$\dot{x}_t = \frac{f(u) \quad f(u^a)}{u(x; t)}: \tag{3.17}$$

Note further that the standard Eulerian problem formulation xes both the in ow boundary, $a(t)$, and the out ow boundary, $b(t)$. This de nes a xed '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 x the 'volume'. This implies that since the in ow boundary, $a(t)$, is the only given boundary condition and xed we must allow $b(t)$ to remain free to move as prescribed by the mass conservation. This is prescribed by

$$\dot{x}_t j_b = \frac{f(u^b) \quad f(u^a)}{u^b(t)}: \tag{3.18}$$

Therefore, in the Lagrangian formulation the net ux across all such coordinates must be equal at any given time, t. This means that the net ux of the in ow boundary condition determines the net ux for every other x which moves with a consistent mass conserving velocity. In the case of the xed in ow boundary, equation (3.12) tells us that the net ux 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 in ow boundary is prescribed to move with the characteristic velocity. This means that the in ow bound-

ary 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

$$\hat{x}$$

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),

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

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

$$\dot{x}_t = \frac{f(u)}{u(\dot{x};t)}; \tag{3.24}$$

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

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

$$\dot{x}_t j_b = \frac{f(u^b)}{u^b(t)}. \tag{3.25}$$

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 in ow boundaries. To determine if a boundary is an in ow or out ow, consider the expression

$$\mathrm{n} \quad f'(u) j_{@} \tag{3.26}$$

where @ is the point on the boundary you are considering and n is the normal unit vector which leaves the domain. Note that in our 1D case 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 in ow or out ow by calculating whether it is positive or negative. If (3.26) is positive then the

boundary is an in ow and if (3.26) is negative then it is an out ow.

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 in ow 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 nd a function to replace $f'(u)$ which changes sign for the two types of boundary?

We propose that a suitable function is

$$f'(u) \quad \dot{x}_t \tag{3.27}$$

where $\dot{x}_t$ is the calculated boundary velocity. This leads to the Lagrangian in-ow/out ow expression being given by

$$\mathrm{n} \quad (f'(u) \quad \dot{x}_t)|_{x=a(t)} : \tag{3.28}$$

This function follows naturally when considering the Eulerian xed boundary case since $f'(u)$ informs us of the velocity of the characteristic trajectories and since the boundary is not moving this is su cient to determine whether the characteristic lines are entering or leaving the domain. In the Lagrangian moving boundary case however it is not su cient 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 in ow and an out ow boundary respectively.

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

Figure 3.2: A moving domain in green where the boundary velocity exceeds the characteristic velocity, hence $f'(u) - \dot{x}_t < 0$ and the left hand boundary is now an out ow 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 in ow nor an out ow boundary. This is the characteristic velocity de ned earlier in this section.

Remark 3.2.2.

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

$$\int_{x_1(t)}^{x_j(t)} u\,dx = A_{(j+1)=2}; \qquad 8j\ 6\ J; \quad j\ 2\ \mathrm{N}; \tag{3.31}$$

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.

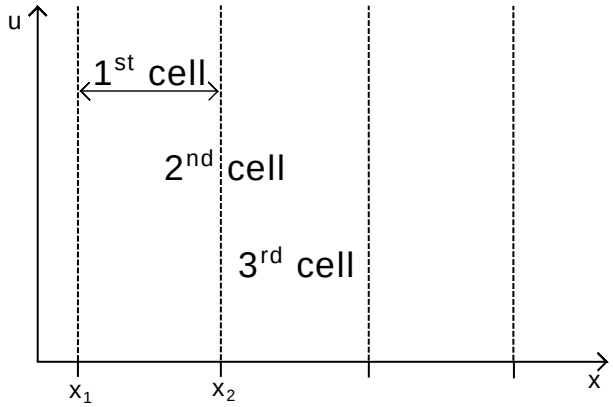


Figure 3.4: The overlapping partition in which subinterval of the domain starts at the in ow boundary of the region.

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,

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

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

where $k_i = \hat{x}(\hat{x}_j^n + t \sum_{l=1}^{s} a_{il} k_l; t^n + c_i t)_t$ and a_{il}, b_i and c_i are the coe cients

form of quadrature approximation is given by

$$\int_{x_p}^{x_q} u(x)dx$$

to work with. Since we are only considering intervals with solutions known at the nodes we simplify the general quadrature (3.36) to

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

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,

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

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

$$u(a;t) = \quad ; \quad t \in \mathrm{R}^+; \tag{3.40}$$

where > 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,

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

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

$$\dot{x}_t = \frac{f(u)_x}{u}; \tag{3.42}$$

where

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

$$\hat{a}(t) \leq f'(\ )t + a; \quad \forall t: \tag{3.43}$$

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 satis es the condition (3.43) then this boundary will always lie within the region where $u(x;t) =$ 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) satis es the condition (3.43), i.e. $f(\ ) 6 \quad f'(\ )$. In this case we can simplify the Lagrangian formulation since the net ux 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 simpli ed velocity which de nes 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

$$u(x;0) = u^0(x); \quad x \ 2 \ (\hat{a}(0);\hat{b}(0)); \tag{3.46}$$

and, the boundary conditions

$$\hat{a}_t = \frac{f(\ )}{\quad} \quad \text{and} \quad u(\hat{a}(t);t) = \ : \tag{3.47}$$

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

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

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 $k_i = \hat{x}_t(\hat{x}_j^n + t \sum_{l=1}^{s} a_{il} k_l \quad a_i$

