UNIVERSITY OF READING
DEPARTMENT OF MATHEMATICS AND STATISTICS

# Truncation Error Estimates for Mesh Refinement in Lagrangian Hydrocodes

Sarah Lianne Cole

Thesis submitted for the degree of
Doctor of Philosophy
August 2013

# Abstract

Compressible fluid flows governed by the Euler Equations of gasdynamics exhibit sharp features such as shocks and expansions which are difficult to capture computationally. In order to focus efficiently on these features an adaptive facility is required.

A standard numerical scheme is the so-called Lagrange-plus-remap method, otherwise known as the ALE (Arbitrary Lagrangian Eulerian) scheme, on a refined mesh using AMR (Adaptive Mesh Refinement). This thesis focuses on the Lagrangian step in the ALE scheme and on an adaptive strategy based on error estimates. The strategy makes use of truncation errors of pairs of schemes of the same order of accuracy to monitor the error and to refine the mesh. Error estimates are obtained for the Isothermal and Euler Equations in both one and two dimensions. The strategy is shown to be effective in refining areas of compressibility.

# Declaration

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

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

Sarah Lianne Cole

# Acknowledgements

Firstly I would like to thank my supervisors. Dr Pete Sweby, for his endless support and encouragement. Thanks to the inspirational Dr Jenny Macey, for sharing her wealth of knowledge and continuously going above and beyond. A very special thank you to Prof. Mike Baines, for constantly going out of his way to offer guidance, sharing his invaluable knowledge and always welcoming my endless ramblings. I could not have done this without you all. Thank you.

Thanks also go to all those at AWE, Dr Andy Barlow, Dr Rob Keevis and Dr Pete Spence for providing additional guidance.

To my two TJs, thank you for always having faith in me. Thank you for all your support, providing me with words of comfort and ensuring I was looked after, especially during the final stages of this work.

To my beautiful girls, thank you for all your understanding and patience when I've not been the best with staying in touch and for reminding me there is a life outside of University.

Last but by no means least, to my wonderfully supportive family, Mum, Chris, Fi, Suz and Caz. Thank you so much for always believing I could do this especially when I'd lost all hope and for all the encouragement and support throughout my studies.

Additionally, I would like to acknowledge the financial support received from EPSRC and AWE.

# Contents

# List of Figures

# List of Tables

# Nomenclature

| | |
|---|---|
| $i$ | Nodal index in $\xi$ direction |
| $j$ | Nodal index in $\eta$ direction |
| $\boldsymbol{\xi}$ | Lagrangian time independent reference vector |
| $h \equiv \Delta\xi$ | Space step in $\xi$ direction |
| $l \equiv \Delta\eta$ | Space step in $\eta$ direction |
| $N$ | Number of spatial steps in $\xi$ direction |
| $M$ | Number of spatial steps in $\eta$ direction |
| $[x_0, x_N]$ | Endpoints of the mesh in the $\xi$ direction |
| $[y_0, y_N]$ | Endpoints of the mesh in the $\eta$ direction |
| $el_{1,2,\dots NM}$ | Elements on a mesh in 2D |
| $t$ | Time |
| $n$ | Time step index |
| $t^n$ | Time at step number $n$ |
| $\Delta t = k$ | Timestep |

| | |
|---|---|
| $\hat{\boldsymbol{x}}$ | Nodal position vector in terms of Eulerian co-ordinates |
| $\hat{\boldsymbol{u}}$ | Nodal velocity vector in terms of Eulerian co-ordinates |
| $\hat{V}$ | Element volume in terms of Eulerian co-ordinates |
| $\hat{\rho}$ | Density in terms of Eulerian co-ordinates |
| $\hat{p}$ | Pressure in terms of Eulerian co-ordinates |
| $\hat{E}$ | Total energy in terms of Eulerian co-ordinates |
| $\hat{\varepsilon}$ | Specific internal energy in terms of Eulerian co-ordinates |
| $\boldsymbol{x}$ | Nodal position vector in terms of Lagrangian co-ordinates |
| $\boldsymbol{u}$ | Velocity vector in terms of Lagrangian co-ordinates |
| $V$ | Element volume in terms of Lagrangian co-ordinates |
| $\rho$ | Density in terms of Lagrangian co-ordinates |

| | |
|---|---|
| $p$ | Pressure in terms of Lagrangian co-ordinates |
| $E$ | Total energy in terms of Lagrangian co-ordinates |
| $\varepsilon$ | Specific internal energy in terms of Lagrangian co-ordinates |
| $m$ | Mass in Lagrangian co-ordinates |
| $\boldsymbol{f}$ | The force vector in the momentum equation |
| $g$ | Internal energy function |
| $\bar{\boldsymbol{u}}$ | Average velocity in time |
| $\boldsymbol{x}_*$ | Scheme 2 nodal position vector |
| $\boldsymbol{u}_*$ | Scheme 2 velocity vector |
| $\bar{\boldsymbol{u}}_*$ | Average velocity vector in time for Scheme 2 |
| $\varepsilon_*$ | Scheme 2 specific internal energy |
| $\boldsymbol{f}_*$ | Scheme 2 force vector |
| $g_*$ | Scheme 2 specific internal energy function |
| $t_*^n$ | The time at step $n$ for Scheme 2 |
| $p_{side}$ | Pressure on the boundary of an element |
| $\boldsymbol{x}_1$ | Nodal position vector corresponding to Scheme 1 |
| $\boldsymbol{x}_2$ | Nodal position vector corresponding to Scheme 2 |
| $p_a$ | The pressure term corresponding to Scheme A |
| $p_b$ | The pressure term corresponding to Scheme B |
| $f_{ij}^{\xi}$ | Force term at a node in the $\xi$ direction |
| $f_{ij}^{\eta}$ | Force term at a node in the $\eta$ direction |
| $u_{ij}^{\xi}$ | Velocity term at a node in the $\xi$ direction |
| $u_{ij}^{\eta}$ | Velocity term at a node in the $\eta$ direction |
| $m_{ij}$ | Nodal mass |
| $s$ | Any scalar |
| $\boldsymbol{v}$ | Any vector |
| $d$ | Dimension |
| $f$ | Any function of space and time |

| | |
|---|---|
| $\partial\hat{V}$ | Boundary of the control volume in Eulerian co-ordinates |
| $\hat{S}$ | A side of the the control volume in Eulerian co-ordinates |
| $\hat{\boldsymbol{n}}$ | Outward normal of the control volume in Eulerian co-ordinates |
| $\Omega$ | Control volume in 2D |
| $S$ | The boundary of $\Omega$ |
| $\Omega_e$ | Control volume of an element in 2D |

| | |
|---|---|
| $\partial\omega_e$ | Boundary of an element in 2D |
| $S_e$ | The side of an element in 2D |

| | |
|---|---|
| $\nabla_{\hat{\boldsymbol{x}}}$ | Spatial derivative in terms of the position vector in Eulerian co-ordinates |
| $\nabla_{\boldsymbol{x}}$ | Spatial derivative in terms of the position vector in Lagrangian co-ordinates |
| $\nabla_{\boldsymbol{\xi}}$ | Spatial derivative in terms of the Lagrangian reference vector |
| $\frac{D}{Dt}$ | Material derivative linking the Eulerian and Lagrangian co-ordinates |
| $\frac{\partial}{\partial t}$ | The time derivative used for Lagrangian co-ordinates |
| $\dot{\boldsymbol{x}} = \frac{d\boldsymbol{x}}{dt}$ | Temporal derivative of $\boldsymbol{x}$ |
| $\dot{\boldsymbol{u}} = \frac{d\boldsymbol{u}}{dt}$ | Temporal derivative of $\boldsymbol{u}$ |
| $\dot{\varepsilon} = \frac{d\varepsilon}{dt}$ | Temporal derivative of $\varepsilon$ |
| $\boldsymbol{J}$ | The Jacobian introduced to map from the Eulerian co-ordinates to the Lagrangian co-ordinates |
| $\boldsymbol{J}^{-1}$ | The inverse of the Jacobian |
| $\lvert\boldsymbol{J}\rvert$ | The determinant of the Jacobian |
| $\lvert\boldsymbol{J}_1\rvert$ | Determinant of the Jacobian for Scheme 1 |
| $\lvert\boldsymbol{J}_2\rvert$ | Determinant of the Jacobian for Scheme 2 |
| $\boldsymbol{C}$ | The adjoint matrix introduced from inverting the Jacobian |

| | |
|---|---|
| $q$ | Artificial viscosity term |
| $c_q$ | Quadratic artificial viscosity term |
| $c_l$ | Linear artificial viscosity term |
| $\Phi$ | Monotonic limiter |
| $L_{hor}$ | Horizontal mesh legs |
| $L_{ver}$ | Vertical mesh legs |
| $C$ | Courant number |
| $c_s = a$ | Sound speed |
| $R = \gamma$ | Ideal gas constant |
| $T$ | Temperature |

| | |
|---|---|
| $\tau_n$ | Temporal truncation error |
| $\tau_i$ | Spatial truncation error |
| $D$ | Constant coefficient in the truncation error |

| | |
|---|---|
| $L$ | Velocity differential operator |
| $L_h$ | Velocity algorithm differential operator |
| $S$ | Pressure differential operator |
| $S_h$ | Pressure algorithm differential operator |
| $\Psi$ | Leading term of the 2D spatial Truncation error |
| $e$ | Error in one step |
| $\sigma$ | General magnitude of truncation error exhibited for a scheme in Milne's approach |
| $\tilde{\sigma}$ | General magnitude of truncation error exhibited for a second scheme in Milne's approach |
| $k$ | Leading term of the truncation error in Milne's approach |
| $D$ | Constant coefficient for a scheme in Milne's approach |
| $\tilde{D}$ | Constant coefficient for a second scheme in Milne's approach |
| $\tilde{x}$ | Error estimated for this variable in Milne's approach |
| $\mu_n$ | The truncation error for a second scheme in Milne's approach |

| | |
|---|---|
| $\omega$ | General test function |
| $\phi$ | Hat function |
| $\psi$ | Test function used in Petrov-Galerkin approach |
| $\boldsymbol{\psi}$ | 2D Petrov-Galerkin function |
| $\boldsymbol{N}$ | Isoparametric Bilinear Quadrilaterals |
| $\delta_{ij}$ | Matrix with zeros everywhere apart from the diagonal |
| $A$ | Matrix of constants |
| $D$ | Diagonal matrix |
| $M_e$ | 2D mass matrix for an element |

| | |
|---|---|
| $F$, $G$ | Functions in the solution of the wave equation |
| $\beta$, $\nu$ | Variables of the wave equation |
| $c_1$, $c_2$ | Constants used in contact verification |
| $\rho_l$, $\rho_r$ | Left and right states of density in jump condition |
| $x_l$, $x_r$ | Left and right states of nodes in jump condition |
| $\varepsilon_l$, $\varepsilon_r$ | Left and right states of $\varepsilon$ in jump condition |

| | |
|---|---|
| $\boldsymbol{r}$ | Quadrilateral vertices in physical space |
| $L$ | Length of side of quadrilateral |
| $r$ | Side direction |
| $A_1$, $A_2$, $A_3$, $A_4$ | Shape functions in terms of $x$ |
| $B_1$, $B_2$, $B_3$, $B_4$ | Shape functions in terms of $y$ |

# Chapter 1

# Introduction

The purpose of this thesis is to develop an error estimate for the Lagrangian phase of an ALE (Arbitrary Lagrangian Eulerian) scheme, which is used to solve the compressible Euler Equations of gasdynamics. When used as a refinement monitor it is computationally cost effective yet retains an accurate numerical solution. In this chapter we discuss the key ideas used to motivate this work and outline the developments towards obtaining the error estimates.

It is often required to obtain accurate numerical solutions for large scale fluid flow problems. Mathematical models of fluids, such as the compressible Euler equations, arise from the understanding of how fluids behave. Ideally we would like to solve problems analytically. A problem with simple geometries, initial data and boundary conditions sometimes have an analytic solution associated to it. However, due to the complexity of such mathematical models it is not always possible to determine an exact solution, hence a numerical solution is sought.

Computational Fluid Dynamics (CFD) is the study of numerical techniques applied to such problems. For compressible flows the solution of the Euler Equations of gasdynamics often exhibit sharp features which can be difficult to model numerically. Advancements towards obtaining accurate and efficient numerical solutions include a combination of Adaptive Mesh Refinement (AMR) and ALE. Although many techniques exist it still remains to obtain a suitable refinement criteria to incorporate with the AMR.

Before investigating numerical approaches we must understand the behaviour of fluids and where we can make assumptions about conservation of mass, momentum and energy.

We are interested in large scale fluid flow problems since they are used to obtain solutions for complex physical problems such as blast waves, gas flows, shock reflections and collisions, flow around airfoils, and many more.

With the introduction of computers a range of Hydrocodes containing the numerical methods used to solve such compressible fluid flow problems have been developed. Hydrocodes are designed to deal with large scale fluid flow problems efficiently.

In order to apply such numerical methods a computational mesh is required, which may move or remain fixed. The form of the mesh has an effect on computational expense and accuracy of the solution. The effects of the mesh on accuracy and efficiency is the main motivation for this thesis.

## 1.1 Motivation

Compressible fluid flow problems exhibit such features as shocks i.e. the solution varies rapidly, whereas the solution elsewhere may vary slowly or remain constant. Modeling shocks numerically is a difficult task. For the sake of efficiency it is desirable to use a fine mesh around flow features which are difficult to capture numerically. However, using a globally fine mesh is extremely computationally expensive.

We wish to use a fine mesh only in the areas of rapidly varying solutions and a coarser mesh elsewhere, where the solution varies less rapidly.

From the motivation the overall aim is to reduce computational expense by keeping the number of nodes and elements to a minimum whilst still preserving the accuracy of the solution by using a method which only refines the mesh around specific flow features.

## 1.2 Techniques

The two main types of refinement considered in this thesis are $r$-refinement and $h$-refinement. We do not consider $p$-refinement in this work, which is where the numerical scheme is refined rather than the mesh.

For $r$-refinement the number of nodes remain fixed and are moved towards the areas of interest. A disadvantage of this approach is that there will be under-resolved areas of the mesh which could lead to ambiguities. The ALE method is widely used by the CFD community and is a type of $r$-refinement. This method combines the advantages of a fixed mesh and moving mesh. The details and developments of this method will be discussed further in

Chapter 2.

It is known as $h$-refinement when elements are subdivided. This leads to an increase in elements and nodes. AMR is a type of $h$-refinement. AMR is used by the CFD community to detect specific flow features using a monitor, then only these areas are refined. The monitor could take the form of density gradients or error estimates amongst others. An overview of refinement monitors is given in Chapter 2.

The requirement of an efficient refinement monitor leads to the aims of this thesis.

## 1.3 Aims

Although a variety of refinement criteria exist, including monitors which may take the form of an error estimate, see [77], there is yet to be an error estimate based entirely on the full form of the truncation error for the utilised schemes. Thus the aim of this thesis is to develop an error estimate based on truncation errors.

The aim is to obtain an error estimate for a system of two equations then advance to estimate an error for the Euler equations, which is to be done by developing the approach in 1D then advancing to 2D. We aim to obtain the error estimate by comparing the truncation error of a widely used method (called here the Main Scheme) from [64] against the truncation error of an alternative method of the same order. The difference between the two methods can then be used to obtain an estimated error using a technique similar to Milne's approach for Ordinary Differential Equations (ODEs), this will be discussed further in Chapter 2.

To develop an appropriate error estimate we concentrate on the Lagrangian step of the ALE approach since this is where the solution initially becomes available.

Since we are dealing with truncation errors and solutions containing discontinuities, we verify that the method can be applied to problems with steep gradients in Appendix C.

By obtaining a monitor which takes the form of an error estimate we aim to detect rapidly varying flow features in the solution automatically. Hence, we use the monitor to trigger which areas of the mesh require refinement. In return we obtain a solution which is computationally cost effective whilst maintaining the accuracy of the solution.

We now discuss the thesis layout along with the developments towards obtaining an error estimate for the 2D compressible Euler equations.

## 1.4 Developments and Thesis Layout

In Chapter 2 we set up the problem. The equations of interest are described along with their formulations in Eulerian and Lagrangian reference frames. The numerical schemes used to solve the system of equations are then discussed. We give an overview of the work that already exists for different methods of refinement and error estimations. This leads on to the discussion of how we develop an error estimate in this work.

We discuss the use of two semi-discrete temporal schemes in Chapter 3. Two schemes for the temporal derivatives are described for systems of two and three equations. We obtain the truncation errors for each scheme which are then used to determine a computable error in one step.

The spatial derivatives are investigated in Chapter 4 for the systems of two and three equations. The Galerkin and Petrov-Galerkin approaches are described in order to obtain two different semi-discrete schemes in space. The truncation error from each spatial scheme is obtained, leading to a spatial approximation to the error in one step.

In Chapter 5 we combine the methods from Chapter 3 and Chapter 4. This leads to the schemes for the two and three systems being expressed in their fully-discrete form, from which we can derive errors for both the density and velocity.

Chapter 6 consists of the results from the approximated errors compared against a test problem, Sod's shock tube, details of which are given in Chapter 6 .

Since the main aim of this work is to investigate error estimates for the system of three equations in 2D we extend the results from Chapter 5 to the 2D case of the Euler equations in Chapter 7.

Chapter 8 is dedicated to the results for the 2D case using a variety of test problems, where the details of such problems are given.

Finally in Chapter 9 we discuss all results, draw some conclusions and highlight areas of further work.

In Appendix A we define some preliminaries including the timestep and artificial viscosity calculation. We discuss the odd-even decoupling phenomenon to verify the choice of a staggered grid used in this work, in Appendix B. In Appendix C we demonstrate a test problem to our approach using the wave equation with a smooth solution of varying gradients in order to investigate the validity of our approach for steep solutions. Some of the difficulties observed with the numerical solution, namely the contact is discussed in Appendix D. Finally we give details of the 2D shape functions in order to express the 2D spatial discretisations in Appendix E.

# Chapter 2

# Background

In this work we look at the Euler equations of gas dynamics in a Lagrangian reference frame. To understand how a Lagrangian reference frame works and the form of the equations we also consider an Eulerian reference frame. Since much of the literature express the Euler equations in an Eulerian reference frame or a hybrid form we wish to clarify what we mean by a Lagrangian reference frame. By doing so we distinguish between the Eulerian reference frame, hybrid form and Lagrangian reference frame.

We begin by discussing the behaviour of fluids in order to understand the mathematical model in different forms.

Once the mathematical model is described we discuss the use of approximating the mathematical model using hydrocodes. Hydrocodes make use of numerical techniques to obtain an approximate solution to the mathematical model. Some of the techniques used in this thesis come from the hydrocode CORVUS, as discussed in [64, 10].

To begin the investigation using numerical methods, we set up the computational mesh. Such numerical methods are then discussed along with the treatment of shocks using artificial viscosity. Shock treatment is an important aspect in this thesis since we aim to refine around shocks efficiently. From this we discuss some of the different refinement strategies that exist. In this thesis we wish to develop a refinement monitor based on error estimates. We discuss the different techniques used to estimate errors. This leads to the methodology behind the technique we develop in this thesis, which is initially discussed in Chapter 3, where the development of an error estimator begins with the semi-discrete in time case.

## 2.1 Behaviour of Fluids

To formulate the model we use continuum mechanics, this describes the motion of a fluid as a continuous mass rather than as discrete particles. Solids, gases and liquids are made up of molecules which are separated by spaces, therefore there are discontinuities in the material. However, since the molecules are dense the material is often modeled as a continuum, where the matter within the material is assumed to be continuously distributed over the entire region.

Problems are characterised by the specific behaviour of the fluid. In this thesis we are particularly interested in the effects of compressibility. A compressible fluid is one in which a change in pressure results in a change in density. Fluids can also be viscous, where the fluid friction has an effect on the motion of the fluid. However, where other features of the fluid are more significant than viscosity then it can be neglected and the fluid assumed to be inviscid. In order to derive the equations which describe the material behaviour, the following assumptions are made:

- conservation of mass
- conservation of momentum (or momentum balance)
- conservation of energy.

In this thesis the fluid is assumed to be inviscid since compressibility dominates. The compressible nature of the fluid often results in shock waves, where there will be discontinuities in the flow variables.

Mathematical models have been proposed using the knowledge of the fluid behaviour. Next we express such models in three forms in order to distinguish between the forms stated in much of the literature and the form we use in this thesis.

## 2.2 Eulerian and Lagrangian Specifications

There are two sets of variables which can be used to describe the kinematic deformation of the continuum. These are known as, Eulerian, which is a spatially based description, and Lagrangian, which is a material based description.

To visualise the differences between an Eulerian and Lagrangian reference frame consider a stationary ball placed on a river bank. The ball is the reference point here and the flow of the river is described at the point it passes the ball. This ball is in an Eulerian reference frame. Now consider the ball to be floating on the river as it flows downstream. The ball is the reference point again, however it is now following the fluid. The ball is in a Lagrangian reference frame.

In a Lagrangian frame the motion of a continuum moving in $\mathbb{R}^d$, where $d$ denotes the dimension, is traced using the function

$$\boldsymbol{x} : \mathbb{R}^d \times \mathbb{R} \to \mathbb{R}^d.$$

We have

$$\hat{\boldsymbol{x}} = \boldsymbol{x}(\boldsymbol{\xi}, t) \tag{2.1}$$

which is the location $\boldsymbol{x}$ of a particle within the continuum described by a fixed co-ordinate $\boldsymbol{\xi}$ at time $t$ which coincides instantaneously with the Eulerian co-ordinate $\hat{\boldsymbol{x}}$. We represent the Eulerian reference point as $\hat{\boldsymbol{x}}$ and the Lagrangian reference point to be a time independent vector $\boldsymbol{\xi}$.

The Lagrangian velocity (or material velocity) is defined as $\boldsymbol{u}(\boldsymbol{\xi}, t)$, where

$$\boldsymbol{u}(\boldsymbol{\xi}, t) = \frac{\partial \boldsymbol{x}(\boldsymbol{\xi}, t)}{\partial t} \tag{2.2}$$

and the Eulerian velocity or spatial velocity is defined as $\hat{\boldsymbol{u}}(\hat{\boldsymbol{x}}, t)$ where in view of eqn. (2.1),

$$\hat{\boldsymbol{u}}(\hat{\boldsymbol{x}}, t) = \hat{\boldsymbol{u}}(\boldsymbol{x}(\boldsymbol{\xi}, t), t) = \boldsymbol{u}(\boldsymbol{\xi}, t). \tag{2.3}$$

Hence the relationship between Lagrangian and Eulerian velocities can be expressed using eqn. (2.2) as

$$\frac{\partial \boldsymbol{x}(\boldsymbol{\xi}, t)}{\partial t} = \hat{\boldsymbol{u}}(\boldsymbol{x}(\boldsymbol{\xi}, t), t) = \boldsymbol{u}(\boldsymbol{\xi}, t) \tag{2.4}$$

for a smooth $\boldsymbol{x}(\boldsymbol{\xi}, t)$.

We now demonstrate how mathematical models for fluid flow problems can be expressed in an Eulerian form and a Lagrangian form, additionally, how the Eulerian and Lagrangian specifications for continuum deformations are connected by the material derivative. This is done as it is often difficult to find a summary of the three representations. The equations are usually expressed in terms of the material derivative, see [65, 10, 17, 68].

## 2.3 Euler Equations of Gasdynamics

The equations for mass, momentum and specific internal energy in Eulerian form may be derived for a volume of fluid within a continuum using an Eulerian reference frame, i.e. a control volume fixed in time. Considering a fixed volume, the equations are derived in accordance to how fluid flows across this volume in time.

### 2.3.1 Mass

Under the assumption that the mass of the fixed material volume increases in accordance with the flux of mass across its boundary, from [40] we have

$$\frac{d}{dt}\int_{\hat{V}} \hat{\rho} d\hat{V} = -\oint_{\partial\hat{V}} \hat{\rho}\hat{\boldsymbol{u}}\cdot\hat{\boldsymbol{n}} d\hat{S}, \tag{2.5}$$

where $\hat{\rho}$ is the density in terms of spatial coordinates, $\partial\hat{V}$ is the surface enclosing $\hat{V}$ in the spatial coordinates and $\hat{\boldsymbol{u}}\cdot\hat{\boldsymbol{n}}$ is the outward normal component of the velocity on the surface, see [60].

Using the divergence theorem we obtain

$$\int_{\hat{V}} [\hat{\rho}_t + \nabla\cdot(\hat{\rho}\hat{\boldsymbol{u}})] d\hat{V} = 0$$

where $\nabla = \nabla_{\hat{x}}$.

Since $\hat{V}$ is arbitrary the integrand vanishes, resulting in the conservation of mass equation in Eulerian coordinates

$$\hat{\rho}_t + \nabla\cdot(\hat{\rho}\hat{\boldsymbol{u}}) = 0. \tag{2.6}$$

### 2.3.2 Momentum

Using Newton's second law for balancing forces where the rate of change of momentum in a control volume is equal to the forces acting on it, from [40] we have

$$\frac{d}{dt}\int_{\hat{V}} \hat{\rho}\hat{\boldsymbol{u}} d\hat{V} = -\oint_{\partial\hat{V}} [\hat{\rho}\hat{\boldsymbol{u}}\hat{\boldsymbol{u}} + \hat{p}\hat{\boldsymbol{n}}] d\hat{S}, \tag{2.7}$$

where $\hat{p}$ is the pressure in terms of spatial co-ordinates.
Using the divergence theorem this becomes

$$\int_{\hat{V}} [(\hat{\rho}\hat{\boldsymbol{u}})_t + \nabla \cdot (\hat{\rho}\hat{\boldsymbol{u}}\hat{\boldsymbol{u}}) + \nabla\hat{p}] d\hat{V} = 0.$$

The integrand vanishes since the limit $\hat{V}$ is arbitrary, resulting in the momentum balance equation written in Eulerian coordinates

$$(\hat{\rho}\hat{\boldsymbol{u}})_t + \nabla \cdot (\hat{\rho}\hat{\boldsymbol{u}}\hat{\boldsymbol{u}}) + \nabla p = 0. \tag{2.8}$$

### 2.3.3 Specific Internal Energy

Using the first law of thermodynamics for energy balance, as stated in [15] the total energy for a material volume is

$$\frac{d}{dt}\int_{\hat{V}} \hat{E} d\hat{V} = -\oint_{\partial\hat{V}} \hat{p}\hat{\boldsymbol{u}} \cdot \hat{\boldsymbol{n}} d\hat{S} - \oint_{\partial\hat{V}} (\hat{\boldsymbol{u}}\hat{E})\hat{\boldsymbol{n}} d\hat{S}, \tag{2.9}$$

where $\hat{E}$ is the total energy in terms of spatial co-ordinates.
Using the divergence theorem we obtain

$$\int_{\hat{V}} [(\hat{E})_t + \nabla \cdot \hat{\boldsymbol{u}}(\hat{E} + p)] d\hat{V} = 0.$$

The integrand vanishes due to the arbitrary nature of $\hat{V}$, giving

$$\hat{E}_t + \nabla \cdot (\hat{\boldsymbol{u}}\hat{E}) + \nabla \cdot (\hat{p}\hat{\boldsymbol{u}}) = 0. \tag{2.10}$$

In this case we wish to gain an expression for specific internal energy since this is the variable used in the Main Scheme from [64]. The internal energy per unit density is expressed as $\hat{\rho}\hat{\varepsilon}$ where the specific internal energy is $\hat{\varepsilon}$.
As stated in [52] we have

$$\hat{E} = \frac{1}{2}\hat{\rho}\hat{\boldsymbol{u}}^2 + \hat{\rho}\hat{\varepsilon},$$

and by substituting into eqn. (2.10) we have

$$\frac{1}{2}(\hat{\rho}\hat{\boldsymbol{u}}^2)_t \quad + \quad (\hat{\rho}\hat{\varepsilon})_t + \nabla \cdot (\frac{1}{2}\hat{\rho}\hat{\boldsymbol{u}}^3) + \nabla \cdot (\hat{\boldsymbol{u}}\hat{\rho}\hat{\varepsilon}) + \nabla \cdot (\hat{\boldsymbol{u}}\hat{p}) = 0. \tag{2.11}$$

Using the identity

$$\nabla \cdot (s\boldsymbol{v}) = s\nabla \cdot \boldsymbol{v} + \boldsymbol{v} \cdot \nabla s \tag{2.12}$$

for any scalar $s$ and any vector $\boldsymbol{v}$, we obtain

$$\begin{aligned}\frac{1}{2}\hat{\boldsymbol{u}}(\hat{\rho}\hat{\boldsymbol{x}})_t \quad &+ \quad \frac{1}{2}\hat{\rho}\hat{\boldsymbol{u}}\hat{\boldsymbol{u}}_t + \hat{\rho}_t\hat{\varepsilon} + \hat{\varepsilon}_t\hat{\rho} + \hat{\rho}\hat{\boldsymbol{u}}^2\nabla \cdot \hat{\boldsymbol{u}} + \frac{1}{2}\hat{\boldsymbol{u}}^2\nabla \cdot \hat{\rho}\hat{\boldsymbol{u}} \\ &+ \quad \hat{\varepsilon}\nabla \cdot \hat{\rho}\hat{\boldsymbol{u}} + \hat{\rho}\hat{\boldsymbol{u}} \cdot \nabla\hat{\varepsilon} + \hat{\boldsymbol{u}} \cdot \nabla\hat{p} + \hat{p}\nabla \cdot \hat{\boldsymbol{u}} = 0. \end{aligned} \tag{2.13}$$

Simplifying and using eqn. (2.6) and eqn. (2.8) we then obtain

$$\begin{aligned} - \quad & \frac{1}{2}\hat{\rho}\hat{\boldsymbol{u}}^2\nabla \cdot \hat{\boldsymbol{u}} - \frac{1}{2}\hat{\boldsymbol{u}}^2\nabla \cdot \hat{\rho}\hat{\boldsymbol{u}} - \frac{1}{2}\hat{\boldsymbol{u}} \cdot \nabla\hat{p} + \frac{1}{2}\hat{\boldsymbol{u}}^2\nabla \cdot \hat{\rho}\hat{\boldsymbol{u}} \\ - \quad & \frac{1}{2}\hat{\rho}\hat{\boldsymbol{u}}^2\nabla \cdot \hat{\boldsymbol{u}} - \frac{1}{2}\hat{\boldsymbol{u}}^2\nabla \cdot \hat{\rho}\hat{\boldsymbol{u}} - \frac{1}{2}\hat{\boldsymbol{u}} \cdot \nabla\hat{p} \\ - \quad & \hat{\varepsilon}\nabla \cdot \hat{\rho}\hat{\boldsymbol{u}} + \hat{\varepsilon}\nabla \cdot \hat{\rho}\hat{\boldsymbol{u}} + \hat{\rho}\hat{\boldsymbol{u}}^2\nabla \cdot \hat{\boldsymbol{u}} + \hat{\varepsilon}_t\hat{\rho} + \frac{1}{2}\hat{\boldsymbol{u}}^2\nabla \cdot \hat{\rho}\hat{\boldsymbol{u}} \\ + \quad & \hat{\rho}\hat{\boldsymbol{u}} \cdot \nabla\hat{\varepsilon} + \hat{\boldsymbol{u}} \cdot \nabla\hat{p} + \hat{p}\nabla \cdot \hat{\boldsymbol{u}} = 0 \qquad (2.14) \\ \Rightarrow \quad & \hat{\rho}(\hat{\varepsilon}_t + \hat{\boldsymbol{u}} \cdot \nabla\hat{\varepsilon}) + \hat{p}\nabla \cdot \hat{\boldsymbol{u}} = 0. \qquad (2.15) \end{aligned}$$

Therefore the equation for specific internal energy written in terms of the Eulerian coordinates is

$$\hat{\varepsilon}_t + \hat{\boldsymbol{u}} \cdot \nabla\hat{\varepsilon} + \frac{\hat{p}}{\hat{\rho}}\nabla \cdot \hat{\boldsymbol{u}} = 0. \tag{2.16}$$

The conservation laws written in Eulerian coordinates are related to the conservation laws in Lagrangian coordinates through the material derivative, we shall call these expressions the Hybrid equations.

We now state the derivation of the Hybrid equations in order to distinguish between the Hybrid equations, which are commonly stated in the literature, and the full Lagrangian form, which we wish to work with in this thesis.

## 2.4 Material Derivative

Consider the evolution of a fluid variable when the velocity field is the flow velocity. For a certain volume of fluid moving along a pathline that follows the velocity of the fluid, the material derivative will describe the evolution of the density of the volume of fluid in time.

Consider $\hat{f} : \mathbb{R}^d \times \mathbb{R} \to \mathbb{R}$ to be a function of space and time coordinates $(\hat{\boldsymbol{x}}, t)$ for some physical property, i.e. pressure, density, energy etc.. Then the corresponding function of $f(\boldsymbol{\xi}, t)$, remembering $\boldsymbol{\xi}$ is the material co-ordinate and $f : \mathbb{R}^d \times \mathbb{R} \to \mathbb{R}$, is defined by

$$f(\boldsymbol{\xi}, t) = \hat{f}(\hat{\boldsymbol{x}}, t) = \hat{f}(\boldsymbol{x}(\boldsymbol{\xi}, t), t).$$

The derivative of $f$ with respect to $t$ following a particle path is known as the material time derivative denoted by $\frac{D}{Dt}$. By the chain rule where $\nabla_{\hat{\boldsymbol{x}}} = \nabla_{\boldsymbol{x}} = \nabla$ here, we have

$$\begin{aligned}
\frac{Df(\boldsymbol{\xi}, t)}{Dt} &= \frac{D\hat{f}(\boldsymbol{x}(\boldsymbol{\xi}, t), t)}{Dt} \\
&= \frac{\partial \hat{f}(\boldsymbol{x}(\boldsymbol{\xi}, t), t)}{\partial t} + \frac{\partial \boldsymbol{x}(\boldsymbol{\xi}, t)}{\partial t} \cdot \nabla \hat{f}(\boldsymbol{x}(\boldsymbol{\xi}, t), t) \\
&= \frac{\partial \hat{f}(\boldsymbol{x}(\boldsymbol{\xi}, t), t)}{\partial t} + \hat{\boldsymbol{u}}(\boldsymbol{x}(\boldsymbol{\xi}, t), t) \cdot \nabla \hat{f}(\boldsymbol{x}(\boldsymbol{\xi}, t), t) \qquad \text{using eqn. (2.4)} \\
&= \frac{\partial \hat{f}(\hat{\boldsymbol{x}}, t)}{\partial t} + \hat{\boldsymbol{u}}(\hat{\boldsymbol{x}}, t) \cdot \nabla \hat{f}(\hat{\boldsymbol{x}}, t) \qquad \text{using eqn. (2.1).} \qquad (2.17)
\end{aligned}$$

So we have the material time derivative of any physical property in $\boldsymbol{\xi}$ co-ordinates written in terms of spatial derivative in $\hat{\boldsymbol{x}}$ co-ordinates as

$$\frac{Df(\boldsymbol{\xi}, t)}{Dt} = \frac{\partial \hat{f}(\hat{\boldsymbol{x}}, t)}{\partial t} + \hat{\boldsymbol{u}}(\hat{\boldsymbol{x}}, t) \cdot \nabla \hat{f}(\hat{\boldsymbol{x}}, t). \qquad (2.18)$$

## 2.5 Hybrid Equations

The equations written in the material derivative form provides a link between the Eulerian and Lagrangian descriptions which we demonstrate next.

### 2.5.1 Mass

From eqn.(2.6) we have the conservation of mass in an Eulerian reference frame,

$$\hat{\rho}_t + \nabla \cdot \hat{\rho} \hat{\boldsymbol{u}} = 0.$$

Now apply eqn. (2.12) to obtain

$$\hat{\rho}_t + \hat{\rho} \nabla \cdot \hat{\boldsymbol{u}} + \hat{\boldsymbol{u}} \cdot \nabla \hat{\rho} = 0. \qquad (2.19)$$

On comparison with eqn. (2.18) this becomes a hybrid equation of the form

$$\frac{D\rho}{Dt} = -\hat{\rho}\nabla \cdot \hat{\boldsymbol{u}}, \tag{2.20}$$

where the time derivative of $\rho$ is in terms of Lagrangian co-ordinates i.e. $\rho = \rho(\boldsymbol{\xi}, t)$ and the right hand side is in terms of Eulerian co-ordinates.
Next the conservation of momentum equation is expressed in terms of the material derivative.

### 2.5.2 Momentum

From eqn. (2.8) we have

$$(\hat{\rho}\hat{\boldsymbol{u}})_t + \nabla \cdot \hat{\rho}\hat{\boldsymbol{u}}^2 + \nabla\hat{p} = 0.$$

We apply eqn. (2.12) and also use the following identity

$$\nabla \cdot (\rho\boldsymbol{u}\boldsymbol{u}) = \rho\boldsymbol{u}(\nabla \cdot \boldsymbol{u}) + (\boldsymbol{u} \cdot \nabla)\rho\boldsymbol{u} \tag{2.21}$$

to obtain

$$\hat{\boldsymbol{u}}(\hat{\rho}_t + \nabla \cdot \hat{\rho}\hat{\boldsymbol{u}}) + \hat{\rho}(\hat{\boldsymbol{u}}_t + \hat{\boldsymbol{u}}\nabla \cdot \hat{\boldsymbol{u}}) + \nabla\hat{p} = 0.$$

Noticing that the first bracketed term is 0 from the conservation of mass eqn. (2.6), the above can be simplified into the hybrid equation as

$$\hat{\rho}\frac{D\boldsymbol{u}}{Dt} = -\nabla\hat{p} \tag{2.22}$$

where the time derivative of $\boldsymbol{u}$ is in terms of Lagrangian co-ordinates i.e. $\boldsymbol{u} = \boldsymbol{u}(\boldsymbol{\xi}, t)$ and the right hand side is in terms of Eulerian co-ordinates.
Finally the conservation of specific internal energy is expressed in terms of the material derivative.

### 2.5.3 Specific Internal Energy

From eqn. (2.16) we have

$$\hat{\varepsilon}_t + \hat{\boldsymbol{u}} \cdot \nabla\hat{\varepsilon} + \frac{\hat{p}}{\hat{\rho}}\nabla \cdot \hat{\boldsymbol{u}} = 0.$$

Notice that on comparison with eqn. (2.18) the above becomes

$$\hat{\rho}\frac{D\varepsilon}{Dt} = -\hat{p}\nabla\cdot\hat{\boldsymbol{u}} \tag{2.23}$$

where the time derivative of $\varepsilon$ is in terms of Lagrangian co-ordinates i.e. $\varepsilon = \varepsilon(\boldsymbol{\xi}, t)$.
The equations in their hybrid form are often what is seen in the literature. However, since we are only interested in the Lagrangian phase, we express the equations entirely in terms of Lagrangian co-ordinates.

## 2.6 Lagrangian Form

To link between the total derivative in the hybrid equations we observe that

$$\frac{D}{Dt} \equiv \frac{\partial}{\partial t}.$$

To transform from Eulerian coordinates to Lagrangian coordinates we have the instantaneous mapping

$$\hat{\boldsymbol{x}} = \boldsymbol{x}(\boldsymbol{\xi}) \to \boldsymbol{\xi}.$$

From this mapping we have

$$d\hat{V} \to |\boldsymbol{J}|dV, \tag{2.24}$$

where $d\hat{V}$ is the control volume in terms of $\hat{\boldsymbol{x}}$, $dV$ is the control volume in terms of $\boldsymbol{\xi}$, and $|\boldsymbol{J}|$ is the determinant of the Jacobian which maps between the Eulerian and Lagrangian coordinates. Additionally, from [13] and [14], we have

$$\nabla_{\hat{\boldsymbol{x}}} \to \boldsymbol{J}^{-1}\nabla_{\boldsymbol{\xi}} = \frac{\boldsymbol{C}}{|\boldsymbol{J}|}\nabla_{\boldsymbol{\xi}}, \tag{2.25}$$

as a result of the chain rule, where $\boldsymbol{C}$ is the adjoint matrix obtained from the inverse of the Jacobian. This has been separated out for ease of cancellations when transforming the equations, as demonstrated later in this work.
All flow variables are now written in terms of Lagrangian co-ordinates, where $\rho = \rho(\boldsymbol{\xi}, t)$, $p = p(\boldsymbol{\xi}, t)$, $\boldsymbol{u} = \boldsymbol{u}(\boldsymbol{\xi}, t)$ and $\varepsilon = \varepsilon(\boldsymbol{\xi}, t)$.

### 2.6.1 Mass

Under the conservation of mass assumption, for $V$ moving with the fluid velocity, from [40] we have

$$\begin{aligned} 0 &= \int_V \frac{D(\rho|\boldsymbol{J}|)}{Dt} dV \\ &\equiv \int_V \frac{\partial(\rho|\boldsymbol{J}|)}{\partial t} dV. \end{aligned}$$

Since $V$ is arbitrary

$$\begin{aligned} \frac{\partial(\rho|\boldsymbol{J}|)}{\partial t} &= 0 \\ \Rightarrow \frac{\partial m}{\partial t} &= 0, \end{aligned} \tag{2.26}$$

where $m = \rho|\boldsymbol{J}|$. In what follows the mass is expressed in Lagrangian co-ordinates, where it is verified in [60] that $m = m(\boldsymbol{\xi}) = \rho|\boldsymbol{J}|$.

### 2.6.2 Momentum

For the momentum equation we integrate the hybrid form of eqn. (2.22),

$$\int_{\hat{V}} \hat{\rho} \frac{D\boldsymbol{u}}{Dt} d\hat{V} = -\int_{\hat{V}} \nabla_{\hat{\boldsymbol{x}}} \hat{p} d\hat{V}.$$

Under the change of variables we obtain

$$\int_V \rho \frac{\partial \boldsymbol{u}}{dt} |\boldsymbol{J}| dV = -\int_V \frac{\boldsymbol{C}}{|\boldsymbol{J}|} \nabla_{\boldsymbol{\xi}} p |\boldsymbol{J}| dV.$$

Since the limits are arbitrary we obtain

$$\rho|\boldsymbol{J}| \frac{\partial \boldsymbol{u}}{\partial t} = -\boldsymbol{C} \nabla_{\boldsymbol{\xi}} p. \tag{2.27}$$

### 2.6.3 Specific Internal Energy

For the specific internal energy equation we integrate the hybrid form of eqn. (2.23),

$$\int_{\hat{V}} \hat{\rho}\frac{D\varepsilon}{Dt}d\hat{V} = -\int_{\hat{V}} \hat{p}\nabla_{\hat{\boldsymbol{x}}} \cdot \hat{\boldsymbol{u}}d\hat{V}.$$

Under the change of variables we obtain

$$\int_{V} \rho\frac{\partial\varepsilon}{\partial t}|\boldsymbol{J}|dV = -\int_{V} p\frac{\boldsymbol{C}}{|\boldsymbol{J}|}\nabla_{\boldsymbol{\xi}} \cdot \boldsymbol{u}|\boldsymbol{J}|dV.$$

Since the limits are arbitrary we obtain

$$\rho|\boldsymbol{J}|\frac{\partial\varepsilon}{\partial t} = -p\boldsymbol{C}\nabla_{\boldsymbol{\xi}} \cdot \boldsymbol{u}. \tag{2.28}$$

## 2.7 Systems of Equations

The overall aim of the thesis is to gain an accurate and efficient numerical method for the Euler Equations of gas dynamics. We begin our investigation by looking at a model problem involving a system of two equations, namely the Isothermal Equations, which is a simplified version of the Euler Equations. This system of equations is useful since the algebra is much simpler whilst maintaining the main features of a hyperbolic system. We investigate the Isothermal Equations in 1D initially, extend the developments to the Euler Equations in 1D, and finally develop the method for the 2D Euler Equations.

### 2.7.1 Isothermal Equations

An isothermal process is used as a model where a tube of gas is considered which is in contact with a heat reservoir. The changes in the gas occur slowly enough for the temperature of the gas to be regulated between the gas and the heat reservoir continually by heat exchange. This ensures that the temperature of the gas remains constant. Isothermal processes occur frequently in living cells. Phase changes such, as evaporation and melting, are also isothermal processes.

The ideal gas law from [52] is

$$p = R\rho T, \tag{2.29}$$

where $p$ is the pressure of the gas, $R$ is the ideal gas constant and $T$ is the temperature. For the isothermal process, $T$ is constant, therefore the ideal gas law eqn. (2.29) reduces to

$$p = a^2\rho, \tag{2.30}$$

$a$ is the sound speed.
Due to the heat transfer between the wall of the filled tube and the heat reservoir, energy is not conserved in this system. However, mass and momentum satisfy conservation laws. Along with the equation of state, eqn. (2.30), the Isothermal equations written in a Lagrangian reference frame are:

- From eqn. (2.4) we have
$$\frac{\partial \boldsymbol{x}}{\partial t} = \boldsymbol{u}.$$
- The conservation of mass from eqn. (2.26) is written as
$$\begin{aligned} \frac{\partial m}{\partial t} &= 0 \\ \Rightarrow m &= m(\boldsymbol{\xi}). \end{aligned} \tag{2.31}$$
- Finally the momentum equation from eqn. (2.27) is written as
$$\begin{aligned} \rho|\boldsymbol{J}|\frac{\partial \boldsymbol{u}}{\partial t} &= -\boldsymbol{C}\nabla p \\ \Rightarrow \frac{\partial \boldsymbol{u}}{\partial t} &= -\frac{\boldsymbol{C}\nabla p}{m(\boldsymbol{\xi})}, \end{aligned} \tag{2.32}$$
where we use eqn. (2.26) and $\nabla = \nabla_{\boldsymbol{\xi}}$.

### 2.7.2 Euler Equations

The Euler equations of gas dynamics, which are simplifications of the compressible Navier Stokes equations of fluid dynamics with the viscosity of the fluid neglected, describe compressible inviscid hydrodynamics. They can be used to model many physical phenomena, such as gas flows and blast waves. It is assumed that the gas is in thermodynamic and chemical equilibrium, where the internal energy is a function of density $\rho$ and pressure $p$. In addition to the Euler equations, an equation of state is specified, which in this case relates specific internal energy and density. The ideal gas equation of state is

$$p = (\gamma - 1)\rho\varepsilon,$$

where $\gamma$ can be modified for different fluids and $\varepsilon$ is specific internal energy. The Euler equations written in a Lagrangian reference frame are,

- From eqn. (2.4) we have
$$\frac{\partial \boldsymbol{x}}{\partial t} = \boldsymbol{u}. \tag{2.33}$$
- The conservation of mass from eqn. (2.26) is written as
$$\begin{aligned} \frac{\partial m}{\partial t} &= 0 \\ \Rightarrow m &= m(\boldsymbol{\xi}). \end{aligned} \tag{2.34}$$
- The momentum equation from eqn. (2.27) is written as
$$\rho|\boldsymbol{J}|\frac{\partial \boldsymbol{u}}{\partial t} = -\boldsymbol{C}\nabla p$$
i.e.
$$\frac{\partial \boldsymbol{u}}{\partial t} = -\frac{\boldsymbol{C}}{m(\boldsymbol{\xi})}\nabla p. \tag{2.35}$$
- Finally the specific internal energy equation from eqn. (2.27) is written as
$$\rho|\boldsymbol{J}|\frac{\partial \varepsilon}{\partial t} = -p\boldsymbol{C}\nabla \cdot \boldsymbol{u}$$
i.e.
$$\frac{\partial \varepsilon}{\partial t} = -\frac{p\boldsymbol{C}\nabla}{m(\boldsymbol{\xi})} \cdot \boldsymbol{u}, \tag{2.36}$$
where $\nabla = \nabla_{\boldsymbol{\xi}}$.

Now the equations are expressed in terms of Lagrangian co-ordinates, we can look at the numerical methods to solve these systems.

## 2.8 Hydrocodes

Due to the complex nature of the systems above, it is not possible to gain an analytic solution for most problems. Computational methods are employed to obtain approximate solutions. They are cheaper and quicker than experiments and can be used on a wide variety of geometries and parameters. When numerical methods are applied to fluid flow problems, the area is known as computational fluid dynamics (CFD). These problems are solved on computers, often supercomputers, which has led to the ongoing development of the numerical models, known as hydrocodes, to improve accuracy and efficiency. The results from the hydrocode must be accurate and stable for meaningful solutions. Once the accuracy and stability of the hydrocode has been tested against known solutions, it can be used for more complex numerical experiments without known solutions.

Hydrocodes are typically large computer programs, which in their purest form were developed for fluids only i.e. no material strength. However, the successful development of hydrocodes has led to the inclusion of strength, equation of state for solids etc.

Hydrocodes have a range of applications in fluid modeling such as flyer plates, burn waves through high explosives, stars, cratering from asteroids etc. Most importantly for this thesis, hydrocodes can be used to simulate shocks [42].

Several hydrocodes exist which include CAVEAT (seen in [1]), DYNA (from [29]), SALE (from [4]) and SALE-3D (see [5]). CORVUS (from [10]) uses a staggered mesh and a Lagrangian reference frame. Temporal derivatives are approximated with a staggered finite difference time step, where spatial derivatives are evaluated using a finite element Galerkin approach. Many aspects of CORVUS form the basis to this work.

## 2.9 Grid Generation

In order to obtain a numerical solution on a physical region, we require the region to be discretised into a set of points or cells and the time window to be similarly discretised.

### 2.9.1 Spatial Grid

In this thesis, non-uniform quadrilateral elements in the physical space $(x, y)$ are mapped onto the uniform grid in computational space $(\xi, \eta)$. This is done to make Taylor expansions easier to manipulate when we use them for truncation errors later on.

The neighbouring points are identified by taking a unit step of the spatial numbering system $(i, j)$ in the $(\xi, \eta)$ direction respectively. The co-ordinate system $(\xi, \eta)$ is assumed to be aligned with the physical boundaries.
For the physical space, the coordinates are

$$(x, y) \in [x_0, x_N] \times [y_0, y_M], \qquad N, M \in \mathbb{Z},$$

where $N$ and $M$ are the number of points in the $x$ and $y$ direction respectively.
We map from the non-uniform physical space to the square in transformed computational space,

$$(x, y) \to (\xi, \eta),$$

where for the computational space we have

$$(\xi, \eta) \in [-1, 1] \times [-1, 1].$$

This gives the uniform grid spacing of $h \equiv \Delta\xi = \frac{2}{N}$ and $l \equiv \Delta\eta = \frac{2}{M}$.
Each point in $(x, y)$ space is mapped to a single point in the square in $(\xi, \eta)$ space (see [88]), where the discrete points on the computational mesh $(\xi_i, \eta_j)$ are defined by

$$\xi_i = hi, \qquad i = 0, \cdots, N$$

$$\eta_j = lj, \qquad j = 0, \cdots, M.$$

To implement the mapping the equations are transformed to the computational variables, thus we will have derivatives in terms of $(\xi, \eta)$ coordinates, as seen in Section 2.6. These transformed equations are then solved numerically on the computational grid.
The scenario also holds for one dimensional problems.

### 2.9.2 Temporal Grid

For the temporal derivatives, the timestep is

$$k \equiv \Delta t.$$

The choice of the timestep is discussed in Appendix A.2. Discrete points in time are defined by

$$t^n = nk, \qquad n = 0, 1, 2 \cdots$$

Next, we describe how the variables of the problem are distributed on the computational mesh.

### 2.9.3 Computational Grid

We apply a staggered grid to the transformed Lagrangian equations. The velocity, position and acceleration are located at the nodes of the elements whereas the density, pressure, internal energy, element mass and volume are stored at the centres of the elements. The CFD community often uses a staggered mesh, since material strength can be discretised easily. The nodal positions along with velocities on a staggered mesh allow for simpler interface tracking.

In addition, when dealing with spatial derivatives on an unstaggered mesh, we observe the phenomena of odd-even decoupling. This is overcome by using a staggered mesh.

The staggered grid method was first developed by F. Harlow and J. Welch in 1965 [31], then known as the Marker and Cell method. One of the advantages they showed was that staggered grids improve stability and accuracy compared to the unstaggered grid which causes 'chequerboard' effects. The odd even decoupling is demonstrated in Appendix B.

In 1D an element is staggered as in Figure 2.1. In 2D, a staggered element looks like Figure 2.2.

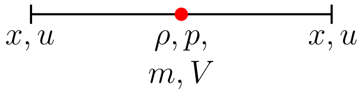


Figure 2.1: 1D Staggered Grid.

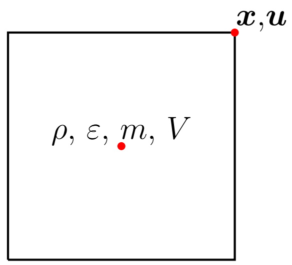

$\boldsymbol{x}$,$\boldsymbol{u}$
$\rho$, $\varepsilon$, $m$, $V$

Figure 2.2: 2D Staggered Element in Space.

## 2.10 Numerical Schemes

There is a wide range of numerical schemes available which can be applied to a variety of fluid flow problems. They can be divided into Finite Difference Methods (FDM), where the solutions are obtained as point values, see [72], Finite Volume Methods (FVM), where the solution is an average of the cells see, [53], and Finite Element Methods (FEM), where the solution is expressed as a linear combination of basis functions, see [100].

For conservation laws these schemes can further be divided into shock capturing schemes, where any shocks in the solution are dealt with automatically, the correct shock-speed is obtained, see [26, 32, 33, 76, 81], and shock fitting schemes where the shocks are detected and the jump condition is imposed to determine the shock speed, see [34].

An excellent overview of different schemes, which are applied to a wide range of fluid flow test problems in 1D and 2D is given in [57]. It is concluded in [57] that some schemes work better than others for specific problems however, there is not a particular scheme which works better universally than any other.

There exists a wide range of schemes applied to staggered grids, as seen in the Marker and Cell method in [98, 97, 69], initially developed in [31]. The method was extended to finite differences in [67]. An overview of some staggered grid methods is given in [63]. In this work a finite element approach is applied to the staggered variables in space, which reduces to a FDM.

Additionally, there are several numerical schemes used to express temporal derivatives. In [78], there is a half timestep applied to the velocity and all other variables, which are then updated to the full timestep from this, whereas in [46] a full timestep predictor step is taken, these values are then corrected to the same full timestep.

The basis of the approach in this thesis comes from [10], which describes the development towards the hydrocode CORVUS. CORVUS uses a two step timestepping scheme where a predicted step is taken at a half timestep and then corrected at the full timestep, which has similarities to the MacCormack scheme described in [52]. The spatial derivatives are evaluated using a Galerkin Finite Element approach.

## 2.11 Shock Waves

As mentioned above hydrocodes can be used to simulate shocks. Consider pressure to be a linear smooth curve connecting three points A, B and C. The wave velocity increases as the pressure increases. Therefore the velocity of point B is greater than the velocity of point C, and the wave velocity of point A is greater than the wave velocity of point B. Hence the curve gets steeper and steeper, until it becomes a straight vertical line, known as a shock wave, see [18].

Due to the discontinuous nature of the material where there is a non-smooth transition between the material at the front of the shock and to the back, it is difficult to deal with this feature numerically. In this work, we incorporate artificial viscosity in order to deal with shock waves which otherwise would lead to oscillations. To improve accuracy further, we look at refining the mesh in the vicinity of shocks.

## 2.12 Artificial Viscosity

As already mentioned in Section 2.11, shocks occur in fluid flow problems and prove difficult to model numerically. Shocks on a staggered grid are traditionally treated by using artificial viscosity, whereas cell centred grids typically use Riemann solvers and more recently Godunov type methods, see [10, 59]. The comparison between staggered grid artificial viscosity methods and cell centred Riemann methods is discussed in [68]. We employ a staggered grid with artificial viscosity in this thesis, since we wish to follow the methodology in [64]. Furthermore, as the methodology is more developed, the results will therefore be applicable to a wider audience.

The first successful numerical approach to dealing with shocks was given by von Neumann [95] in the 1950's. The idea was to smear the shock over several zones in order to decrease the oscillations around the shock. The method treats shocks automatically, where and when they occur. Von Neumann makes use of how viscosity and heat conduction are dissipative in shocks. Viscosity smears out shocks by replacing the discontinuity with a continuous and rapidly varying solution. Hence, the idea is to introduce 'artificial' viscosity into the problem. The artificial viscosity increases the mesh distance around the discontinuity but allows it to remain comparable to what we already have. This allows for numerical techniques to be applied to the entire domain and improves the stability of the solution.

In this work, we use a combination of the quadratic artificial viscosity term proposed by von Neumann [95], along with a linear term from [51]. Christensen's limiting procedure in [16] is used, as seen in [64], which is not computationally expensive and reduces excessive dissipation. The full form of the artificial viscosity used in this thesis is described in Appendix A.1.

To further increase accuracy around shocks and other flow features, we look at an appropriate refinement criteria to resolve the mesh around such features.

## 2.13 Refinement

It is desirable to gain an accurate numerical solution for fluid flow problems which is well resolved on the computational mesh throughout the evolution.

An obvious approach to resolving the solution is to refine the entire domain. However, this results in a large number of nodes and elements and is extremely computationally expensive.

A way to overcome the expense is to resolve only around the areas of interest hence, keeping the number of nodes and elements to a minimum. In a compressible fluid flow problem, the areas of interest are in the vicinity of shocks.

The two main types of refinement methods that can be used to resolve the solution around areas of interest are $r$-refinement and $h$-refinement. As mentioned in the Introduction we do not consider $p$-refinement in this work.

For $r$-refinement the number of nodes remains fixed and are moved towards the areas of interest. A disadvantage of this approach is that there will be under-resolution away from some areas and an under-resolved solution could lead to ambiguities.

For $h$-refinement the elements are subdivided in the areas of interest, leading to an increased number of nodes and elements.

### 2.13.1 $r$-refinement

The ALE method developed by Hirt *et al.* [39] combines the advantages of Lagrangian and Eulerian formulations, demonstrated in [12]. During the Lagrangian step, the mesh moves with the fluid flow, but may become tangled in areas of shocks or high vorticity. A relaxation technique may then be used to preserve a good quality mesh throughout the computation, the solution then being remapped to the new mesh. Provided that the Lagrangian nature of the mesh is preserved in the re-map phase, ALE behaves as an $r$-refinement method, i.e. moving the nodes to areas of interest. This can cause disadvantages, such as under-resolution away from features of interest. Therefore, methods to combine ALE with $h$-refinement have been developed in [64] .

### 2.13.2 $h$-refinement

AMR is a type of $h$-refinement, see [11]. There are two practical approaches to $h$-refinement. One approach is where elements, which are required to be refined, are clustered together into separate grids and then refined together, creating a hierarchy of refinement levels. The alternative method is cell by cell refinement, where individual, or small groups, of elements are refined. Once elements are subdivided, they are inserted into the mesh. From this the solution is obtained from the dynamic unstructured mesh. This approach is less computationally expensive than solving at each level on the hierarchy, but requires extra machinery to deal with the unstructured mesh.

Cell by cell refinement can also include anisotropic refinement, where the element is divided in only one direction. Since many of the features of interest only change in one direction, anisotropic refinement reduces wasted refinement in the other directions. Work has been carried out on combining ALE with cell by cell refinement, anisotropically and isotropically in [64].

Density gradients are used in [64] to trigger when and where the refinement should occur. However, in this work, we are interested in developing alternative refinement criterion. In [64], the refinement occurs at the end of the Lagrangian step. Due to this, the focus of this work is on the Lagrangian step.

## 2.14 Refinement Monitor

The main aim of this thesis is to develop an accurate and efficient method to determine where and when to refine or derefine elements, leading to the ultimate aim of generating an automatic technique.

There are several methods used as refinement sensors, with much debate over the variables

which are best suited in detecting certain flow features.
Pressure or density is used by Anderson *et al* in [6], where the sensor is the normalised second difference of these variables.
Aftosmis [2], suggests the necessity of refining both smooth features and shocks for convergence towards the correct solution. It is suggested in [2] that different flow features should be detected by different sensors, due to the relationship between shocks and element size. For shocks, Aftosmis suggests using an undivided second difference of pressure. This is normalised by a weighted local average of the pressure to deal with weak and strong shocks. Smooth features are then detected using undivided differences of total velocity or density.
Undivided differences are also used for shocks by Kallinderis [43], where it is suggested pressure or velocity should be used, since density leads to too much refinement.
Further combinations of refinement sensors are used by Khokhlov [49] and van der Vegt [92].
Three categories, which most refinement criteria fall under, are described by Keats [48];

1. Gradient sensors - where the absolute value of the difference in density between a centre element and neighbouring elements, based on work by Ripley [74] and Lien [56]. This is simple but is not sensitive to changes in the second derivative.

2. Higher order sensors - the ratio of the second and first order terms of the Taylor expansion is used as a sensor, implemented by Sun in [85]. Keats [48] suggests this is best used for Isotropic refinement as it is noisy when applied to anisotropic refinement.

3. Estimation of optimal cell size - Ham [30], uses first and second order terms from Taylor series. However, the scheme reduces to first order around shocks when using second order terms. Therefore, Keats [48] suggests using first order terms from the Taylor series.

Dannenhoffer [20] also suggests not using second differences as it leads to inappropriate refinement. It is also suggested in [20], that density is the best flow variable to use with respect to accuracy, computational time and all flow features of interest being detected.
Previous work by Morrell [64], uses an undivided first order difference due to the simplicity. It also does not suffer from excessive noise like second differences, as stated in [48] and [20]. The flow variable used by Morrell in [64] is density since it is cell centred, unlike velocity, and detects contacts as well as shocks. The density change between a centred element and its neighbours is used. This is compared to a refinement or derefinement tolerance, which will keep the number of elements to a minimum, whilst still refining around features of

interest. The user inputs these tolerances, creating a drawback to this method. To overcome this problem slightly, Morrell manually uses the cumulative frequency of density change at different times per problem to determine the most accurate tolerance, as suggested in [20].

## 2.15 Error Estimations

We are ultimately interested in gaining an automatic method to determine where and when to refine the grid. Refinement methods require an appropriate tolerance to be defined to compare against refinement criteria.

The first step is to recognise what information we have available from the solution and use this to define a tolerance. This leads to the investigation of the use of errors in the numerical solution as a refinement criteria, assuming that the numerical scheme will be less accurate around the areas we wish to refine.

A variety of techniques to estimate errors is discussed in [77]. These techniques include using higher order methods and estimates based on the residual of the numerical approximation. The bound on an error estimate usually relies on a function of various derivatives of the exact solution. As discussed in [77], this function is difficult to estimate. Therefore, the term is either eliminated by using the Richardson Extrapolation technique or the Residual Method, which uses an approximate to the exact solution.

In this thesis, we aim to estimate the term involving derivatives of the exact solution by using the relationship between two numerical schemes where the approximation to the error is achieved using a method similar to Milne's Device for Ordinary Differential Equations (ODEs), see [27] and [80].

## 2.16 Choosing a Method to Estimate the Error

It is often impossible to determine an exact solution to CFD problems. Hence, the need for numerical schemes. Therefore, a true error will be impossible to determine. Here, we investigate methods to estimate the true error.

There is a considerable amount of work on error estimates, using a variety of techniques. One area of research is based on using the Hessian matrix to estimate an error, see [86]. Another approach is to use the adjoint for error estimates as discussed in [35, 55, 54, 66]. It is stated in [66, 93, 23, 44] that the adjoint based method is good for steady CFD problems, however for time-dependent problems, as we have here, it is computationally expensive since it is required to solve the adjoint problem backwards. Aftosmis and Berger [3], also point out the issue with the adjoint method being computationally expensive, especially when the Jacobian is not already computed and is only introduced for the sole purpose of adaption.

Energy norms are also used as error estimates as seen in [47, 71, 99]. As originally proposed by Babuška and Rheinboldt in [8, 9], local residuals would be obtained and the error estimated by the energy norm. Apel *et al*, see [7], suggest that error estimates based on energy norms can be used when isotropic or anisotropic refinement should be triggered.

Even with the application of energy norms the question still remains how can we determine a residual to the numerical scheme. It is suggested in [47], to use local truncation errors (LTEs). The LTE is obtained from the weak form of the PDE where localised quadratic B-Splines are used as test functions. The LTE is converted into the local infinity norm, $L^{\infty}$, following the convergence theory by Tadmor in [87], and then compared to the LTE. From [47], the LTE and local infinity norm, $L^{\infty}$, exhibit similar behaviour. The outcome of [47] suggests using the LTE for adaptive methods.

Berger and Colella discuss an alternative method to determine the LTE in [11], where test functions not B-splines are used, where the ultimate aim is automatic refinement. The method is based on Richardson extrapolation. The method makes use of an original grid (at the beginning of a timestep), which is referred to as the fine grid and is then projected onto a coarser grid. This is achieved by coarsening the fine grid by a factor two in all spatial directions. The original fine grid is advanced in time by two time steps and the coarse grid is advanced in time by taking one large timestep. An estimated LTE at time $t$ is obtained, using the difference of the scheme advanced on the fine grid between the scheme advanced on the coarse grid. The exact form of the leading terms in the truncation error are not important in [11], only the order is required. The difference between coarse and fine grid solutions is obtained from the computation which is proportional to the estimated error. Refinement is then triggered if the difference values exceed a tolerance, which is set as 0.02 in [11].

In hierarchical AMR every level is refined and solved upon, storing more data and solving for more cells than an unstructured cell by cell approach, but this can be beneficial to implement Richardson extrapolation, see [11]. However, since we implement a cell by cell approach, we would require significant data structure changes therefore, it would be expensive to use Richardson extrapolation. Since every level must be stored it is wasteful to refine, solve and store the coarser grids, if it is only used to obtain an error estimate. The hierarchy of levels can also cause conservation problems at mesh boundaries. The theoretical work may be simple, since only the order of the scheme is important. However, this only leads to an estimate in terms of the order of the error.

As discussed in [77], Richardson Extrapolation is used to eliminate the derivatives to the exact solution in the leading term of the error, using the ratio of the coarse and fine grids. This method produces a global error and can be applied to any variable.

Residual based methods are also discussed in [77]. These methods are related to the truncation error, i.e. the difference between the discrete solution and the mathematical model which is being discretised. Two approaches in [77], which use the Residual method include; using an exact solution to substitute into the discrete model or substituting the discrete solution into the continuous mathematical model.

In this thesis we develop a method which is Residual based. We do substitute the exact solution into the discrete form of the mathematical model. However, we do not estimate the exact solution as is done in [77]. The aim here is to estimate the leading terms in which the derivative to the exact solution are present. We do not use a fine and coarse grid as is used in the Richardson Extrapolation, instead we use two schemes which are evolved from the same initial mesh. Thus, only one timestep is required, unlike the Richardson Extrapolation in [77]. We also do not need an estimate to the exact solution from a fine grid as is described in [77]. We use the relationship between the two schemes to solve for the leading term of the truncation error, which contains the derivatives. The requirement is that the schemes are related by their order. By doing this, the derivative terms are estimated from the computable differences between schemes. It is stated in [77] that the Richardson Extrapolation method will break down around discontinuities. Since we do not neglect the function containing derivatives, we find our approach does not break down. This is verified in Appendix C, where the wave equation is used on a curve of varying gradients to demonstrate this approach holds for steep solutions.

## 2.17 Milne's Approach

Milne's device is a well known method used to estimate local errors for multistep schemes see [36, 80, 25]. Milne's device estimates the local truncation error of a predictor-corrector scheme in ordinary differential equations, by looking at the error in one step, where both the predictor and corrector are of the same order. It is widely used to control the timestep of numerical schemes for ODEs. If we use a timestep which is too large, corresponding to a large truncation error, we may encounter inaccuracies in our solution and, if the timestep is too small, this could be extremely computationally expensive, and the accuracy may not be improved at all. Therefore, the aim of Milne's device is to adapt the timestep so that the truncation error is uniform. Similarly, in the present work, we are concerned with how to adapt our spatial/temporal step size in order to gain an accurate solution around discontinuities, where the time and space steps are related through a CFL condition. An overview of Milne's device is given in [27, 80], and is summarised here.

Milne's device is implemented by comparing two approximations of the same order, as remarked in [80, 36, 25]. One of the approximations is denoted here by $\tilde{}$ and the other is not. To apply the method consider the ODE defined as

$$\frac{dx}{dt} = u$$

where $x = x(t)$ and $u = u(x, t)$. Suppose that the numerical solution in one step is quantified by $x^n$. Let the numerical solution at timestep $n + 1$ be

$$x^{n+1}. \tag{2.37}$$

Suppose now that the numerical approximation applied to the exact solution in one step is given by

$$x(t^n + k), \tag{2.38}$$

where $k$ is the time step. The difference between eqn. (2.37) and eqn. (2.38) is the local error in one step, given by

$$x(t^n + k) - x^{n+1} = k\tau_n = Dk^\sigma x^\sigma(t^n) + \mathcal{O}(k^{\sigma+1}), \tag{2.39}$$

where $\tau_n$ is the truncation error of the scheme, $\sigma$ is the order of accuracy and $D$ is the coefficient of the leading term. Similarly, for the second approximation we have

$$x(t^n + k) - \tilde{x}^{n+1} = k\mu_n = \tilde{D}k^{\tilde{\sigma}} x^{\tilde{\sigma}}(t^n) + \mathcal{O}(k^{\tilde{\sigma}+1}) \tag{2.40}$$

where $\tilde{D}$ is the coefficient of the leading term.

In order to proceed with Milne's device it is assumed that the numerical solution is equal to the approximated solution at time level $n$ , i.e. $x^n = x(t^n)$ and that $\sigma = \tilde{\sigma}$. Under these assumptions we can obtain the local error in one step by subtracting eqn. (2.40) from eqn.

(2.39), resulting in

$$\tilde{x}^{n+1} - x^{n+1} = (D - \tilde{D})k^{\sigma} x^{\sigma}(t^n) + \mathcal{O}(k^{\sigma+1}). \tag{2.41}$$

Therefore, we can obtain an estimate of the higher order derivative, which is difficult to compute, from

$$\frac{\tilde{x}^{n+1} - x^{n+1}}{(D - \tilde{D})} = k^{\sigma} x^{\sigma}(t^n) + \mathcal{O}(k^{\sigma+1}). \tag{2.42}$$

By substituting this into eqn. (2.39) we have

$$x(t^n + k) - x^{n+1} = D\frac{\tilde{x}^{n+1} - x^{n+1}}{(D - \tilde{D})} + \mathcal{O}(k^{\sigma+1}) \tag{2.43}$$

which is a computable estimate to the local error. We use a similar device in this thesis.

## 2.18 Truncation Error

Numerical algorithms make use of a finite number of steps, matching as far as possible the infinite Taylor series. The truncation error is what is left over from truncating the Taylor series in order to form the finite algorithm. The LTE is the measure of the truncation error in one step.

Consider the ODE

$$\frac{dx}{dt} = u,$$

and define the operator $L$ as

$$L(x) \;\; = \;\; \frac{dx}{dt} - u = 0. \tag{2.44}$$

Define the algorithm approximating the ODE as $L_h$ and the numerical scheme $x_i^n$ applied to the numerical solution as $L_h(x_i^n) = 0$. By applying the numerical algorithm in one step to the exact solution $x(\xi_i, t^n)$ sampled at time $t^n$, we have $L_h(x(\xi_i, t^n))$.

The local truncation error to the ODE is then defined as the difference between the numerical scheme applied to the exact solution $L_h(x(\xi_i, t^n)$, at time $t^n$, using the infinite Taylor series, and the numerical solution itself, from the finite Taylor series, in one step, i.e.

$$\tau_n = L_h(x(\xi_i, t^n)) - L_h(x_i^n). \tag{2.45}$$

This definition can be applied to estimate an error along with Milne's approach mentioned above.

## 2.19 The Requirement of Two Schemes

We can use a similar approach to Milne's device for any two schemes of the same order. We shall then be able to calculate a computable error estimate for the true error, defined as the error between the exact solution and the approximation in one step.

We use the scheme given in [64], as our main reference scheme, which we refer to as the Main Scheme throughout this thesis. It is the true error of this scheme that we are interested in estimating throughout this work. This scheme uses a staggered finite difference temporal discretisation, along with a Galerkin finite element approach to the spatial discretisation. The temporal discretisation is referred to as Scheme 1 and the spatial discretisation is referred to as Scheme A.

We require a second scheme to compare against our main scheme. We shall investigate schemes with temporal discretisations from [78, 46], referred to here generically as Scheme 2. For the second spatial discretisation we use a Petrov-Galerkin finite element approach, giving a scheme which will be referred to as Scheme B in this work.

We shall investigate temporal and spatial errors separately then combined, leading to the understanding of how temporal and spatial errors behave.

To estimate the error using the available schemes we first calculate the truncation error in each scheme, for the individual variable that we are interested in estimating. The truncation errors are then used to calculate an estimated error for the Main Scheme from [64], known as Scheme 1. This gives an expression for the estimated error of Scheme 1 in terms of the two schemes, yielding a computable value of the error.

Therefore, the approach we wish to develop will be for an estimate to the true error, not just the order of the error. The theory will be sightly more complicated than in [11], since we must determine an exact form for the local truncation error. However, we use a cell by cell approach where there is only one mesh. This mesh is solved upon and refined, the data being stored in connectivity arrays and disjoint nodes are used. This will be more cost effective than solving for every level seen in [11]. Once the exact form of the truncation error is known we can gain an estimate to the true error in the scheme using a method similar to Milne's device.

Since spatial and temporal derivatives exist in the system of equations, we investigate an error estimate by considering the truncation errors for time and space separately. We begin the development of an error estimator with the semi-discrete in time scheme.

# Chapter 3

# Semi-Discrete Temporal Schemes in 1D

As observed in the previous chapter, the equations of interest contain temporal and spatial derivatives which require approximation by numerical methods.

The numerical scheme will have an LTE associated with it which includes both time and space derivatives. Therefore, we will observe truncation errors as a result of the temporal approximations and as a result of the spatial approximations.

In order to understand the impact each derivative has on the numerical scheme, and hence the truncation error, we look at the schemes in their semi-discrete form before advancing to the fully-discrete form.

We begin our investigation by looking at two semi-discrete in time schemes, in 1D. We do this to initially understand the effect of the time derivative approximations on the LTE and to demonstrate the strategy used in this thesis.

As we have already mentioned, we shall be using a strategy similar to that of Milne's device. We estimate the true error based on a comparison of LTEs from two schemes. This requires the introduction of a second temporal scheme of the same order.

A system of two equations are studied in order to understand how the variables associated with the equations have an effect on the truncation errors. The findings are then advanced to a system of three equations, where specific internal energy is introduced.

For each system we study the temporal scheme used in [64], known as the Main Scheme, and determine its truncation error. We then introduce a second semi-discrete in time scheme, which has the same order as the Main Scheme. The LTE of this scheme is also calculated.

The two schemes are then used to determine an approximation to the true error in one step. This chapter leads to an understanding of how different variables and temporal derivatives affect the truncation error. This is used later to understand the effect of combining temporal and spatial derivatives, before advancing to the 2D case.

## 3.1 System of Two Equations

In the previous chapter we have seen how the equations written in a Lagrangian reference frame are obtained. Since, in the time dependent semi-discrete case, the spatial dependencies are not approximated, we can write the system of two equations in 1D formally as;

- Conservation of Mass

  $$\frac{dm}{dt} = 0, \tag{3.1}$$

  where the mass is expressed as

  $$m = \rho x_{\xi}, \tag{3.2}$$

  in 1D.

- Momentum Equation

  $$\frac{du}{dt} = f, \tag{3.3}$$

where $m$ is the mass which is constant in time, $u = u(t)$ is the velocity and $f = f(t)$ is the acceleration, expressed in its full form in eqn. (2.27).

We have the connection to the physical mesh through the velocity of the physical nodes on the Lagrangian mesh

$$\frac{dx}{dt} = u, \tag{3.4}$$

where $x = x(t)$ is the Lagrangian co-ordinate in terms of computational nodes in the semi-discrete form.

With this system of 1D conservation laws written in Lagrangian form, we now look at appropriate semi-discrete numerical schemes to approximate the solutions to the equations.

Motivated by the scheme expressed in [64, 10], we consider a two step temporal scheme related to the Verlet scheme.

## 3.2 Derivation of the Verlet Schemes

From Newton's equation of motion for conserved physical systems, we have

$$\frac{d^2x}{dt^2} = \frac{du}{dt} = f. \tag{3.5}$$

To numerically solve eqn. (3.5), a timestep $k > 0$ is chosen and a single point is considered at $t^n = nk$. The aim is to obtain a sequence of points $x^n$ that closely follow the points $x(t^n)$ which lie on the trajectory of the exact solution.

Verlet in [94, 28], used a central difference scheme to approximate the second derivative i.e.

$$\frac{x^{n+1} - 2x^n + x^{n-1}}{k^2} = f^n. \tag{3.6}$$

Störmer [28, 78, 94, 24] used eqn. (3.6) to obtain the current position from the previous two known positions, such that

$$x^{n+1} = 2x^n - x^{n-1} + k^2 f^n. \tag{3.7}$$

This was popularised by Verlet in 1967, see [94], and is often called the Störmer-Verlet method.

It can be shown that the Störmer-Verlet method is more accurate than the simple Taylor expansion method. Consider the local error which is obtained by substituting the exact values $x(t^n - k)$, $x(t^n)$ and $x(t^n + k)$ into eqn. (3.7) and expanding about $t^n$ for $x(t^n \pm k)$ using Taylor expansions,

$$\begin{aligned} x(t^n + k) &= x + ku + \frac{k^2 f}{2} + \frac{k^3 b}{6} + \mathcal{O}(k^4) && (3.8) \\ x(t^n - k) &= x - ku + \frac{k^2 f}{2} - \frac{k^3 b}{6} + \mathcal{O}(k^4) && (3.9) \end{aligned}$$

where $x = x(t^n)$ is the position, $u = u(t^n) = \frac{dx}{dt}$ is the velocity, $f = f(t) = \frac{d^2x}{dt^2}$ is the acceleration and $b = b(t)$ is the third derivative of $x$. Adding and re-arranging eqn. (3.8) and eqn. (3.9) we obtain

$$x(t^n + k) = 2x(t^n) - x(t^n - k) + f(t^n)k^2 + \mathcal{O}(k^4). \tag{3.10}$$

This is the exact solution substituted into the Störmer-Verlet iteration from eqn. (3.7) to order $\mathcal{O}(k^4)$. Since the first and third order terms cancel in eqn. (3.10), this makes the Störmer-Verlet iteration an order more accurate than the approximation directly from eqn. (3.8).

Velocity is not directly generated from eqn. (3.7), but it is required to determine certain physical properties. The velocity can be obtained from subtracting eqn. (3.9) from eqn. (3.8), see [78], resulting in

$$\frac{x(t^n+k)-x(t^n-k)}{2k} = u(t^n) + \mathcal{O}(k^2). \tag{3.11}$$

In terms of a central difference numerical scheme, eqn. (3.11) becomes

$$u^n = \frac{x^{n+1}-x^{n-1}}{2k} \tag{3.12}$$

The Störmer-Verlet schemes, for position eqn. (3.7) and velocity eqn. (3.12), can be used to determine two different schemes known as Position Verlet and Velocity Verlet, see [78], where each scheme makes the transformation from $\{x^n, u^n\}$ to $\{x^{n+1}, u^{n+1}\}$. In [21] the Position Verlet is favoured for larger timesteps, whereas the Velocity Verlet is considered to be better than the Position Verlet in determining a more accurate velocity. The velocity approximate in the Velocity Verlet uses old and new values of the position to update the velocity, i.e. it is implicit, whereas the Position Verlet only uses old values, i.e. it is explicit. Implicit schemes are more favourable for increased stability, while explicit schemes have more simplistic algorithms.

The crucial point here is that the Position Verlet scheme is similar to the scheme in [64], i.e. the Main Scheme, whereas the Velocity Verlet scheme can be used as our additional scheme for estimating the error. We now look at how the above results are used to obtain two different numerical schemes in time.

### 3.2.1 Position Verlet Derivation

The Position Verlet scheme is a numerical method used for time dependent ODEs, which is a second order finite difference method consisting of a predictor step followed by a corrector step.

We use the above results and a mapping from $\{x^n, u^n\}$ to $\{x^{n+1}, u^{n+1}\}$ to define the Position Verlet Scheme.

- The scheme begins with a mapping from $\{x^n, u^n\}$ to $\left\{x^{n+\frac{1}{2}}, u^{n+\frac{1}{2}}\right\}$ and uses eqn.(3.8) for the expansion of $x(t^n + \frac{k}{2})$ to first order, resulting in

$$x^{n+\frac{1}{2}} = x^n + \frac{ku^n}{2}. \tag{3.13}$$

- The velocity is now updated to the full timestep using the mapping $\left\{u^n, f^{n+\frac{1}{2}}\right\}$ to $\left\{u^{n+1}, f^{n+\frac{1}{2}}\right\}$. By subtracting $u^n$ from $u^{n+1}$ and using eqn. (3.12) we get

$$\begin{aligned} u^{n+1} - u^n &= \frac{1}{2k}(x^{n+2} - x^n - x^{n+1} + x^{n-1}) \\ \Rightarrow u^{n+1} &= u^n + \frac{k}{2}(f^n + f^{n+1}) \qquad \text{using eqn. (3.6).} \end{aligned}$$

Here the average of the force term, $f$, is expressed as $\frac{1}{2}(f^n + f^{n+1}) = f^{n+\frac{1}{2}}$. The velocity in one step is then expressed as

$$u^{n+1} = u^n + kf^{n+\frac{1}{2}}. \tag{3.14}$$

- The positions to a full timestep are then updated where the mapping goes from $\left\{x^n, u^n, f^{n+\frac{1}{2}}\right\}$ to $\left\{x^{n+1}, u^n, f^{n+\frac{1}{2}}\right\}$. The result in eqn. (3.12) is used to obtain

$$\begin{aligned} \frac{x^{n+1} - x^n}{k} &= u^{n+\frac{1}{2}} \\ \Rightarrow x^{n+1} &= x^n + ku^{n+\frac{1}{2}}. \end{aligned}$$

where the average of the velocity $u$ is expressed as

$$u^{n+\frac{1}{2}} = \frac{u^{n+1} + u^n}{2}, \tag{3.15}$$

hence

$$x^{n+1} = x^n + k\left(\frac{u^{n+1} + u^n}{2}\right). \tag{3.16}$$

From the result in eqn. (3.14) we obtain

$$x^{n+1} = x^n + ku^n + \frac{k^2}{2}f^{n+\frac{1}{2}}. \tag{3.17}$$

In [64] and [10] this is precisely the scheme used for the temporal discretisation.
Incidentally, this scheme also exhibits similarities with the MacCormack scheme, [58], due to the characteristic feature of the MacCormack scheme being outlined in eqn. (3.16), where the predicted full step velocity is used to determine a half step velocity which in turn is used to obtain the corrected full step variables. We will see the above results expressed as a scheme for the system of equations later, where it will be referred to as the Main Scheme.
Our attention now turns to choosing another second order scheme which is different from the Main Scheme.

### 3.2.2 Velocity Verlet Derivation

We now consider the Velocity Verlet scheme as a possible alternative. The approach to determining this scheme is similar to the Position Verlet however, we begin with a half time step velocity rather than a half time step position.
The Störmer-Verlet method for determining the velocity in eqn. (3.12) is approximated a timestep behind the position approximation eqn. (3.7). To circumvent this, positions and velocities are obtained at time $t^n + k$ using the same quantities at time $t^n$, i.e. we carry out the mapping $\{x^n, u^n\}$ to $\{x^{n+1}, u^{n+1}\}$ as before.

- The half step velocity is obtained by using eqn. (3.12), where

$$u^{n+\frac{1}{2}} = \frac{x^{n+1} - x^n}{k}. \tag{3.18}$$

By using eqn. (3.8) to second order for $x^{n+1}$, we obtain the approximation

$$\begin{aligned} u^{n+\frac{1}{2}} &= \frac{x^n + ku^n + \frac{k^2}{2} f^n - x^n}{k} \\ &= u^n + \frac{kf^n}{2}. \end{aligned} \tag{3.19}$$

- Eqn. (3.18) then gives

$$x^{n+1} = ku^{n+\frac{1}{2}} + x^n. \tag{3.20}$$

To demonstrate the mapping from $\{x^n, u^n\}$ to $\{x^{n+1}, u^{n+1}\}$, eqn. (3.19) is used to obtain

$$x^{n+1} = x^n + ku^n + \frac{k^2 f^n}{2}. \tag{3.21}$$

- To improve stability of the velocity an implicit method is used to determine the full step velocity. From eqn. (3.12), we have

  $$u^{n+1} = \frac{x^{n+2} - x^{n}}{2k}. \tag{3.22}$$

  Using eqn. (3.21)

  $$x^{n+2} = x^{n+1} + ku^{n+1} + \frac{k^2 f^{n+1}}{2}, \tag{3.23}$$

  and from eqn.(3.7)

  $$x^{n} = \frac{x^{n+1} + x^{n-1} - k^2 f^{n}}{2} \tag{3.24}$$

  by substituting both into eqn. (3.22), we have

  $$u^{n+1} = \frac{1}{2k}\left(\frac{x^{n+1} - x^{n-1}}{2} + ku^{n+1} + \frac{k^2}{2}(f^{n+1} + f^{n})\right). \tag{3.25}$$

  By using eqn. (3.12) and re-arranging the above, we obtain

  $$u^{n+1} = u^{n} + \frac{k}{2}(f^{n+1} + f^{n}). \tag{3.26}$$

The above two schemes are now applied to the Lagrangian system of two equations from equations (3.1)-(3.4).

## 3.3 Numerical Schemes for the System of Two Equations

The semi-discrete numerical schemes in Section 3.2.1 and Section 3.2.2 are applied to the system of equations (3.1)-(3.4).

### 3.3.1 Scheme 1

Scheme 1 follows from the Position Verlet scheme, which is the scheme described in [64], i.e. the Main Scheme. The predictor and corrector steps are as follows.