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

$$\frac{d}{dt}\int_{x_1(t)}^{x_2(t)} u\,dx = 0. \tag{3.51}$$

Applying Leibniz integral rule to equation (3.51) gives

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

and appealing to the conservation law (3.50) yields

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

The fundamental theorem of calculus can be used to show that

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

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

$$\dot{\hat{x}}_t = a: \tag{3.55}$$

Having chosen mass conservation and found the associated conservative veloc-

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

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

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 veri able 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 de ning the constant

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

the initial condition

$$u^0(x) = \begin{cases} (x^2 \quad 1)^2 + 0{:}5 & 1 \ 6 \ x \ 6 \ 1 \\ 0{:}5 & \text{otherwise} \end{cases}; \tag{3.59}$$

the in ow boundary trajectory

$$\hat{x}_0(t) = t \quad 2; \tag{3.60}$$

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 $f'(u) = {}^{f(u)}$

It is clear from Figure 3.5 that the notes made in remark 3.6.1 hold since the solution does not di use 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

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

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

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

$$u^0(x) = \begin{cases} 1:1 + x & 1 < x \leq 0 \\ 1:1 - x & 0 < x < 1 \\ 0:1 & \text{otherwise} \end{cases} ; \tag{3.66}$$

the in ow boundary condition

$$u(\hat{x}_0(t); t) = 0:1; \tag{3.67}$$

the in ow 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 char-

time $t = 1$. The pre-shock solution for $t < 1$ is then given as

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

and the post-shock solution for $t > 1$ is

$$u(x;t) = \begin{cases} \dfrac{x + 1.1}{t + 1} & 0.1t \quad 1 < x \leq 0.1t + \sqrt{2t + 2} \quad 1 \\ 0.1 & \text{otherwise} \end{cases}: \tag{3.70}$$

The numerical method (3.64)-(3.65) is run with 41 computational nodes over the initial domain $[\ 2; 2]$ with a timestep of $t = 0.05$. 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.

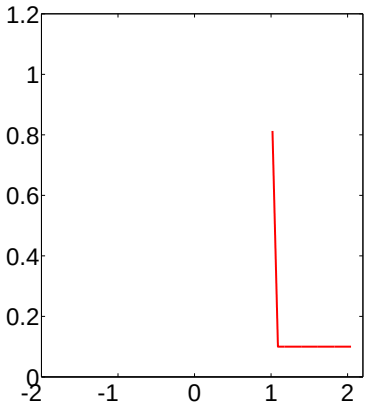


Figure 3.6: These graphs show the comparison between the numerical solution (Blue) to Inviscid Burgers Equation obtained from the scheme (3.64)-(3.65) and the exact solution (Red). Comparisons are taken at the pre-shock time regime $t = 0.9$ (A) and the post shock time regime $t = 1.5$ (B).

Figure 3.6A shows that unlike the linear example the Inviscid Burgers' scheme is prone to numerical di usion. 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 nal 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

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

and $M > 0$ is a given constant.

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

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

where we note that the $u \leq 0$ 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 $t = 0{:}0001$ over the region [2;

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 di usion problems to yield e ective 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 xed mesh methods before introducing a transformation to a xed 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.

exact, and $u(\hat{x}(\ _j; t^n); t^n)$ 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.

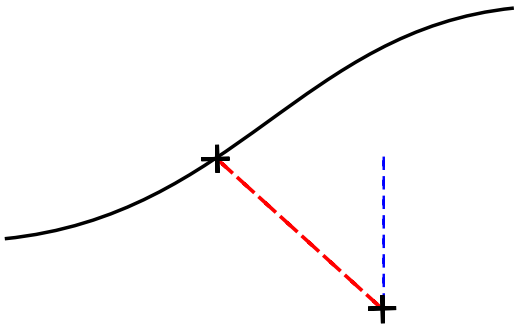
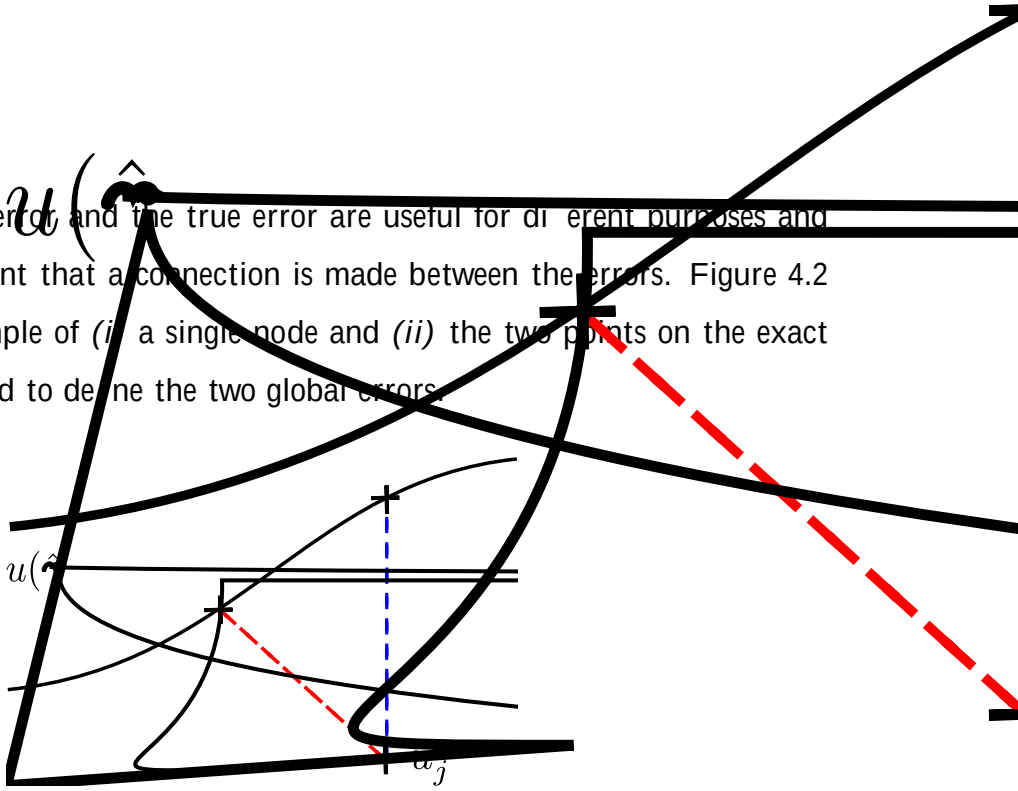


Figure 4.1: The True Error in the scheme broken down into solution error, U^n

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.



(4.3) gives

$$\sqrt{U_j^{n\ 2} + X_j^{n\ 2}} \to 0; \tag{4.7}$$

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,

$$|u(\hat{x}_j^n; t^n) - u_j^n| \le |u(\hat{x}_j^n; t^n) - u(\hat{x}(\xi_j; t^n); t^n)| + |u(\hat{x}(\xi_j; t^n); t^n) - u_j^n|; \tag{4.9}$$

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

$$|u(\hat{x}_j^n; t^n) - u_j^n| \le |u(\hat{x}_j^n; t^n) - u(\hat{x}(\xi_j; t^n); t^n)| + U_j^n: \tag{4.10}$$

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

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

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

$$|u(\hat{x}_j^n; t^n) - u(\hat{x}(\xi_j; t^n); t^n)| = \frac{\partial u}{\partial x}(\eta) \, |\hat{x}(\xi_j; t^n) - \hat{x}_j^n| \to 0; \tag{4.12}$$

where η is a position between $\hat{x}(\xi_j; t^n)$ 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.

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

$$\rho_t + \frac{\rho f(\rho)_x - f(\rho)\rho_x}{\rho^2 x_t} = 0$$

$$\Rightarrow \rho_t + \frac{\rho}{x_t}\left(\frac{f(\rho)}{\rho}\right)_x = 0.$$

We note that (4.20) is not a classical conservation law due to the factor $\frac{u^2}{K}$. 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 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 $w = \frac{1}{u}$.

Using this new variable the transformed PDE (4.20) becomes

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

All that remains in order for the transformed PDE to be a classical conservation law is to show that K is a constant in space. In order to show this we begin by noting that we have made no assumptions on $$ 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 $0 \leq \leq 1$

$$\int_{x_0}^{x(\,;t)} u\,dx = \int_{x_0}^{x_F} u\,dx. \tag{4.22}$$

Note here that the LHS of the definition (4.22) defines a proportion of the integral over the whole region $(x_0; x_F)$ in physical space which has a constant value due to our requirement of mass conservation.

If we let the reference space coordinate be as de ned by the equidistribution principle and transform equation (4.22) to the xed reference space then

$$\int_0^{\hat{x}}$$

Using the local conservation principle (3.64) to eliminate the Δx terms gives

$$\frac{c}{u_j^{n+1}} \quad \frac{c}{u_j^n} = \quad t \quad \frac{f_j^n}{u_j^n} \quad \frac{f_{j-1}^n}{u_{j-1}^n} \quad ; \tag{4.28}$$

where c is the local mass constant.

at the start of this chapter.

4.3.1 Comparing the Standard Error and the True Error

At the beginning of this chapter we brie y 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 insu cient 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

$$S_j^n = u(\hat{x}(\ _j; t^n); t^n) \quad u_j^n \quad \text{and} \quad X_j^n = \hat{x}(\ _j; t^n) \quad \hat{x}_j^n: \tag{4.30}$$

The true error is given by

$$T_j^n = \sqrt{\quad}$$

and

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

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

$$T_j^n = \sqrt{(O(\ x^k) + O(\ t^m))^2 + (O(\ x^l) + O(\ t^n))^2}$$

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

$$U_j^n = \frac{1}{w(\ _j;\ ^n)w_j^n}(w(\ _j;\ ^n) \quad w_j^n): \tag{4.39}$$

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

$$U_j^n = O(\quad ^p) + O(\quad ^q); \tag{4.40}$$

which can be changed to physical space parameters to give

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

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 x. This is due to the fact that the scheme replaces the notion of xed 'volumes' with xed masses and hence the quantity x is not useful since we cannot say what size it has in general. This is still consistent with regular de nitions of order since both a reduction in spatial step x 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 nd the position error. Recall from Section 4.2 that

$$\tilde{x} = K w; \tag{4.42}$$

where K is a constant as de ned in equation (4.19).

Since equation (4.42) shows that $\tilde{x}$ 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 $\tilde{x}$ is accurate to the p^{th} order in space and the q^{th} order in time.