### Predictor

- Eqn. (3.4) can be approximated using the result in eqn. (3.13). This is used to obtain the nodal positions at the half timestep

$$x^{n+\frac{1}{2}} = x^n + \frac{k}{2}u^n.$$

- This result can now be used to gain an approximation to the density at $t^n + \frac{k}{2}$ by using the conservation of mass from eqn. (3.1),

$$m^{n+\frac{1}{2}} = m^n, \tag{3.27}$$

  where in 1D the mass is expressed as

$$m = \rho x_\xi, \tag{3.28}$$

  density is updated to

$$\rho^{n+\frac{1}{2}} = \frac{m^n}{x_\xi^{n+\frac{1}{2}}}. \tag{3.29}$$

- The pressure can then be determined at the half step by using the equation of state from Section 2.7.1,

$$p^{n+\frac{1}{2}} = a^2 \rho^{n+\frac{1}{2}}.$$

- Full timestep velocities are then calculated from the half step pressures using eqn. (3.14),

$$u^{n+1} = u^n + kf^{n+\frac{1}{2}}. \tag{3.30}$$

### Corrector

- Averaged velocities from eqn. (3.16) are used to update the nodal positions to the full timestep,

$$x^{n+1} = x^n + ku^n + \frac{k^2}{2}f^{n+\frac{1}{2}}. \tag{3.31}$$

- Following this the density can be calculated at the full timestep from the conservation of mass in eqn. (3.1),

$$\begin{aligned} m^{n+1} &= m^n \\ \Rightarrow \rho^{n+1} &= \frac{m^n}{x_\xi^{n+1}}. \end{aligned}$$

- Finally calculate the pressures from the equation of state at the full timestep

$$p^{n+1} = a^2 \rho^{n+1}.$$

This comprises the mapping from $\{x^n, u^n\}$ to $\{x^{n+1}, u^{n+1}\}$ for the entire system of equations, where we have used the half timestep, $t^n + \frac{k}{2}$, as an intermediate step to achieve this mapping.

### 3.3.2 Scheme 2

We now approximate the system using the Velocity Verlet Scheme in Section 3.2.2, which is initiated with the approximation to the velocity at the half timestep as follows.

#### Predictor

- We apply eqn. (3.19) to the conservation of momentum eqn. (3.3), to obtain the half timestep velocities

$$u^{n+\frac{1}{2}} = u^n + \frac{kf^n}{2}. \tag{3.32}$$

This result leads to updating the solutions to the full timestep.

#### Corrector

- We apply eqn. (3.21) to eqn. (3.4) in order to update the nodal positions to the full timestep by

$$x^{n+1} = x^n + ku^n + \frac{k^2 f^n}{2}. \tag{3.33}$$

- The density is calculated at the full timestep using the conservation of mass equation,

$$\begin{aligned} m^{n+1} &= m^n \\ \Rightarrow \rho^{n+1} &= \frac{m^n}{x_\xi^{n+1}}. \end{aligned}$$

- Calculate the pressures at the full timestep from the equation of state,

$$p^{n+1} = a^2 \rho^{n+1}.$$

- Finally update the velocities to the full timestep using eqn. (3.26)

$$u^{n+1} = u^n + \frac{k}{2}(f^{n+1} + f^n). \tag{3.34}$$

We have obtained a second scheme which completes the mapping from $\{x^n, u^n\}$ to $\{x^{n+1}, u^{n+1}\}$. Although this scheme differs from Scheme 1 they are both second order schemes.
This leads on to the investigation of an error estimate based on both schemes.

## 3.4 Local Truncation Error

The definition for the LTE has already been given in Section 2.18, for the general case. Here we deal with the semi-discrete case where we have the differential equations (3.1)-(3.4).
For the ODE,

$$\frac{dx}{dt} = u,$$

the differential operator for nodal position is expressed as

$$L(x) \quad = \quad \frac{dx}{dt} - u = 0, \tag{3.35}$$

where $x = x(t)$ and $u = u(t)$. The algorithm approximating the differential equation from Section 3.2.1 is defined as $L_h$ for eqn. (3.35).
The truncation error in one step for a temporal semi-discrete scheme is defined as

$$\tau_n = L_h(x(t^n)) - L_h(x^n), \tag{3.36}$$

where $L_h(x^n) = 0$ from eqn. (3.35).

### 3.4.1 Scheme 1

The differential equation is

$$L(x(t)) = \frac{dx(t)}{dt} - u(t) = 0,$$

and the scheme used to obtain a numerical approximation to $x$ from eqn. (3.31) is

$$L_h(x^n) = \frac{x^{n+1} - x^n}{k} - u^n - \frac{k}{2} f^{n+\frac{1}{2}} = 0. \tag{3.37}$$

To obtain the LTE apply eqn. (3.37) to the exact solution at time $t^n$, giving

$$L_h(x(t^n)) = \frac{x(t^n + k) - x(t^n)}{k} - u(t^n) - \frac{k}{2} f\left(t^n + \frac{k}{2}\right). \tag{3.38}$$

The LTE for position is therefore,

$$\tau_n(L_1) = \frac{x(t^n + k) - x(t^n)}{k} - u(t^n) - \frac{k}{2} f\left(t^n + \frac{k}{2}\right), \tag{3.39}$$

where we have used eqn. (3.37) and eqn. (3.38). Where $\tau_n(L_1)$ is the truncation error for eqn. (3.37).

The exact differential equations are;

$$x' = \frac{dx(t)}{dt} = u(t) \tag{3.40}$$

$$x'' = \frac{d^2x(t)}{dt^2} = \frac{du(t)}{dt} = f(t), \tag{3.41}$$

therefore, eqn. (3.39) becomes

$$\tau_n(L_1) = \frac{x(t^n + k) - x(t^n)}{k} - x'(t^n) - \frac{k}{2} x''\left(t^n + \frac{k}{2}\right). \tag{3.42}$$

Expanding $x(t^n + k)$ and $x''(t^n + \frac{k}{2})$ in the above about $x(t^n) = x$, by Taylor series,

$$x(t^n + k) = x + kx' + \frac{k^2}{2!}x'' + \frac{k^3}{3!}x''' + \cdots \tag{3.43}$$

$$x''(t^n + \frac{k}{2}) = x'' + k\frac{1}{2}x''' + \frac{k^2}{2^2.2!}x'''' + \cdots \tag{3.44}$$

the LTE becomes

$$\begin{aligned} k\tau_n(L_1) &= x + kx' + \frac{k^2}{2!}x'' + \frac{k^3}{3!}x''' - x - kx' - \frac{k^2}{2}\left(x'' + k\frac{1}{2}x''' + \frac{k^2}{2^2.2!}x''''\right) + \cdots \\ &= \left(\frac{1}{6} - \frac{1}{4}\right)k^3x''' + \mathcal{O}(k^4) \\ \Rightarrow \tau_n(L_1) &= -\frac{k^2}{12}x''' + \mathcal{O}(k^3) \qquad (3.45) \end{aligned}$$

which is of second order.

### 3.4.2 Scheme 2

For Scheme 2 the differential equation in eqn. (3.35) is approximated by eqn. (3.33), so define

$$L_h(x^n) = \frac{x^{n+1} - x^n}{k} - u^n - \frac{k}{2}f^n = 0. \qquad (3.46)$$

Apply eqn. (3.46) to the exact solution $x(t^n)$,

$$L_h(x(t^n)) = \frac{x(t^n + k) - x(t^n)}{k} - u(t^n) - \frac{k}{2}f(t^n). \qquad (3.47)$$

Using eqn. (3.46) and eqn. (3.47), along with eqn. (3.40) and eqn. (3.41), where $\tau_n(L_2)$ is the LTE for eqn. (3.46), we obtain

$$\tau_n(L_2) = \frac{x(t^n + k) - x(t^n)}{k} - x'(t^n) - \frac{k}{2}x''(t^n). \qquad (3.48)$$

Using eqn. (3.43) to expand $x(t^n + k)$ about $x(t^n) = x$ in the above by Taylor series, the local truncation error becomes

$$\begin{aligned} k\tau_n(L_2) &= x + kx' + \frac{k^2}{2!}x'' + \frac{k^3}{3!}x''' - x - kx' - \frac{k^2}{2}x'' + \cdots \\ &= \frac{k^3}{6}x''' + \mathcal{O}(k^4) \qquad (3.49) \\ \Rightarrow \tau_n(L_2) &= \frac{k^2}{6}x''' + \mathcal{O}(k^3), \qquad (3.50) \end{aligned}$$

hence, the scheme is of second order.

## 3.5 Error Estimation

We have seen in eqn. (2.43) how the truncation errors can be used to gain an estimated error in one step. Here we can obtain an estimated error for the nodal positions in a similar way since the order of the truncation errors for Scheme 1 and Scheme 2 are the same, but the leading coefficient is different.

### 3.5.1 Estimated Error for Position in One Step

From Section 3.2.1 and Section 3.2.2 we have the leading terms of the LTE for the nodal positions as

$$\begin{aligned} x(t^n + k) - x_1^{n+1} &= k\tau_n(L_1) = -\frac{k^3}{12}x_{ttt}(t^n) + \mathcal{O}(k^4) && (3.51) \\ x(t^n + k) - x_2^{n+1} &= k\tau_n(L_2) = \frac{k^3}{6}x_{ttt}(t^n) + \mathcal{O}(k^4), && (3.52) \end{aligned}$$

where $x_1^{n+1}$ and $x_2^{n+1}$ represent Scheme 1 and Scheme 2 approximations to $x$ respectively. Now subtract eqn. (3.52) from eqn. (3.51)

$$\begin{aligned} x_2^{n+1} - x_1^{n+1} &= -\frac{k^3}{4}x_{ttt}(t^n) + \mathcal{O}(k^4), \\ \Rightarrow k^3 x_{ttt} &= 4\left(x_1^{n+1} - x_2^{n+1}\right) + \mathcal{O}(k^4). && (3.53) \end{aligned}$$

We now have a computable estimate to the uncomputable term, which can be substituted into the local error term, eqn. (3.51),

$$x(t^n + h) - x_1^{n+1} = \frac{\left(x_2^{n+1} - x_1^{n+1}\right)}{3} + \mathcal{O}(k^4) \quad (3.54)$$

leading to the truncation error in one step being estimated by

$$\tau_n(L_1) = \frac{\left(x_2^{n+1} - x_1^{n+1}\right)}{3k} + \mathcal{O}(k^3). \quad (3.55)$$

Hence, we now have a computable local error due to the information introduced from both schemes.

## 3.6 Semi-Discrete Error Estimate - Density and Velocity

From eqn. (3.55) we have the LTE for the differential equation $\frac{dx}{dt} = u$, which is

$$\tau(L_1) = \frac{\left((x_2)^{n+1} - (x_1)^{n+1}\right)}{3k} + \mathcal{O}(k^3). \tag{3.56}$$

Additionally, from this we obtained, in Section 3.5.1, an estimated semi-discrete LTE in one time-step for the variable $x$. From eqn. (3.56) this is

$$x(t^n + k) - (x_1)^{n+1} = \frac{\left((x_2)^{n+1} - (x_1)^{n+1}\right)}{3} + \mathcal{O}(k^4), \tag{3.57}$$

where all values are held the same point on the computational grid.

### 3.6.1 Semi-Discrete in Time - Density Error Estimate

From eqn. (3.2) we have

$$m = \rho x_\xi,$$

and using eqn. (3.1), where mass is conserved, i.e.

$$\begin{aligned} \frac{dm}{dt} &= 0 \\ \Rightarrow m^{n+1}(\xi) &= m^n(\xi), \\ \Rightarrow \rho^{n+1}(\xi) &= \frac{m^n(\xi)}{x_\xi^{n+1}}, \end{aligned}$$

the semi-discrete in time LTE of density in one time step can be expressed by

$$\begin{aligned} e[\rho] = \rho(t^n + k) - (\rho_1)^{n+1} &= \frac{m^n}{x_\xi(t^n + k)} - \frac{m^n}{(x_\xi)^{n+1}} \\ &= \frac{m^n\left((x_\xi)^{n+1} - x_\xi(t^n + k)\right)}{x_\xi(t^n + k)\,(x_\xi)^{n+1}}. \end{aligned} \tag{3.58}$$

Let $e$ denote the error.
Differentiating eqn. (3.57) with respect to $\xi$,

$$e[x_\xi] = x_\xi(t^n + k) - \left((x_1)_\xi\right)^{n+1} = \frac{\left(\left((x_2)_\xi\right)^{n+1} - \left((x_1)_\xi\right)^{n+1}\right)}{3} + \mathcal{O}(k^4)$$

where $x_\xi$ is just the divided difference of $x$ at two points on the computational mesh, i.e. $\frac{(x_i - x_{i-1})}{h}$. Therefore, for the LTE of density in one timestep, by substituting the above into eqn. (3.58) we obtain the expression

$$e[\rho] = \rho(t^n + k) - (\rho_1)^{n+1} = \frac{m^n\left(\left((x_1)_\xi\right)^{n+1} - \left((x_2)_\xi\right)^{n+1}\right)}{\left((x_1)_\xi\right)^{n+1}\left(2\left((x_1)_\xi\right)^{n+1} + \left((x_2)_\xi\right)^{n+1}\right)}. \qquad (3.59)$$

### 3.6.2 Semi-Discrete in Time - Velocity Error Estimate

Using eqn. (3.55), we have the LTE for the differential equation where $\frac{dx}{dt} = \dot{x}$, such that

$$e\left[\dot{x}\right] = \tau(L_1) = \frac{\left((x_2)^{n+1} - (x_1)^{n+1}\right)}{3k} + \mathcal{O}(k^3).$$

Using the above and eqn. (3.4), where

$$\frac{dx}{dt} = u,$$

we can obtain the LTE for the variable $u$ in one time-step, such that

$$e[u] = u(t^n + k) - (u_1)^{n+1} = \tau(L_1) = \frac{\left((x_2)^{n+1} - (x_1)^{n+1}\right)}{3k} + \mathcal{O}(k^3) \qquad (3.60)$$

## 3.7 System of Three Equations

The previous idea is extended to the more complicated system of three equations in 1D. In addition to the conservation of mass, momentum and the trajectory equation in Section 3.1, we introduce the conservation of specific internal energy

$$\frac{d\varepsilon}{dt} = pg, \tag{3.61}$$

where $g = g(t)$ is defined and expressed in its full form in eqn. (2.28). The pressure $p$ is not included in $g$. Later we see $p$ and $g$ expressed at different timesteps.

As before, we wish to investigate two numerical schemes, where the first scheme is as proposed in Section 3.3.1, the second scheme is an alternative scheme for the system of three equations.

### 3.7.1 Scheme 1

As already mentioned, the first scheme we introduce for the 1D system of three equations is as stated in Section 3.3.1, with the addition of a step for the specific internal energy.

The approximation of the specific internal energy occurs after the density update eqn. (3.29) and before the pressure update. In the predictor step this is given by

$$\varepsilon^{n+\frac{1}{2}} = \varepsilon^n + \frac{k}{2} p^n g^n.$$

The specific internal energy is then advanced to the full timestep in the corrector step by

$$\varepsilon^{n+1} = \varepsilon^n + kp^{n+\frac{1}{2}} g^{n+\frac{1}{2}}. \tag{3.62}$$

We now look at a second scheme for the 1D system of three equations.

### 3.7.2 Scheme 2

As in Section 3.3.2 a second scheme is chosen which is of the same order as Scheme 1.

The alternative scheme used is stated in [45] and [46]. This scheme is chosen since it is widely used and published by the Los Alamos National Laboratory (LANL), for the system of three equations, [46]. It follows the desired characteristics of being of predictor-corrector form and is of second order. This scheme uses the full timestep as the corrector along with averaged velocities to correct the results at the full timestep.

### Predictor

We shall use the suffix $_*$ to denote the predictor stage here, since the predictor and corrector are calculated at the same timesteps.

- Full timestep velocities are predicted using the momentum equation in eqn. (3.3),

$$u_*^{n+1} = u^n + kf^n. \tag{3.63}$$

- A predicted average velocity is obtained from,

$$\begin{aligned} \bar{u}_* = u_*^{n+\frac{1}{2}} &= \frac{1}{2}(u^n + u_*^{n+1}) \\ &= u^n + \frac{k}{2}f^n. \end{aligned} \tag{3.64}$$

- Nodal positions are predicted using eqn. (3.4)

$$x_*^{n+1} = x^n + k\bar{u}_* \tag{3.65}$$

$$= x^n + ku^n + \frac{k^2}{2}f^n. \tag{3.66}$$

- Density can be predicted at the full timestep by using the conservation of mass in eqn. (3.1).
- The specific internal energy can now be predicted at the full timestep using eqn. (3.61)

$$\varepsilon_*^{n+1} = \varepsilon^n + kp^n g_*^{n+\frac{1}{2}}. \tag{3.67}$$

- The equation of state is then used to obtain the pressure at the predicted full timestep.

### Corrector

- Corrected full timestep velocities are calculated using the predicted pressures

$$u^{n+1} = u^n + kf_*^{n+1}. \tag{3.68}$$

- Averaged velocities are corrected by

$$\begin{aligned}\bar{u} &= u^{n+\frac{1}{2}} = \frac{1}{2}(u^n + u^{n+1}) \\ &= u^n + \frac{k}{2} f_*^{n+1}. \end{aligned} \tag{3.69}$$

- The averaged velocities are used to update the nodal positions

$$\begin{aligned} x^{n+1} &= x^n + k\bar{u} \\ &= x^n + ku^n + \frac{k^2}{2} f_*^{n+1}. \end{aligned} \tag{3.70}$$

- Corrected densities are obtained from the conservation of mass in eqn. (3.1).
- Energies are corrected at the full timestep using

$$\varepsilon^{n+1} = \varepsilon^n + kp_*^{n+1} g^{n+\frac{1}{2}}. \tag{3.71}$$

- Finally full timestep pressures are updated using the equation of state.

Now that we have two schemes, for the system of three equations, we can obtain their truncation errors.

## 3.8 Local Truncation Error

The differential operators for nodal position have already been defined in Section 3.4. From Section 3.4, the truncation error in one step for a semi-discrete scheme is defined as

$$\tau_n = L_h(x(t^n)) - L_h(x^n), \tag{3.72}$$

where $L_h(x^n) = 0$ from eqn. (3.35).
We now obtain the truncation errors to the schemes for the system of three equations.

### 3.8.1 Scheme 1

The truncation error of the nodal position as a result of applying Scheme 1 has already been defined in Section 3.4.1 as

$$\tau_n(L_1) = -\frac{k^2}{12}x''' + \mathcal{O}(k^3) \tag{3.73}$$

from eqn. (3.45).

### 3.8.2 Scheme 2

We have the differential equation

$$L(x(t)) = \frac{dx(t)}{dt} - u(t) = 0$$

and the scheme used to obtain a numerical approximation to $x$ from eqn. (3.70) is

$$L_h(x^n) = \frac{x^{n+1} - x^n}{k} - u^n - \frac{k}{2}f_*^{n+1} = 0. \tag{3.74}$$

To obtain the LTE, apply eqn. (3.74) to the exact solution at time $t^n$, giving

$$L_h(x(t^n)) = \frac{x(t^n + k) - x(t^n)}{k} - u(t^n) - \frac{k}{2}f\left(t_*^{n+1}\right). \tag{3.75}$$

Using eqn. (3.74), and eqn. (3.75) along with eqn. (3.40) and eqn. (3.41), where the truncation error for eqn. (3.74) is $\tau_n(L_2)$, we obtain

$$\tau_n(L_2) = \frac{x(t^n + k) - x(t^n)}{k} - x'(t^n) - \frac{k}{2}x''\left(t_*^{n+1}\right). \tag{3.76}$$

Expanding $x(t^n + k)$ and $x''(t_*^{n+1})$ in the above about $x(t^n) = x$, as stated in eqn. (3.43), the LTE becomes

$$\begin{aligned} k\tau_n(L_2) &= x + kx' + \frac{k^2}{2!}x'' + \frac{k^3}{3!}x''' - x - kx' - \frac{k^2}{2}\left(x'' + kx''' + \frac{k^2}{2!}x''''\right) + \cdots \\ &= \left(\frac{1}{6} - \frac{1}{2}\right)k^3x''' + \mathcal{O}(k^4) \\ \Rightarrow \tau_n(L_2) &= -\frac{k^2}{3}x''' + \mathcal{O}(k^3), \end{aligned} \tag{3.77}$$

the scheme is of second order.

## 3.9 Error Estimation

We have already seen in eqn. (2.43) and Section 3.5 how the truncation errors are used to determine an estimated error in one step. Here, we obtain an estimated error for the nodal positions. However, we can not obtain an estimated error to the specific internal energy. This is because the second scheme is only 1st order accurate for the specific internal energy approximation, where we require second order accuracy in order to use our error estimation. For now the error estimation for the nodal positions is adequate to continue the investigation of error estimates.

### 3.9.1 Estimated Error for Velocity and Density in One Step

From Section 3.8.1 and Section 3.8.2 we have the leading terms of the LTE for the nodal positions as

$$x(t^n + k) - x_1^{n+1} = k\tau_n(L_1) = -\frac{k^3}{12}x_{ttt}(t^n) + \mathcal{O}(k^4) \tag{3.78}$$

$$x(t^n + k) - x_2^{n+1} = k\tau_n(L_2) = -\frac{k^3}{3}x_{ttt}(t^n) + \mathcal{O}(k^4), \tag{3.79}$$

where $x_1^{n+1}$ and $x_2^{n+1}$ represent Scheme 1 and Scheme 2 approximations to $x$ respectively for the system of three equations.
Now subtract eqn. (3.79) from eqn. (3.78)

$$\begin{aligned} x_2^{n+1} - x_1^{n+1} &= \frac{k^3}{4}x_{ttt}(t^n) + \mathcal{O}(k^4) \\ \Rightarrow k^3 x_{ttt} &= 4\left(x_2^{n+1} - x_1^{n+1}\right) + \mathcal{O}(k^4). \end{aligned} \tag{3.80}$$

We now have a computable approximation to the uncomputable terms, which can be substituted into the local error term eqn. (3.78),

$$x(t^n + k) - x_1^{n+1} = \frac{\left(x_1^{n+1} - x_2^{n+1}\right)}{3} + \mathcal{O}(k^4), \tag{3.81}$$

leading to the truncation error, in one step, of the derivative $\frac{dx}{dt}$ being estimated by

$$\tau_n(L_1) = \frac{\left(x_1^{n+1} - x_2^{n+1}\right)}{3k} + \mathcal{O}(k^3). \tag{3.82}$$

Hence, we now have a computable local error due to the information introduced from both schemes.

### 3.9.2 Semi-Discrete in Time Estimated Error for $x$

In Section 3.7 we saw how to approximate the LTE for the differential equation

$$\frac{dx}{dt} = \dot{x} = u,$$

which is

$$\tau(L_1) = \frac{\left((x_1)^{n+1} - (x_2)^{n+1}\right)}{3k} + \mathcal{O}(k^3), \tag{3.83}$$

from eqn. (3.82). We obtained the computable LTE in one step for the variable $x$ in the form

$$x(t^n + h) - (x_1)^{n+1} = \frac{\left((x_1)^{n+1} - (x_2)^{n+1}\right)}{3} + \mathcal{O}(k^4). \tag{3.84}$$

From this result, we are able to gain the LTEs for density and velocity.

### 3.9.3 Semi-Discrete in Time Estimated Error for Density

Using the result in eqn. (3.58) we are able to obtain the semi-discrete LTE for density using

$$e[\rho] = \rho(t^n + k) - (\rho_1)^{n+1} = \frac{m^n \left((x_\xi)^{n+1} - x_\xi(t^n + k)\right)}{x_\xi(t^n + k)\,(x_\xi)^{n+1}}. \tag{3.85}$$

By differentiating eq. (3.84) with respect to $\xi$ we obtain

$$e[x_\xi] = x_\xi(t^n + k) - \left((x_1)_\xi\right)^{n+1} = \frac{\left(\left((x_1)_\xi\right)^{n+1} - \left((x_2)_\xi\right)^{n+1}\right)}{3} + \mathcal{O}(k^4), \tag{3.86}$$

which can be substituted into eqn. (5.9), resulting in

$$e[\rho] = \rho(t^n + k) - (\rho_1)^{n+1} = \frac{m^n \left(\frac{\left(\left((x_2)_\xi\right)^{n+1} - \left((x_1)_\xi\right)^{n+1}\right)}{3}\right)}{\left(\left((x_1)_\xi\right)^{n+1} + \frac{\left(\left((x_1)_\xi\right)^{n+1} - \left((x_2)_\xi\right)^{n+1}\right)}{3}\right)\left((x_1)_\xi\right)^{n+1}} + \mathcal{O}(k^4). \tag{3.87}$$

### 3.9.4 Semi-Discrete in Time Estimated Error for Velocity

We are able to obtain the LTE approximation to velocity directly from eqn. (3.82). Since this is the truncation error of the derivative $\dot{x}$, using eqn. (3.4) we have

$$e[\dot{x}] = e[u] = \tau(L_1)$$

hence

$$e[u] = u(t^n + k) - (u_1)^{n+1} = \frac{\left((x_1)^{n+1} - (x_2)^{n+1}\right)}{3k} + \mathcal{O}(k^3). \tag{3.88}$$

## 3.10 Chapter 3 Summary

In this chapter we have introduced two schemes to approximate the temporal derivatives. This has been done for a system of two equations and a system of three equations.

The truncation errors for each scheme have been obtained. These results have been utilised in a method similar to Milne's approach. This results in an error estimate for density and velocity for the Main Scheme, which was described in Section 3.3.1.

We now turn our attention to investigating the error in spatial derivatives by looking at the different forms of discretising these derivatives and how this can lead to an error estimate.

# Chapter 4

# Semi-Discrete Spatial Schemes in 1D

Previously we have seen the system of two and three equations in 1D expressed as semi-discrete in time schemes. The schemes are written in a finite difference form. Now continue by investigating the spatial derivatives for the two and three system of equations in 1D, where the spatial derivatives are approximated using a finite element approach.

The Finite Element Method (FEM), has been used for several decades, with its development accredited to several mathematicians. An overview of these mathematicians and their contributions to the development of the FEM is given by Robinson in [75]. After the initial development by Courant in 1943, see [19], the next significant advancement came from H. C. Martin in the 1950's, [41], as a result of the company Boeing approaching Martin to develop a solution to the structural analysis problems for aircraft. The success of Martin's work came from his academic and industrial experience. Martin's first publication on the FEM came out in 1956, see [91], which was followed by two books on his FEM developments, see [62] and [61].

FEMs are used as an alternative to finite difference methods (FDM), to approximate solutions of partial differential equations (PDEs), in particular where it is difficult to obtain a solution on a non-rectangular domain. The idea of FEMs is to simplify a complex problem by using small building blocks, i.e. elements. FEMs are used to solve PDEs over complicated domains, where boundaries are irregular or moving. This is done by subdividing the region into smaller elements, yielding problems which are solved in relation to each other.

Due to the flexibility of FEMs there is an extensive list of applications that include mechanical, aerospace, civil and automotive engineering, structural analysis, fluid dynamics and electromagnetics, amongst many others. Since the discretisation is flexible it allows for more detailed visualisations of structures, in particular where awkward features occur, whereas FDMs are basically restricted to rectangular domains. FEMs have greatly improved the standard of Engineering designs, since they can be optimised and refined before prototypes are developed, increasing productivity and reducing manufacturing costs. Although FDMs may be simpler to implement, it is FEMs that are favoured for structural analysis.

During this chapter we look at the discretisation of spatial derivatives in 1D, for the two and three system of equations. We begin by looking at the system of two equations where the spatial derivative arises in the momentum equation. This is discretised using a standard Galerkin method, as in [64, 10]. We also seek an alternate scheme to estimate the error. For this we introduce a second test function and use a Petrov-Galerkin method. Both methods can then be used to determine an estimated error. The method is advanced to the system of three equations, with the addition of specific internal energy.

## 4.1 System of Two Equations

In Section 2.6 we saw the system of equations expressed in terms of Lagrangian co-ordinates. For the system of two equations, the momentum equation contains a spatial derivative. From eqn. (2.35) and eqn. (2.32), the momentum equation is of the form

$$m\frac{\partial \boldsymbol{u}}{\partial t} = -\boldsymbol{C}\nabla_{\boldsymbol{\xi}}p, \tag{4.1}$$

where $m = m(\boldsymbol{\xi})$, $\boldsymbol{u} = \boldsymbol{u}(\boldsymbol{\xi}, t)$, $p = p(\boldsymbol{\xi}, t)$, $\boldsymbol{C}$ is the adjoint matrix from the inverse of the Jacobian and $\nabla$ is the spatial derivative, in terms of the reference co-ordinates $\boldsymbol{\xi} \in [-1, 1] \times [-1, 1]$.

In 1D, the momentum equation is expressed as

$$m\dot{u} = -\frac{\partial p}{\partial \xi}, \tag{4.2}$$

since $\boldsymbol{C} = I$ in 1D. We drop the temporal dependencies and use $m = m(\xi)$, $\dot{u} = \dot{u}(\xi)$ and $p = p(\xi)$, where $\xi \in [-1, 1]$.

To gain a spatial approximation for eqn. (4.2), consider a spatial step $h > 0$ and a single point at $\xi_i = ih$, where $i = 0, 1, \cdots N + 1$. The aim is to obtain a sequence of points $x_i$ that closely follow the points $x(\xi_i)$, which are on the trajectory of the exact solution, where $\xi_i = \xi$.

To gain an expression for the 1D semi-discrete momentum equation in terms of finite elements, we follow the scheme in [64, 10] and apply a finite element approach using a weak form of eqn. (4.2).

## 4.2 Weak Form of Momentum Equation

A weak form of the momentum equation is obtained by multiplying eqn. (4.2) by a test function of the form $\omega(\xi)$, where $\omega = \omega(\xi) \in H^1(-1,1)$. Integrate over a control volume, which in 1D is $\xi \in (-1,1)$, resulting in

$$\int_{-1}^{1} \omega m \dot{u} d\xi = -\int_{-1}^{1} \omega p_\xi d\xi. \tag{4.3}$$

The weak form of the momentum equation can be approximated using a variety of test functions. Here we look at a Galerkin method and a Petrov-Galerkin method.

### 4.2.1 Elementwise Galerkin Approach

Integration by parts of eqn. (4.3) yields the weak form of the momentum equation, where the interval $[-1,1]$ is subdivided into equal elements of size $h$ such that the interior boundaries are $(\xi_{i-1}, \xi_i)$, therefore,

$$\int_{\xi_{i-1}}^{\xi_i} \omega m \dot{u} d\xi = -\omega p \Big|_{\xi_{i-1}}^{\xi_i} + \int_{\xi_{i-1}}^{\xi_i} \omega_\xi p d\xi. \tag{4.4}$$

We assume $m$ and $p$ to be piecewise constant functions within each element $\in (\xi_{i-1}, \xi_i)$. The acceleration $\dot{u}$ is approximated by the piecewise linear function,

$$\dot{u} = \sum_{i=0}^{N+1} \dot{u}_i \phi_i(\xi). \tag{4.5}$$

The basis function $\phi_i(\xi)$ is chosen to be a piecewise linear hat function,

$$\phi_i(\xi) = \begin{cases} \frac{\xi - \xi_{i-1}}{h} & \xi_{i-1} < \xi < \xi_i \\ \frac{\xi_{i+1} - \xi}{h} & \xi_i < \xi < \xi_{i+1} \\ 0 & \text{elsewhere} \end{cases} \tag{4.6}$$

for $i = 1, 2 \cdots N$. We have the end-point basis functions $\phi_0(\xi)$ and $\phi_{N+1}(\xi)$ as half hat functions.

### 4.2.2 Scheme A

For Scheme A we use a Galerkin approach, such that $\omega(\xi) = \phi_j(\xi)$ is used in the weak form eqn. (4.4), where $i = 1 \cdots N$. Using this and the weak form eqn. (4.4) of the momentum equation, over an element $(\xi_{i-1}, \xi_i)$ we obtain,

$$m_{i-\frac{1}{2}} \int_{\xi_{i-1}}^{\xi_i} \sum_{j=i-1}^{i} \dot{u}_j \phi_j \phi_i d\xi = -\phi_i p \Big|_{\xi_{i-1}}^{\xi_i} + p_{i-\frac{1}{2}} \int_{\xi_{i-1}}^{\xi_i} \frac{\partial \phi_j}{\partial \xi} d\xi. \tag{4.7}$$

We have taken $m_{i-\frac{1}{2}}$ and $p_{i-\frac{1}{2}}$ outside the integral as they are constant across an element. The integral and sum are interchanged, resulting in

$$m_{i-\frac{1}{2}} \sum_{j=i-1}^{i} \dot{u}_j \int_{\xi_{i-1}}^{\xi_i} \phi_j \phi_i d\xi = -\phi_i p \Big|_{\xi_{i-1}}^{\xi_i} + p_{i-\frac{1}{2}} \int_{\xi_{i-1}}^{\xi_i} \frac{\partial \phi_i}{\partial \xi} d\xi. \tag{4.8}$$

Using eqn. (4.8) and including the end nodes of the element, we obtain the following system, for each element,

$$m_{i-\frac{1}{2}} \begin{pmatrix} \int_{\xi_{i-1}}^{\xi_i} \phi_{i-1}\phi_{i-1} d\xi & \int_{\xi_{i-1}}^{\xi_i} \phi_{i-1}\phi_i d\xi \\ \int_{\xi_{i-1}}^{\xi_i} \phi_{i-1}\phi_i d\xi & \int_{\xi_{i-1}}^{\xi_i} \phi_i\phi_i d\xi \end{pmatrix} \begin{pmatrix} \dot{u}_{i-1} \\ \dot{u}_i \end{pmatrix}$$

$$= - \begin{pmatrix} \phi_{i-1} p \Big|_{\xi_{i-1}}^{\xi_i} \\ \phi_i p \Big|_{\xi_{i-1}}^{\xi_i} \end{pmatrix} + p_{i-\frac{1}{2}} \begin{pmatrix} \int_{\xi_{i-1}}^{\xi_i} (\phi_\xi)_{i-1} d\xi \\ \int_{\xi_{i-1}}^{\xi_i} (\phi_\xi)_i d\xi \end{pmatrix}. \tag{4.9}$$

On evaluating the integrals in eqn. (4.9), we have,

$$m_{i-\frac{1}{2}} h \begin{pmatrix} \frac{1}{3} & \frac{1}{6} \\ \frac{1}{6} & \frac{1}{3} \end{pmatrix} \begin{pmatrix} \dot{u}_{i-1} \\ \dot{u}_i \end{pmatrix} = \begin{pmatrix} p_{i-1} \\ -p_i \end{pmatrix} + \begin{pmatrix} -p_{i-\frac{1}{2}} \\ p_{i-\frac{1}{2}} \end{pmatrix}, \tag{4.10}$$

where $h$ is the distance between two points and is equal to $\xi_i - \xi_{i-1}$.

Work carried out in [64, 10] deals with the above mass matrix by 'mass lumping'. Mass lumping is a technique used to transfer all the off-diagonal information onto the diagonal. This only leads to a higher order error. The advantage is we have a diagonal matrix which is less computationally expensive to solve than the full system. Mass lumping is discussed further in [96]. For the system in eqn. (4.10), the mass lumped system becomes

$$m_{i-\frac{1}{2}}h\begin{pmatrix}\frac{1}{2} & 0\\ 0 & \frac{1}{2}\end{pmatrix}\begin{pmatrix}\dot{u}_{i-1}\\ \dot{u}_i\end{pmatrix}=\begin{pmatrix}p_{i-1}\\ -p_i\end{pmatrix}+\begin{pmatrix}-p_{i-\frac{1}{2}}\\ p_{i-\frac{1}{2}}\end{pmatrix}.$$

In order to gain a solution at node $i$ we require information from both the neighbouring elements, $i-\frac{1}{2}$ and $i+\frac{1}{2}$, see Figure 4.1.

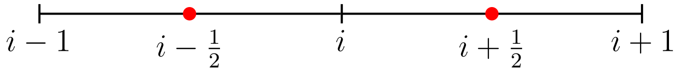


Figure 4.1: Neighbouring Elements to Node $i$ Required for Assembly.

The assembly process overlaps the systems for elements $i-\frac{1}{2}$ and $i+\frac{1}{2}$ and preserves the diagonal nature of the mass matrix, such that

$$h\begin{pmatrix}\frac{m_{i-\frac{1}{2}}}{2} & 0 & 0\\ 0 & \frac{m_{i-\frac{1}{2}}}{2}+\frac{m_{i+\frac{1}{2}}}{2} & 0\\ 0 & 0 & \frac{m_{i+\frac{1}{2}}}{2}\end{pmatrix}\begin{pmatrix}\dot{u}_{i-1}\\ \dot{u}_i\\ \dot{u}_{i+1}\end{pmatrix}=\begin{pmatrix}p_{i-1}\\ -p_i+p_i\\ -p_{i+1}\end{pmatrix}+\begin{pmatrix}-p_{i-\frac{1}{2}}\\ p_{i-\frac{1}{2}}-p_{i+\frac{1}{2}}\\ p_{i+\frac{1}{2}}\end{pmatrix}.$$

At node $i$, the solution is extracted easily, resulting in

$$\frac{1}{2}(m_{i-\frac{1}{2}}+m_{i+\frac{1}{2}})h\dot{u}_i \quad = \quad (-p_i+p_i)+(p_{i-\frac{1}{2}}-p_{i+\frac{1}{2}}). \tag{4.11}$$

The boundary terms cancel. We define the average of the mass as the nodal mass $m_i$, such that

$$m_i=\frac{1}{2}(m_{i-\frac{1}{2}}+m_{i+\frac{1}{2}}).$$

Therefore, the spatial derivative in the momentum equation in a semi-discrete form is approximated by

$$\dot{u}_i=-\frac{p_{i+\frac{1}{2}}-p_{i-\frac{1}{2}}}{hm_i}, \tag{4.12}$$

which will be referred to as Scheme A.

It is required to develop a second scheme for the spatial derivative, for use later in an error estimation. Hence, we look at a Petrov-Galerkin method to obtain an alternative to Scheme A.

### 4.2.3 Elementwise Petrov-Galerkin Approach for the Momentum Equation

For the Petrov-Galerkin method, we choose a different test function where $\psi_i \neq \phi_i$, see [100], such that the weak form in eqn. (4.4) becomes

$$\int_{\xi_{i-1}}^{\xi_i} \psi_i m \dot{u} d\xi = -\psi_i p \Big|_{\xi_{i-1}}^{\xi_i} + \int_{\xi_{i-1}}^{\xi_i} p \frac{\partial \psi_i}{\partial \xi} d\xi. \tag{4.13}$$

Previously we have seen the interval $[-1, 1]$ divided into elements and choose $\dot{u}$ as a piecewise linear function, such that

$$\dot{u} = \sum_{j=0}^{N+1} \dot{u}_j \phi_j(\xi),$$

where the basis functions $\phi(\xi)$ have previously been defined in eqn. (4.6). For an element $(\xi_{i-1}, \xi_i)$, eqn. (4.13) becomes

$$m_{i-\frac{1}{2}} \sum_{j=i-1}^{i} \dot{u}_j \int_{\xi_{i-1}}^{\xi_i} \phi_j \psi_i d\xi = -\psi_i p \Big|_{\xi_{i-1}}^{\xi_i} + p_{i-\frac{1}{2}} \int_{\xi_{i-1}}^{\xi_i} \frac{\partial \psi_i}{\partial \xi} d\xi. \tag{4.14}$$

We take $m_{i-\frac{1}{2}}$ and $p_{i-\frac{1}{2}}$ outside the integral since they are constant over an element, and interchange the integral and the sum.

We determine $\psi(\xi)$. To avoid mass lumping, we require $\psi(\xi)$to be diagonal. Hence, the integral on the left hand side of eqn. (4.14) is of the form

$$\int_{\xi_{i-1}}^{\xi_i} \psi_j \phi_i d\xi = \delta_{ij}, \tag{4.15}$$

where $\delta_{ij}$ is a matrix of zeros everywhere apart from the diagonal. This is achieved by multiplying the mass matrix, in eqn. (4.10), by its inverse.

Using the above and the mass matrix in eqn. (4.10), for an element $i - \frac{1}{2}$ we obtain,

$$\begin{pmatrix} \psi_{i-1} \\ \psi_i \end{pmatrix} = \begin{pmatrix} \frac{1}{3} & \frac{1}{6} \\ \frac{1}{6} & \frac{1}{3} \end{pmatrix}^{-1} \begin{pmatrix} 1 & 0 \\ 0 & 1 \end{pmatrix} \begin{pmatrix} \phi_{i-1} \\ \phi_i \end{pmatrix} \tag{4.16}$$

$$= \begin{pmatrix} 2 & -1 \\ -1 & 2 \end{pmatrix} \begin{pmatrix} 1 & 0 \\ 0 & 1 \end{pmatrix} \begin{pmatrix} \phi_{i-1} \\ \phi_i \end{pmatrix}, \tag{4.17}$$

which gives

$$\begin{pmatrix} \psi_{i-1} \\ \psi_i \end{pmatrix} = \begin{pmatrix} 2\phi_{i-1} - \phi_i \\ 2\phi_i - \phi_{i-1} \end{pmatrix}. \tag{4.18}$$

Note that the basis function $\psi$ takes a form which is non-zero on the element boundaries, as illustrated in Figure 4.2.

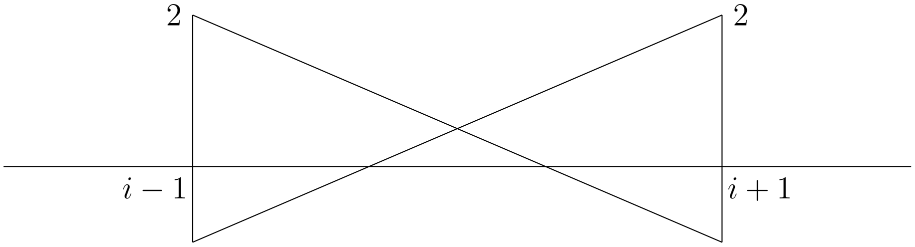


Figure 4.2: Basis Function $\psi_i^{\pm}$ in Element $i - \frac{1}{2}$

### 4.2.4 Scheme B

The weak form of the momentum equation, for an element, using the Petrov-Galerkin basis function from [100], is stated in eqn. (4.14). Using the hat basis function, $\phi$, in eqn. (4.6) and the new basis function, $\psi$, from eqn. (4.18), the system for an element becomes

$$m_{i-\frac{1}{2}} \begin{pmatrix} \int_{\xi_{i-1}}^{\xi_i} \phi_{i-1}\psi_{i-1} d\xi & \int_{\xi_{i-1}}^{\xi_i} \phi_i \psi_{i-1} d\xi \\ \int_{\xi_{i-1}}^{\xi_i} \phi_{i-1}\psi_i d\xi & \int_{\xi_{i-1}}^{\xi_i} \phi_i \psi_i d\xi \end{pmatrix} \begin{pmatrix} \dot{u}_{i-1} \\ \dot{u}_i \end{pmatrix} \tag{4.19}$$

$$= - \begin{pmatrix} p(2\phi_{i-1} - \phi_i)\Big|_{\xi_{i-1}}^{\xi_i} \\ p(2\phi_i - \phi_{i-1})\Big|_{\xi_{i-1}}^{\xi_i} \end{pmatrix} + p_{i-\frac{1}{2}} \begin{pmatrix} \int_{\xi_{i-1}}^{\xi_i} (\psi_\xi)_{i-1} d\xi \\ \int_{\xi_{i-1}}^{\xi_i} (\psi_\xi)_i d\xi \end{pmatrix} \tag{4.20}$$

giving

$$m_{i-\frac{1}{2}}h\begin{pmatrix}\frac{1}{2} & 0\\ 0 & \frac{1}{2}\end{pmatrix}\begin{pmatrix}\dot{u}_{i-1}\\ \dot{u}_i\end{pmatrix} \tag{4.21}$$

$$= -\begin{pmatrix}-\frac{3}{2}p_{i-\frac{1}{2}} - p_{i-\frac{3}{2}} - \frac{1}{2}p_{i+\frac{1}{2}}\\ \frac{3}{2}p_{i-\frac{1}{2}} + \frac{1}{2}p_{i-\frac{3}{2}} + p_{i+\frac{1}{2}}\end{pmatrix} + p_{i-\frac{1}{2}}\begin{pmatrix}-3\\ 3\end{pmatrix}. \tag{4.22}$$

Note the element mass matrix is diagonal. For the pressures at the element boundaries, we assume an average of the pressures in neighbouring cells, i.e.

$$p_{i-1} = \frac{1}{2}(p_{i-\frac{1}{2}} + p_{i-\frac{3}{2}})$$

and

$$p_i = \frac{1}{2}(p_{i-\frac{1}{2}} + p_{i+\frac{1}{2}}).$$

Compared with the right hand side of the scheme from eqn. (4.12) in Section 4.2.1, this provides a different stencil for the pressure term expressed as,

$$m_{i-\frac{1}{2}}h\begin{pmatrix}\frac{1}{2} & 0\\ 0 & \frac{1}{2}\end{pmatrix}\begin{pmatrix}\dot{u}_{i-1}\\ \dot{u}_i\end{pmatrix} \tag{4.23}$$

$$= \begin{pmatrix}-\frac{3}{2}p_{i-\frac{1}{2}} + p_{i-\frac{3}{2}} + \frac{1}{2}p_{i+\frac{1}{2}}\\ \frac{3}{2}p_{i-\frac{1}{2}} - \frac{1}{2}p_{i-\frac{3}{2}} - p_{i+\frac{1}{2}}\end{pmatrix}. \tag{4.24}$$

After assembly the scheme is

$$\dot{u}_i = \frac{-p_{i-\frac{3}{2}} + 5p_{i-\frac{1}{2}} - 5p_{i+\frac{1}{2}} + p_{i+\frac{3}{2}}}{2m_ih}, \tag{4.25}$$

referenced in this work as Scheme B.

Notice that since the alternative basis function, $\psi$, does not vanish at boundaries, element boundary terms are included in the approximation. Other approximations could be generated using different assumptions of the pressure at the boundaries.

Since we now have an alternative discretisation to the momentum equation it can be used to estimate an error.

We have obtained a Finite Difference (FD) scheme by applying a FEM. Although this may seem an elaborate way of obtaining a FD approximation this approach is useful when extending the problem to 2D.

## 4.3 Local Truncation Error

The definition for the LTE has already been given in Section 2.18 for the general case, and Section 3.4 for the semi-discrete in time case. Here, we deal with eqn. (4.2), which is written for convenience as

$$\frac{\partial p}{\partial \xi} = -m\dot{u}. \tag{4.26}$$

The differential operator for the momentum equation is expressed as,

$$S(p) \quad = \quad \frac{\partial p}{\partial \xi} + m\dot{u} = 0, \tag{4.27}$$

where $p = p(\xi)$, $\dot{u} = \dot{u}(\xi)$ and $m = m(\xi)$. The semi-discrete in space numerical schemes approximating the differential equation from Section 4.2.1 and Section 4.2.3 are defined as $S_h$, for eqn. (4.27).

The truncation error in one step for a spatial semi-discrete scheme is defined as

$$\tau_i(S_h) = S_h(p(\xi_i)) - S_h(p_i), \tag{4.28}$$

where $S_h(p_i) = 0$ from eqn. (4.27).

### 4.3.1 Scheme A

From eqn. (4.27), we have the spatial differential equation for $p(\xi)$,

$$S(p) = \frac{\partial p}{\partial \xi} + m\dot{u} = 0, \tag{4.29}$$

while from eqn. (4.12) we have a numerical approximation to $p_i$,

$$S_h(p_i) = \frac{p_{i+\frac{1}{2}} - p_{i-\frac{1}{2}}}{h} + m_i\dot{u}_i = 0. \tag{4.30}$$

To obtain the LTE apply eqn. (4.30) to the exact solution at the spatial point $\xi_i$, giving

$$S_h(p(\xi_i)) = \frac{p(\xi_i + \frac{h}{2}) - p(\xi_i - \frac{h}{2})}{h} + m(\xi_i)\dot{u}(\xi_i). \tag{4.31}$$

Using the definition of the truncation error, as stated in eqn.(4.28), along with eqn. (4.30) and eqn. (4.31), where $\tau_i(S_a)$ is the LTE to eqn. (4.30) we obtain,

$$\tau_i(S_a) = \frac{p(\xi_i + \frac{h}{2}) - p(\xi_i - \frac{h}{2})}{h} + m(\xi_i)\dot{u}(\xi_i). \tag{4.32}$$

Using the exact differential equation, eqn. (4.26),

$$m(\xi_i)\dot{u}(\xi_i) = -p_\xi(\xi_i),$$

and expanding $p(\xi_i + \frac{h}{2})$ and $p(\xi_i - \frac{h}{2})$ in the above about $p(\xi_i) = p$ by Taylor series, where

$$p\left(\xi_i + \frac{h}{2}\right) = p + \frac{h}{2}p_\xi + \frac{1}{2!}\left[\frac{h^2}{4}p_{\xi\xi}\right] + \frac{1}{3!}\left[\frac{h^3}{8}p_{\xi\xi\xi}\right] + .. \tag{4.33}$$

$$p\left(\xi_i - \frac{h}{2}\right) = p - \frac{h}{2}p_\xi + \frac{1}{2!}\left[\frac{h^2}{4}p_{\xi\xi}\right] - \frac{1}{3!}\left[\frac{h^3}{8}p_{\xi\xi\xi}\right] + .. \tag{4.34}$$

the local truncation error becomes

$$\begin{aligned} h\tau_i(S_a) &= p + \frac{h}{2}p_\xi + \frac{1}{2!}\left[\frac{h^2}{4}p_{\xi\xi}\right] + \frac{1}{3!}\left[\frac{h^3}{8}p_{\xi\xi\xi}\right] \\ &- \left(p - \frac{h}{2}p_\xi + \frac{1}{2!}\left[\frac{h^2}{4}p_{\xi\xi}\right] - \frac{1}{3!}\left[\frac{h^3}{8}p_{\xi\xi\xi}\right]\right) - hp_\xi + \cdots \\ &= \frac{h^3}{24}p_{\xi\xi\xi} + \mathcal{O}(h^4) \\ \Rightarrow \tau_i(S_a) &= \frac{h^2}{24}p_{\xi\xi\xi} + \mathcal{O}(h^3). \end{aligned} \tag{4.35}$$

Hence, the scheme is of second order.

### 4.3.2 Scheme B

From eqn. (4.27) we have the spatial differential equation

$$S(p) = \frac{\partial p}{\partial \xi} + m\dot{u} = 0.$$

From eqn. (4.25) we have a numerical approximation to $p$ as

$$S_h(p_i) = \frac{p_{i-\frac{3}{2}} - 5p_{i-\frac{1}{2}} + 5p_{i+\frac{1}{2}} - p_{i+\frac{3}{2}}}{2h} + m_i\dot{u}_i = 0. \tag{4.36}$$

To obtain the LTE apply eqn. (4.36) to the exact solution at the spatial point $\xi_i$, giving

$$S_h(p(\xi_i)) = \frac{p(\xi_i - \frac{3h}{2}) - 5p(\xi_i - \frac{h}{2}) + 5p(\xi_i + \frac{h}{2}) - p(\xi_i + \frac{3h}{2})}{2h} + m(\xi_i)\dot{u}(\xi_i). \tag{4.37}$$

The LTE of eqn. (4.36) is defined as $\tau_i(S_b)$, when using eqn.(4.28), this becomes

$$\tau_i(S_b) = S_h(p(\xi_i)) - S_h(p_i). \tag{4.38}$$

Using eqn. (4.36) and eqn.(4.37), eqn. (4.38) becomes

$$\tau_i(S_b) = \frac{p(\xi_i - \frac{3h}{2}) - 5p(\xi_i - \frac{h}{2}) + 5p(\xi_i + \frac{h}{2}) - p(\xi_i + \frac{3h}{2})}{2h} + m(\xi_i)\dot{u}(\xi_i).$$

Using the exact differential equation, eqn. (4.26) at $\xi_i$, i.e.

$$m(\xi_i)\dot{u}(\xi_i) = -p_\xi(\xi_i),$$

and the expansions from eqn. (4.33) and eqn. (4.34), along with $p(\xi_i + \frac{3h}{2})$ and $p(\xi_i - \frac{3h}{2})$ in the above, expanded about $p(\xi_i) = p$ by Taylor series, where

$$p\left(\xi_i + \frac{3h}{2}\right) = p + \frac{3h}{2}p_\xi + \frac{1}{2!}\left[\frac{9h^2}{4}p_{\xi\xi}\right] + \frac{1}{3!}\left[\frac{27h^3}{8}p_{\xi\xi\xi}\right] + .. \tag{4.39}$$

$$p\left(\xi_i - \frac{3h}{2}\right) = p - \frac{3h}{2}p_\xi + \frac{1}{2!}\left[\frac{9h^2}{4}p_{\xi\xi}\right] - \frac{1}{3!}\left[\frac{27h^3}{8}p_{\xi\xi\xi}\right] + .., \tag{4.40}$$

the local truncation error, $\tau_i(S_b)$, is given by

$$\begin{aligned}
2h\tau_i(S_b) &= p - \frac{3h}{2}p_\xi + \frac{1}{2!}\left[\frac{9h^2}{4}p_{\xi\xi}\right] - \frac{1}{3!}\left[\frac{27h^3}{8}p_{\xi\xi\xi}\right] \\
&- 5\left(p - \frac{h}{2}p_\xi + \frac{1}{2!}\left[\frac{h^2}{4}p_{\xi\xi}\right] - \frac{1}{3!}\left[\frac{h^3}{8}p_{\xi\xi\xi}\right]\right) \\
&+ 5\left(p + \frac{h}{2}p_\xi + \frac{1}{2!}\left[\frac{h^2}{4}p_{\xi\xi}\right] + \frac{1}{3!}\left[\frac{h^3}{8}p_{\xi\xi\xi}\right]\right) \\
&- \left(p + \frac{3h}{2}p_\xi + \frac{1}{2!}\left[\frac{9h^2}{4}p_{\xi\xi}\right] + \frac{1}{3!}\left[\frac{27h^3}{8}p_{\xi\xi\xi}\right]\right) \\
&- 2hp_\xi + \cdots \\
&= hp_\xi\left(-\frac{3}{2} + \frac{5}{2} + \frac{5}{2} - \frac{3}{2} - 2\right) + h^2p_{\xi\xi}\left(\frac{9}{8} - \frac{5}{8} + \frac{5}{8} - \frac{9}{8}\right) \\
&+ h^3p_{\xi\xi\xi}\left(-\frac{27}{48} + \frac{5}{48} + \frac{5}{48} - \frac{27}{48}\right) + \mathcal{O}(h^4) \\
&= -\frac{11}{12}h^3p_{\xi\xi\xi} + \mathcal{O}(h^4) \\
\Rightarrow \tau_i(S_b) &= -\frac{11h^2}{24}p_{\xi\xi\xi} + \mathcal{O}(h^3). \qquad (4.41)
\end{aligned}$$

Hence, this scheme is also of second order.

## 4.4 Error Estimation

As we have seen in Section 3.5, we can obtain a computable error estimate in one step when we have two schemes of the same order.

We now obtain a spatial error estimate for the pressure in one step.

### 4.4.1 Estimated Spatial Error for Pressure in One Step

From Section 4.3.1 and Section 4.3.2, we have the leading terms of the LTE for the pressure as

$$p(\xi_i + h) - (p_a)_{i+1} = h\tau_n(S_a) = \frac{h^3}{24}p_{\xi\xi\xi}(\xi_n) + \mathcal{O}(h^4) \qquad (4.42)$$

$$p(\xi_i + h) - (p_b)_{i+1} = h\tau_n(S_b) = -\frac{11h^3}{24}p_{\xi\xi\xi}(\xi_n) + \mathcal{O}(h^4) \qquad (4.43)$$

where $(p_a)_{i+1}$ and $(p_b)_{i+1}$ represent Scheme A and Scheme B approximations to $p$, respectively. Subtract eqn. (4.42) from eqn. (4.43)

$$\begin{aligned} (p_b)_{i+1} - (p_a)_{i+1} &= \frac{12h^3}{24} p_{\xi\xi\xi}(\xi_i) + \mathcal{O}(h^4) \\ \Rightarrow h^3 p_{\xi\xi\xi}(\xi_i) &= 2\left((p_b)_{i+1} - (p_a)_{i+1}\right) + \mathcal{O}(h^4). \end{aligned} \tag{4.44}$$

Resulting in a computable estimate to the uncomputable third derivative of $p$, this can be substituted into the local error term, eqn. (4.42), to give

$$p(\xi_i + h) - (p_a)_{i+1} = \frac{(p_b)_{i+1} - (p_a)_{i+1}}{12} + \mathcal{O}(h^4), \tag{4.45}$$

leading to the truncation error, for the derivative $p_\xi$. In one step this is estimated by

$$\tau_n(S_a) = \frac{(p_b)_{i+1} - (p_a)_{i+1}}{12h} + \mathcal{O}(h^3). \tag{4.46}$$

Hence, we have a computable local error due to the information introduced from both schemes.

## 4.5 Semi-Discrete in Space Error Estimate for the Density

In Section 4.4 we obtained an estimation to the LTE for pressure in one spatial step, i.e. from eqn. (4.45),

$$e[p] = p(\xi_i + h) - (p_a)_{i+1} = \frac{(p_b)_{i+1} - (p_a)_{i+1}}{12} + \mathcal{O}(h^4). \tag{4.47}$$

Since pressure and density are related, in the case of the Isothermal equations, by

$$p = a^2 \rho,$$

from eqn. (2.30), we are able to obtain an estimate to the LTE of density in one spatial step, simply by dividing eqn. (4.47) by $a^2$, resulting in

$$e[\rho] = \rho(\xi_i + h) - (\rho_a)_{i+1} = \frac{(\rho_b)_{i+1} - (\rho_a)_{i+1}}{12a^2} + \mathcal{O}(h^4). \tag{4.48}$$

We now look at the additional spatial derivative in the system of three equations.

## 4.6 System of Three Equations

The Lagrangian system of three equations contains spatial derivatives in the momentum equation and the specific internal energy equation.

We have seen in, Section 4.1, how we deal with the spatial derivatives in the 1D momentum equation, see eqn. (4.2).

The specific internal energy equation, written in its 2D full form, in Lagrangian co-ordinates, from eqn. (2.28) is

$$\rho|\boldsymbol{J}|\frac{d\varepsilon}{dt} = -p\boldsymbol{C}\nabla\cdot\boldsymbol{u}, \tag{4.49}$$

where $\varepsilon = \varepsilon(\boldsymbol{\xi}, t)$, $m = m(\boldsymbol{\xi})$, $\boldsymbol{u} = \boldsymbol{u}(\boldsymbol{\xi}, t)$, $p = p(\boldsymbol{\xi}, t)$ and $\nabla$ is the spatial derivative in terms of the Lagrangian co-ordinates $\boldsymbol{\xi}$.

In 1D, eqn. (4.49) takes the form

$$m\dot{\varepsilon} = -p\frac{\partial u}{\partial \xi}, \tag{4.50}$$

since $C = I$ in 1D, where $m = m(\xi)$, $u = u(\xi)$, $\dot{\varepsilon} = \dot{\varepsilon}(\xi)$ and $p = p(\xi)$.

We now seek a numerical approximation to this equation using a finite element approach.

## 4.7 Weak form of Specific Internal Energy

We multiply eqn. (4.50) by a test function $\omega$ and integrate, which gives

$$\int_{\xi_{i-1}}^{\xi_i} \omega m\dot{\varepsilon}d\xi = -\int_{\xi_{i-1}}^{\xi_i} \omega p\frac{\partial u}{\partial \xi}d\xi. \tag{4.51}$$

We look at the discrete form of eqn. (4.51), using the Galerkin method and the Petrov-Galerkin method.

### 4.7.1 Specific Internal Energy using a Galerkin Approach

Integrating by parts over a single element, $(\xi_{i-1}, \xi_i)$, we obtain the weak form of the specific internal energy equation. We choose $p$ to be piecewise constant over the element, such that

$p$ can be taken outside the integral and $\frac{\partial p}{\partial \xi} = 0$, hence eqn. (4.51) becomes

$$\int_{\xi_{i-1}}^{\xi_i} \omega m \dot{\varepsilon} d\xi = p_{i-\frac{1}{2}} \left( -u\omega \Big|_{\xi_{i-1}}^{\xi_i} + \int_{\xi_{i-1}}^{\xi_i} \omega u \frac{\partial(\omega)}{\partial \xi} d\xi \right). \tag{4.52}$$

For the Galerkin approach in eqn. (4.52), we choose $\omega = \phi_i$, and integrate over an element, $(\xi_{i-1}, \xi_i)$, and let $u$ be a piecewise linear function of the form

$$u = \sum_{j=1}^{N+1} u_j \phi_j(\xi). \tag{4.53}$$

The weak form of the specific internal energy over an element is

$$\int_{\xi_{i-1}}^{\xi_i} \phi_{i-\frac{1}{2}} m_{i-\frac{1}{2}} \dot{\varepsilon}_{i-\frac{1}{2}} d\xi = -p_{i-\frac{1}{2}} \left( u\phi \Big|_{\xi_{i-1}}^{\xi_i} \right) + p_{i-\frac{1}{2}} \int_{\xi_{i-1}} \sum_{j=i-1}^{i} u_i \phi_i \frac{\partial \phi_i}{\partial \xi} d\xi, \tag{4.54}$$

where $m$ and $\dot{\varepsilon}$ are piecewise constant within the element. Interchanging the integral and the sum,

$$m_{i-\frac{1}{2}} \dot{\varepsilon}_{i-\frac{1}{2}} \int_{\xi_{i-1}}^{\xi_i} \phi_i d\xi = -p_{i-\frac{1}{2}} \left( u\phi \Big|_{\xi_{i-1}}^{\xi_i} \right) + p_{i-\frac{1}{2}} \sum_{j=i-1}^{i} u_j \int_{\xi_{i-1}}^{\xi_i} \phi_i \frac{\partial \phi_j}{\partial \xi} d\xi. \tag{4.55}$$

The basis functions are chosen from eqn. (4.6), where their derivatives are,

$$\frac{\partial \phi_i(\xi)}{\partial \xi} = \begin{cases} \frac{1}{h} & \xi_{i-1} < \xi < \xi_i \\ -\frac{1}{h} & \xi_i < \xi < \xi_{i+1} \quad (4.56) \\ 0 & \text{elsewhere}, \end{cases}$$

and $h = \xi_i - \xi_{i-1}$.

Using eqn. (4.6) and eqn. (4.56), the weak form in eqn. (4.55) becomes

$$\frac{h m_{i-\frac{1}{2}} \dot{\varepsilon}_{i-\frac{1}{2}}}{2 p_{i-\frac{1}{2}}} = -\frac{(u_i - u_{i-1})}{2}. \tag{4.57}$$

The boundary terms vanish. Hence eqn. (4.57) can be rewritten as

$$\dot{\varepsilon}_{i-\frac{1}{2}} = -\frac{(u_i - u_{i-1})\, p_{i-\frac{1}{2}}}{h m_{i-\frac{1}{2}}}. \tag{4.58}$$

This is our specific internal energy equation discretised in space using the test function (4.6).

### 4.7.2 Specific Internal Energy using a Petrov Galerkin Approach

We have developed an alternative basis function in order to carry out a Petrov-Galerkin method in Section 4.2.3. This basis function can be applied to eqn. (4.52) by using $\omega = \psi_i$. We integrate over an element on the interval $(\xi_{i-1}, \xi_i)$ and choose $u$ to be a piecewise linear function as eqn. (4.53).

The weak form of the specific internal energy equation, using a Petrov Galerkin approach, in an element becomes

$$m_{i-\frac{1}{2}}\dot{\varepsilon}_{i-\frac{1}{2}} \int_{\xi_{i-1}}^{\xi_i} \psi_i d\xi = p_{i-\frac{1}{2}} \left( -u\psi\Big|_{\xi_{i-1}}^{\xi_i} + \sum_{j=i-1}^{i} u_j \int_{\xi_{i-1}}^{\xi_i} \phi_i \frac{\partial \psi_j}{\partial \xi} d\xi \right). \tag{4.59}$$

The basis function for $\phi_i$ is given by eqn. (4.6), whereas $\psi_j$ is given by eqn. (4.18), the derivative is expressed as

$$\frac{\partial \psi_i(\xi)}{\partial \xi} = \begin{cases} \frac{3}{h} & \xi_{i-1} < \xi < \xi_i \\ -\frac{3}{h} & \xi_i < \xi < \xi_{i+1} \\ 0 & \text{elsewhere.} \end{cases} \tag{4.60}$$

Once computed, the weak form in eqn. (4.59) becomes

$$\begin{aligned} \frac{h}{2} m_{i-\frac{1}{2}}\dot{\varepsilon}_{i-\frac{1}{2}} &= p_{i-\frac{1}{2}} \left( 2(u_{i-1} - u_i) + \frac{3}{2}(u_i - u_{i-1}) \right) \\ \Rightarrow \frac{h}{2} m_{i-\frac{1}{2}}\dot{\varepsilon}_{i-\frac{1}{2}} &= -\frac{p_{i-\frac{1}{2}}}{2} (u_i - u_{i-1}) \\ \Rightarrow \dot{\varepsilon}_{i-\frac{1}{2}} &= -\frac{(u_i - u_{i-1})\, p_{i-\frac{1}{2}}}{h m_{i-\frac{1}{2}}}. \end{aligned} \tag{4.61}$$

This is exactly the same discretisation as seen in eqn. (4.58). Since the velocity values $u$ are already held on the boundary we do not require an averaging process here, as we did for the momentum case. Therefore the stencil is not spread differently.

## 4.8 Semi-Discrete in Space Estimated Error for Pressure

From the previous result eqn. (4.47), we have the approximated LTE of the variable pressure in one spatial step,

$$e[p] = p(\xi_i + h) - (p_a)_{i+1} = \frac{(p_b)_{i+1} - (p_a)_{i+1}}{12} + \mathcal{O}(h^4). \tag{4.62}$$

We do not use this solution to obtain the approximated LTE for density in one spatial step, as we did previously for the system of two equations. This is because the pressure is related to the density in a different way for the system of three equations, i.e.

$$p = (\gamma - 1)\rho\varepsilon.$$

Although we have an estimated error for $p$ in one spatial step, we do not have the required information for $\varepsilon$, due to the two schemes having the same leading term in the LTE, as is seen in Section 4.7.2. Therefore, we restrict our results to the approximated LTE for pressure in one spatial step, see eqn. (4.62).

## 4.9 Chapter 4 Summary

In this chapter we have introduced a semi-discrete form of the spatial derivatives. This has been done for a system of two and three equations.

The approximations to the spatial derivatives are obtained through a Galerkin approach, using standard FEMs. Additionally, an alterntive approximation is developed in this work, which is of a Petrov-Galerkin form.

Using the variation in the approximations, we calculate the LTE for a system of two and three equations. Along with Milne's approach, we obtain an error estimate.

An error estimate for density and pressure is obtained for the system of two equations, whereas only a pressure error estimate is obtained for the system of three equations.

We shall use the above results to progress the investigation onto the fully-discrete form, along with the results from Chapter 3.

# Chapter 5

# Fully-Discrete Schemes in 1D

We have seen in Chapters 3 and 4, how truncation errors for the system of two and three equations in 1D give rise to error estimates for the temporal and spatial derivatives separately. In Chapter 3, we obtained the LTE for the Lagrangian equation $\frac{dx}{dt} = u$ for two temporal schemes, which were then used to obtain the LTE in one time step, for the variable $x$. From this, an estimate of the leading term of the LTE for the variable $x$ was found in terms of the computed $x$ values, obtained from both schemes.

Similarly, in Chapter 4, we obtained the LTE for the two spatial schemes of the momentum equation. This was used to obtain an estimate of the leading term of the LTE in terms of computed $p$ values from both schemes.

The results are extended to obtain the fully-discrete case for the system of two equations and the system of three equations.

## 5.1 Fully-Discrete Error Estimate System of Two Equations

As demonstrated in Section 3.6.1, we are able to obtain an estimate to the LTE of density and velocity using the LTE error of $x$ and $\frac{dx}{dt}$ respectively.

The fully-discrete form of the differential equation, using eqn. (3.31) and eqn. (4.12), for Scheme 1 in time and Scheme A in space, is given by

$$L_h(x_i^n) = \frac{x_i^{n+1} - x_i^n}{k} - u_i^n + \frac{k}{2hm_i}\left(p_{i+\frac{1}{2}}^{n+\frac{1}{2}} - p_{i-\frac{1}{2}}^{n+\frac{1}{2}}\right). \tag{5.1}$$

We can make use of the LTE errors obtained in Sections 3.4.1 and 4.3. From eqn.(3.39) we have

$$\tau(L_1) = \frac{x(t^n + k) - x(t^n)}{k} - u(t^n) - \frac{k}{2} f\left(t^n + \frac{k}{2}\right), \tag{5.2}$$

once expanded, we see the LTE from eqn. (3.45) becomes

$$\tau_n(L_1) = -\frac{k^2}{12} x''' + \mathcal{O}(k^3). \tag{5.3}$$

From eqn. (4.32), we have

$$\tau(S_a) = \frac{p(\xi_i + \frac{h}{2}) - p(\xi_i - \frac{h}{2})}{h} + m(\xi_i)\dot{u}(\xi_i), \tag{5.4}$$

once expanded from eqn. (4.35), we have

$$\tau(S_a) = \frac{h^2}{24} p_{\xi\xi\xi} + \mathcal{O}(h^3). \tag{5.5}$$

By substituting eqn. (5.3) and eqn. (5.5) into eqn. (5.1), we obtain the fully-discrete LTE of the differential equation

$$\frac{\partial x}{\partial t} = u,$$

For one spatial step the error takes the form

$$\tau(L_1) = -\frac{k^2}{12} x''' + \frac{h^2 k}{48m} p_{\xi\xi\xi} + \mathcal{O}(k^3) + \mathcal{O}(kh^3). \tag{5.6}$$

From eqn. (3.53), an estimate to the LTE error was obtained using

$$k^3 x_{ttt} = 4\left((x_1)^{n+1} - (x_2)^{n+1}\right),$$

and from eqn. (4.44), we have

$$h^3 p_{\xi\xi\xi} = 2\left((p_b)_{i+1} - (p_a)_{i+1}\right).$$

Therefore, eqn. (5.6) becomes

$$\tau(L_1) = -\frac{(x_2)^{n+1} - (x_1)^{n+1}}{3k} + \frac{k\left((p_b)_{i+1} - (p_a)_{i+1}\right)}{24hm} + \mathcal{O}(k^3) + \mathcal{O}(kh^3). \tag{5.7}$$

From this we obtain an estimated LTE for the variable $x$ in terms of computable values, such that

$$\begin{aligned} e[x] &= x(\xi_i, t^n + k) - (x_1)_i^{n+1} = k\tau(L_1) \\ &= -\frac{\left((x_2)^{n+1} - (x_1)^{n+1}\right)}{3} + \frac{k^2\left((p_b)_{i+1} - (p_a)_{i+1}\right)}{24hm} + \mathcal{O}(k^3) + \mathcal{O}(kh^3). \end{aligned} \tag{5.8}$$

The information is available to determine an approximated LTE for density and velocity, from the fully discrete scheme.

### 5.1.1 Estimated LTE for Density

As previously seen, the LTE for density can be obtained by using eqn. (3.58), where

$$\begin{aligned} e[\rho] = \rho(t^n + k)_i - (\rho_s 1)_i^{n+1} &= \frac{m^n}{x_\xi(t^n + k)} - \frac{m^n}{(x_\xi)^{n+1}} \\ &= \frac{m^n\left((x_\xi)^{n+1} - x_\xi(t^n + k)\right)}{x_\xi(t^n + k)\,(x_\xi)^{n+1}}. \end{aligned} \tag{5.9}$$

By differentiating eqn. (5.8) with respect to $\xi$, we obtain

$$\begin{aligned} e[x] &= x_\xi(\xi_i, t^n + k) - \left((x_1)_\xi\right)_i^{n+1} = k\tau(L_1) \\ &= -\frac{\left(\left((x_2)_\xi\right)^{n+1} - \left((x_1)_\xi\right)^{n+1}\right)}{3} + \frac{k^2\left(\left((p_b)_\xi\right)_{i+1} - \left((p_a)_\xi\right)_{i+1}\right)}{24hm} \\ &+ \mathcal{O}(k^3) + \mathcal{O}(kh^3), \end{aligned} \tag{5.10}$$

this can be substituted into eqn. (5.9) to obtain

$$e[\rho] = \rho_i(t^n + k) - (\rho_1)_i^{n+1}$$

$$= -\frac{m^n\left(-\frac{\left(\left((x_2)_\xi\right)^{n+1} - \left((x_1)_\xi\right)^{n+1}\right)}{3} + \frac{k^2\left(\left((p_b)_\xi\right)_{i+1} - \left((p_a)_\xi\right)_{i+1}\right)}{24hm}\right)}{\left(\left((x_1)_\xi\right)^{n+1} - \frac{\left(\left((x_2)_\xi\right)^{n+1} - \left((x_1)_\xi\right)^{n+1}\right)}{3} + \frac{k^2\left(\left((p_b)_\xi\right)_{i+1} - \left((p_a)_\xi\right)_{i+1}\right)}{24hm}\right)\left((x_1)_\xi\right)^{n+1}}. \tag{5.11}$$

### 5.1.2 Estimated LTE for Velocity

From eqn. (5.7), we have

$$\begin{aligned} e\left[\dot{x}\right] &= \tau(L_1) \\ &= -\frac{\left((x_2)^{n+1} - (x_1)^{n+1}\right)}{3k} + \frac{k\left((p_b)_{i+1} - (p_a)_{i+1}\right)}{24hm} + \mathcal{O}(k^3) + \mathcal{O}(kh^3). \end{aligned} \tag{5.12}$$

Using eqn. (3.4), where

$$\frac{\partial x}{\partial t} = u$$

we obtain the estimated LTE for the variable $u$ in terms of computable values, such that

$$\begin{aligned} e[u] &= u(\xi_i, t^n + k) - (u_1)_i^{n+1} \\ &= -\frac{\left((x_2)^{n+1} - (x_1)^{n+1}\right)}{3k} + \frac{k\left((p_b)_{i+1} - (p_a)_{i+1}\right)}{24hm} + \mathcal{O}(k^3) + \mathcal{O}(kh^3). \end{aligned} \tag{5.13}$$

## 5.2 Fully-Discrete Estimated Error in a System of Three Equations

The fully discrete approximated LTE from eqn. (5.6), is

$$\tau(L_1) = -\frac{k^2}{12}x''' + \frac{h^2 k}{48m}p_{\xi\xi\xi} + \mathcal{O}(k^3) + \mathcal{O}(kh^3). \tag{5.14}$$

A computable estimation to the LTE error can be obtained by using eqn. (3.80)

$$k^3 x_{ttt} = 4\left((x_2)^{n+1} - (x_1)^{n+1}\right)$$

and eqn. (4.44)

$$h^3 p_{\xi\xi\xi} = 2\left((p_b)_{i+1} - (p_a)_{i+1}\right)$$

resulting in

$$\tau(L_1) = -\frac{\left((x_1)^{n+1} - (x_2)^{n+1}\right)}{3k} + \frac{k\left((p_b)_{i+1} - (p_a)_{i+1}\right)}{24hm} + \mathcal{O}(k^3) + \mathcal{O}(kh^3). \tag{5.15}$$

From this we determine the LTE for the variable $x$ by

$$k\tau(L_1) = x(t^n+h)-(x_1)^{n+1} = -\frac{\left((x_1)^{n+1}-(x_2)^{n+1}\right)}{3}+\frac{k^2\left((p_b)_{i+1}-(p_a)_{i+1}\right)}{24hm}+\mathcal{O}(k^3)+\mathcal{O}(kh^3). \tag{5.16}$$

The above can be used in obtaining an estimation to the LTE, for density and velocity.

### 5.2.1 Estimated LTE for Density

Using eqn. (3.58), we require eqn. (5.16) differentiated by $\xi$. Thus, resulting in,

$$e[x_\xi] = x_\xi(t^n+h) - \left((x_1)_\xi\right)^{n+1} =$$

$$-\frac{\left(\left((x_1)_\xi\right)^{n+1}-\left((x_2)_\xi\right)^{n+1}\right)}{3}+\frac{k^2\left(\left((p_b)_\xi\right)_{i+1}-\left((p_a)_\xi\right)_{i+1}\right)}{24hm}+\mathcal{O}(k^3)+\mathcal{O}(kh^3). \tag{5.17}$$

Substituting the above into eqn. (3.58) we obtain

$$e[\rho] = \rho_i(t^n+k) - (\rho_1)_i^{n+1}$$

$$= -\frac{m^n\left(-\frac{\left(\left((x_1)_\xi\right)^{n+1}-\left((x_2)_\xi\right)^{n+1}\right)}{3}+\frac{k^2\left(\left((p_b)_\xi\right)_{i+1}-\left((p_a)_\xi\right)_{i+1}\right)}{24hm}\right)}{\left(\left((x_1)_\xi\right)^{n+1}-\frac{\left(\left((x_1)_\xi\right)^{n+1}-\left((x_2)_\xi\right)^{n+1}\right)}{3}+\frac{k^2\left(\left((p_b)_\xi\right)_{i+1}-\left((p_a)_\xi\right)_{i+1}\right)}{24hm}\right)\left((x_1)_\xi\right)^{n+1}}. \tag{5.18}$$

### 5.2.2 Estimated LTE for Velocity

From eqn. (5.7) we have

$$e\left[\dot{x}\right] = \tau(L_1)$$

$$= -\frac{\left((x_1)^{n+1}-(x_2)^{n+1}\right)}{3k}+\frac{k\left((p_b)_{i+1}-(p_a)_{i+1}\right)}{24hm}+\mathcal{O}(k^3)+\mathcal{O}(kh^3). \tag{5.19}$$

Using eqn. (3.4),

$$\dot{x} = u,$$

we obtain the approximated LTE for velocity, such that

$$e[u] = u(\xi_i, t^n+k) - (u_1)_i^{n+1}$$

$$= -\frac{\left((x_1)^{n+1} - (x_2)^{n+1}\right)}{3k} + \frac{k\left((p_b)_{i+1} - (p_a)_{i+1}\right)}{24hm} + \mathcal{O}(k^3) + \mathcal{O}(kh^3). \tag{5.20}$$

## 5.3 Chapter 5 Summary

This chapter consists of combining the results from Chapters 3 and 4. The combinations provide error estimates for the system of equations in their fully-discrete form.

We have successfully obtained error estimates for density and velocity, for both two and three systems of equations.

Using these results we determine how well the error approximation captures the features of the solution. Sod's Shock tube is used as a test problem since an analytic solution is known.

# Chapter 6

# 1D Results

Using the results from Chapter 3, we are able to obtain the results for the semi-discrete in time case for the system of two equations. The variables we investigate are density and velocity. Additionally, we have these error estimates for the Euler equations.

In Chapter 4, we determined an error estimate for density and pressure from the semi-discrete in space case for the system of two equations. We also obtained an error estimate for pressure in the semi-discrete space case for the Euler equations.

Using the information from Chapters 3 and 4, the error estimates are advanced to the fully-discrete case form Chapter 5. We obtained an error estimate for density and velocity for the system of two equations and the Euler equations.

In this chapter, we illustrate the results of the error estimates and determine how well the error estimates behave against a true error. A test case is chosen for the Isothermal equations and the Euler equations, which have analytic solutions associated to them. The test case also exhibits the flow features that we are interested in capturing, i.e. shocks, contacts and expansions. Since an analytic solution exists for the test problems, we are able to compare and contrast against the error estimates obtained in the previous chapters and determine their suitability as a choice of refinement criteria.

## 6.1 Riemann Problem

As a test problem for the 1D case we use a Riemann problem, see [52], to determine how well the estimated error behaves. Although the Riemann problem is a simple test it exhibits all fundamental characteristics of fluid dynamics, i.e. shocks, contact discontinuities and rarefactions. This enables hydrocodes to be tested for their suitability in modeling such features. Since an analytic solution is available for the Riemann problem, for details see [52], we are able to compare the estimated error against the true error of the Main Scheme from [64, 10], and defined in Sections (3.3.1) and (4.2.1).

The initial conditions for the test problem consists of piecewise-constant data with a single discontinuity, which is illustrated in Figure 6.1, with $\rho_l = 1.0$, $p_l = 2.25$, $u_l = 0$, $\rho_r = 0.125$, $p_r = 0.28125$ and $u_r = 0$.

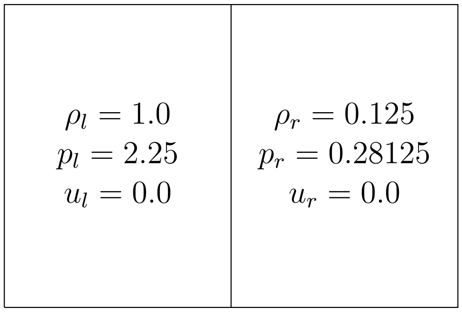


Figure 6.1: 1D Riemann Problem

### 6.1.1 Isothermal Equations

The test case for the Isothermal problem is run to a final time of 0.2 on a mesh of 400 points for the region $[0, 2]$.

The solution consists of three constant states which are connected by a rarefaction fan and a shock, see Figures 6.2 and 6.3.

The numerical scheme used to drive the numerical solution in Figure 6.2, is taken from [64, 10], while the numerical solution in Figure 6.3 is driven by a combination of Scheme 2 and Scheme B. Christensen artificial viscosity has been used with $c_l = 0.5$ and $c_q = 0.75$, and is added to all pressure terms. The analytic solution is based on the solution to the Riemann problem taken from [52]. The Rankine-Hugoniot jump condition is used to form a Hugoniot locus to gather information from the shock waves, and integral curves are used to obtain information for the rarefaction fan.

In Figures 6.2 and 6.3, the numerical solution is illustrated by a dotted red line and the analytic by the solid blue line. We see the top of the expansion fan at $x = 0.70$ and the bottom is at $x = 1.02$. These regions are connected to the shock by an intermediate constant state, where $\rho = 0.35$. The shock front spans $x \in [1.497, 1.499]$.

In the Main Scheme, Sections (3.3.1) and (4.2.1), see Figure 6.2, the solution is in good agreement to the analytic solution, apart from the undershoot at the base of the rarefaction, this is due to "Wall heating", see [10]. Details of Wall heating are beyond the scope of this thesis, however it is discussed in [73].

Scheme 2 is in good agreement to the analytic solution. However, Figure 6.3 illustrates an overshoot before the expansion in the density and an undershoot after shock on the velocity. This is the opposite to the result for the Main Scheme in Figure 6.2.

Scheme 1 and Scheme 2, from Figures 6.2 and 6.3, are both comparable to the analytic solution, and each other. Since the analytic solution is available, we further investigate their suitability, by calculating the error norms, see Table 6.1. The norms are comparable, which demonstrates they are suitable schemes to use to determine an error estimate.

Now that we have demonstrated the suitability of both schemes to be used in estimating the error, we wish to explore the results for the variety of estimated errors we have obtained in the previous chapters.

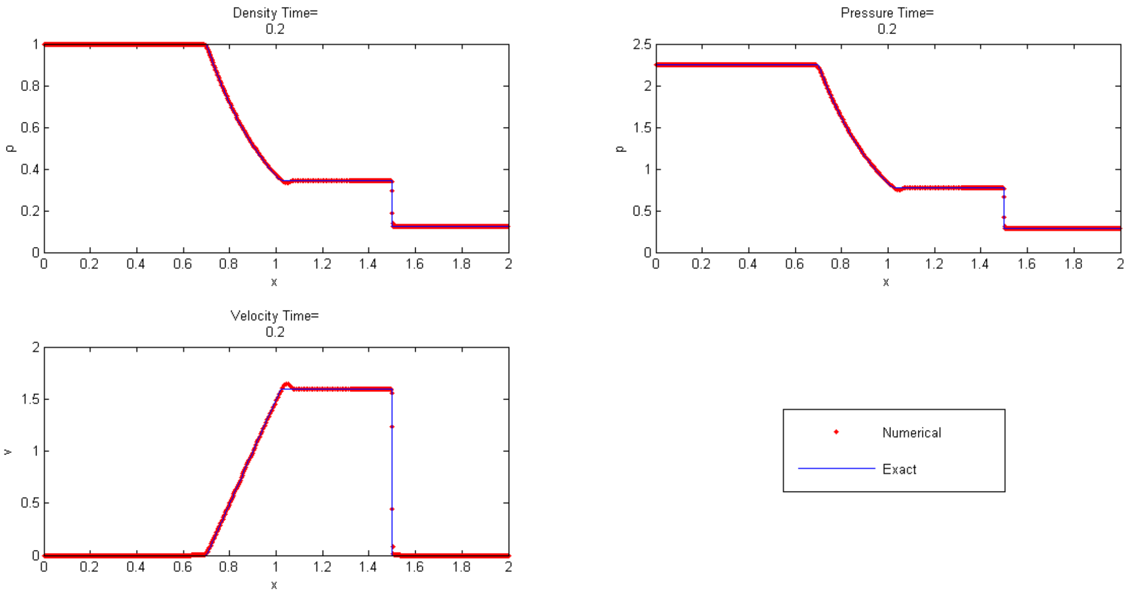


Figure 6.2: Riemann problem, density, pressure and velocity results at at $T = 0.2$ using Main Scheme and analytic solution for the Isothermal Equations on a fine mesh in 1D.

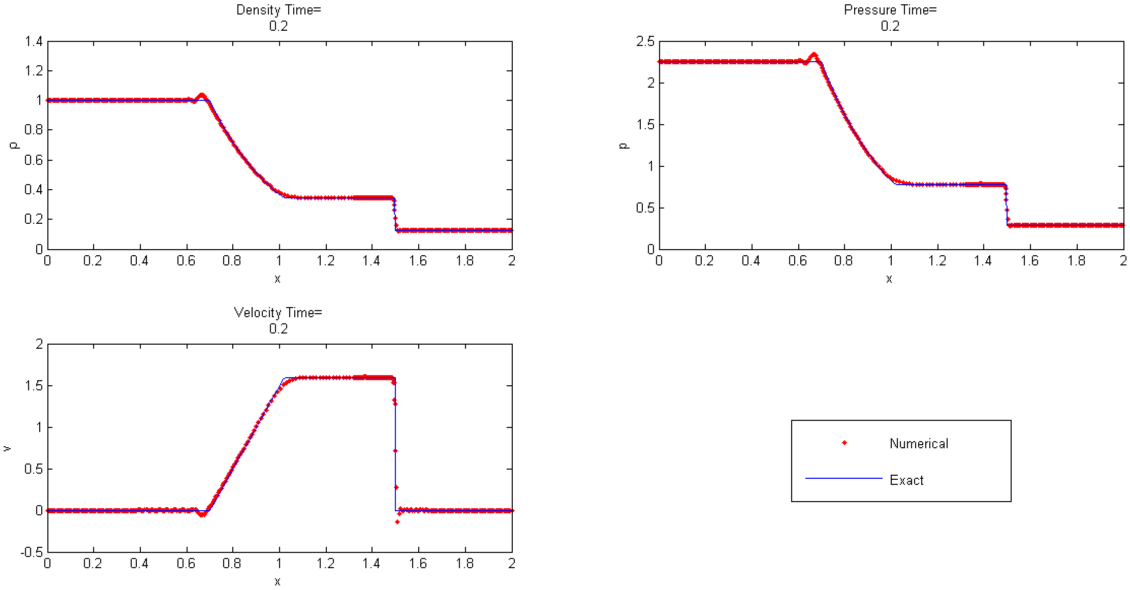


Figure 6.3: Riemann problem, density, pressure and velocity results at $T = 0.2$ using a combination of Scheme 2 and Scheme B against the analytic solution for the Isothermal Equations in 1D on a fine mesh.