We now need to nd the order of the error in $\tilde{x}$ from the error in $\tilde{x}$. It is an established result that if an approximation is n^{th} order then the error in the derivative is n 1^{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 e ects cancel out and thus

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

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

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

and

$$S_j^n \ 6\ O(\ \Delta x^p) + O(\ \Delta t^q): \tag{4.45}$$

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

$$x_j^{n+1} > x_{j-1}^{n+1}. \tag{4.46}$$

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

$$x_j^n + \quad tv_j^{n+1=2} > x_{j-1}^n + \quad tv_{j-1}^{n+1=2} \tag{4.47}$$

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 nd the restriction on t to ensure that (4.46) holds. This restriction is given by

$$t < \frac{x_j^n \quad x_{j-1}^n}{v_{j-1}^{n+1=2} \quad v_j^{n+1=2}}; \tag{4.48}$$

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

Note that the inequality (4.48) only accounts for the possibility that the j 1^{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

$$t_{j-1=2}^{n+1=2} \; 6 \; \frac{\hat{x}_j^n \quad \hat{x}_{j-1}^n}{v_{j-1}^{n+1=2} \quad v_j^{n+1=2}} \qquad \text{if} \quad v_{j-1}^{n+1=2} > v_j^{n+1=2}; \tag{4.49}$$

and no requirement otherwise.

De nition (4.49) allows $t^{n+1=2}$ to be calculated for the entire mesh. Explicitly this is

$$t^{n+1=2} = \min_j \quad t_{j-1=2}^{n+1=2}: \tag{4.50}$$

The timestep de nition (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 de nition 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

$$t_{j-1=2}^{n+1=2} < \frac{\hat{x}_j^n \quad \hat{x}_{j-1}^n}{v_{j-1}^{n+1=2}}: \tag{4.51}$$

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

approximation to the local mass conservation (3.44),

$$u_j^n \left(x_j^n - x_{j-1}^n \right) = A_{j-1=2}; \tag{4.52}$$

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

The choice of adaptive timestep (4.50) ensures that $x_j^n - x_{j-1}^n > 0$, however it places no limit on how small this can become. It follows directly from the quadrature choice (4.52) that since $x_j^n - x_{j-1}^n$ 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*

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*

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

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* $t \leq T$, *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 $t \leq T$ and that in the moving mesh numerical scheme (3.48)-(3.49) there is at least one J such that $\hat{x}_J^{n+1} > \hat{x}_{J+1}^{n+1}$ (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 $u_{\hat{\jmath}}^{n+1} \leq 0$ for some $\hat{\jmath}$ since $\hat{x}_{J+1}^{n+1} - \hat{x}_J^{n+1} \leq 0$. Then the corresponding $w_{\hat{\jmath}}^{n+1}$ 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* $t \leq T$,

then the original moving mesh scheme is also TVD for $t \leq T$.

Proof. We have already shown in the previous Lemma that for this choice of t the

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

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

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

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

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

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

De nition 4.5.1 (Entropy and Entropy Flux). Two smooth functions (u) and $q(u)$ form an entropy/entropy ux pair of the conservation law (3.1) provided that (u) is convex and

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

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

$$q(u) = \int^{Z} '(ʁ)f'(ʁ)dʁ: \tag{4.62}$$

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

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

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

$$(u)_t + q(u)_x \leq 0; \tag{4.64}$$

which leads to the de nition 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,

$$\int_0^\infty \int_a^b (\phi_t u + \phi_x f(u))\,dx\,dt + \int_a^b \phi_0 u_0\,dx = 0; \tag{4.65}$$

and the entropy inequality,

$$\int_0^\infty \int_a^b (\psi_t \eta(u) + \psi_x q(u))\,dx\,dt + \int_a^b \psi_0 \eta(u_0)\,dx \geq 0; \tag{4.66}$$

where ϕ and ψ are periodic Lipschitz continuous test functions and $\psi \geq 0$. 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

$$\int_0^\infty \int_a^b (\psi_t \eta(u^\varepsilon) + \psi_x q(u^\varepsilon))\,dx\,dt + \int_a^b \psi_0 \eta(u_0)\,dx = 0. \tag{4.67}$$

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*

$$u^\varepsilon \to u \quad \text{almost everywhere as} \quad \varepsilon \to 0; \tag{4.68}$$

then u

Multiplying the regularised PDE (4.58) by the test function and integrating over $(a; b)$ $[0; 1)$, it follows that by integration by parts

$$\int_0^{\infty}\int_a^b (\partial_t u + \partial_x f(u) + u_{xx})\,dx\,dt + \int_a^b u_0\,dx = 0. \tag{4.69}$$

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

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

$$w + \tilde{f}(w) = w : \tag{4.77}$$

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

$$w + \tilde{f}(w) = 0: \tag{4.78}$$

4.5.3 Regularised Numerical Scheme

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

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

$$C_{j-1=2} = \frac{}{c} \quad \frac{1}{2w_j^n w_{j-1}^n} + \frac{}{c} \qquad \text{and} \quad D_{j+1=2} = \frac{}{c^2} \tag{4.84}$$

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

$$1 \quad C_{j-1=2} \quad D_{j-1=2} > 0: \tag{4.85}$$

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

$$t \; 6 \; \frac{2c^2}{4 \; + cu_j^n u_{j-1}^n}: \tag{4.86}$$

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.

De nition 4.5.7. The $L^\infty([0;T];L^p(\ ))$ Bochner norm is de ned as

$$jjujj_{L^1\ ([0;T];L^p(\ ))} = \text{ess sup}_{t\in[0;T]} jju(t)jj_{L^p(\ )} \tag{4.87}$$

where ess sup is the essential supremum which is the supremum over all but nitely many points. Furthermore $p > 1$ and is the spatial domain.