Table 6.1: 1D Error norms comparing Scheme 1 and Scheme 2 for the Sod Shock Tube for the Isothermal Equations.

| Calculation | $\|\|e\|\|_1$ | $\|\|e\|\|_2$ | $\|\|e\|\|_\infty$ |
|---|---|---|---|
| Scheme 1 | 0.00296 | 0.14917 | 0.08550 |
| Scheme 2 | 0.00534 | 0.24900 | 0.08417 |

### Comparison of the True Error and Estimated Error

We wish to determine the estimated error for density and velocity. We have been able to obtain a range of these errors by means of a semi-discrete in time scheme, semi-discrete in space scheme and fully-discrete scheme. Note, since velocity is not a conserved variable in space, we do not have the information for a semi-discrete in space scheme for the velocity.
The two features that we are interested in resolving in this problem are the rarefaction and shock. The results below illustrate how well the estimated error captures these features. In order to compare the entire region for both true error and estimated error we look at the results plotted on a log scale.

From Figure 6.4, we see how the semi-discrete in time estimated error in density directly compares to the true error. The magnitude difference at the top and bottom of the fan is visible, there is less variation between the two schemes used to evaluate the estimated error at the top and bottom of the fan, as demonstrated by the two peaks at $x = 0.7$ and $x = 1.02$. The fan is not tracked well by the estimated error. The solution exhibits oscillations before the shock front. However, as desired, the shock front is detected extremely well in position and value.

Figure 6.5 demonstrates the semi-discrete in time estimated error for the velocity. The top and bottom of the fan are detected, where the values are closer to the true error here than in the density case. Additionally, the fan is detected better than the previous case. In front of the shock, between $x = 1.3$ and $x = 1.49$, oscillations are seen in both the true error and estimated error.

It is not clear the reason why oscillations are appearing before the shock front in both the true error and estimated error. Since the oscillations are present in the true error case this indicates the Main Scheme stated in Sections (3.3.1) and (4.2.1), exhibits oscillations at this point. One possible cause of the oscillations is the effect of the artificial viscosity. The artificial viscosity is triggered around the compressible feature where the nodal spacing is small. The artificial viscosity could have an ‘artificial’ effect on the nodes around the shock, therefore causing oscillations. The possible causes of the oscillations should be considered for further work.

The shock front is in excellent agreement in terms of position and strength, once again.

The semi-discrete in space estimated error for density is seen in Figure 6.6. The top and bottom of the fan are detected, however not as well as for the time semi-discrete case. The bottom of the fan is detected better than the top. Again, oscillations are shown before the shock, and yet again the shock is captured extremely well.

The results for the fully-discrete estimated error in density from Figure 6.7, are similar to the semi-discrete in space density results, in the way the bottom of the fan is captured better than the top, and the shock being captured extremely well. However, in this case the estimation lies closer to the true error, especially the oscillations prior to the shock. These are similar to the semi-discrete time only case.

From Figure 6.8, the fully-discrete estimated error for velocity, we see that the bottom of the fan is greater than the top of the fan, which is different to the time semi-discrete case. The fan is not captured well here, in a similar manner to the time semi-discrete case. The oscillations prior to the shock are further from the true error, than they are in the density fully-discrete case. The shock is detected and evaluated very well once again.

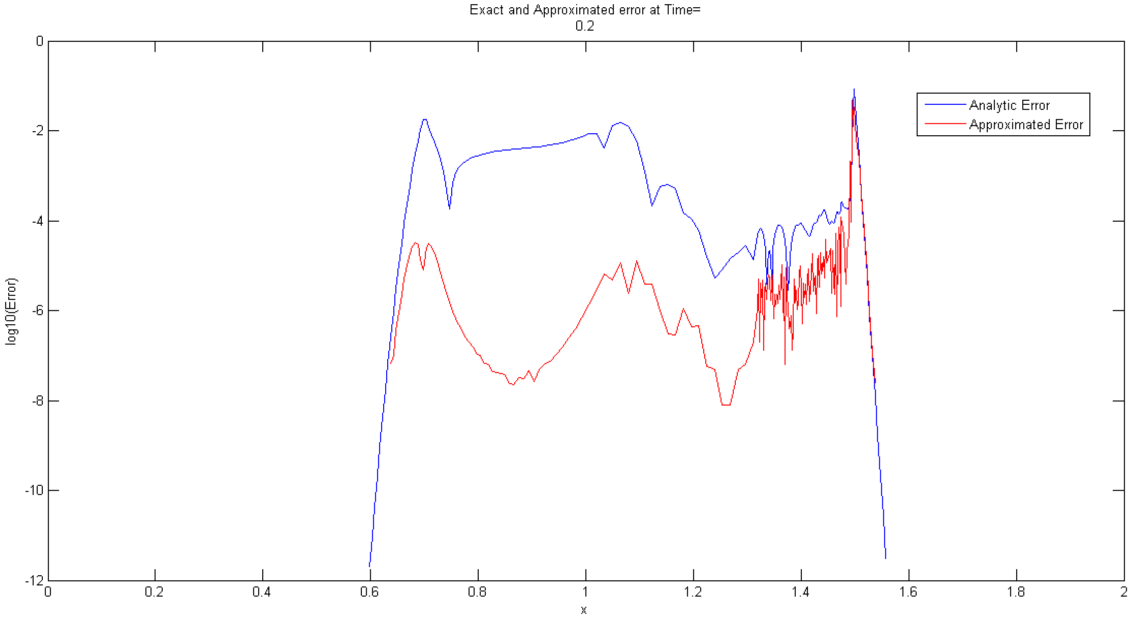


Figure 6.4: 1D Isothermal Equations, Semi-Discrete in Time Density Errors.

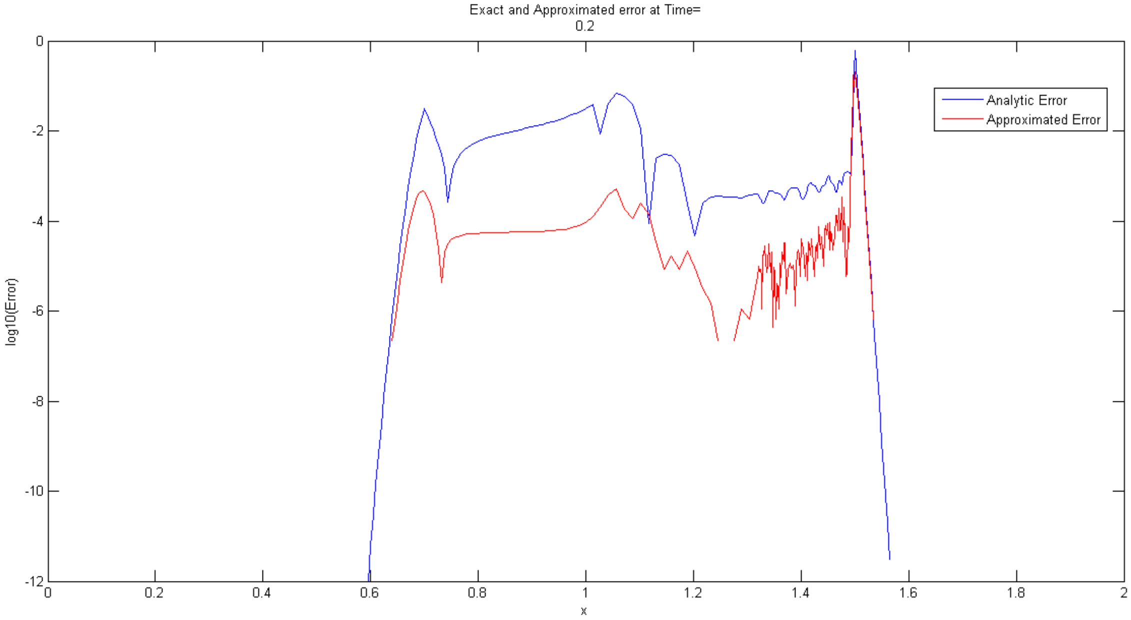


Figure 6.5: 1D Isothermal Equations, Semi-Discrete in Time Velocity Errors.

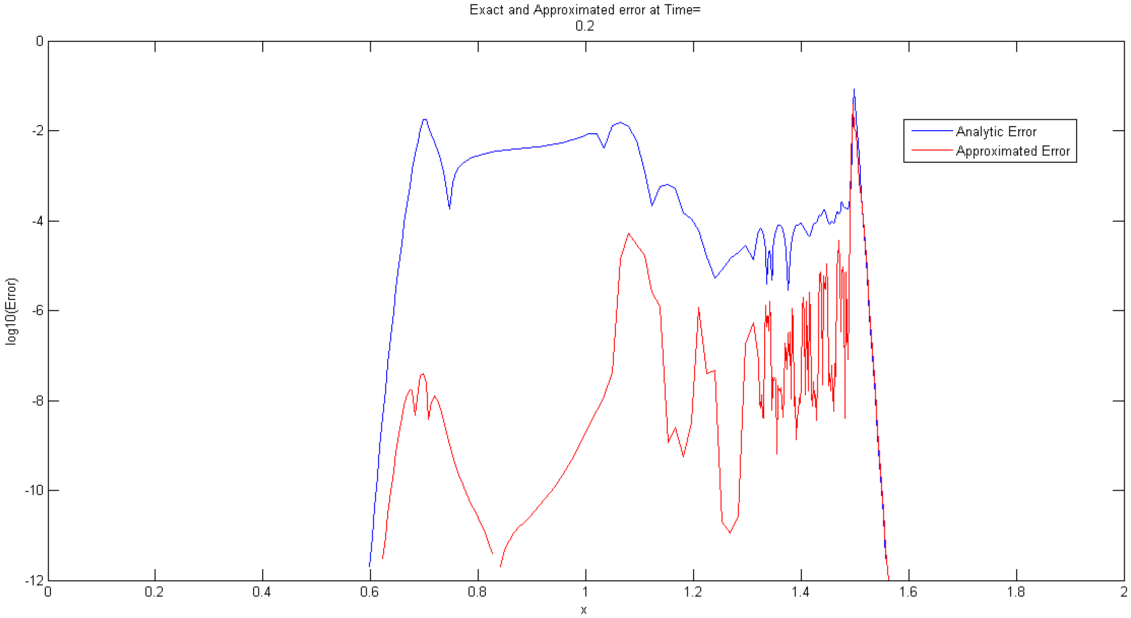


Figure 6.6: 1D Isothermal Equations, Semi-Discrete in Space Density Errors.

To further investigate how well the estimated errors in density capture the shock we compare Figure 6.9 against Figures 6.11, 6.15, 6.17. We see that the shock is captured best by using the fully-discrete estimated error for density, in Figure 6.17. The true error is 0.088 and the estimated error is 0.078. The estimated error in density captures the shock front least well in the semi-discrete space case, where the true error is 0.088 and the estimated error is 0.04. Comparing Figure 6.10 with Figures 6.12 and 6.18 we see that the estimated error in velocity captures the shock front in the fully-discrete case better than the semi-discrete in time case, where the true error is 0.63 and the estimated error is 0.25 and 0.2 respectively. The results indicate that the estimated error in density with the fully-discrete scheme captures the shock front best.

Note how the shock front true error and estimated errors in all cases are of the same magnitude. In order to investigate the expansion we zoom in on Figures 6.11, 6.12, 6.15, 6.17, 6.18. This is because the magnitude of the estimated error for the expansion is not in agreement with the true error. In all cases of the estimated error we see that the fan is not captured, only the top and bottom of the fan is, see Figures 6.13, 6.14, 6.16, 6.19, 6.20. The top of the fan is captured best by the estimated error of velocity using the semi-discrete in time scheme, as seen in Figure 6.14. However, this error is still out by two orders of magnitude.

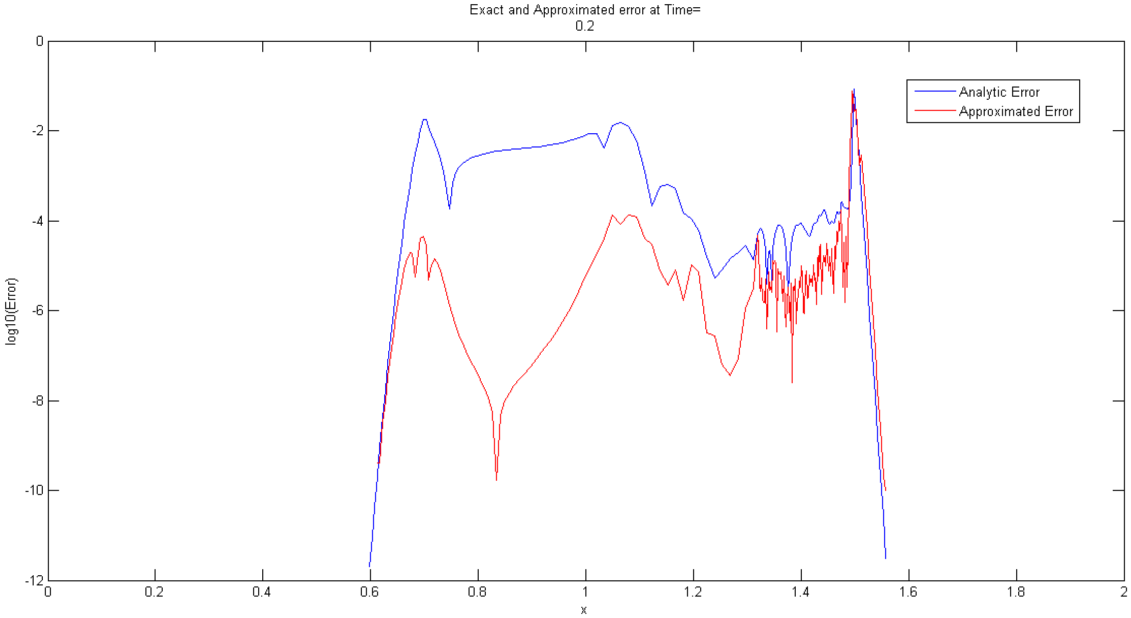


Figure 6.7: 1D Fully-Discrete Density Errors for the Isothermal Equations.

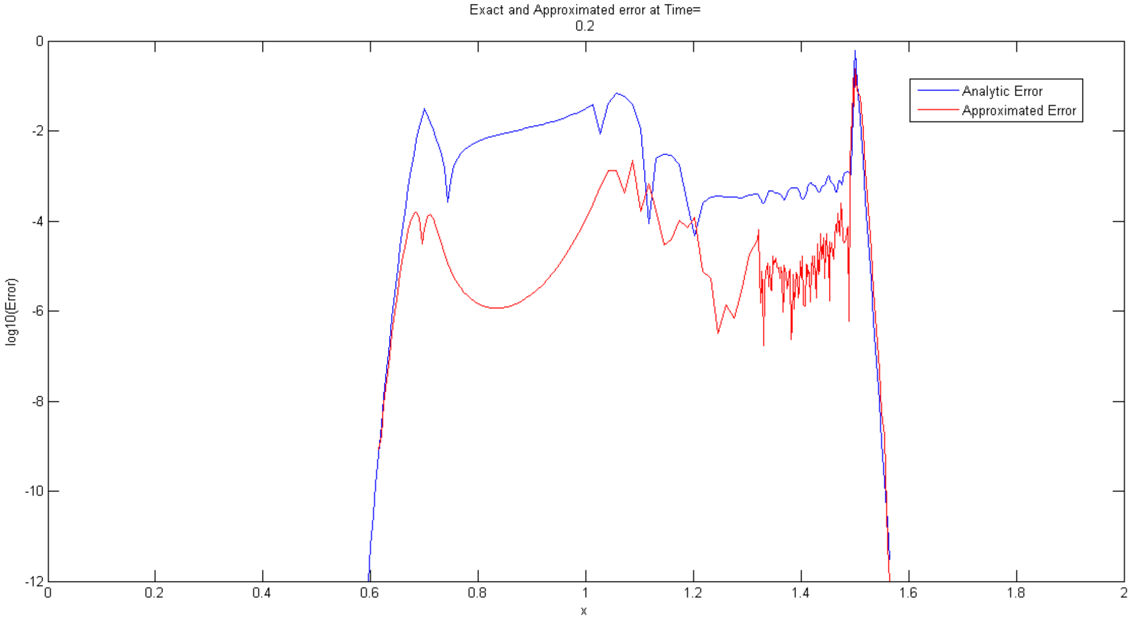


Figure 6.8: 1D Fully-Discrete Velocity Errors for the Isothermal Equations.

The bottom of the expansion fan is captured best by the estimated error for velocity in the fully-discrete case, see Figure 6.20, but this is still out by one order of magnitude.
The density estimated error does not capture the top and bottom of the fan as well as the velocity. However, the best results for the density case can be seen in the fully-discrete density, see Figure 6.19, as would be expected. However, in this case the top of the fan is out by three orders of magnitude, which is worse than any of the velocity results. This suggests using the velocity error estimate to capture the top and bottom of the fan.
It is still not clear why the magnitude of the error estimate is not in agreement with the true error around the expansion. One explanation for this could be due to the oscillations. Since mass is conserved, the mass used around these oscillations has been taken from another region i.e. around the expansion. If this is the case then the magnitude of the true error around the fan will be affected. Hence, when comparing the true error and the estimated error, the magnitude for the true error may be larger because of the oscillations.

To compare the true error and the estimated errors further we calculate the $1-$norm, $2-$norm and infinity-norm, on a coarse grid of 100 elements and a fine grid of 400 elements. The error norms for the semi-discrete cases in Tables 6.2-6.4 are significantly different, although, the infinity-norm demonstrates the same magnitude in all the fine cases. The differ-

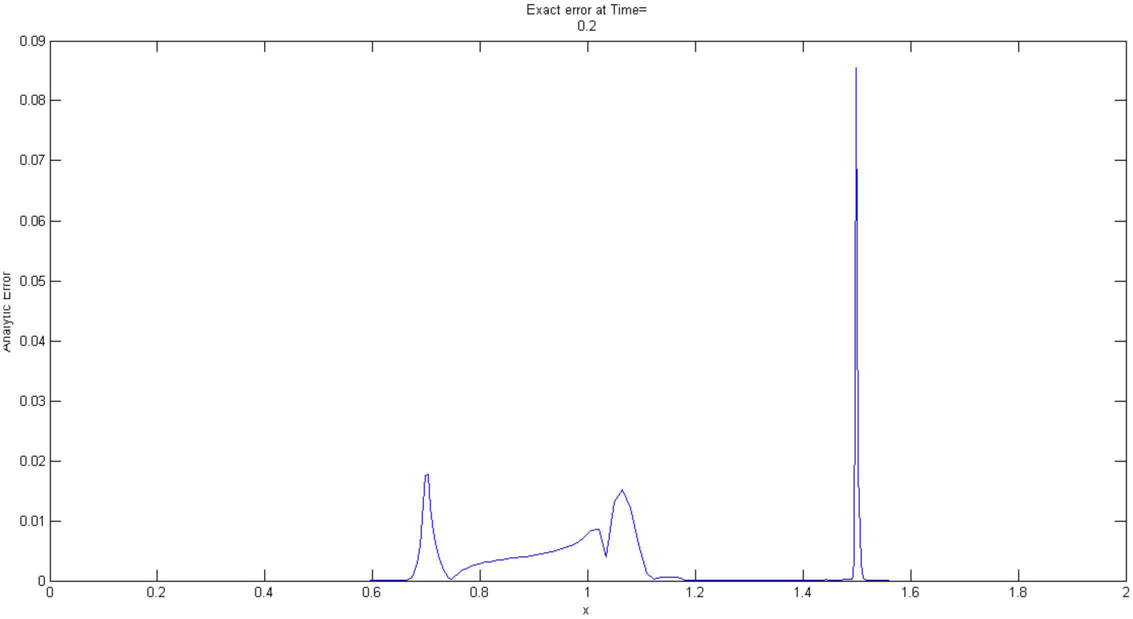


Figure 6.9: True Error in Density for the Isothermal Equations in 1D.

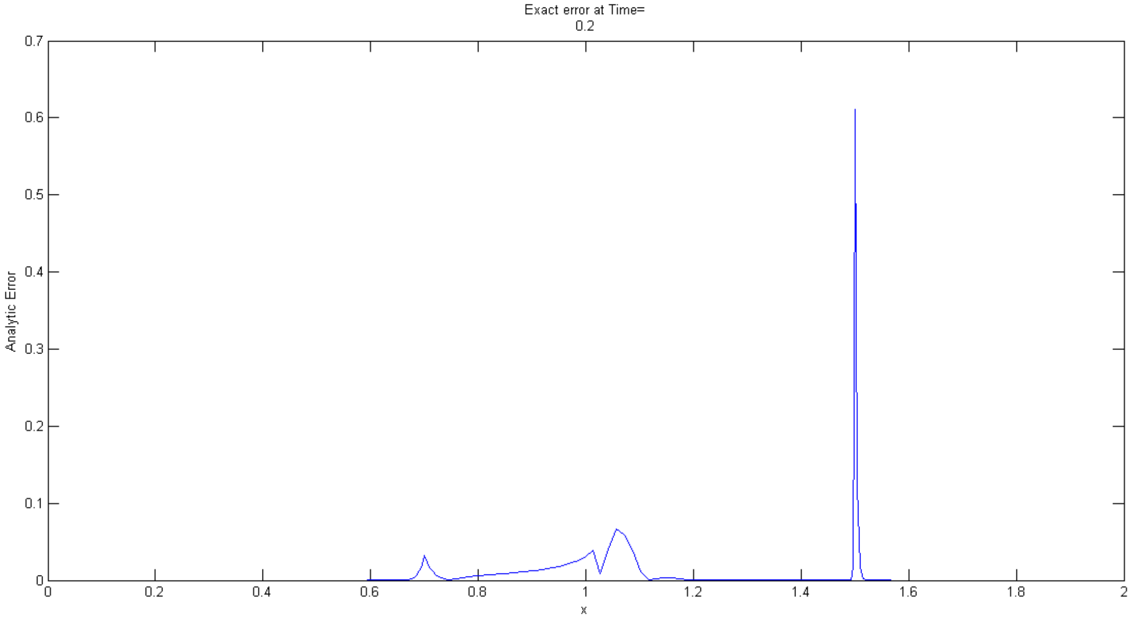


Figure 6.10: True Error in Velocity for the Isothermal Equations in 1D.

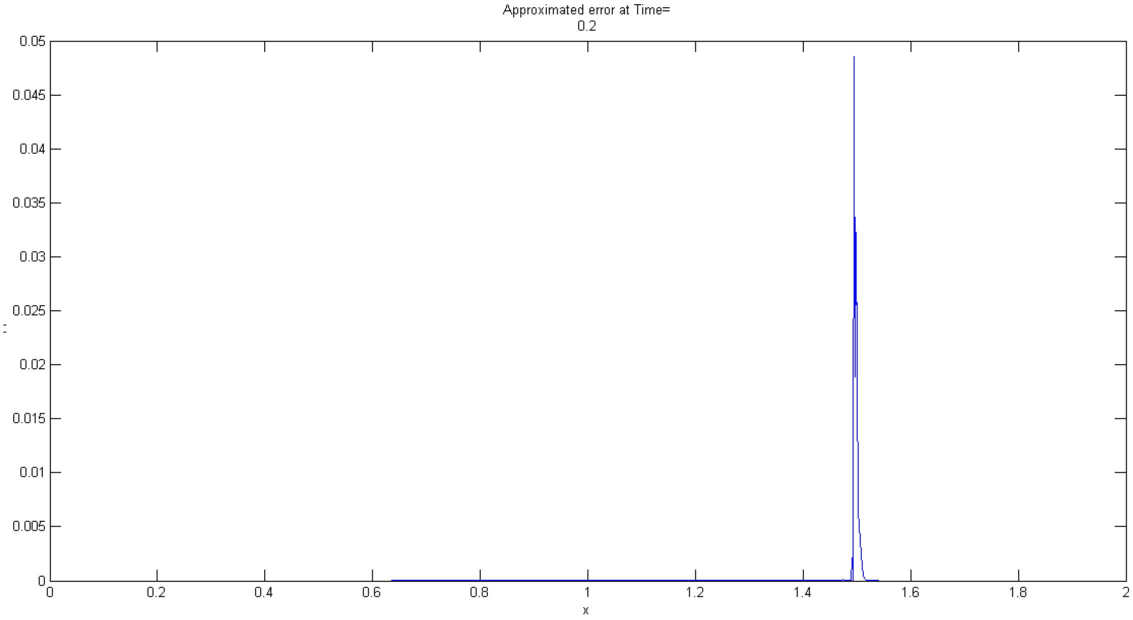


Figure 6.11: Semi-Discrete in Time Estimated Error for Density for the Isothermal Equations in 1D.

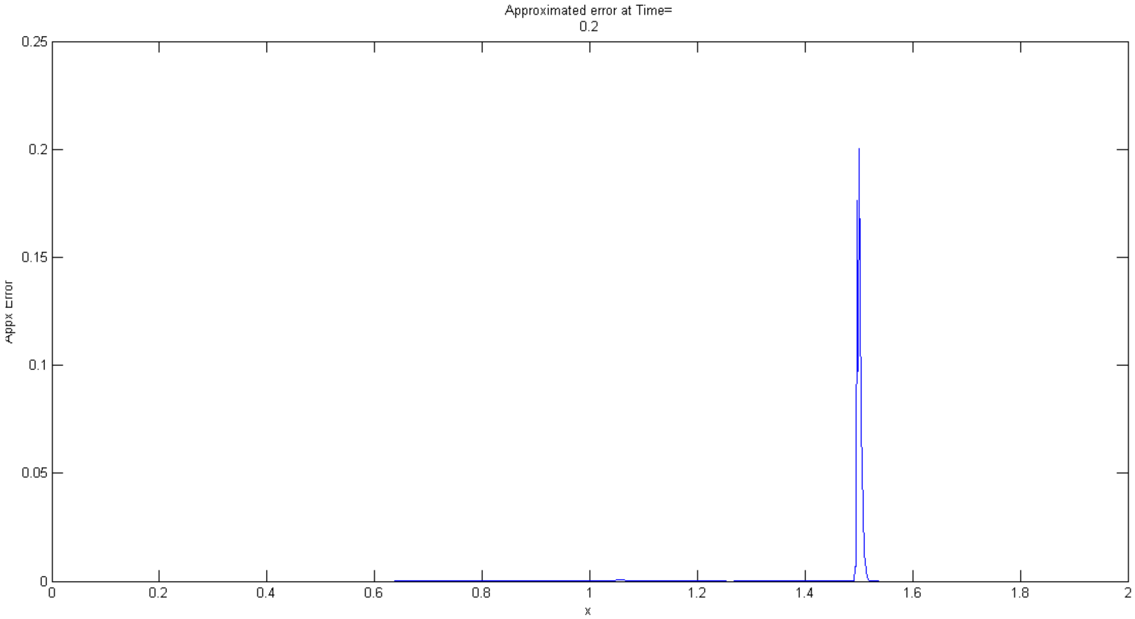


Figure 6.12: Semi-Discrete in Time Estimated Error for Velocity for the Isothermal Equations in 1D.

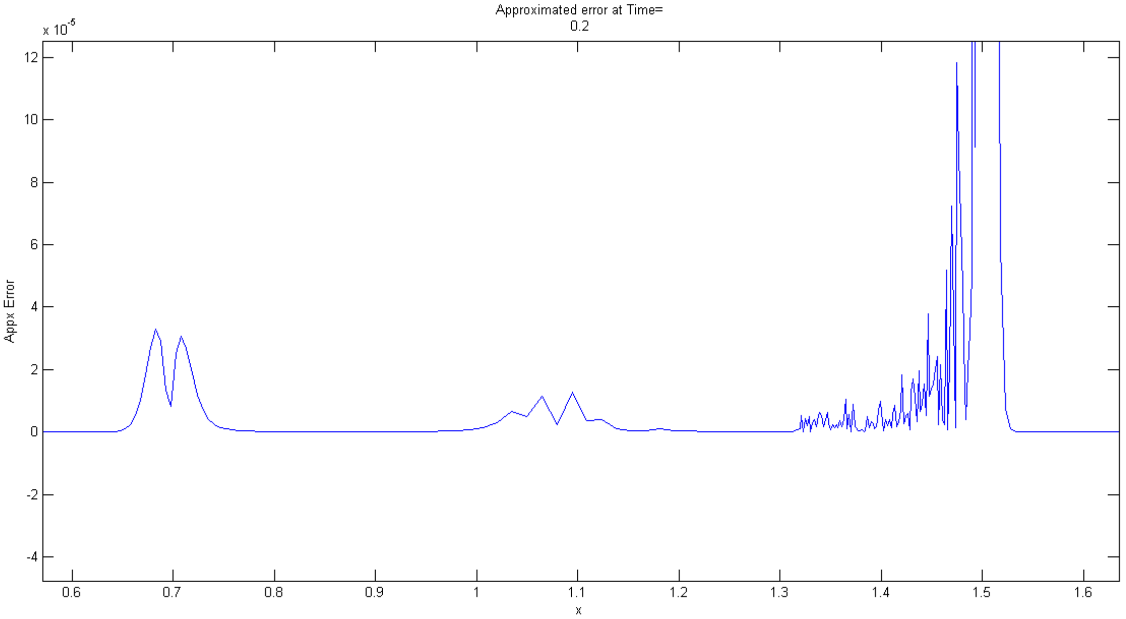


Figure 6.13: Semi-Discrete in Time Estimated Error for Density for the Isothermal Equations in 1D Zoomed.

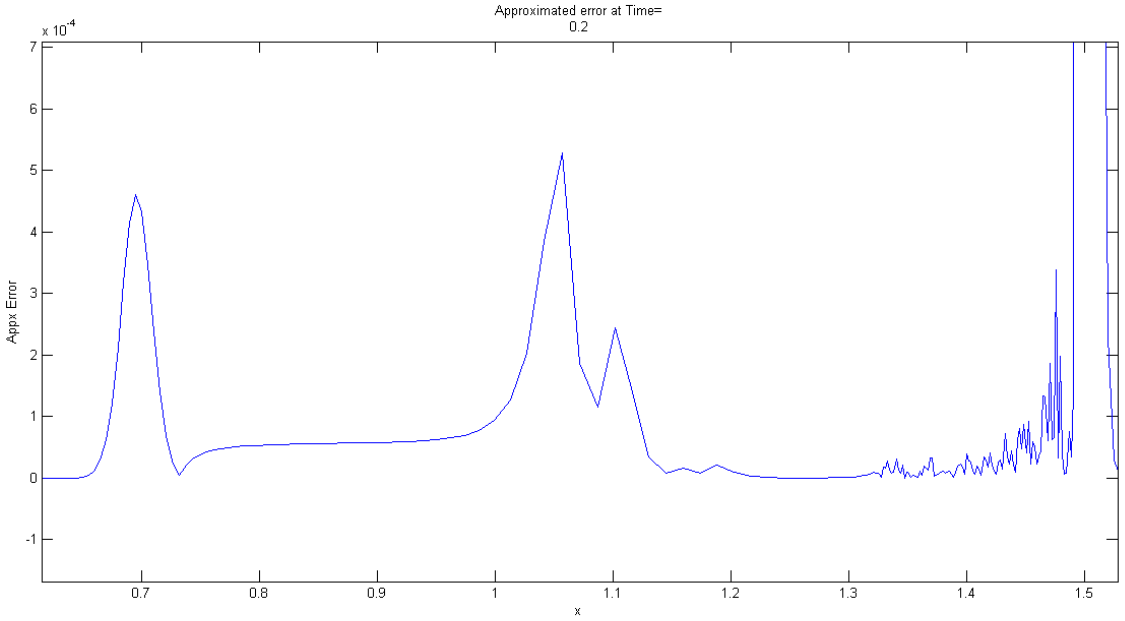


Figure 6.14: Semi-Discrete in Time Estimated Error for Velocity for the Isothermal Equations in 1D Zoomed.

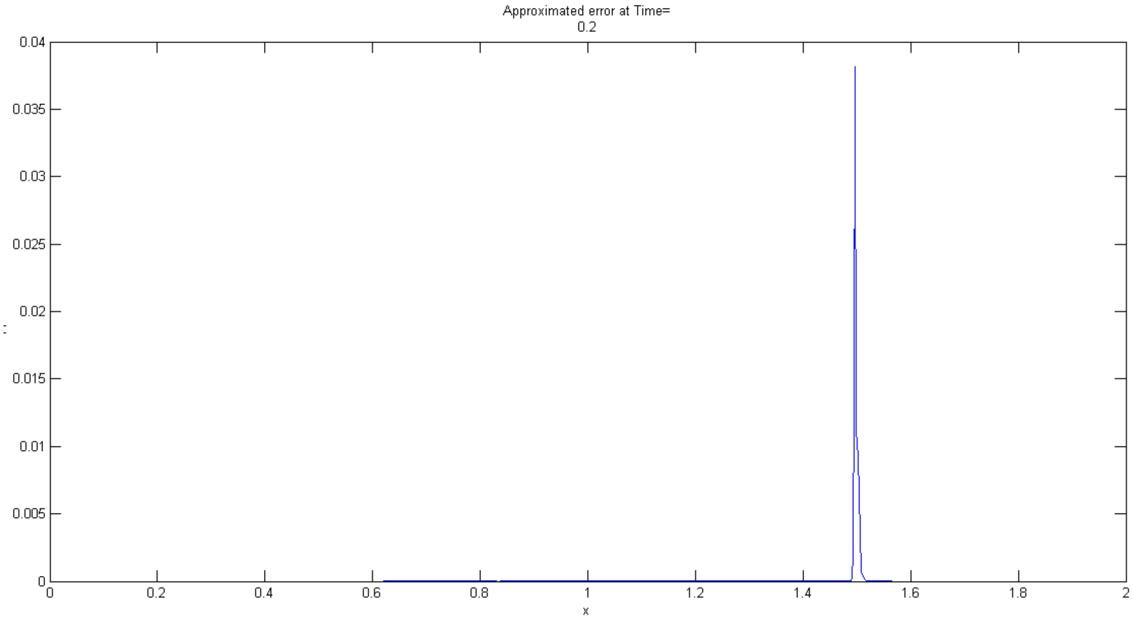


Figure 6.15: Semi-Discrete in Space Estimated Error for the Isothermal Equations in 1D for Density

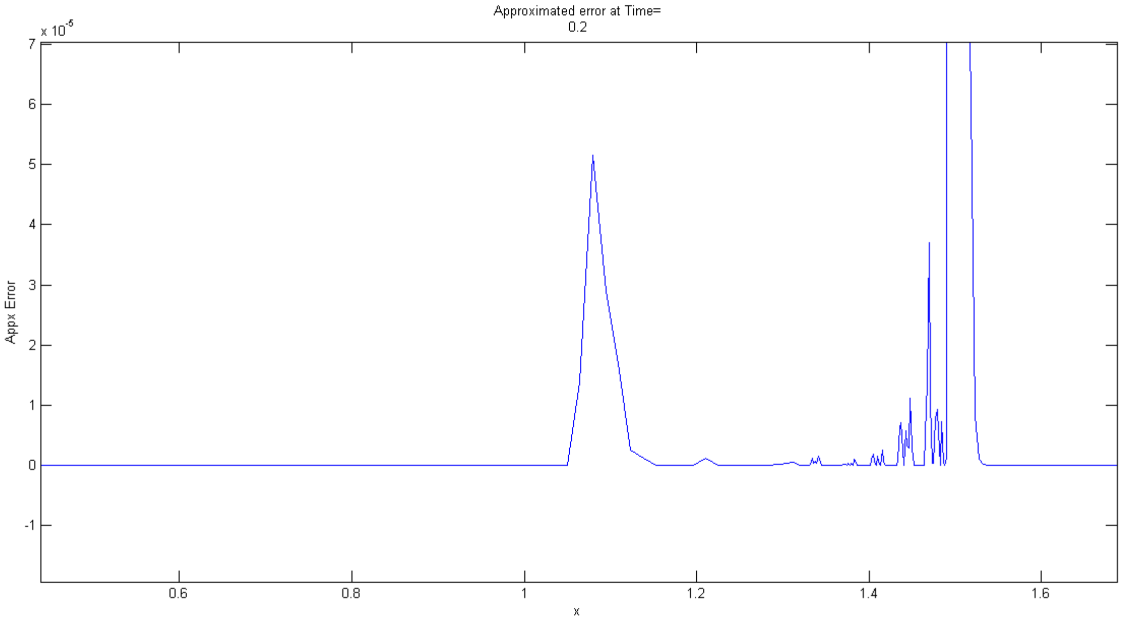


Figure 6.16: Semi-Discrete in space Estimated Error for Density for the Isothermal Equations in 1D Zoomed.

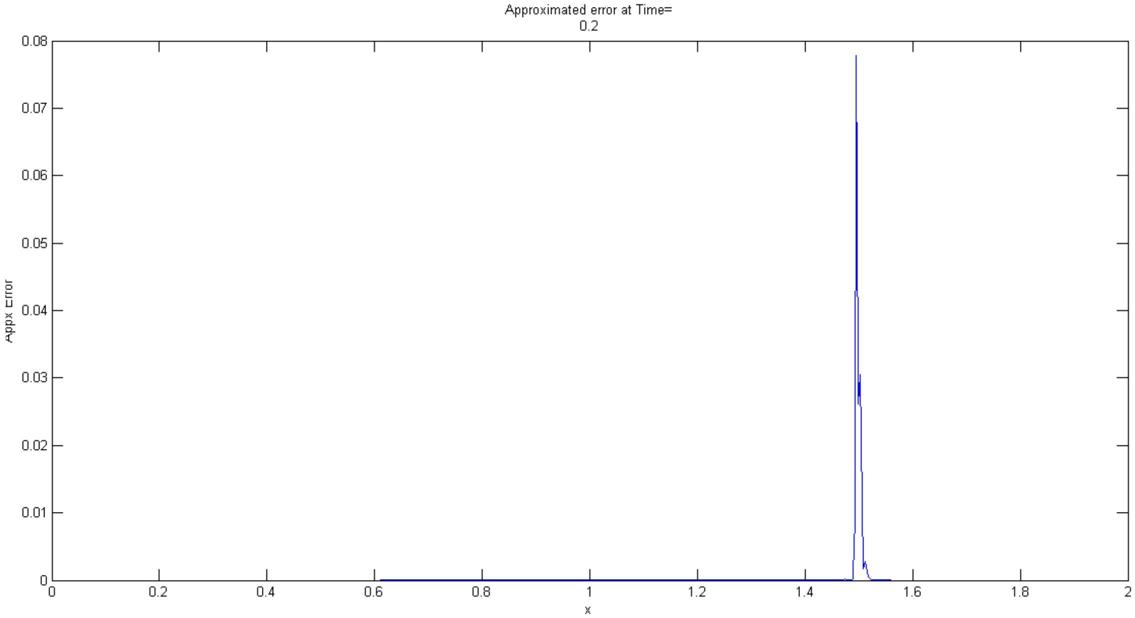


Figure 6.17: Fully-Discrete Estimated Error in Density for the Isothermal Equations in 1D.

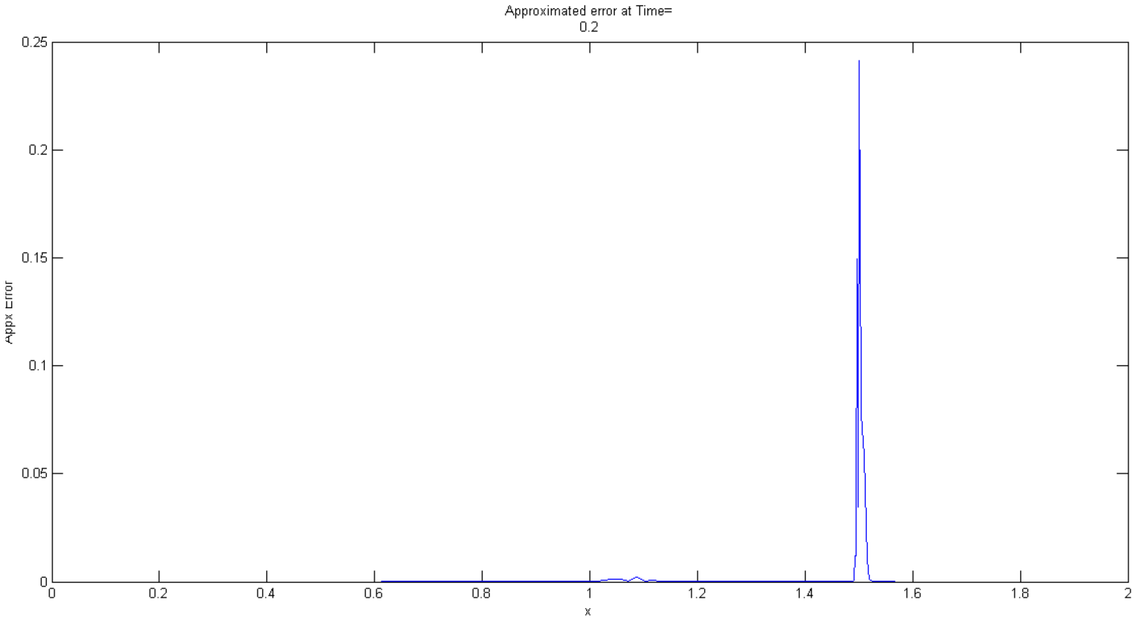


Figure 6.18: Fully-Discrete Estimated Error in Velocity for the Isothermal Equations in 1D.

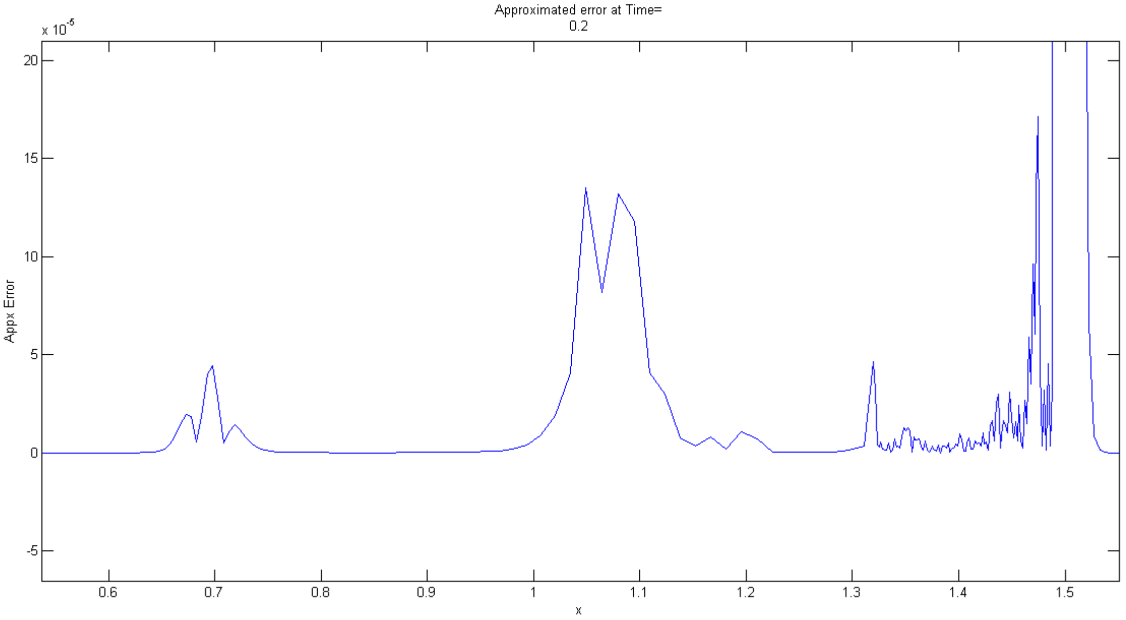


Figure 6.19: Fully-Discrete Estimated Error for Density for the Isothermal Equations in 1D Zoomed.

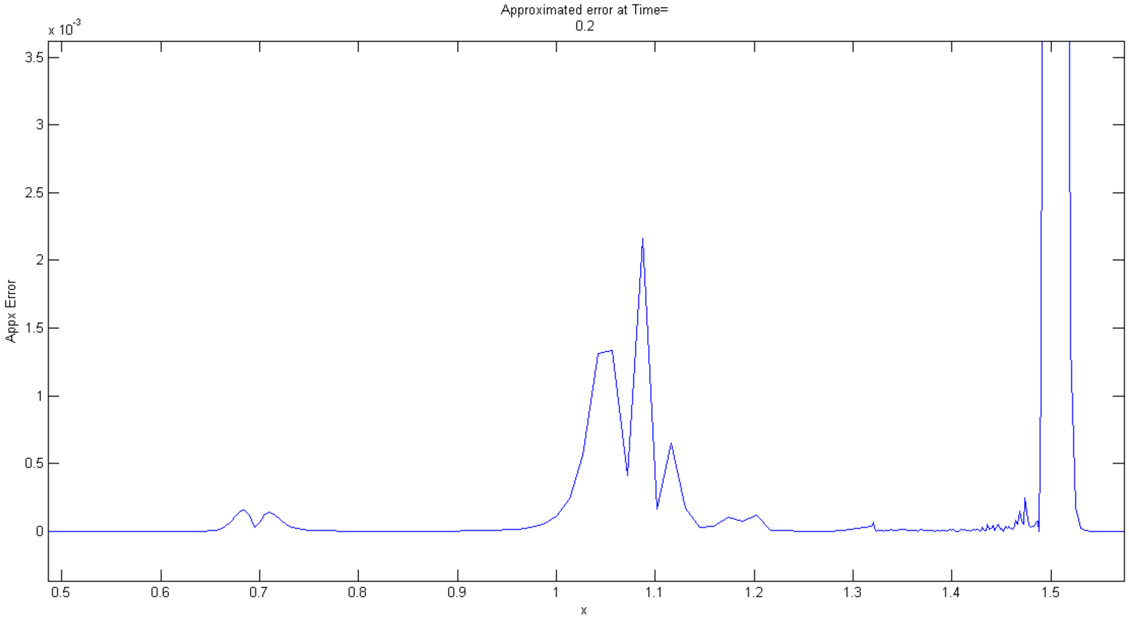


Figure 6.20: Fully-Discrete Estimated Error for Velocity for the Isothermal Equations in 1D Zoomed.

Table 6.2: Norms for Semi-Discrete in Time Density Errors

| | $\lvert\lvert e\rvert\rvert_1$ | | $\lvert\lvert e\rvert\rvert_2$ | | $\lvert\lvert e\rvert\rvert_\infty$ | |
|---|---|---|---|---|---|---|
| | 100 | 400 | 100 | 400 | 100 | 400 |
| **True Error** | 0.45066 | 0.42377 | 0.13987 | 0.10548 | 0.11366 | 0.08550 |
| **Estimated Error** | 6.21645 | 0.11256 | 3.31151 | 0.06167 | 2.83615 | 0.04854 |

Table 6.3: Norms for Semi-Discrete in Time Velocity Errors

| | $\lvert\lvert e\rvert\rvert_1$ | | $\lvert\lvert e\rvert\rvert_2$ | | $\lvert\lvert e\rvert\rvert_\infty$ | |
|---|---|---|---|---|---|---|
| | 100 | 400 | 100 | 400 | 100 | 400 |
| **True Error** | 2.16705 | 1.81174 | 0.88916 | 0.68360 | 0.81492 | 0.61086 |
| **Estimated Error** | 1.52606 | 0.58422 | 0.70488 | 0.29341 | 0.44780 | 0.20012 |

Table 6.4: Norms Semi-Discrete in Space for Density Errors

| | $\lvert\lvert e\rvert\rvert_1$ | | $\lvert\lvert e\rvert\rvert_2$ | | $\lvert\lvert e\rvert\rvert_\infty$ | |
|---|---|---|---|---|---|---|
| | 100 | 400 | 100 | 400 | 100 | 400 |
| **True Error** | 0.45066 | 0.42376 | 0.13987 | 0.10548 | 0.11366 | 0.08550 |
| **Estimated Error** | 0.14150 | 0.07647 | 0.08692 | 0.04326 | 0.08053 | 0.03816 |

Table 6.5: Norms of Fully-Discrete Density Errors

| | $\lvert\lvert e\rvert\rvert_1$ | | $\lvert\lvert e\rvert\rvert_2$ | | $\lvert\lvert e\rvert\rvert_\infty$ | |
|---|---|---|---|---|---|---|
| | 100 | 400 | 100 | 400 | 100 | 400 |
| **True Error** | 0.45066 | 0.42377 | 0.13987 | 0.10548 | 0.11366 | 0.08550 |
| **Estimated Error** | 0.32961 | 0.21427 | 0.15586 | 0.10567 | 0.11060 | 0.07788 |

Table 6.6: Norms Fully-Discrete Velocity Errors

| | $\lvert\lvert e\rvert\rvert_1$ | | $\lvert\lvert e\rvert\rvert_2$ | | $\lvert\lvert e\rvert\rvert_\infty$ | |
|---|---|---|---|---|---|---|
| | 100 | 400 | 100 | 400 | 100 | 400 |
| **True Error** | 2.16705 | 1.81174 | 0.88916 | 0.68360 | 0.81492 | 0.61086 |
| **Estimated Error** | 1.27366 | 0.60981 | 0.62063 | 0.30265 | 0.55336 | 0.24132 |

ence in norms is due to the fact that these errors are estimated using semi-discrete schemes, whereas the fully-discrete form contributes more information to the estimated error.

The results in Table 6.5 for the fully-discrete estimated error in density give the best agreement, particularly the infinity-norm, however there are still differences due to the fact that we are using an estimation, and as we have seen in the log plots some features were not detected with the same magnitude as the true error. Therefore, the $1-$norm and $2-$norm will differ significantly even if the infinity-norm is in good agreement.

For the fully-discrete estimated error in velocity, the results in Table 6.6 are not as good as the density case, where there is a significant difference in the infinity-norms. Since the infinity-norm is effectively the error at the shock, it appears from this result that if the shock is not captured well the rest of the solution will suffer.

To conclude the results for the 1D system of two equations, it appears that the shock is best captured using a temporal error. From Tables 6.2, 6.4 and 6.5 the infinity-norm, which demonstrates the values for the shock, are in best agreement here. These results are all density related. In the analytic equation the density is obtained from the conservation of mass, i.e. eqn. (2.26)

$$\frac{dm}{dt} = 0.$$

Since this only contains a temporal derivative we conclude the shock is captured best by a temporal error. Additionally in Figures 6.5 and 6.8 we see the expansion is captured best by the velocity. From the analytic equation, in which the velocity is obtained, as stated in eqn. (2.27), for the 1D case we have

$$m\frac{du}{dt} = -p_\xi. \tag{6.1}$$

Since a spatial derivative is present we conclude that the expansion is related to an error in space. The shock is also captured in Figures 6.5 and 6.8, which is in agreement the conclusion that the shock is captured by temporal errors, as there is also a temporal derivative in eqn. (6.1).

### 6.1.2 Euler Equations

We use the well known Sod shock tube problem from [83] as a test problem for the Euler equations. We choose this problem for the same reasons as choosing a Riemann problem in the Isothermal case. The problem is simple and an analytic solution exists. The Sod shock tube problem consists of the desired features, i.e. shock, contact and expansion, therefore, making it is an ideal problem to test the numerical method.

The initial data for the Euler equations is given in Figure 6.21. The problem consists of a rigid walled tube, which contains two phases separated by a membrane. The gas of lower pressure is called the test gas and the higher pressure gas is the driver. At the beginning of the problem the membrane is removed and assumed to have no further influence. Once the membrane is removed a shock wave forms which moves to the right, as a result of the driver gas being on the left of the test gas. A rarefaction propagates to the left.

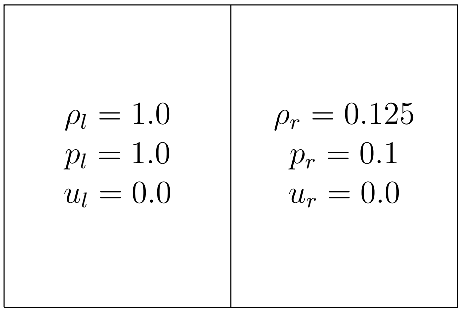


Figure 6.21: Sod Shock Tube Problem in 1D.

The problem is run to a final time of 0.2 on a uniform mesh of 200 points in a region $[0, 1]$. For the Euler equations, the solution consists of four constant states, which are connected by a rarefaction fan, a contact discontinuity, and a shock wave.

Christensen artificial viscosity has been used. The analytic solution is based on the Riemann solver, from [82], which allows for true error to be compared against the estimated error.

In Figures 6.22 and 6.23, the analytic solution is illustrated with the blue solid line, the numerical solution is illustrated with the red dotted line. We can see the top of the expansion fan at $x = 0.26$ and the bottom is at $x = 0.49$, the fan is connected to the contact discontinuity by a constant state at $\rho = 0.43$. The contact spans the interval $x \in [0.6795, 0.6866]$. There is another constant state at $\rho = 0.2656$ which connects the contact and the shock. The shock spans the interval $x \in [0.8499, 0.8533]$.

From Figure 6.22, the solutions from the Main Scheme, seen in Sections (3.3.1) and (4.2.1), and the analytic are in good agreement, as previously stated the undershoot at the bottom of the fan, due to wall heating is visible. Similarly for Figure 6.23, where a combination of Scheme 2 and Scheme B is used, the analytic and numerical solutions are in good agreement. However, the opposite to the result illustrated in Figure 6.22 is demonstrated, where there is an overshoot at the top of the fan. Additionally, the solution around the shock appears to be smoothed out. There is also possible evidence of odd-even decoupling, which could lead to the artificial viscosity behaving badly, hence causing the overshoots. This is an area of further work, where the $c_q$ and $c_l$ coefficients in the artificial viscosity terms should be investigated.

Since the analytic solution is available we calculated the error norms for Scheme 1 and Scheme 2 on a fine mesh, see Table 6.7. The norms are not as comparable as in the Isothermal case. However, we must take into consideration that there is an extra feature in the solution here, i.e. the contact, therefore, it would be expected for the solutions of each scheme to be more varied. It is promising that the $\infty$-norm is comparable in magnitude, as this is the main feature we wish to capture.

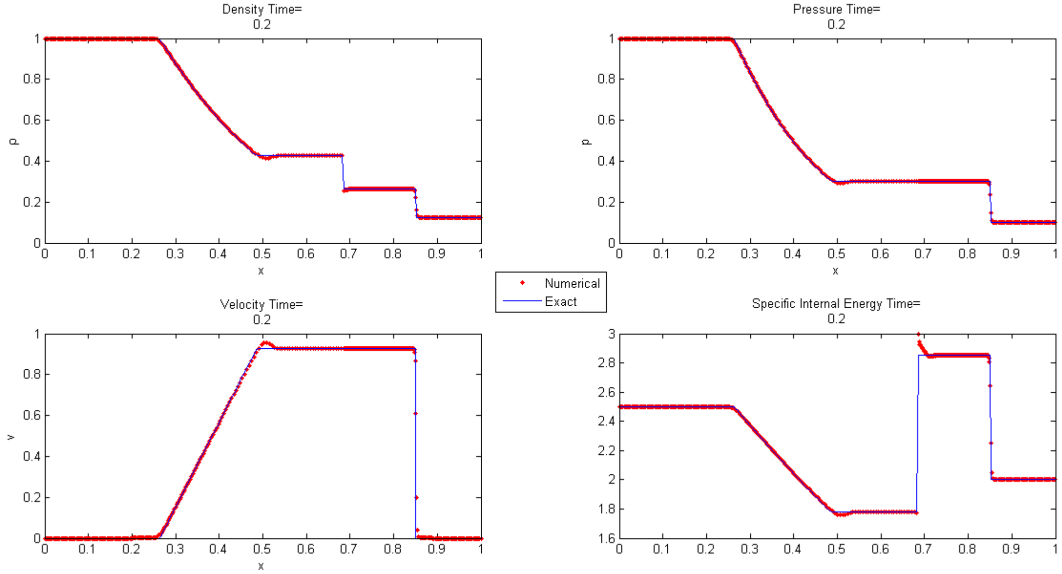


Figure 6.22: Density, Pressure and Velocity Results at $T = 0.2$ using Scheme 1 and the Sod Shock Tube Problem for the Euler Equations in 1D.

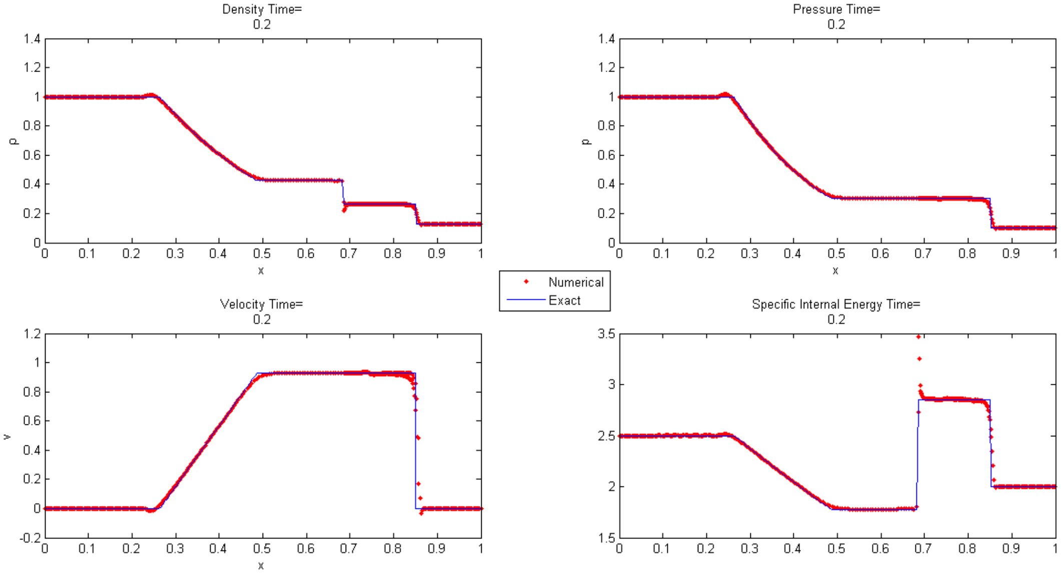


Figure 6.23: Density, Pressure and Velocity Results at $T = 0.2$ using Scheme 2 and the Sod Shock Tube Problem for the Euler Equations in 1D.

Table 6.7: 1D Euler Equations Error Norms Comparing Scheme 1 and Scheme 2 for the Sod Shock Tube.

| Calculation | $\|\|e\|\|_1$ | $\|\|e\|\|_2$ | $\|\|e\|\|_\infty$ |
|---|---|---|---|
| Scheme 1 | 0.00262 | 0.07790 | 0.04582 |
| Scheme 2 | 0.00484 | 0.16845 | 0.07995 |

Since both schemes are comparable, they are suitable to be used as the estimated error. We are now able to investigate the estimated errors obtained in the previous chapters.

### Comparison of the True Error and Estimated Error

We have a variety of errors produced from the semi-discrete in time scheme, semi-discrete in space scheme and the fully-discrete scheme.

We are interested in capturing the flow features i.e. rarefaction, contact and shock with the estimated errors. The results of the estimated error can be compared against the true error on a log scale.

From Figure 6.24, we see that the top and bottom of the expansion are detected well although, not in size. As we saw in the Isothermal case, the peaks around the top and bottom of the fan are peaks themselves, suggesting that the two schemes used to estimate the error are similar to each other in this area, resulting in a lower estimated error, i.e. the peaks.
To the left of $x = 0.7$ we see the contact is detected as having the same magnitude at the top and bottom of the fan. Across the contact peaks form in the same manner as the endpoints of the fan. The top of the shock is detected correctly at $x = 0.8499$, where the size is also in good agreement.
The semi-discrete in time estimation for velocity, in Figure 6.25, demonstrates that the top and bottom of the fan is captured very well, where the magnitude at these locations is better than the density case. Additionally, the fan itself has been captured.
The contact is captured in the correct location and is of a similar value to the true error. The shock is also captured well here.

The semi-discrete in space estimated error in pressure is illustrated in Figure 6.26. The

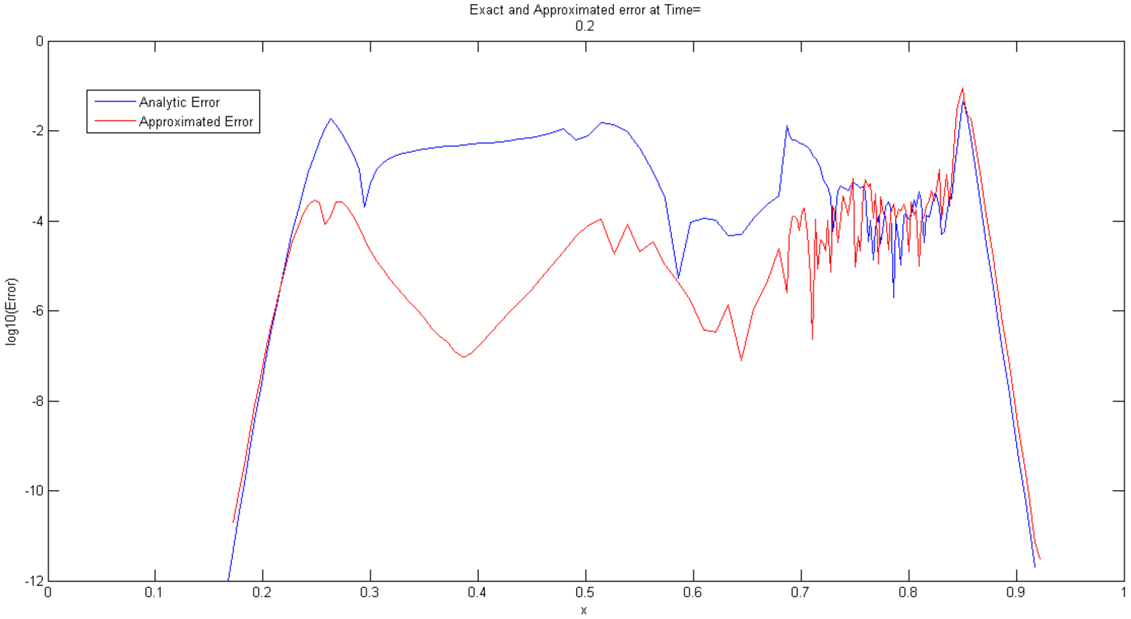


Figure 6.24: 1D Euler Equations, Semi-Discrete in Time Density Errors.

features of the solution are not as distinguished as in previous results. Although, the top and bottom of the fan are captured, they blend into the results of the fan itself. The contact is captured at the correct location with a similar value. Again, the top of the shock is in good

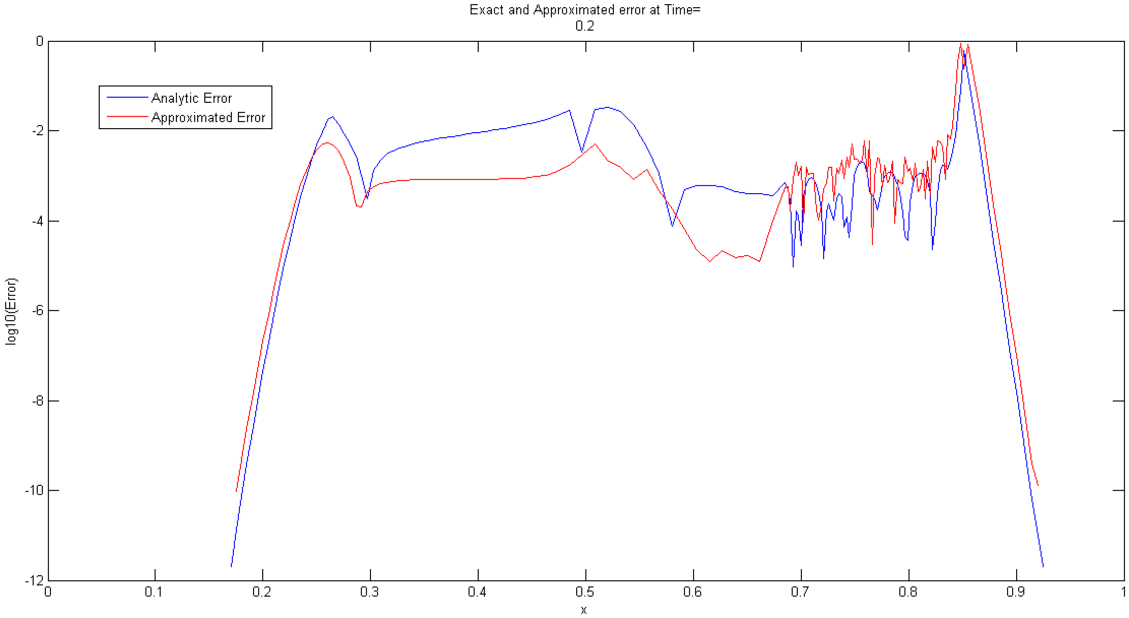


Figure 6.25: 1D Euler Equations, Semi-Discrete in Time Velocity Errors.

agreement. From Figure 6.27, the top and bottom of the fan are detected well, however, they are not as distinguished as for the semi-discrete in time case. However, the bottom of the fan is in better agreement than the semi-discrete in time case. There is less variation between the two schemes at the top of the fan, demonstrated by the peaks. The contact is detected just as well as the semi-discrete in time case. The shock is again in good agreement.

Figure 6.28 illustrates the fully-discrete estimated error in velocity. The information across the fan is detected in a similar manner to the semi-discrete in time case, however, this semi-discrete case has a greater magnitude. Again the contact is detected in the correct position but there is a greater difference. The shock is again in good agreement here.

For all cases in Figures 6.24-6.28, the shock is captured very well in position and value. Although, some variables are better than others at detecting the expansion, there is still a difference in magnitudes between the true error and estimate error, as was also observed in the Isothermal case. We have discussed the undershoot at the bottom of the fan and the oscillations before the shock. Due to the conservation of mass, these features will be counteracted elsewhere in the solution. Therefore, if the two schemes behave equally as poor around the fan, the true errors will be affected, but there will be no difference seen in the estimated error. Therefore the true error may be more than anticipated where the estimated

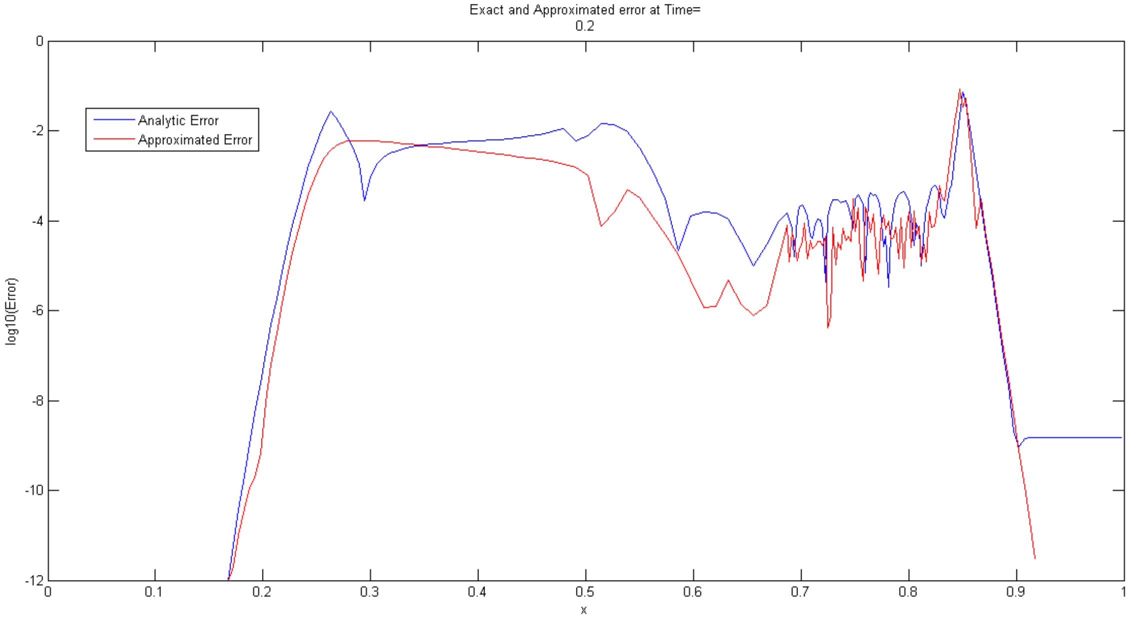


Figure 6.26: 1D Euler Equations, Semi-Discrete in Space Pressure Errors.

error may be behaving better than appears. This is seen to be illustrated in Figures 6.24-6.28, since the estimated error is always less than the true error.

On comparison of Figure 6.36 with Figure 6.31, we see that the semi-discrete in space estimated error for pressure captures the shock best, where the true error reaches 0.072 and the estimated error is at 0.085. This is closely followed by the fully-discrete approximation of velocity from Figures 6.39, where the true error is at 0.600 and the estimated error is at 0.724.

The contact is best captured by the semi-discrete in time estimated error in velocity, from Figure 6.33, Figure 6.35 and Figure 6.30, the true error is at 0.0007 and the estimated error is at 0.0005.

The top of the rarefaction is best captured by the semi-discrete in time estimated error in velocity with the true error reaching 0.02 and the estimated error is at 0.005, from Figure 6.33, Figure 6.35 and Figure 6.30. The bottom of the fan is best captured by the semi-discrete in space estimated error for pressure, from Figure 6.36, Figure 6.31 and Figure 6.37, where the true error reaches 0.006 and the estimated error is at 0.0015.

We see that the contact appears at $x = 0.7$ for the true error in Figure 6.29. On comparison of Figures 6.32 and 6.38, we can not detect the contact, therefore on closer inspection we

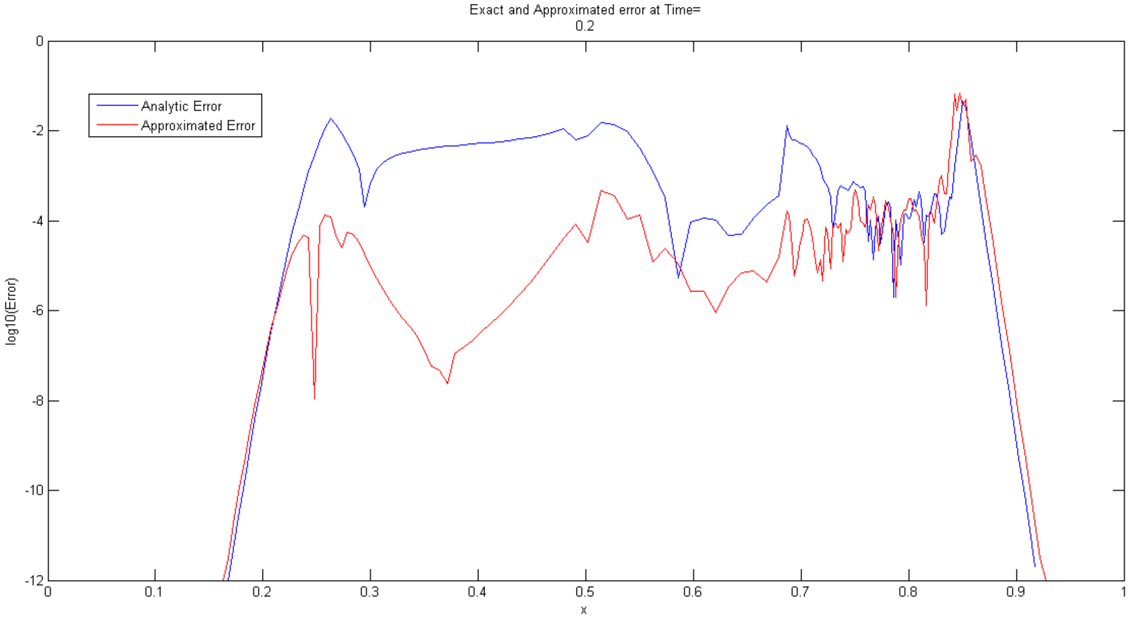


Figure 6.27: 1D Euler Equations, Fully-Discrete Density Errors.

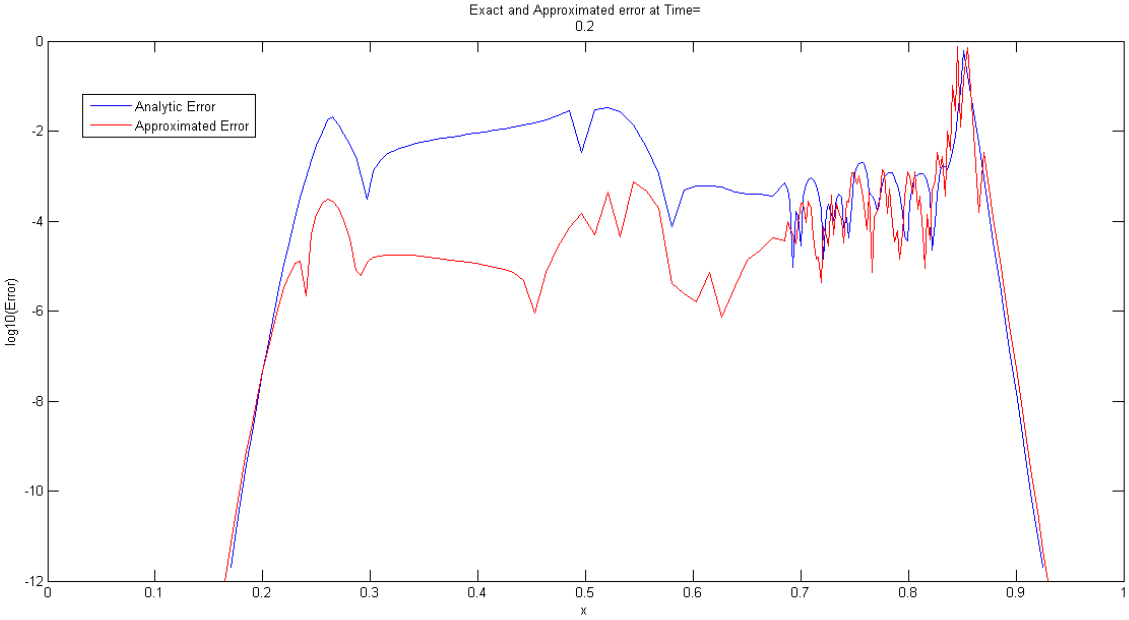


Figure 6.28: 1D Euler Equations, Fully-Discrete Velocity Errors.

obtain Figures 6.34 and 6.40. The semi-discrete in time and fully-discrete error estimates for density capture the contact least well. However, for all cases it is difficult to distinguish whether it is the contact we are detecting or the oscillations. This is discussed further in Appendix D.

## 6.2 Chapter 6 Summary

In summary for this problem it is best to use a combination of the semi-discrete in space approximation of pressure to capture the shock front and bottom of the expansion, whereas it is best to use the semi-discrete in time approximation of velocity to capture the top of the fan and contact. Pressure and velocity may be behaving better here since they are natural variables of the derivatives we gain an estimated error for. Whereas, the estimated error in $x$ needs to be manipulated to obtain an estimated error for density. It appears that the contact will always be difficult to detect using this method. It is a difficult task to capture all features with a single variable, as was discussed in Chapter 2. We also found the overall results for the error norms in the Euler equation case exhibit a similar pattern to the Isothermal case. Since the results are promising for detecting the fluid flow features of interest we extended to the 2D system of Euler equations.

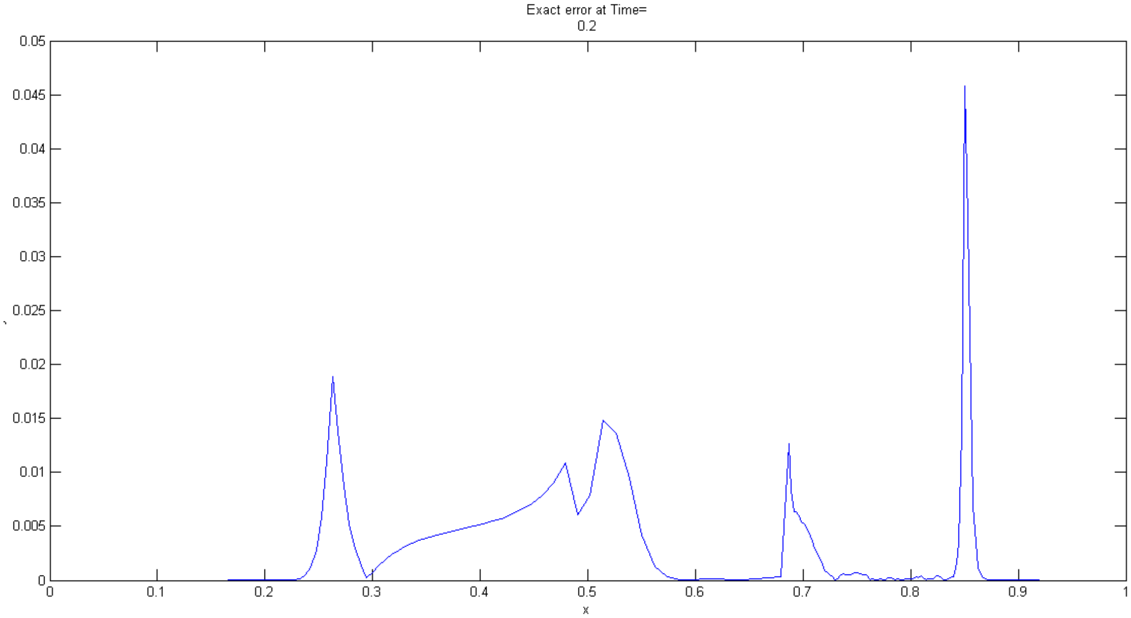


Figure 6.29: True Error for Density for the Euler Equations in 1D.

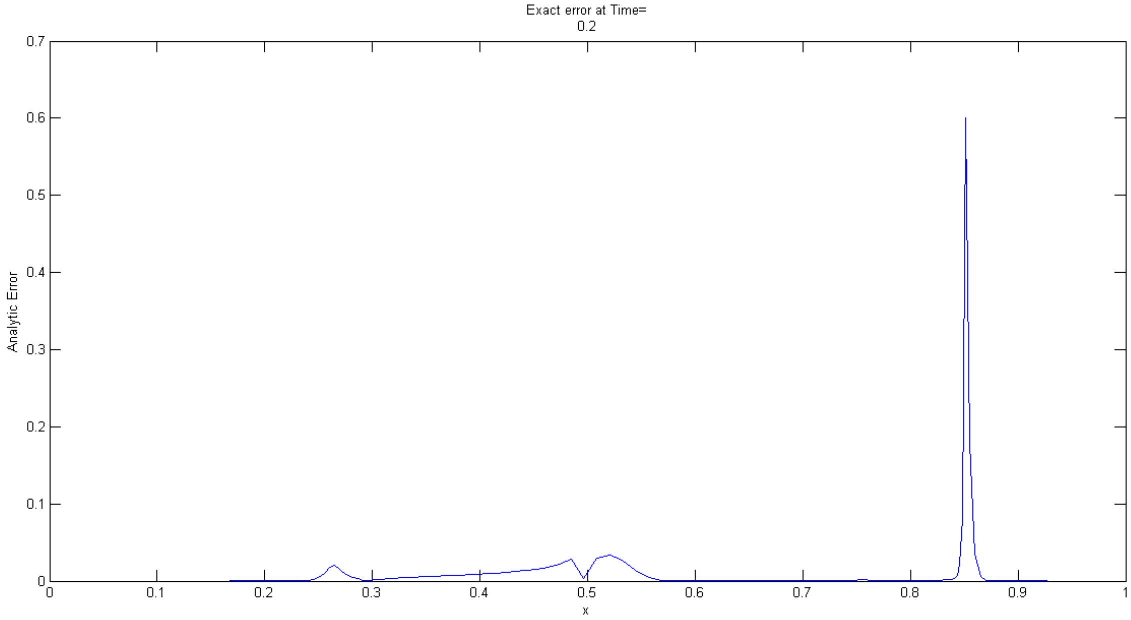


Figure 6.30: True Error for Velocity for the Euler Equations in 1D.

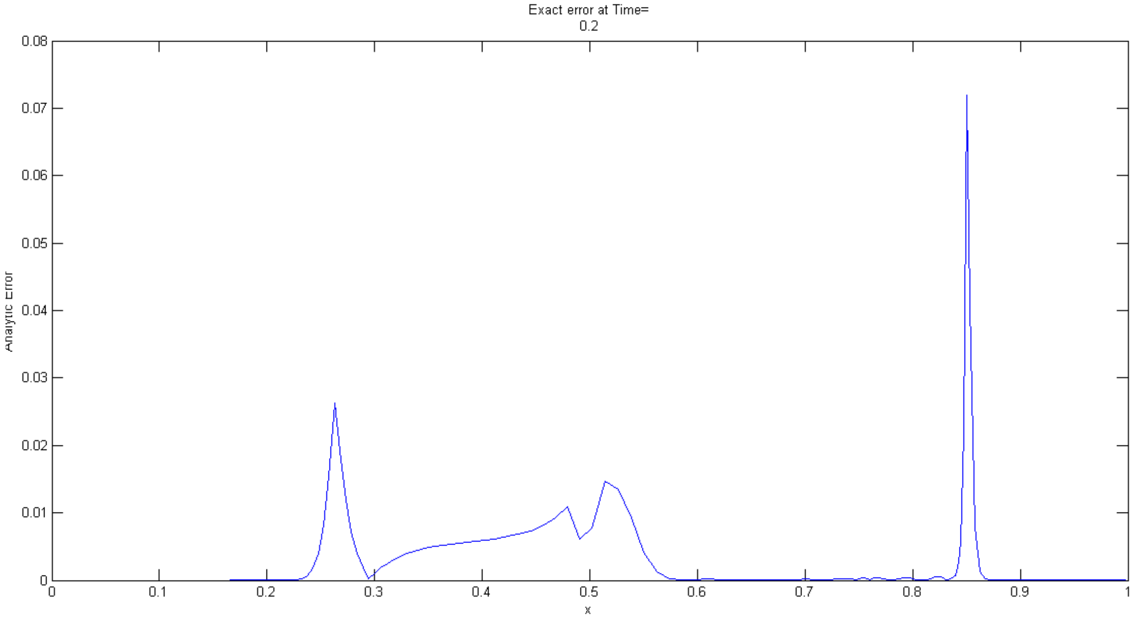


Figure 6.31: True Error for Pressure for the Euler Equations in 1D.

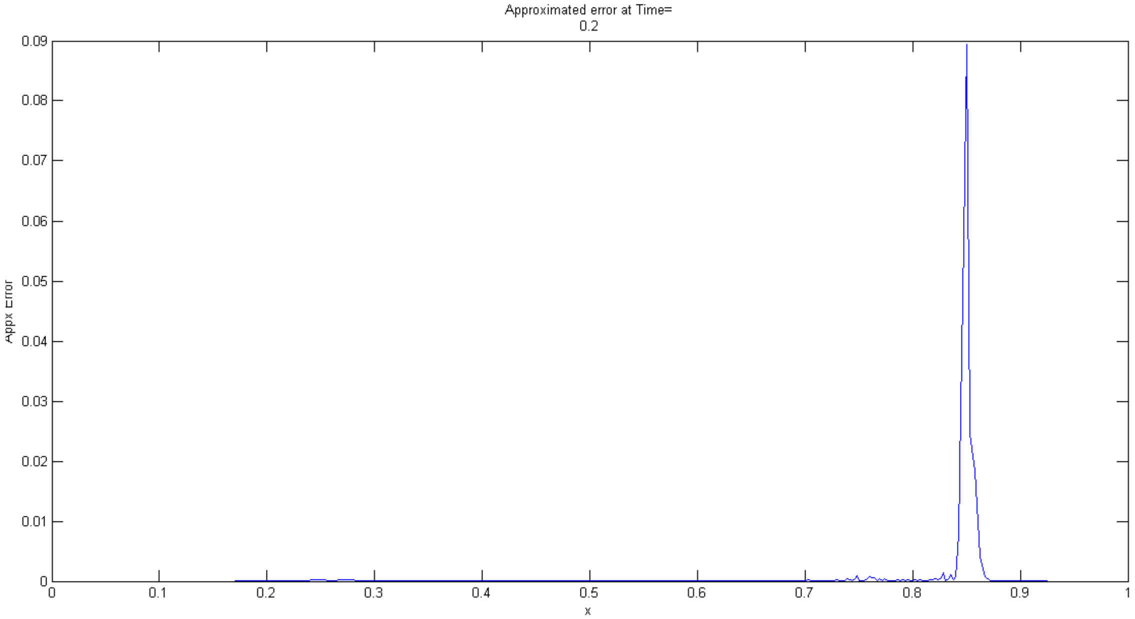


Figure 6.32: Semi-Discrete in Time Estimated Error for Density for the Euler Equations in 1D.

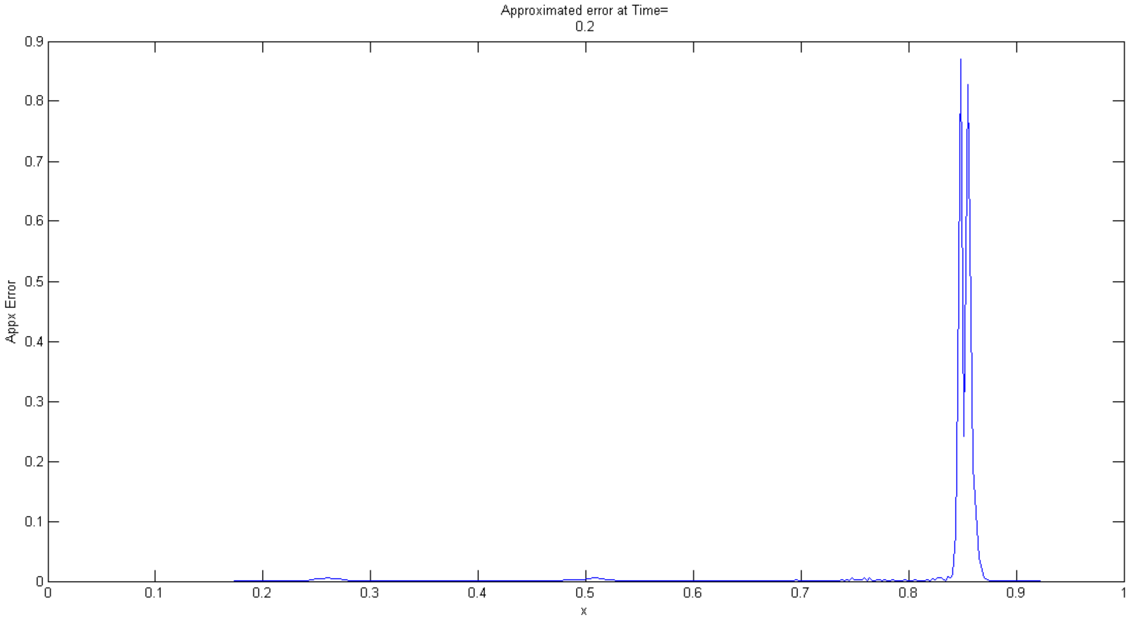


Figure 6.33: Semi-Discrete in Time Estimated Error for Velocity for the Euler Equations in 1D.