To simplify notation we denote the Bochner norm by

$$\|u\|_{L^1(L^p)} \tag{4.88}$$

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

$$\|u - u_h\|_{L^1(L^1)} \leq E(\Delta) \tag{4.89}$$

where u_h is the regularised moving mesh numerical solution, u is the entropy solution to the conservation law (3.1), and $E(\Delta)$ 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 be the numerical approximation (4.82). Further assume that w is bounded and sufficiently smooth so that $w < C_1$, $w_x < C_2$, $w_{xx} < C_3$ and $w_t < C_4$. Then the error between u and u_h is given by*

$$\|w - w_h\|_{L^1(L^1)} \leq C_5 \|u_0\|_{TV} \sqrt{\Delta t} + \frac{C_4}{2} + \frac{1}{4K}\left(2\left(C_1^2\|u_0^3\|_{L^1} + C_2\|u_0^2\|_{L^1}\right)^2 C_3\right) \tag{4.90}$$

a smooth solution of the regularised PDE then

$$\|u - u_\varepsilon\|_{L^1([0,T);L^1(a(t),b(t)))} \leq C_5 \|u_0\|_{TV} \sqrt{t}. \tag{4.91}$$

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 be the regularised transformed numerical solution given by (4.79), u be the entropy solution of the conservation law (3.1), and u_h be the moving mesh numerical*

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

$$jjw \quad w_h jj_{L^1(L^1(0;1))} > \frac{1}{jju_0 jj_{L^1} K} \text{ess sup}_{t\in[0;T]} \int_{a(t)}^{b(t)} ju \quad u_h jd\mathcal{X}: \tag{4.96}$$

Rearranging and using the de nition 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

$$jjw \quad w_h jj_{L^1(L^1)} 6 \; jjw \quad w \, jj_{L^1(L^1)} + jjw \quad w_h jj_{L^1(L^1)}; \tag{4.97}$$

where w is the solution to the regularised transformed PDE.

Since w is an entropy solution to a conservation law and w is the vanishing viscosity regularisation we may apply Theorem 4.5.9 to obtain a bound on jjw $w \, jj_{L^1(L^1)}$. Hence

$$jjw \quad w_h jj_{L^1(L^1)} 6 \; C_5 jw_0 j_{TV} \sqrt{t} + jjw \quad w_h jj_{L^1(L^1)}: \tag{4.98}$$

The nal 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

$$jjw \quad w_h jj_{L^1(L^1)} 6 \; C_5 jw_0 j_{TV} \sqrt{t} + \frac{}{2} w \quad + \frac{}{4K} \quad \frac{1}{w} \qquad \frac{}{12K^2} w \quad : \tag{4.99}$$

Directly di erentiating $\frac{1}{w}$ gives

$$\frac{1}{w} \quad = \frac{2(w)^2}{(w)^3} \quad \frac{w}{(w)^2} \tag{4.100}$$

which can be inserted into (4.99) to give

$$jjw \quad w_h jj_{L^1\ (L^1)} 6\ C_5 jw_0 j_{TV} \sqrt{t} + \frac{}{2} w \quad + \frac{}{4K} \quad \frac{2(w\ )^2}{(w\ )^3} \quad \frac{w}{(w\ )^2} \qquad \frac{}{12K^2} w \quad : \tag{4.101}$$

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

$$jjw \quad w_h jj_{L^1\ (L^1)} 6\ C_5 jw_0 j_{TV} \sqrt{t} + \frac{C_4}{2} + \frac{}{4K} \quad \frac{2C_1^2}{(w\ )^3} \quad \frac{C_2}{(w\ )^2} \qquad \frac{}{12K}$$

We run the schemes multiple times doubling the number of intervals N each time and calculating the error, $e_N = |u(x_j^n; t^n) - u_j^n|$, 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

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

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

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

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.

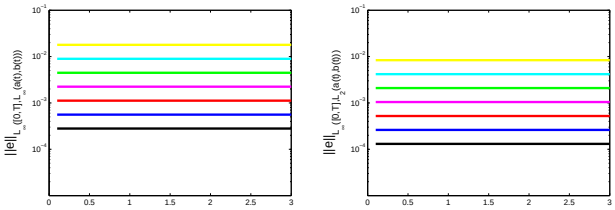


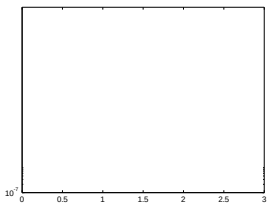
Figure 4.3: Global Errors and associated EOC for the numerical scheme (4.106) applied the linear advection equation with initial data (4.107). The $L^\infty(L^\infty)$ error is on the left, the $L^\infty(L^2)$ error is in the middle and the $L^\infty(L^1)$ is on the right.

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

$$u(x;t) = \begin{cases} 0.15 & x \leq 3 \\ 0.05 & x > 3 \end{cases}; \tag{4.108}$$

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


10^{-7}
0
0.5
1
1.5
2
2.5
3

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 $t = 1$.

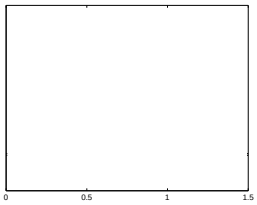


Figure 4.5: Global Errors and associated EOC for the numerical scheme (4.110) applied the linear advection equation with initial data (4.111). The $L^\infty(L^\infty)$ error is on the left, the $L^\infty(L^2)$ error is in the middle and the $L^\infty(L^1)$ is on the right.