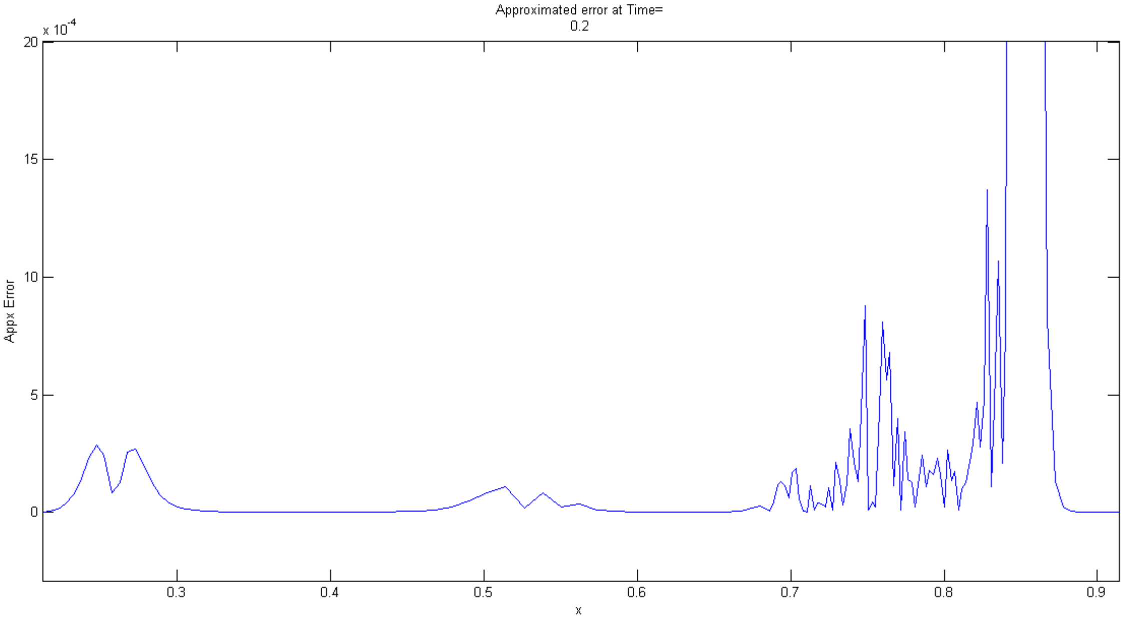


Figure 6.34: Semi-Discrete in Time Estimated Error for Density for the Euler Equations in 1D Zoomed.

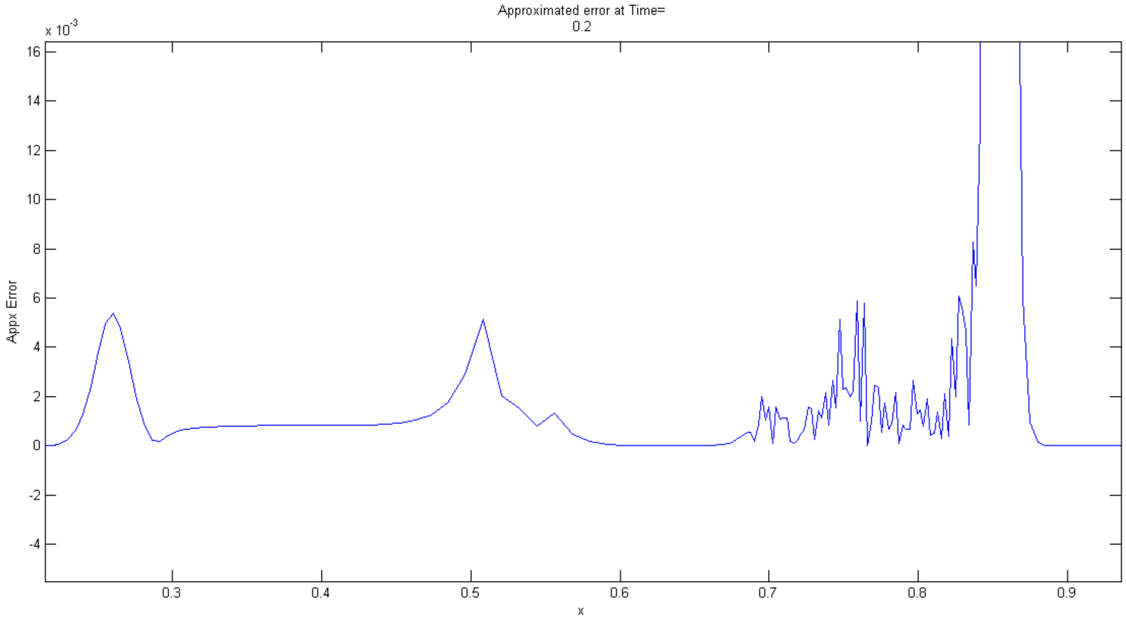


Figure 6.35: Semi-Discrete in Time Estimated Error for Velocity for the Euler Equations in 1D Zoomed.

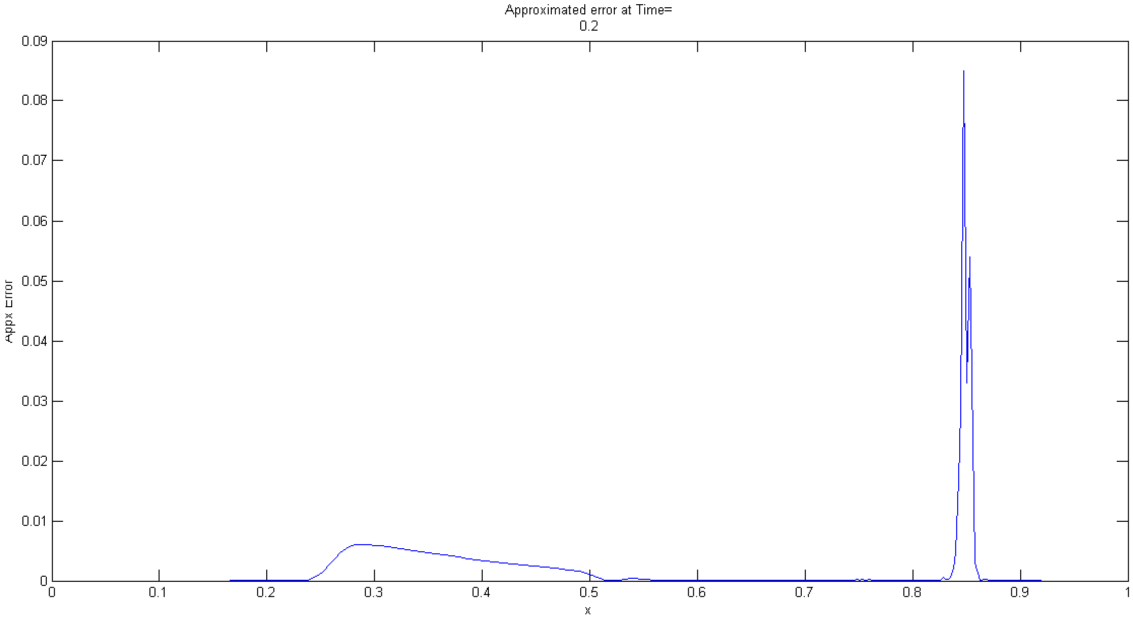


Figure 6.36: Semi-Discrete in Space Estimated Error for Pressure for the Euler Equations in 1D.

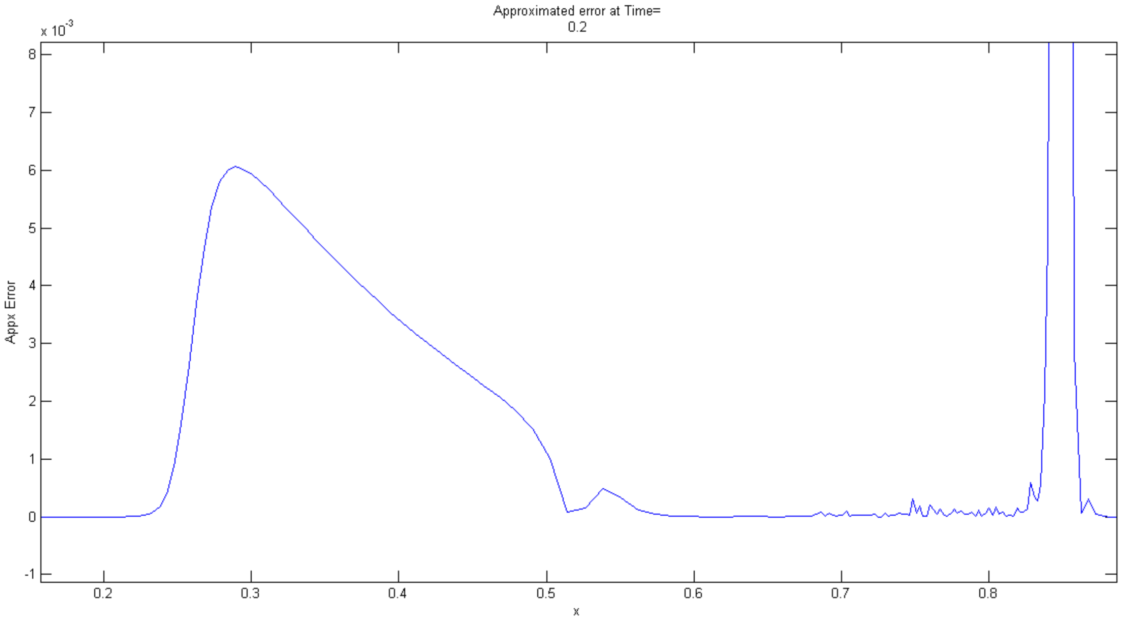


Figure 6.37: Semi-Discrete in Space Estimated Error for Pressure for the Euler Equations in 1D Zoomed.

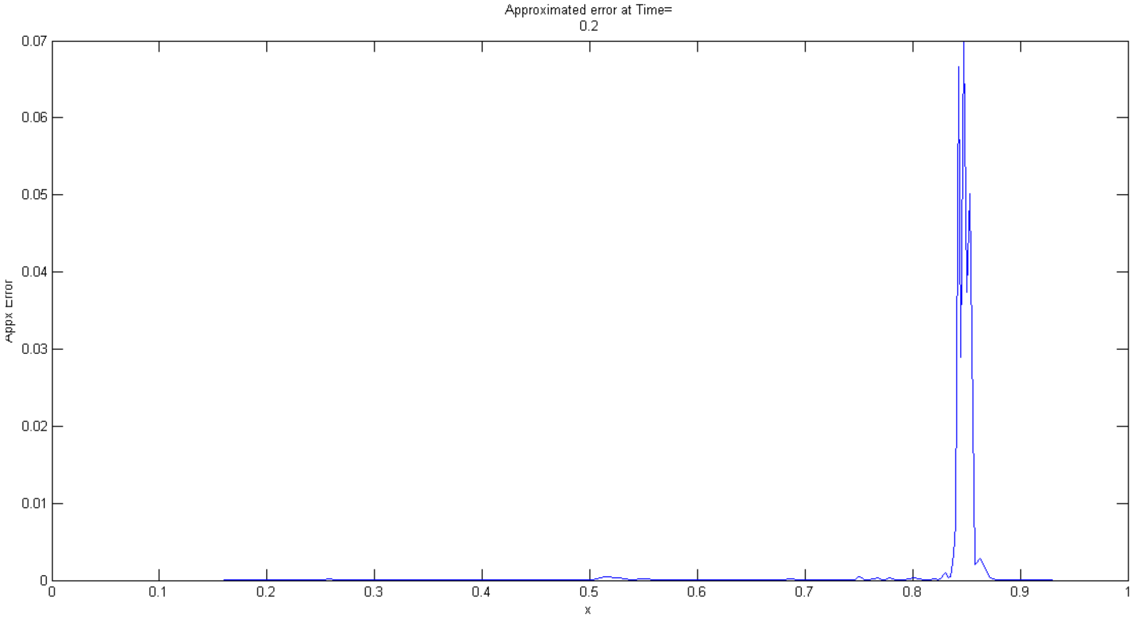


Figure 6.38: Fully-Discrete Estimated Error for Density for the Euler Equations in 1D.

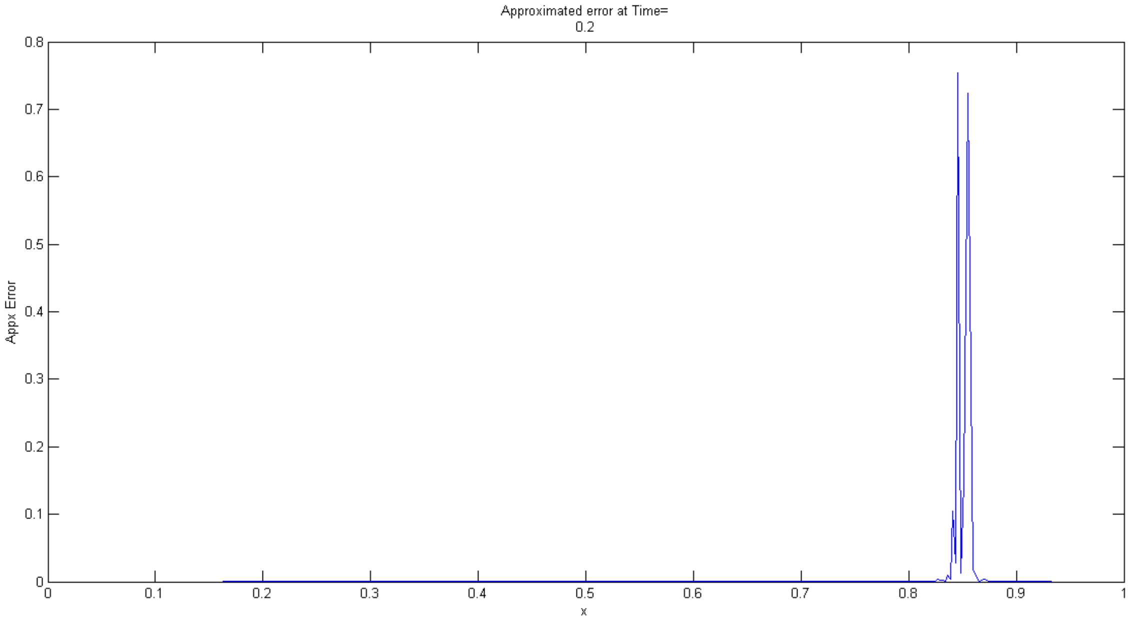


Figure 6.39: Fully-Discrete Estimated Error for Velocity for the Euler Equations in 1D.

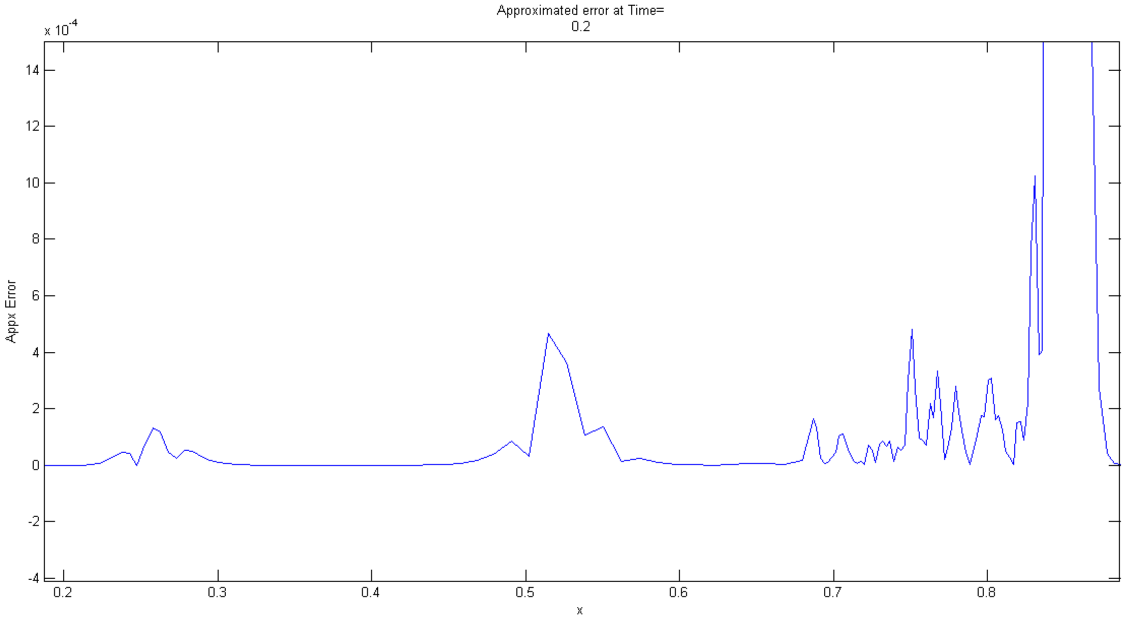


Figure 6.40: Fully-Discrete Estimated Error for Density for the Euler Equations in 1D Zoomed.

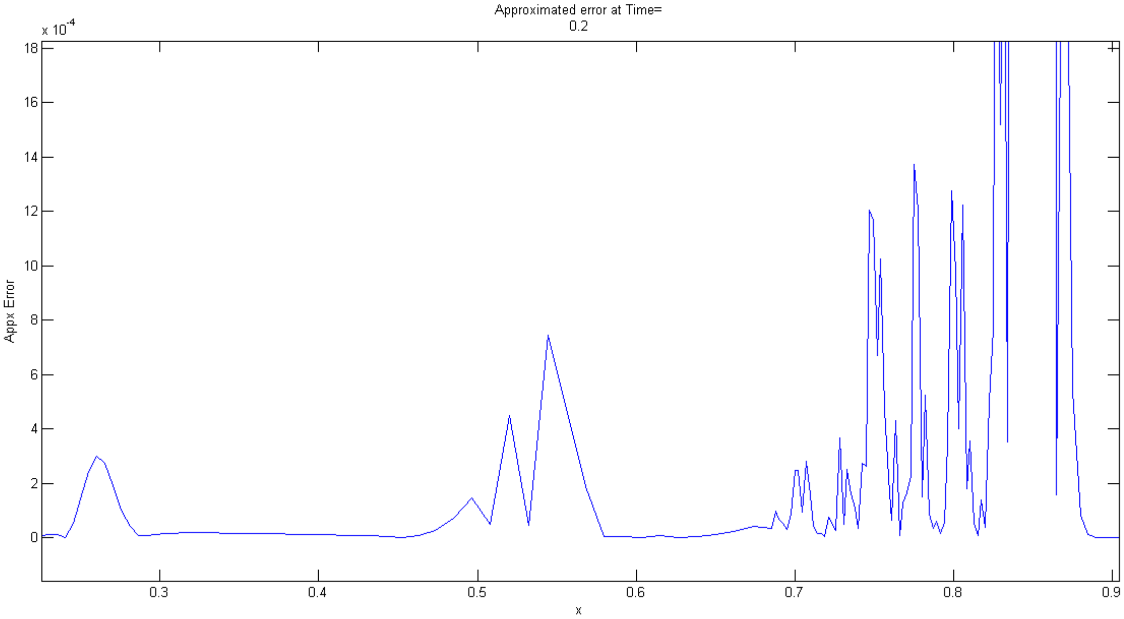


Figure 6.41: Fully-Discrete Estimated Error for Velocity for the Euler Equations in 1D Zoomed.

# Chapter 7

# 2D System of Three Equations

In Chapter 5 we saw how to express the 1D system of the Euler Equations in a fully-discrete form. We now wish to extend the approach to 2D.

The Main Scheme from [64], uses the Position Verlet scheme from Section (3.3.1) for the temporal discretisations and a Galerkin approach from Section (4.2.1) for the spatial discretisations.

In Chapters 3 and 4 we developed a method to estimate the error, in the nodal positions in one step for the Main Scheme in 1D. This was achieved by using the Main Scheme for the temporal derivatives and an alternative scheme, of the same order, for the spatial derivative, which produces an estimated computable term for the spatial derivative in the leading term of the truncation error. Likewise, by using the Main Scheme for the spatial discretisations and a different temporal scheme, we achieved a computable estimate to temporal derivatives in the truncation error. Furthermore, by combining a variation in both space and time approximations an entirely computable estimate to the error in one step was achieved.

We now look at the fully-discrete system of the Euler Equations in 2D. We will first determine an error estimate in one step for the position $\boldsymbol{x}$.

The next section will be dedicated to two different temporal schemes in 2D. Following on from this, two spatial schemes in 2D will be demonstrated.

The fully-discrete truncation errors for the nodal position will be obtained by fixing the temporal scheme and varying the spatial scheme and vice versa. From the truncation errors we determine an estimation to the error in one step for the fixed temporal scheme and spatial scheme.

By using different schemes in both space and time we obtain a truncation error in which

both temporal and spatial terms in the truncation error are computable.
We begin the extension by investigating the 2D semi-discrete in time schemes followed by the 2D semi-discrete in space schemes.

## 7.1 System of Three Equations

The Lagrangian form of the Euler Equations, which connects the moving point $\boldsymbol{x}(\boldsymbol{\xi}, t)$ to the velocity, $\boldsymbol{u}(\boldsymbol{\xi}, t)$, is

$$\frac{\partial \boldsymbol{x}}{\partial t} = \boldsymbol{u}. \tag{7.1}$$

From Section (2.6), we have the Euler Equations written for a general case. In 2D these are expressed as

- Conservation of Mass

$$\frac{\partial m}{\partial t} = 0, \tag{7.2}$$

where $m = \rho|\boldsymbol{J}|$.

- Momentum Balance,

$$\frac{\partial \boldsymbol{u}}{\partial t} = -\frac{\boldsymbol{C}\nabla p}{m}. \tag{7.3}$$

- Specific Internal Energy Balance,

$$\frac{\partial \varepsilon}{\partial t} = -\frac{p\boldsymbol{C}}{m}\nabla \cdot \boldsymbol{u}, \tag{7.4}$$

where $\boldsymbol{\xi} = (\xi, \eta)$, we have $\boldsymbol{x} = \boldsymbol{x}(\boldsymbol{\xi}, t) = (x(\boldsymbol{\xi}, t), y(\boldsymbol{\xi}, t))$, $\boldsymbol{u} = \boldsymbol{u}(\boldsymbol{\xi}, t) = (u(\boldsymbol{\xi}, t), v(\boldsymbol{\xi}, t))$, $m = m(\boldsymbol{\xi})$, $p = p(\boldsymbol{\xi}, t)$, $\nabla = \nabla_{\boldsymbol{\xi}} = \left(\frac{\partial}{\partial \xi}, \frac{\partial}{\partial \eta}\right)$ and $\varepsilon = \varepsilon(\boldsymbol{\xi}, t)$. Here, $\boldsymbol{C}$ is the adjoint matrix which is introduced as a result of inverting the Jacobian when transforming the co-ordinates into $(\xi, \eta)$ space. It takes the form

$$\boldsymbol{C} = \begin{pmatrix} \frac{\partial y}{\partial \eta} & -\frac{\partial y}{\partial \xi} \\ -\frac{\partial x}{\partial \eta} & \frac{\partial x}{\partial \xi} \end{pmatrix}. \tag{7.5}$$

In order to look at the temporal and spatial schemes separately, the schemes for the system of three equations will be expressed in their semi-discrete form.

## 7.2 Temporal Discretisation

To discretise the system of three equations in a semi-discrete manner, we drop the spatial dependencies from the Euler Equations, such that

$$\frac{d\boldsymbol{x}}{dt} = \boldsymbol{u} \tag{7.6}$$

From Section 2.6, we have the system of three equations written for a general case. In 2D this is expressed as;

- Conservation of Mass
  $$\frac{dm}{dt} = 0, \tag{7.7}$$
  where $m = \rho|\boldsymbol{J}|$.
- Momentum Equation
  $$\frac{d\boldsymbol{u}}{dt} = \boldsymbol{f}, \tag{7.8}$$
  where $\boldsymbol{f} = -\frac{\boldsymbol{C}\nabla p}{m}$.
- Specific Internal Energy Equation,
  $$\frac{d\varepsilon}{dt} = pg, \tag{7.9}$$
  where $g = -\frac{p\boldsymbol{C}\nabla\cdot\boldsymbol{u}}{\rho|\boldsymbol{J}|}$.

Since we have eliminated the spatial dependencies, the partial derivatives now become ordinary derivatives, where we have $\boldsymbol{x} = \boldsymbol{x}(t)$, $\boldsymbol{u} = \boldsymbol{u}(t)$, $\boldsymbol{f} = \boldsymbol{f}(t)$, $g = g(t)$, $p = p(t)$, and $\varepsilon = \varepsilon(t)$. We now look at discretising the temporal derivatives.

### 7.2.1 Scheme 1

In Section 3.7.1 we saw how the Position Verlet Scheme was extended to the system of three equations in 1D. By extending this to 2D we obtain the scheme.

### Predictor

- Nodal positions are calculated at a half timestep by the semi-discrete step

$$\boldsymbol{x}^{n+\frac{1}{2}} = \boldsymbol{x}^n + \frac{k}{2}\boldsymbol{u}^n. \tag{7.10}$$

- The density, $\rho$, is calculated at the half timestep using the conservation of mass,

$$\begin{aligned} m^{n+\frac{1}{2}} &= m^n \\ \Rightarrow \rho^{n+\frac{1}{2}}|\boldsymbol{J}|^{n+\frac{1}{2}} &= m^n \\ \Rightarrow \rho^{n+\frac{1}{2}} &= \frac{m^n}{|\boldsymbol{J}|^{n+\frac{1}{2}}}. \end{aligned} \tag{7.11}$$

- Half timestep energies can be calculated from,

$$\varepsilon^{n+\frac{1}{2}} = \varepsilon^n + k\frac{p^n g^n}{2}. \tag{7.12}$$

- Using the equation of state the pressure is now updated at the half timestep using,

$$p^{n+\frac{1}{2}} = (\gamma - 1)\rho^{n+\frac{1}{2}}\varepsilon^{n+\frac{1}{2}}. \tag{7.13}$$

### Corrector

- Full timestep velocities are calculated from the half step densities using

$$\boldsymbol{u}^{n+1} = \boldsymbol{u}^n + k\boldsymbol{f}^{n+\frac{1}{2}}. \tag{7.14}$$

- The averaged velocities over the full timestep is,

$$\begin{aligned} \bar{\boldsymbol{u}} &= \frac{1}{2}(\boldsymbol{u}^n + \boldsymbol{u}^{n+1}) \\ &= \boldsymbol{u}^n + \frac{k}{2}\boldsymbol{f}^{n+\frac{1}{2}}, \end{aligned} \tag{7.15}$$

  which is used to update the nodal positions at the full timestep by

$$\boldsymbol{x}^{n+1} = \boldsymbol{x}^n + k\boldsymbol{u}^n + \frac{k^2}{2}\boldsymbol{f}^{n+\frac{1}{2}}. \tag{7.16}$$

- The density, $\rho$, can now be calculated at the full timestep using the conservation of mass,

$$\begin{aligned} m^{n+1} &= m^n && (7.17)\\ \Rightarrow \rho^{n+1}|\boldsymbol{J}|^{n+1} &= m^n \\ \Rightarrow \rho^{n+1} &= \frac{m^n}{|\boldsymbol{J}|^{n+1}}. && (7.18) \end{aligned}$$

- Energies are evaluated at the end of the time step by,

$$\varepsilon^{n+1} = \varepsilon^n + kp^{n+\frac{1}{2}}g^{n+\frac{1}{2}}. \qquad (7.19)$$

- Finally, calculate the pressures at $t^{n+1}$, from the equation of state, where,

$$p^{n+1} = (\gamma - 1)\rho^{n+1}\varepsilon^{n+1}. \qquad (7.20)$$

### 7.2.2 Scheme 2

We extend the Scheme, from Section 3.7.2, to the 2D case, obtaining the following scheme.

#### Predictor

We use the suffix $_*$, to denote the predictor stage, since the predictor and corrector are calculated at the same timestep.

- Full timestep velocities are predicted by,

$$\boldsymbol{u}_*^{n+1} = \boldsymbol{u}^n + k\boldsymbol{f}^n. \qquad (7.21)$$

- A predicted average velocity is obtained from,

$$\begin{aligned} \bar{\boldsymbol{u}}_* &= \frac{1}{2}(\boldsymbol{u}^n + \boldsymbol{u}_*^{n+1}) \\ &= \boldsymbol{u}^n + \frac{k}{2}\boldsymbol{f}^n. && (7.22) \end{aligned}$$

- Nodal positions are predicted using,

$$\begin{aligned}\boldsymbol{x}_*^{n+1} &= \boldsymbol{x}^n + k\bar{\boldsymbol{u}}_* \\ &= \boldsymbol{x}^n + k\boldsymbol{u}^n + \frac{k^2}{2}\boldsymbol{f}^n. \end{aligned} \tag{7.23}$$

- The density can be predicted at the full timestep by using the conservation of mass,

$$\begin{aligned} m_*^{n+1} &= m^n \\ \Rightarrow \rho_*^{n+1}|\boldsymbol{J}_*|^{n+1} &= m^n \\ \Rightarrow \rho_*^{n+1} &= \frac{m^n}{|\boldsymbol{J}_*|^{n+1}}. \end{aligned} \tag{7.24}$$

- The specific internal energy can now be predicted at the half timestep using,

$$\varepsilon_*^{n+1} = \varepsilon^n + kp^n g_*^{n+\frac{1}{2}}. \tag{7.25}$$

- The equation of state is used to obtain the pressure at the predicted full timestep.

### Corrector

- Corrected full timestep velocities are calculated using the predicted pressures such that,

$$\boldsymbol{u}^{n+1} = \boldsymbol{u}^n + k\boldsymbol{f}_*^{n+1}. \tag{7.26}$$

- Averaged velocities are corrected by

$$\begin{aligned}\bar{\boldsymbol{u}} &= \frac{1}{2}(\boldsymbol{u}^n + \boldsymbol{u}^{n+1}) \\ &= \boldsymbol{u}^n + \frac{k}{2}\boldsymbol{f}_*^{n+1}. \end{aligned} \tag{7.27}$$

- The averaged velocities are used to update the nodal positions,

$$\begin{aligned}\boldsymbol{x}^{n+1} &= \boldsymbol{x}^n + k\bar{\boldsymbol{u}} \\ &= \boldsymbol{x}^n + k\boldsymbol{u}^n + \frac{k^2}{2}\boldsymbol{f}_*^{n+1}. \end{aligned} \tag{7.28}$$

- Corrected densities are again obtained from the conservation of mass, where,

$$\begin{aligned} m^{n+1} &= m^n \\ \Rightarrow \rho^{n+1}|\boldsymbol{J}|^{n+1} &= m^n \\ \Rightarrow \rho^{n+1} &= \frac{m^n}{|\boldsymbol{J}|^{n+1}}. \end{aligned} \tag{7.29}$$

- Energies are corrected at the full timestep using

$$\varepsilon^{n+1} = \varepsilon^n + kp_*^{n+1}g_*^{n+1}. \tag{7.30}$$

- Finally, full timestep pressures are updated using the equation of state.

Now we have stated the two schemes for the temporal derivatives, we seek two schemes to approximate the spatial derivatives in the momentum and specific internal energy equation.

## 7.3 Spatial Discretisation

Spatial discretisations occur in the momentum equation and the specific internal energy equation. The spatial approximations are obtained using a finite element approach. As we have seen in Chapter 4, we seek two spatial schemes in order to obtain a computable error estimate later.

Firstly, we will look at the two different spatial discretisations for the momentum equation, then these methods will be applied to the specific internal energy equation.

### 7.3.1 Momentum Equation

Where $\dot{\boldsymbol{u}} = \frac{\partial \boldsymbol{u}}{\partial t}(\boldsymbol{\xi}, t)$, the momentum equation becomes

$$m\dot{\boldsymbol{u}} = -\boldsymbol{C}\nabla p, \tag{7.31}$$

where we drop the temporal dependencies, such that $m = m(\boldsymbol{\xi})$, $\dot{\boldsymbol{u}}(\boldsymbol{\xi})$, $\nabla = \left(\frac{\partial}{\partial \xi}, \frac{\partial}{\partial \eta}\right)$ and $p = p(\boldsymbol{\xi})$.

We now express the momentum equation in a weak form.

### 7.3.2 Weak Form of Momentum Equation

The momentum equation is multiplied by a test function $\omega(\xi,\eta)$, where $\omega \in H^1(\Omega)$, $\Omega$ is a unit square in $(\xi,\eta)$ space, resulting in

$$\int_\Omega \omega m \dot{\boldsymbol{u}} d\Omega = -\int_\Omega \omega \boldsymbol{C} \nabla p d\Omega. \tag{7.32}$$

Using Green's theorem we obtain,

$$\int_\Omega \omega m \dot{\boldsymbol{u}} d\Omega = -\oint_S \omega p d\boldsymbol{S} + \int_\Omega p \boldsymbol{C} \nabla \omega d\Omega, \tag{7.33}$$

where $\boldsymbol{S}$ is the boundary of $\Omega$. This can now be used to obtain two different expressions for the spatial derivative.
We evaluate the integrals separately in Appendix E.1. To use the weak form of the momentum equation we introduce a mesh of square elements, as in Figure (7.1), and let $p$ and $m$ be piecewise constant functions within each element. The velocity $\boldsymbol{u}$, and acceleration $\dot{\boldsymbol{u}}$ are taken to be the piecewise bilinear functions,

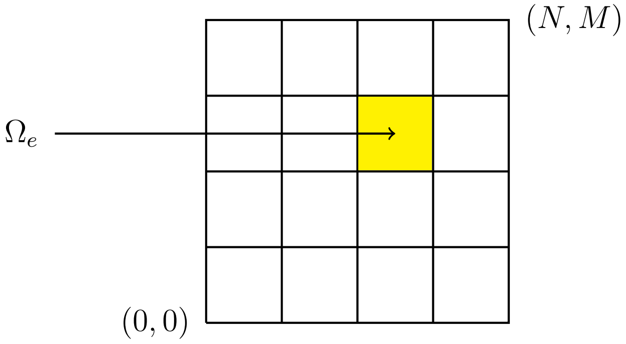


Figure 7.1: Computational Element as a Subregion of Transformed Region in 2D.

$$\boldsymbol{u} = \sum_{i=1}^{N+1} \boldsymbol{u}_i N_i(\boldsymbol{\xi}) \quad , \quad \dot{\boldsymbol{u}} = \sum_{i=1}^{N+1} \dot{\boldsymbol{u}}_i N_i(\boldsymbol{\xi}), \tag{7.34}$$

where the bilinear basis functions are,

$$N_1 = \frac{1}{4}(1-\xi)(1-\eta), \tag{7.35}$$

$$N_2 = \frac{1}{4}(1+\xi)(1-\eta), \tag{7.36}$$

$$N_3 = \frac{1}{4}(1+\xi)(1+\eta), \tag{7.37}$$

$$N_4 = \frac{1}{4}(1-\xi)(1+\eta). \tag{7.38}$$

## 7.4 Galerkin Approach

For the Galerkin approach the test function in the weak form of the momentum equation is chosen as $\omega = N_j$. Using the above information and substituting into eqn. (7.33), where the integral is now evaluated over an element $\Omega_e$, we have

$$\int_{\Omega_e} mN_j \sum_{i=1}^{4} \dot{\boldsymbol{u}}_i N_i d\xi d\eta = -\oint_{\partial\Omega_e} N_j p d\boldsymbol{S}_e + \int_{\Omega_e} p\boldsymbol{C}\nabla N_j d\xi d\eta, \tag{7.39}$$

for $j = 1, \cdots, 4$, where $d\boldsymbol{S}_e$ is the side differential vector for each normal direction. From this we can take $m$ and $p$ outside the integrals, since they are constant within the element. We interchange the integral and the sum. For the boundary term, we take $p$ outside the integral as an average of the neighbouring elements, at the mid-point of the relevant side, we call this $p_{side}$. This results in,

$$m\sum_{i=1}^{4} \dot{\boldsymbol{u}}_i \int_{\Omega_e} N_j N_i d\xi d\eta = -p_{side} \oint_{\partial\Omega_e} N_j d\boldsymbol{S}_e + p\int_{\Omega_e} \boldsymbol{C}\nabla N_j d\xi d\eta. \tag{7.40}$$

### 7.4.1 System Assembly

When assembled, the bilinear basis functions all vanish on the internal boundaries. Hence, the boundary terms disappear in eqn. (7.40). In order to solve the resulting system conveniently, as in [64, 10], we lump the off-diagonals of the mass matrix adding them to the diagonal, leading to

$$m\dot{\boldsymbol{u}}_i \int_{\Omega_e} N_i d\xi d\eta = p\int_{\Omega_e} \boldsymbol{C}\nabla N_j d\xi d\eta. \tag{7.41}$$

The shape function integrals are evaluated in Appendix (E.1).

The system is assembled by calculating the forcing term for each element. The element forcing terms are summed for all elements surrounding the required node, as illustrated in Figure 7.2.

This results in the following solution which we call Scheme A.

### 7.4.2 Scheme A

Using the labeling system, such that the nodes of an element are labeled in an anticlockwise manner, as in Figure 7.3, we can obtain the forcing term from the right hand side of eqn. (7.41).

For a node $(i,j)$ the contribution to the forcing term in the $\xi$ direction is

$$\begin{aligned} f^{\xi}_{ij} \;&=\; p1\int_{\eta_{i-1}}^{\eta_i}\int_{\xi_{i-1}}^{\xi_i}\left(C_{11}\frac{\partial N_1}{\partial \xi}+C_{12}\frac{\partial N_1}{\partial \eta}\right)d\xi d\eta + p2\int_{\eta_{i-1}}^{\eta_i}\int_{\xi_{i-1}}^{\xi_i}\left(C_{11}\frac{\partial N_2}{\partial \xi}+C_{12}\frac{\partial N_2}{\partial \eta}\right)d\xi d\eta \\ &+p3\int_{\eta_{i-1}}^{\eta_i}\int_{\xi_{i-1}}^{\xi_i}\left(C_{11}\frac{\partial N_3}{\partial \xi}+C_{12}\frac{\partial N_3}{\partial \eta}\right)d\xi d\eta + p4\int_{\eta_{i-1}}^{\eta_i}\int_{\xi_{i-1}}^{\xi_i}\left(C_{11}\frac{\partial N_4}{\partial \xi}+C_{12}\frac{\partial N_4}{\partial \eta}\right)d\xi d\eta \end{aligned} \quad (7.42)$$

and in the $\eta$ direction we have

$$\begin{aligned} f^{\eta}_{ij} \;&=\; p1\int_{\eta_{i-1}}^{\eta_i}\int_{\xi_{i-1}}^{\xi_i}\left(C_{21}\frac{\partial N_1}{\partial \xi}+C_{22}\frac{\partial N_1}{\partial \eta}\right)d\xi d\eta + p2\int_{\eta_{i-1}}^{\eta_i}\int_{\xi_{i-1}}^{\xi_i}\left(C_{21}\frac{\partial N_2}{\partial \xi}+C_{22}\frac{\partial N_2}{\partial \eta}\right)d\xi d\eta \\ &+ p3\int_{\eta_{i-1}}^{\eta_i}\int_{\xi_{i-1}}^{\xi_i}\left(C_{21}\frac{\partial N_3}{\partial \xi}+C_{22}\frac{\partial N_3}{\partial \eta}\right)d\xi d\eta + p4\int_{\eta_{i-1}}^{\eta_i}\int_{\xi_{i-1}}^{\xi_i}\left(C_{21}\frac{\partial N_4}{\partial \xi}+C_{22}\frac{\partial N_4}{\partial \eta}\right)d\xi d\eta. \end{aligned} \quad (7.43)$$

On computation in the $\xi$ direction, and using the nodal stencil in the $(\xi,\eta)$ space, from Figure 7.4, we obtain

$$\begin{aligned} m_{ij}\dot{u}^{\xi}_{ij} = f^{\xi}_{ij} \;&=\; \frac{1}{2}p_{i+\frac{1}{2},j+\frac{1}{2}}(y_{i+1,j}-y_{i,j+1}) \\ &+\; \frac{1}{2}p_{i-\frac{1}{2},j+\frac{1}{2}}(y_{i,j+1}-y_{i-1,j}) \\ &+\; \frac{1}{2}p_{i-\frac{1}{2},j-\frac{1}{2}}(y_{i-1,j}-y_{i,j-1}) \\ &+\; \frac{1}{2}p_{i+\frac{1}{2},j-\frac{1}{2}}(y_{i,j-1}-y_{i+1,j}), \end{aligned} \quad (7.44)$$

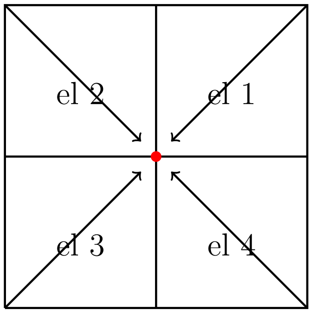


Figure 7.2: Force Contributions From The Elements Surrounding a Node in 2D.

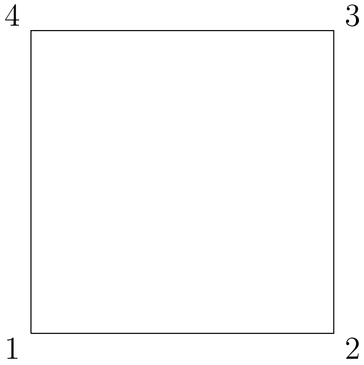


Figure 7.3: Node Labeling in a 2D Element.

the nodal mass $m_{ij}$, is the contribution of mass, $m_e \int_{\Omega_e} N_i d\xi d\eta$, from each surrounding element to the node $ij$.

Similarly, we can obtain the force term in the $\eta$ direction.