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 conservationuvnot

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

$$w_t + \tilde{f}(w)_x = 0, \tag{5.2}$$

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

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

Theorem 5.1.2. *If a nite di erence 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

$$w_j^{n+1} = w_j^n$$

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,

$$\hat{x}_j^{n+1} \quad \hat{x}_{j-1}^{n+1} \quad \hat{x}_j^n + \hat{x}_{j-1}^n = \quad G(w_{j-p}^n; w_{j-p+1}^n; \quad ; w_{j+q}^n)$$

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

$$Aw_j^n = \tilde{x}_{j+1}^n \quad \tilde{x}_j^n : \tag{5.12}$$

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

$$\tilde{x}_j^{n+1} = \tilde{x}_j^n + \quad tG(w_{j-1-p}^n; w_{j-p}^n; \quad ; w_{j+q-1}^n): \tag{5.13}$$

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 rst order upwind approximation,

$$u_j^{n+1} = u_j^n + \frac{t}{x} \; F(u_j^n) \quad F(u_{j-1}^n) \; ; \tag{5.14}$$

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,

$$w \quad \frac{1}{2K} \quad \frac{1}{w} \quad = 0: \tag{5.16}$$

Applying the rst order upwind approximation (5.14) to the transformed PDE (5.16) and noting that $K \quad = A$ gives the reference space scheme,

$$w_j^{n+1} = w_j^n \quad \frac{}{2A} \quad \frac{1}{w_j^n} \quad \frac{1}{w_{j-1}^n} \quad ; \tag{5.17}$$

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

$$Aw_j^n = \not{x}_j^n \quad \not{x}_{j-1}^n; \tag{5.18}$$

yields

$$\not{x}_j^{n+1} \quad \not{x}_{j-1}^{n+1} \quad \not{x}_j^n + \not{x}_{j-1}^n = \frac{}{2} \quad \frac{1}{w_j^n} \quad \frac{1}{w_{j-1}^n} \quad : \tag{5.19}$$

The anchor point, the fact that $\quad = \quad t$ and $u_j^n = \frac{1}{w_j^n}$ gives the timestepping scheme to be

$$\not{x}_j^{n+1} = \not{x}_j^n + \frac{t}{2} u_j^n: \tag{5.20}$$

The overall moving mesh scheme to solve the Inviscid Burgers equation (5.15) is then

$$\hat{x}_j^{n+1} = \hat{x}_j^n + \frac{t}{2} u_j^n; \tag{5.21}$$

$$A = u_j^n \quad \hat{x}_j^n \quad \hat{x}_{j-1}^n \quad ; \tag{5.22}$$

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 rst 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

$$\hat{x}_j^{n+1} = \hat{x}_j^n + \frac{t}{2} u_{j-1}^n: \tag{5.23}$$

The alternative scheme is then given by

$$\hat{x}_j^{n+1} = \hat{x}_j^n + \frac{t}{2} u_{j-1}^n; \tag{5.24}$$

$$A = u_j^n \quad \hat{x}_{j+1}^n \quad \hat{x}_j^n \quad : \tag{5.25}$$

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.

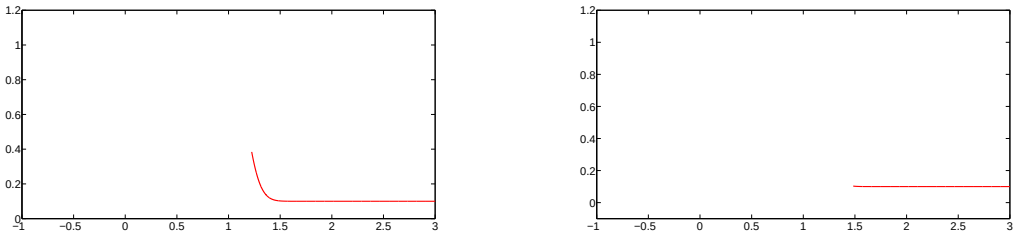
In this section we compare the results of directly applying the fixed grid schemes with the results of the resulting moving mesh schemes as numerical motivation for this extra work.

The schemes we will consider are the first order upwind scheme

$$u_j^{n+1} = u_j^n \quad \frac{t}{x}(F_j^n \quad F_{j-1}^n); \tag{5.26}$$

and the second order upwind scheme

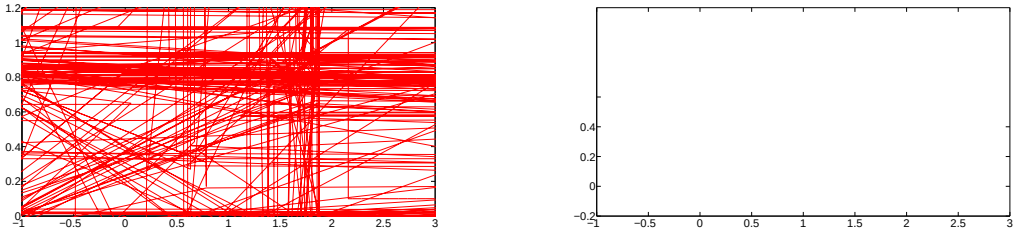
$$u_j^{n+1} = u_j^n \quad \frac{t}{2\ x}(3F_j^n \quad 4F_{j-1}^n \text{ n}$$



(a) First Order Upwind

(b) Second Order Upwind

Figure 5.1: Numerical comparison of the moving mesh schemes (blue) and the Eule-



(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 brie y 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 brie y cover some of the issues that occur when attempting to solve systems of equations with our conservation based moving mesh methods.

The rst major di erence 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 nd 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 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 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 Δx 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 is one-signed, implies that $\hat{x}_t = v$. The Lagrangian formulation given by the density monitor is therefore

$$\hat{x}_t = v; \tag{6.6}$$

$$\int_{\hat{x}_2(t)}^{\hat{x}_1(t)} dx = A(\hat{x}_1(t); \hat{x}_2(t)); \tag{6.7}$$

$$\frac{d}{dt}\int_{\hat{x}_2(t)}^{\hat{x}_1(t)} v dx = [P]_{\hat{x}_2(t)}^{\hat{x}_1(t)} \tag{6.8}$$

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 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

Introduce a set of discrete points $f\hat{x}_j(t)g$ at time t and let $\hat{x}_1(t) = \hat{x}_j(t)$ and $\hat{x}_2(t) = \hat{x}_{j+1}(t)$ de ne an individual cell of the discrete scheme. Assuming that 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

$$_j(\hat{x}_{j+1}(t) \quad \hat{x}_j(t)) = A_{j+1=2} \tag{6.11}$$

and

$$_j v_j(\hat{x}_{j+1}(t) \quad \hat{x}_j(t)) = B_{j+1=2}(t); \tag{6.12}$$

where $B_{j+1=2}(t)$ is the semi-discrete approximation to the momentum integral $B(t) = \int_{\hat{x}_j(t)}^{\hat{x}_{j+1}(t)} (t)v(t)dx$.

We note that in the above, all constant approximations of 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

$$\hat{x}_j^{n+1} = \hat{x}_j^n + \quad tv_j^n; \tag{6.14}$$

The transformation of the independent variables is given by $\hat{x}(\ ;\ ) \ !\ \ , t \ !\ \ ,$ following from Equation (4.13), leading to the transformations

$$@_t = @ \quad \frac{\hat{x}}{\hat{x}}$$

point $\tilde{} \, 2 \, (0; 1]$ giving

$$A(\tilde{}) = \int_0^{\tilde{}} \, dx : \tag{6.26}$$

We may now di erentiate (6.26) to give

$$A \; = \; dx ; \tag{6.27}$$

where we note that A is a constant. This follows from the scalar case and involves comparing the right hand side of (6.26) with the equidistribution principle for .

Inserting (6.27) into equations (6.23) and (6.25) gives the reference space isothermal equations,

$$+ \frac{^2}{A} v \; = 0; \tag{6.28}$$

$$v \; + \frac{a^2}{A} \; = 0: \tag{6.29}$$

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 , v and the derivatives

of our mesh variable x, namely

$$x = v \quad \text{and} \quad x = \frac{A}{}:$$

Di erentiating (6.6) with respect to and using (6.29) and (6.27) eliminates and v and gives

$$x = v = \frac{a^2}{A} = a^2 \frac{1}{x} \tag{6.30}$$

which may be rearranged to give the mesh PDE

$$x = a^2 \frac{x}{x^2}: \tag{6.31}$$

Since equation (6.31) is a non-linear wave equation we could use existing theory concerning nite di erence 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 $\frac{a^2}{}$

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

$$\frac{@^2u}{@t^2} = \quad \frac{G}{u}; \tag{6.32}$$

where $G = G(u;u_x)$ is a function of both u and u_x and $\frac{G}{u} = \frac{@G}{@u} \quad \frac{@}{@x} \quad \frac{@G}{@u_x}$ is the variational derivative of G with respect to u. Furthermore $x \ 2 \ [0;L]; L < 1$ is the one-dimensional spatial variable and t is the time variable.

De nition 6.2.1. Given $u(x;t)$ and a function G of u, u_x the energy integral is de ned to be

$$I = \int_0^L \quad \frac{1}{2}u_t^2 + G \quad dx: \tag{6.33}$$

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.

$$\frac{d}{dt}\int_0^L \quad \frac{1}{2}u_t^2 + G \quad dx = 0: \tag{6.35}$$

Proof. Applying Leibniz integral rule to the left hand side of equation (6.35),

$$\frac{d}{dt}\int_0^L \quad \frac{1}{2}u_t^2 + G \quad dx = \int_0^L (u_{tt}u_t + G_u u_t + G_{u_x}u_{xt})\,dx: \tag{6.36}$$

Now use integration by parts on the final term of the integrand and apply the definition of the variational derivative,

$$\frac{d}{dt}\int_0^L \left(\frac{1}{2}u_t^2 + G\right) dx = \int_0^L u_t$$

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 di erence and $g_l^-(u_j)$ are functions of u_j which approximate u_x using a downwind di erence.

Having found a consistent approximation G_d we now use it to calculate a discrete equivalent to the variational derivative $\frac{G}{u}$ which we denote $\frac{G_d}{(u_j ; v_j)}$. This is achieved by considering the following property of the variational derivative, namely

$$\int_0^L G(u)dx \quad \int_0^L G(v)dx \quad \int_0^L \frac{G}{u}(u \quad v)dx + [G_{u_x}(u \quad v)]_0^L : \tag{6.41}$$

Before we can consider a discrete equivalent to (6.41) we rst consider a discrete equivalent of integration by parts.

Theorem 6.2.3 (Summation by Parts).

Equation (6.43) can easily be veri ed 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

a stable scheme for the mesh movement and use this in a new Lagrangian scheme for solving the isothermal equations.

The main di erence 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,

$$\dot{x} = a^2 \frac{\partial x}{\partial x^2}; \tag{6.46}$$

$$\dot{x} = \frac{A}{\rho}; \tag{6.47}$$

$$\dot{x} = v: \tag{6.48}$$

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 nd the Furihata scheme we must rst relate equation (6.46) with the general nonlinear wave equation (6.32). It can be easily veri ed that equation (6.46) is indeed of the desired form with

$$G(x; \dot{x}) = a^2 \ln(\dot{x}): \tag{6.49}$$

We approximate the function G using (6.40) in the form

$$G_d(x_j) = \frac{a^2}{2} \ln(\quad + x_j) \mp \ln(\quad$$

Inserting (6.50) into the left hand side of equation (6.43) gives

$$\sum_{j=0}^{N} (G_d(x_j^n) - G_d(x_j^{n-1})) = \frac{a^2}{2} \sum_{j=0}^{N} (\ln(\ ^+x_j^n) + \ln(\ ^-x_j^n) - \ln(\ ^+x_j^{n-1}) - \ln(\ ^-x_j^{n-1})) \quad ; \quad (6.51)$$

and applying the summation by parts formula gives

$$\sum_{j=0}^{N} (G_d(x_j^n) - G_d(x_j^{n-1})) = \frac{a^2}{2} \sum_{j=0}^{N} - \frac{\ln(\ ^+x_j^n) - \ln(\ ^+x_j^{n-1})}{\ ^+x_j^n - \ ^+x_j^{n-1}} ! + \ ^+ \frac{\ln(\ ^-x_j^n) - \ln(\ ^-x_j^{n-1})}{}$$

In order to get around this issue we rewrite (6.53) as

$$\frac{G_d}{(\hat{x}_j^n;\hat{x}_j^{n-1})} = a^2 \left(\frac{\ln(1+\ )}{\hat{x}_{j-1=2}^n} \quad \frac{\ln(1+\ )}{\hat{x}_{j+1=2}^n} \right)^{!} \tag{6.54}$$

where $= \frac{\hat{x}_{j\ 1=2}^{n} - \hat{x}_{j\ 1=2}^{n\ 1}}{\hat{x}_{j\ 1=2}^{n}}$ and $= \frac{\hat{x}_{j+1\ =2}^{n} - \hat{x}_{j+1\ =2}^{n\ 1}}{\hat{x}_{j+1\ =2}^{n}}$.

Assuming that we are taking su ciently small timesteps such that and are small then we may approximate (6.54) using the Taylor expansion of $\ln(1+\ )$. Hence

$$\frac{G_d}{(\hat{x}_j^n;\hat{x}_j^{n-1})} \quad a^2 \left(\frac{1}{\hat{x}_{j-1=2}^n} \quad 1 \quad \frac{}{2} \quad \frac{1}{\hat{x}_{j+1=2}^n} \quad 1 \quad \frac{}{2} \right)^{!}$$

$$= a^2 \quad \frac{3}{2} \quad \frac{1}{\hat{x}_{j-1=2}^n} \quad \frac{n\ 1}{j-1=2} \quad \frac{1}{\hat{x}_j^n} \quad \ln(1+$$

velocity everywhere. The initial condition is given by

$$(x;0) = \begin{cases} (1 \quad x^2)^2 + 0{:}1 & jxj \ 6 \ 1 \\ 0{:}1 & \text{otherwise} \end{cases} ; \tag{6.56}$$

and

$$v(x;0) = 0{:} \tag{6.57}$$

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 nd 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 su cient 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 brie y 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 re nement (r-re nement) 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 nal 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

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

di cult and care needs to be taken since vorticity can add new ways of tangling

Bibliography

[AO97] Mark Ainsworth and J Tinsley Oden, *A posteriori error estimation in nite element analysis*, Computer Methods in Applied Mechanics and Engineering 142 (1997), no. 1, 1{88.

[Aro] DG Aronson, *The porous medium equaion. in nonlinear di usion problems*, Lecture notes in mathematics, no. 1224.

[Bai94] MJ Baines, *Moving nite elements*, Oxford University Press, Inc., 1994.

[Bai987(1224.)]TJ 0 87K(Inc.,)-314(1994.)]TJecDai94]

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 re nement 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 Dor and L. Drury, *Simple adaptive grids for 1d initial value problems*, Journal of Computational Physics 69 (1987), pp. 175 { 195.

[Eva10] Lawrence Evans, *Partial di erential equations*, AMS, 2010, ISBN: 0821849743, 9780821849743.

[Fla73] Harley Flanders, *Di erentiation 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 di erential equations*, Chapman & Hall/CRC Press, 2011, ISBN: 1420094467, 9781420094466.

[Fur01] Daisuke Furihata, *Finite-di erence 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 di erence 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 di erential 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 di erential equations (mmpdes) based on the equidistribution principle*, SIAM Journal on Numerical Analysis 31 (1994), pp. 709 { 730.

[KHDB03] E Kuhl, S Hulsho , and R De Borst, *An arbitrary lagrangian eulerian nite-element approach for uid{structure interaction phenomena*, Inter-

[Ole63] O. Oleinik, *Discontinuous solutions of non-linear di erential 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 ows with moving boundaries and uid-structure interaction*, International Journal for Numerical Methods in Engineering **71** (2007), no. 9, 1009{1050.

[Shu09] Chi-Wang Shu,

[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 di erence 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 nite element method for the numerical solution of partial di erential equations and systems*, Ph.D. thesis, University of Reading, 2004.

[Wes01] Pieter Wesseling, *Principles of computational uid dynamics*, 1 ed., Springer Series in Computational Mathematics 29, Springer-Verlag Berlin Heidelberg, 2001.