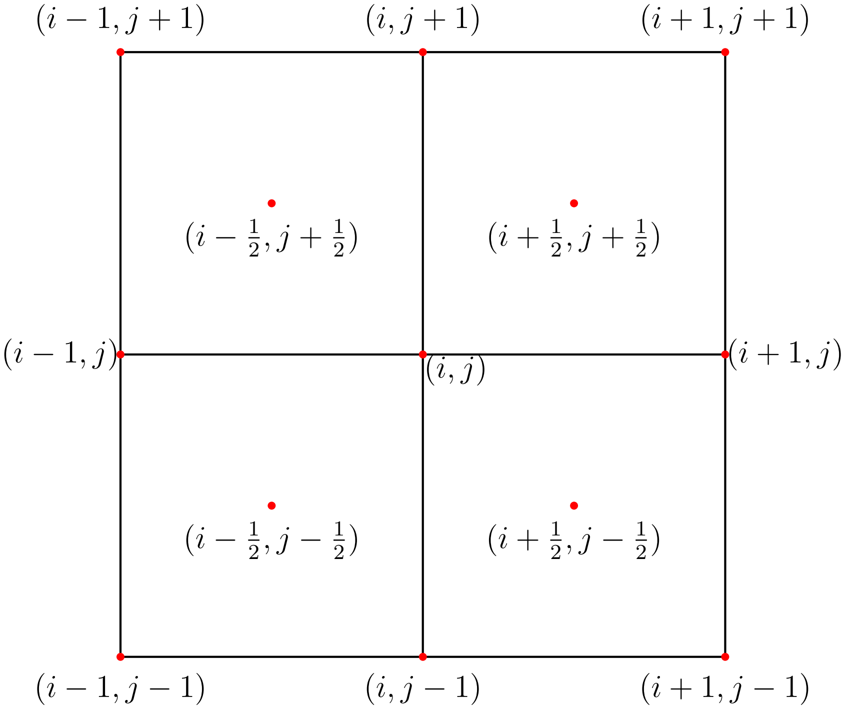
The discretised forcing terms shall be used later in obtaining an error estimation.

## 7.5 Petrov-Galerkin Approach

An alternative scheme to the above has been investigated in Section 4.2.3. This idea is now extended to the 2D problem.

To determine the alternative scheme we seek a different basis function $\psi$, which still gives a diagonal matrix due to the orthogonality expressed in

$$\int \psi_i N_j d\Omega = \delta_{ij}, \tag{7.45}$$



$(i-1,j+1)$
$(i,j+1)$
$(i+1,j+1)$
$(i-\frac{1}{2},j+\frac{1}{2})$
$(i+\frac{1}{2},j+\frac{1}{2})$
$(i-1,j)$
$(i,j)$
$(i+1,j)$
$(i-\frac{1}{2},j-\frac{1}{2})$
$(i+\frac{1}{2},j-\frac{1}{2})$
$(i-1,j-1)$
$(i,j-1)$
$(i+1,j-1)$

Figure 7.4: Nodal and Element Points Surrounding the Node of Interest in 2D.

where $N_j$ is the bilinear finite element function and $\delta_{ij}$ is a matrix with zeros everywhere apart from the diagonal.

From the integral on the left hand side of eqn. (7.40) and using, $\psi$, as a test function over an element in $(\xi,\eta)$ space, the left hand side of eqn. (7.40) becomes,

$$\int_{\Omega_e} \psi N d\xi d\eta.$$

Let $\psi_i = \sum_{i=1}^{4} a_{ik} N_k$ and,

$$\psi = A\mathbf{N}, \tag{7.46}$$

where $A$ is a $4 \times 4$ matrix of constants and $\mathbf{N}$ is a vector of $N$'s. We obtain

$$\int_{\Omega_e} A\mathbf{N}\mathbf{N}^T d\xi d\eta = AM_e, \tag{7.47}$$

where $M_e = \int_{\Omega_e} \mathbf{N}\mathbf{N}^T d\xi d\eta$ is the standard element mass matrix. We seek, $\boldsymbol{A}$ such that $\boldsymbol{AM}_e = \boldsymbol{D}$ where $\boldsymbol{D}$ is a diagonal matrix or identity matrix. Hence, $\boldsymbol{A} = \boldsymbol{DM}_e^{-1}$ which allows the alternative basis functions to be calculated in a vector as,

$$\boldsymbol{\psi} = \boldsymbol{A}\mathbf{N} = \boldsymbol{D}\boldsymbol{M}_e^{-1}\mathbf{N}.$$

The bilinear finite element functions have been defined above, allowing $\boldsymbol{M}_e^{-1}$ to be calculated as

$$\boldsymbol{M}_e^{-1} = \begin{pmatrix} 4 & -2 & 1 & -2 \\ -2 & 4 & -2 & 1 \\ 1 & -2 & 4 & -2 \\ -2 & 1 & -2 & 4 \end{pmatrix}. \tag{7.48}$$

Therefore, we can calculate the alternative basis functions, in terms of the bilinear basis functions, $\mathbf{N}$, as

$$\boldsymbol{\psi} = \boldsymbol{D}\boldsymbol{M}_e^{-1}\mathbf{N} \tag{7.49}$$

$$= \begin{pmatrix} 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{pmatrix} \begin{pmatrix} 4 & -2 & 1 & -2 \\ -2 & 4 & -2 & 1 \\ 1 & -2 & 4 & -2 \\ -2 & 1 & -2 & 4 \end{pmatrix} \begin{pmatrix} N_1 \\ N_2 \\ N_3 \\ N_4 \end{pmatrix}, \tag{7.50}$$

where we use $\boldsymbol{D} = \boldsymbol{I}$. Therefore, we have

$$\begin{pmatrix} \psi_1 \\ \psi_2 \\ \psi_3 \\ \psi_4 \end{pmatrix} = \begin{pmatrix} 4N_1 - 2N_2 + N_3 - 2N_4 \\ -2N_1 + 4N_2 - 2N_3 + N_4 \\ N_1 - 2N_2 + 4N_3 - 2N_4 \\ -2N_1 + N_2 - 2N_3 + 4N_4 \end{pmatrix} = \begin{pmatrix} \frac{1}{4}(1-3\xi)(1-3\eta) \\ \frac{1}{4}(1+3\xi)(1-3\eta) \\ \frac{1}{4}(1+3\xi)(1+3\eta) \\ \frac{1}{4}(1-3\xi)(1+3\eta) \end{pmatrix}. \tag{7.51}$$

From eqn. (7.33), the weak form of the momentum equation in an element is

$$m\int_{\Omega_e} \omega\dot{\boldsymbol{u}}d\Omega = -\oint_{\partial\Omega_e} \omega p d\boldsymbol{S} + \int_{\Omega_e} p\boldsymbol{C}\nabla\omega d\Omega. \tag{7.52}$$

To evaluate the weak form of the momentum equation, we introduce the alternative basis function $\omega = \psi_j$. We still assume $m$ and $p$ to be piecewise constant functions within each element. The velocity, $\boldsymbol{u}$, and acceleration $\dot{\boldsymbol{u}}$ are denoted by the piecewise bilinear functions,

$$\boldsymbol{u} = \sum_{i=1}^{N+1} \boldsymbol{u}_i N_i(\boldsymbol{\xi}) \quad , \quad \dot{\boldsymbol{u}} = \sum_{i=1}^{N+1} \dot{\boldsymbol{u}}_i N_i(\boldsymbol{\xi}). \tag{7.53}$$

The integral is now evaluated over an element, giving

$$m\int_{\Omega_e}\psi_j\sum_{i=1}^{4}\dot{\boldsymbol{u}}_iN_i d\xi d\eta = -\oint_{\partial\Omega_e}\psi_j p d\boldsymbol{S} + \int_{\Omega_e} pC\nabla\psi_j d\xi d\eta. \tag{7.54}$$

From this we can take $m$ and $p$ outside the integrals since they are constant within the element. Interchanging the sum and the integral results in the acceleration vector, $\dot{\boldsymbol{u}}$, being taken outside the integral.

### 7.5.1 System Assembly

In this case the basis function $\psi$ does not vanish at the element boundaries.

For the boundary term, we take $p$ outside the integral as an average of the neighbouring elements to the side we are on, calling this $p_{side}$. By orthogonality, the result for an element is

$$m\dot{\boldsymbol{u}}_i\int_{\Omega_e}\psi_i N_i d\xi d\eta = -p_{side}\oint_{\partial\Omega_e}\psi_j d\boldsymbol{s} + p\int_{\Omega_e}\boldsymbol{C}\nabla\psi_j d\xi d\eta. \tag{7.55}$$

The system is assembled by summing the information from the elements surrounding a node. Using the shape function integrals, which are evaluated in Appendix (E.2), for this case, we obtain the following scheme, which we call Scheme B.

### 7.5.2 Scheme B

Using Appendix (E.2), we are able to obtain a discretised form of eqn. (7.55). This is done by summing boundary terms for an element along with the element's forcing contribution to node $ij$. In the $\xi$ direction we have

$$\begin{aligned} m_{ij}\dot{u}^{\xi}_{ij} = f^{\xi}_{ij} &= \frac{1}{4}(y_2(p_{11} - p_{12} + 15p_8 - 15p_9) \\ &+ y_3(-p_{11} + p_7) + y_4(p_3 + 15p_4 - p_7 - 15p_8) \\ &+ y_5(-p_1 + p_2 - 15p_4 + 15p_5) + y_6(p_1 - p_3) \\ &+ y_7(-p_2 + p_6) + y_8(p_{10} - 15p_5 - p_6 + 15p_9) \\ &+ y_9(-p_{10} + p_{12})). \end{aligned} \tag{7.56}$$

where the positions of the values are illustrated in Figures (E.3). The nodal mass $m_{ij}$, is the contribution of mass, $m_e \int_{\Omega_e} \psi_j N_i d\xi d\eta$, from each surrounding element to the node $ij$.

The two schemes we have obtained for the spatial derivatives will be applied later to estimate an error. First, we look at determining an error estimate in terms of the temporal derivatives.

## 7.6 Semi-Discrete Time Error

We have seen in Section 3.9.2 how to use the truncation error for the nodal position $\boldsymbol{x}$ to obtain an estimated error in one time step for velocity and density. We extend this method to obtain a semi-discrete in time error estimate for density. We do not extend the result for velocity in the 2D case, since it is a nodal value and is not clear which elements would require refining.

### 7.6.1 Estimated Error for Density

To evaluate the estimated error for density in 2D, we require an estimated error to the Jacobian $|\boldsymbol{J}|$.

Following on from the 1D result for the nodal truncation error, from Section (3.8.1), we extend the result to 2D for Scheme 1, using eqn. (3.73), such that

$$\tau_n(L_1) = -\frac{k^2}{12}\boldsymbol{x}''' + \mathcal{O}(k^3), \tag{7.57}$$

where the Jacobian follows as

$$\tau_n(L_1) = -\frac{k^2}{12}\boldsymbol{J}''' + \mathcal{O}(k^3). \tag{7.58}$$

Similarly, for Scheme B expressed in Section (3.8.2), the result can be extended to 2D using eqn. (3.77), resulting in

$$\tau_n(L_2) = -\frac{k^2}{3}\boldsymbol{x}''' + \mathcal{O}(k^3), \tag{7.59}$$

where the Jacobian follows

$$\tau_n(L_2) = -\frac{k^2}{3}|\boldsymbol{J}|''' + \mathcal{O}(k^3). \tag{7.60}$$

From Section 3.9, we obtain an estimated error in one time-step for the nodal position $x$ in eqn. (3.81). This can be extended to the 2D case, using equations (7.58) and (7.60), resulting in

$$e[|\boldsymbol{J}|] = |\boldsymbol{J}|(t^n + k) - (|\boldsymbol{J}|_1)^{n+1} = k\tau_n(L_1) = \frac{\left((|\boldsymbol{J}|_2)^{n+1} - (|\boldsymbol{J}|_1)^{n+1}\right)}{3} + \mathcal{O}(k^4). \tag{7.61}$$

The estimated error in one time-step for the derivative of $|\boldsymbol{J}|$ becomes

$$e\left[\frac{d|\boldsymbol{J}|}{dt}\right] = \tau_n(L_1) = \frac{\left((|\boldsymbol{J}|_2)^{n+1} - (|\boldsymbol{J}|_1)^{n+1}\right)}{3k} + \mathcal{O}(k^3). \tag{7.62}$$

The result in eqn. (7.62) will be applied in the fully-discrete case, whereas eqn. (7.61) can now be applied to obtain an estimated error for density.
In 2D the mass is

$$m = \rho|\boldsymbol{J}|.$$

Therefore, from eqn. (7.7) we have

$$\begin{aligned} \frac{dm}{dt} &= 0 \\ \Rightarrow m^{n+1} &= m^n \\ \Rightarrow \rho^{n+1} &= \frac{m^n}{|\boldsymbol{J}|^{n+1}}. \end{aligned}$$

The 1D result is demonstrated in eqn. (3.58), extending this result to 2D we obtain

$$e[\rho] = \rho(t^n + k) - (\rho_1)^{n+1} \quad = \quad \frac{m^n\left(|\boldsymbol{J}|^{n+1} - |\boldsymbol{J}|(t^n + k)\right)}{|\boldsymbol{J}|(t^n + k)|\boldsymbol{J}|^{n+1}}. \tag{7.63}$$

Using eqn. (7.61) we obtain the estimated error of density in one step such that

$$e[\rho] = \rho(t^n + k) - (\rho_1)^{n+1} = \frac{m^n\left((|\boldsymbol{J}|_1)^{n+1} - (|\boldsymbol{J}|_2)^{n+1}\right)}{(|\boldsymbol{J}|_1)^{n+1}\left(2\left(|\boldsymbol{J}|_1\right)^{n+1} + (|\boldsymbol{J}|_2)^{n+1}\right)}. \tag{7.64}$$

## 7.7 Semi-Discrete Spatial Error

The results in eqn. (7.44) and eqn. (7.56) can be applied to determine an error estimate in one step.

By expanding eqn. (7.44) and eqn. (7.56) using Taylor series, we obtain the truncation error for the spatial derivative in eqn. (7.31).

The leading term of the truncation error for eqn. (7.44) is

$$\begin{aligned} \tau(S_a) &= h^2\left(\frac{y_{\xi\xi\xi}p_\eta}{6} + \frac{y_\xi p_{\xi\xi\eta}}{8} - \frac{y_\eta p_{\xi\xi\xi}}{24} + \frac{y_{\xi\xi}p_{\xi\eta}}{4}\right) \\ &- l^2\left(\frac{y_{\eta\eta\eta}p_\xi}{6} + \frac{y_\eta p_{\xi\eta\eta}}{8} - \frac{y_\xi p_{\eta\eta\eta}}{24} + \frac{y_{\eta\eta}p_{\xi\eta}}{4}\right) + \cdots \end{aligned} \tag{7.65}$$

Similarly, for eqn. (7.56), the leading term of the truncation error is

$$\begin{aligned} \tau(S_b) &= h^2\left(3\frac{y_{\xi\xi\xi}p_\eta}{2} + 9\frac{y_\xi p_{\xi\xi\eta}}{8} - 7\frac{y_\eta p_{\xi\xi\xi}}{8} + 9\frac{y_{\xi\xi}p_{\xi\eta}}{4} - \frac{y_{\xi\xi\eta}p_\xi}{2} - y_{\xi\eta}p_{\xi\xi}\right) \\ &- l^2\left(3\frac{y_{\eta\eta\eta}p_\xi}{2} + 9\frac{y_\eta p_{\xi\eta\eta}}{8} - 7\frac{y_\xi p_{\eta\eta\eta}}{8} + 9\frac{y_{\eta\eta}p_{\xi\eta}}{4} - \frac{y_{\xi\eta\eta}p_\eta}{2} - y_{\xi\eta}p_{\eta\eta}\right) + \cdots \end{aligned} \tag{7.66}$$

As has previously been seen we have simplified the leading term to a single derivative, where an estimate to that derivative has been obtained. It is difficult to extend that method since we have several derivatives in the leading term of the truncation error.

For now we assume the mesh remains orthogonal such that we can disregard the mixed derivatives, resulting in the following truncation errors

$$\tau(S_a) = \frac{1}{24}\left(l^2 y_\xi p_{\eta\eta\eta} - h^2 y_\eta p_{\xi\xi\xi}\right) + \cdots \tag{7.67}$$

and for eqn. (7.56)

$$\tau(S_b) = \frac{7}{8}\left(l^2 y_\xi p_{\eta\eta\eta} - h^2 y_\eta p_{\xi\xi\xi}\right) + \cdots \tag{7.68}$$

Alternative methods to obtain the leading term of the truncation error more appropriately shall be discussed in Chapter 9.

### 7.7.1 Error Estimation

Using eqn. (7.67) and eqn. (7.68), we have the following leading terms of the truncation error in one step for $p$

$$p(\xi_i + h, \eta_j + l) - (p_a)_{(i+1,j+1)} = hl\tau_n(S_a) = \frac{1}{24}\left(hl^3 y_\xi p_{\eta\eta\eta} - lh^3 y_\eta p_{\xi\xi\xi}\right) + \cdots \tag{7.69}$$

$$p(\xi_i + h, \eta_j + l) - (p_b)_{(i+1,j+1)} = hl\tau_n(S_b) = \frac{7}{8}\left(hl^3 y_\xi p_{\eta\eta\eta} - lh^3 y_\eta p_{\xi\xi\xi}\right) + \cdots \tag{7.70}$$

where we will say $\Psi(\xi_i, \eta_j) = (hl^3 y_\xi p_{\eta\eta\eta} - lh^3 y_\eta p_{\xi\xi\xi})$.
By subtracting eqn. (7.70) from eqn. (7.69) we obtain

$$\begin{aligned} (p_b)_{(i+1,j+1)} - (p_a)_{(i+1,j+1)} &= -\frac{5}{6}\Psi(\xi_i, \eta_j) + \cdots \\ \Rightarrow \Psi(\xi_i, \eta_j) &= \frac{6\Big((p_a)_{(i+1,j+1)} - (p_b)_{(i+1,j+1)}\Big)}{5} + \cdots \end{aligned} \qquad (7.71)$$

Now substitute the computable term from eqn. (7.71) into eqn. (7.69), to obtain an estimated error in one spatial step for the pressure term, such that

$$e[p] = p(\xi_i + h, \eta_j + l) - (p_a)_{(i+1,j+1)} = \frac{\Big((p_a)_{(i+1,j+1)} - (p_b)_{(i+1,j+1)}\Big)}{20} + \cdots \qquad (7.72)$$

Following from this, we obtain an estimated error in one step for the spatial derivative of $p$ from eqn. (7.31), such that

$$e[\boldsymbol{C}\nabla p] = \frac{\Big((p_a)_{(i+1,j+1)} - (p_b)_{(i+1,j+1)}\Big)}{20hl} + \cdots \qquad (7.73)$$

We use eqn. (7.72) as a refinement criteria for the semi-discrete in space case, where eqn. (7.73) shall be applied to the fully-discrete case.

## 7.8 Fully-Discrete Error Estimate for Density

To estimate the error for the Jacobian in onestep we use eqn. (7.16). We have

$$\boldsymbol{x}^{n+1} = \boldsymbol{x}^n + k\boldsymbol{u}^n + \frac{k^2}{2}\boldsymbol{f}^{n+\frac{1}{2}}. \qquad (7.74)$$

The error in one step for $\boldsymbol{x}$ becomes

$$\boldsymbol{x}(t^n + k) - (\boldsymbol{x}_1)^{n+1} = e[\boldsymbol{x}] = k(e[\boldsymbol{u}]) + \frac{k^2 e([\boldsymbol{f}])}{2} \qquad (7.75)$$

where $|\boldsymbol{J}|$ follows, such that

$$|\boldsymbol{J}|(t^n + k) - (|\boldsymbol{J}|_1)^{n+1} = e[|\boldsymbol{J}|] = k(e[\boldsymbol{u}]) + \frac{k^2 e([\boldsymbol{f}])}{2} + \cdots \tag{7.76}$$

where $\boldsymbol{f} = -\frac{\boldsymbol{C}\nabla p}{m}$. Using eqn. (7.62) we have

$$e[\boldsymbol{u}] = e\left[\frac{\partial \boldsymbol{x}}{\partial t}\right] = e\left[\frac{d|\boldsymbol{J}|}{dt}\right] = \tau_n(L_1) = \frac{\left((|\boldsymbol{J}|_2)^{n+1} - (|\boldsymbol{J}|_1)^{n+1}\right)}{3k} + \mathcal{O}(k^3). \tag{7.77}$$

Using eqn. (7.73) we also have

$$e[\boldsymbol{f}] = e[\boldsymbol{C}\nabla p] = \frac{\left((p_a)_{(i+1,j+1)} - (p_b)_{(i+1,j+1)}\right)}{20hl} + \cdots. \tag{7.78}$$

By substituting eqn. (7.77) and eqn. (7.78) into eqn. (7.76) we obtain

$$e[|\boldsymbol{J}|] = \frac{\left((|\boldsymbol{J}|_2)^{n+1} - (|\boldsymbol{J}|_1)^{n+1}\right)}{3} - \frac{k^2\left((p_a)_{(i+1,j+1)} - (p_b)_{(i+1,j+1)}\right)}{40hlm} + \cdots \tag{7.79}$$

which is the fully-discrete error estimate in one time step for the Jacobian. This can now be applied to obtain an error estimate for density.

By substituting eqn. (7.79) into eqn. (7.63) we obtain

$$e[\rho] = \rho(t^n + k) - (\rho_1)^{n+1} = \tag{7.80}$$

$$-\frac{m^n\left(\frac{\left((|J|_2)^{n+1} - (|J|_1)^{n+1}\right)}{3} - \frac{k^2\left((p_a)_{(i+1,j+1)} - (p_b)_{(i+1,j+1)}\right)}{40hlm}\right)}{\left((|J|_1)^{n+1} + \frac{\left((|J|_2)^{n+1} - (|J|_1)^{n+1}\right)}{3} - \frac{k^2\left((p_a)_{(i+1,j+1)} - (p_b)_{(i+1,j+1)}\right)}{40hlm}\right)(|J|_1)^{n+1}} + \cdots \tag{7.81}$$

which can now be used as a refinement criteria for 2D problems.

## 7.9 Chapter 7 Summary

In this chapter we have extended all results, from previous chapters, to the 2D Euler Equations. We have obtained two temporal schemes and two spatial schemes.

The truncation errors for the Jacobian and momentum equations were obtained. These were applied in Milne's approach in order to estimate the error. The estimated errors for semi-discrete in time density, semi-discrete in space pressure and a fully-discrete density were derived.

In the following chapter, we utilise these error estimates as refinement criterion and implement refinement for a range of 2D problems.

# Chapter 8

# Refinement

## 8.1 Results Overview

In Chapter 6 we used the Sod shock tube problem, which has an analytic solution, to determine how well the approach developed acts as an error estimate. The features we wished to capture were the shock front, the top and bottom of the fan and the contact. The method proved to capture all these features. Although it was shown to capture the top and bottom of the fan and the contact with a lower magnitude than the true error, the approach captured the shock position and value extremely well for all cases. We demonstrated how different variables captured different features, i.e. the velocity captured the entire rarefaction the best. From these results it would seem beneficial to choose the error estimate as a refinement criterion, since it is capturing all the features we wish to resolve.

The refinement is implemented by flagging elements using the error estimate and these elements are divided isotropically. A full description is given in [64].

We apply the 2D approach to the 1D problem, Sod's shock tube, which is implemented in a 2D manner. From this we can judge how well the approach works as a refinement criteria.

The approach can then be applied to more genuine 2D problems such as the Square Sod problem (see [64]) and the 2D Riemann problem (see [50]) to determine how well the approach works for detecting flow features in 2D.

We implement the method into a full ALE method for the 2D Riemann problem, Radial Sod problem (see [90]) and Sedov Blast wave (see [79]), since these problems exhibit mesh tangling when run in a purely Lagrangian manner, see [64]. We also wish to demonstrate that the approach can be applied to the ALE method.

## 8.2 Lagrangian Results

We run the Sod shock tube problem, Square Sod problem and 2D Riemann problem in a purely Lagrangian manner, and implement the error estimates as a refinement criteria.

### 8.2.1 Sod Shock Tube

We have already described the Sod shock tube problem, from [83], in Section 6.1. This problem is used to test and compare the method developed in this project since an analytic solution is available. We now look at running the Sod shock tube problem (which is a 1D problem) with a 2D code, and implement refinement of elements. For this problem we will also investigate how well the error estimation behaves as a refinement criteria against existing results using density gradients as a refinement criteria, which was implemented in [64].

The Sod shock tube problem is run to a final time of 0.2 for the region $(0, 1) \times (0, 1)$ with $100 \times 10$ mesh points. From Figures 8.1, 8.2 and 8.3 we see how the uniformly fine mesh shows expansion from $x = 0.2$, and then compression at $x = 0.69$ and the shock at $x = 0.8$. These are the regions we wish to resolve using the refinement criteria. We compare the Scheme 1 and Scheme 2 solutions in Figures 8.2 and 8.3, as we wish to demonstrate that both schemes behave well in order to use for the error estimate. From Figures 8.2 and 8.3, the results are extremely comparable. One visible difference between the schemes is in the specific internal energy plot, Scheme 1 overshoots the analytic solution while Scheme 2 undershoots the analytic solution. Since the analytic solution is available for this problem we can investigate the error norms of both schemes, see Table 8.1. Although there are differences between the schemes the error norms are comparable. The largest difference is in the $\infty$-norm. This is a good feature since the $\infty$-norm demonstrates the error norm at the shock. If the largest difference is with the $\infty$-norm then the estimated error will be largest around the shock, as we desire. These results demonstrate that both schemes are suitable to be used to determine an error estimate which will trigger refinement. We now investigate how well the the error estimate performs as a refinement criterion.

The profiles in Figures 8.4 and 8.5, which demonstrate the results from using the density gradient and the semi-discrete in time error estimate as a refinement criteria respectively, provide excellent agreement to the uniformly fine solution. However from this alone it is not clear how well each refinement criterion performs in relation to the other. Figure 8.7 shows that using the semi-discrete in time error estimate as a refinement criteria captures the top and bottom of the fan and the shock extremely well. The results for the density gradient in Figure 8.6 also demonstrates that these regions are well resolved by the refinement criteria. However, we see in Figure 8.6 the entire expansion is refined due to its varying gradient, whereas Figure 8.7 only detects the top and bottom of the fan. Such refinement is wasteful since the solution is smooth along the expansion, and we do not require refinement here. However on comparison of the number of elements present over the evolution of the problem in Figure 8.8, the density gradient criteria has fewer elements than the semi-discrete in time error estimate criterion.

The fine grid and both AMR grids required 322 timesteps. For the fine grid, a total of $322,000$ elements were required for the entire evolution. For the estimated error criterion, only 40% of this amount is required, and for the density gradient criterion, 37% is required. The additional elements are coming from the shock when the semi-discrete in time error estimate is used, which is evident in Figure 8.7 where there is more refinement around the shock than in Figure 8.6. The runtime is reduced by 40% for the error AMR and 77% less for the density gradient AMR, both a significant saving, however the error estimate is taking 5.13 seconds longer than the density gradient AMR.

Although the density gradient performs better overall, the approach devised in this thesis still performs significantly better than the uniformly fine mesh. It is promising that the error estimate is not wasting refinement on smooth features i.e. the rarefaction, as the density gradient is doing. Although the estimated error seems to be over refining around the shock, this is not necesseraly a bad feature. When shocks pass through refinement boundaries, oscillations often arise, (see [64]). The over refinement could prove to be beneficial in reducing such oscillations, hence improving the accuracy of the solution.

Since an analytic solution is available here we are able to determine the error norms to compare the fine calculation against the calculations using AMR, see Table 8.2. From the error norms we see that the solutions of the fine calculation and AMR using the error estimate devised in this work are more comparable than the solution obtained using the density gradient as an AMR trigger. This is an extremely promising result.

Having demonstrated how the estimated error compares to the density gradient refinement criteria, we look at the estimated error as a refinement criteria for genuine 2D problems, and investigate how well the approach works against the uniformly fine grid. Additionally to the semi-discrete in time error estimate for pressure, for the 2D problems we have a semi-discrete in space error estimate for density and a fully-discrete error estimate. We begin the 2D investigation with a Square Sod problem.

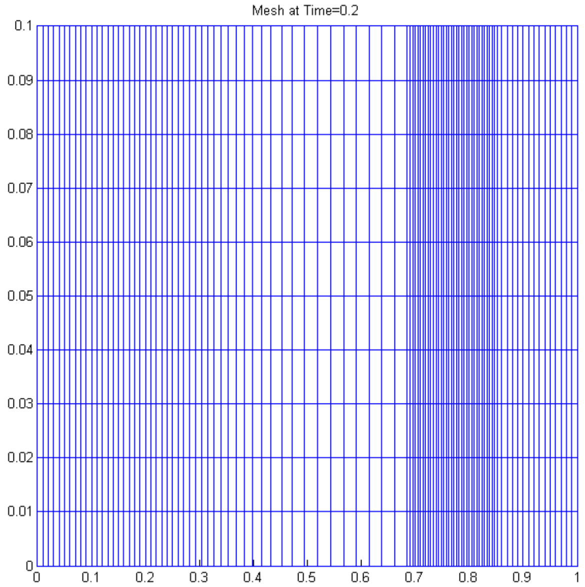


Figure 8.1: Sod's Shock Tube Problem Lagrangian Fine Mesh at $T = 0.2$.

Table 8.1: Error Norms Comparing Scheme 1 and Scheme 2 for the Sod Shock Tube.

| Calculation | $\|\|e\|\|_1$ | $\|\|e\|\|_2$ | $\|\|e\|\|_\infty$ |
|---|---|---|---|
| Scheme 1 | 0.00524 | 0.00907 | 0.04462 |
| Scheme 2 | 0.00587 | 0.00995 | 0.0618 |

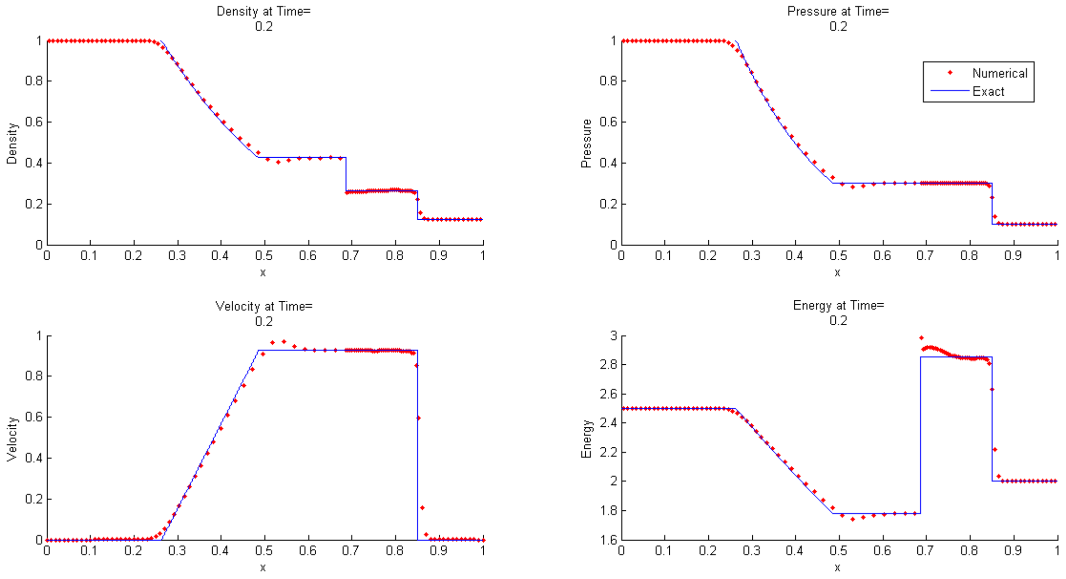


Figure 8.2: Sod's Shock Tube Problem at $T = 0.2$. Density, Pressure, Velocity and Energy. Scheme 1 on a Fine Mesh Compared to Analytic Solution.

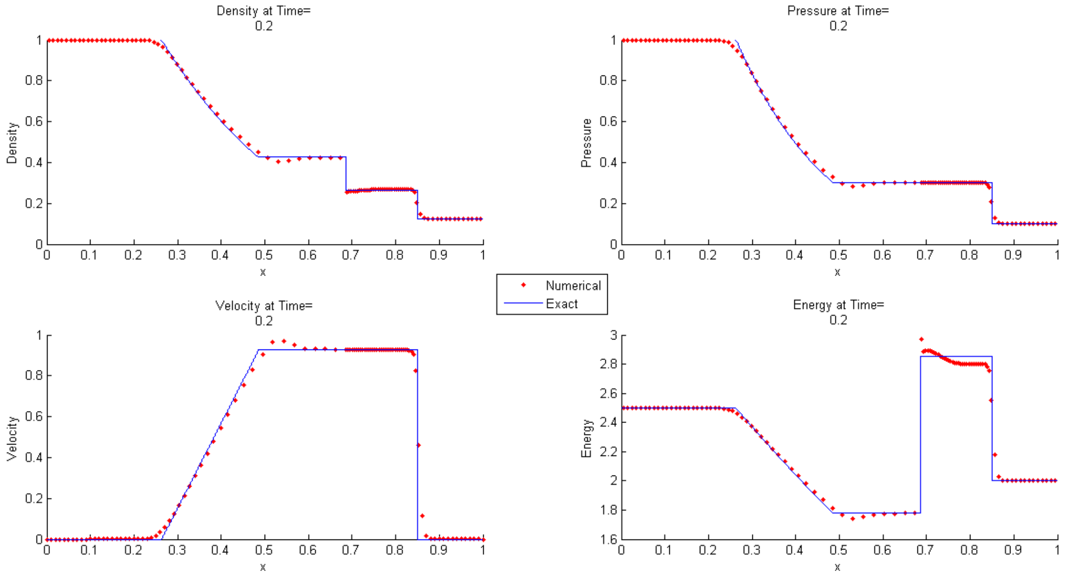


Figure 8.3: Sod's Shock Tube Problem at $T = 0.2$. Density, Pressure, Velocity and Energy. Scheme 2 on a Fine Mesh Compared to Analytic Solution.

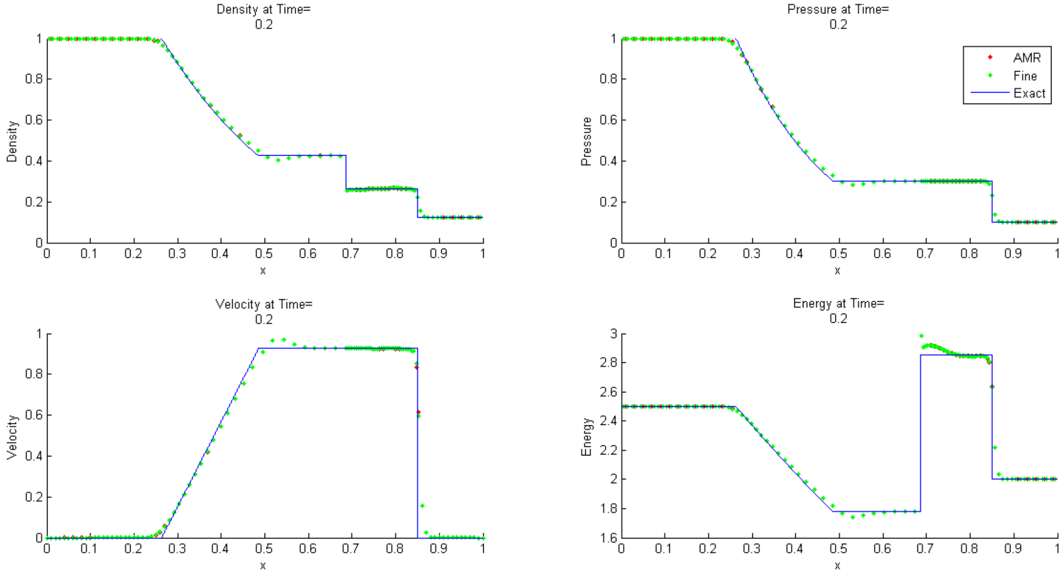


Figure 8.4: Sod's Shock Tube Problem Comparing Fine, Analytic and AMR using Density Gradient at $T = 0.2$.

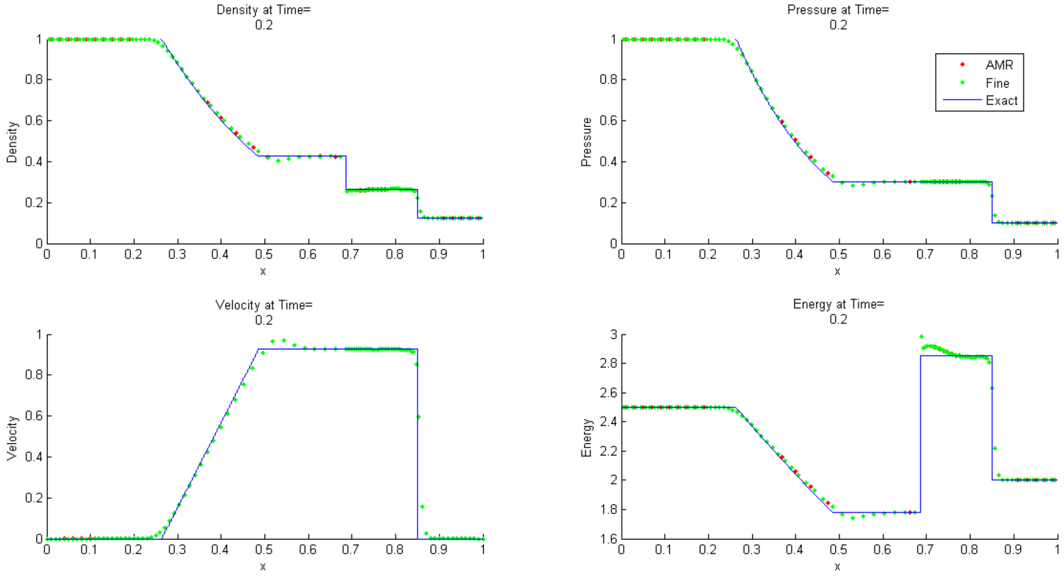


Figure 8.5: Sod's Shock Tube Problem Comparing Fine, Analytic and AMR Results using a Semi-Discrete in Time Error Estimate at $T = 0.2$.

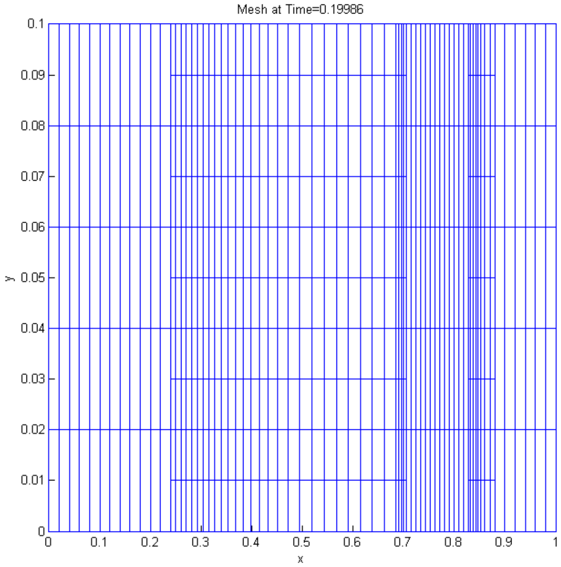


Figure 8.6: Sod's Shock Tube Problem AMR using Density Gradients at $T = 0.2$.

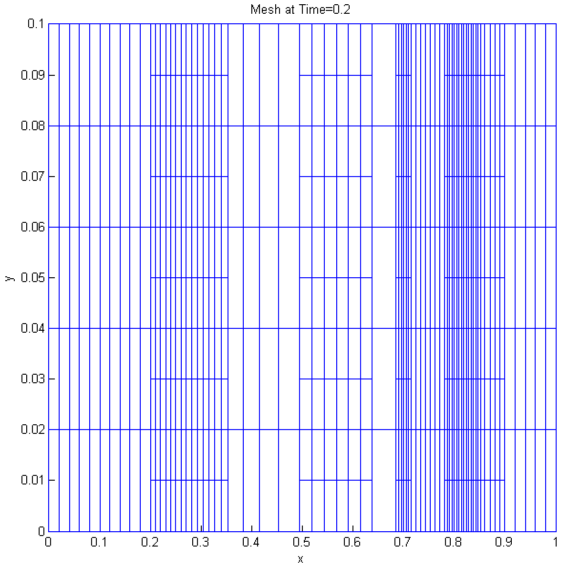


Figure 8.7: Sod's Shock Tube Problem AMR using a Semi-Discrete in Time Error Estimate for Density at $T = 0.2$.

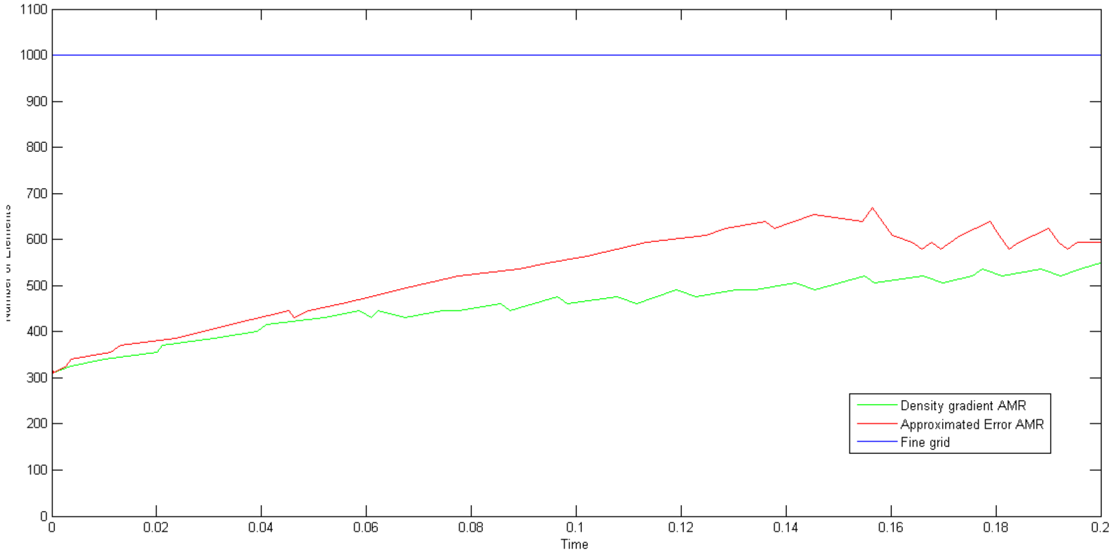


Figure 8.8: Sod's Shock Tube Problem Comparing Number of Elements.

Table 8.2: Error Norms Comparing Fine Calculation against AMR.

| Calculation | $\|\|e\|\|_1$ | $\|\|e\|\|_2$ | $\|\|e\|\|_\infty$ |
|---|---|---|---|
| Fine | 0.00524 | 0.00907 | 0.04462 |
| Density Gradient AMR | 0.00078 | 0.00316 | 0.03722 |
| Semi-Discrete in Time AMR | 0.00146 | 0.00456 | 0.03736 |

### 8.2.2 Square Sod Problem

The Square Sod problem consists of a square domain, here a $(0,1) \times (0,1)$ region, with a smaller square inside on the domain, $(0.3, 0.7) \times (0.3, 0.7)$. The initial conditions within the smaller square are $\rho = 1$, $p = 1$ and $\mathbf{u} = 0$. Outside this region the initial conditions are $\rho = 0.125$, $p = 0.1$ and $\mathbf{u} = 0$. This is illustrated in Figure 8.9.

A fine calculation was run on a $100 \times 100$ mesh to a final time of $t = 0.1$, as is illustrated in Figures 8.10 and 8.11. The artificial viscosity coefficients are taken as $c_l = 0.3$ and $c_q = 0.65$ as in [64].

The problem behaves like a Sod shock tube along the centre of each side, with Radial Sod

behaviour at the corners, (see [64]).
We compare the fine grid results against the AMR results, where the initial mesh is taken as $50 \times 50$ when refinement is implemented.

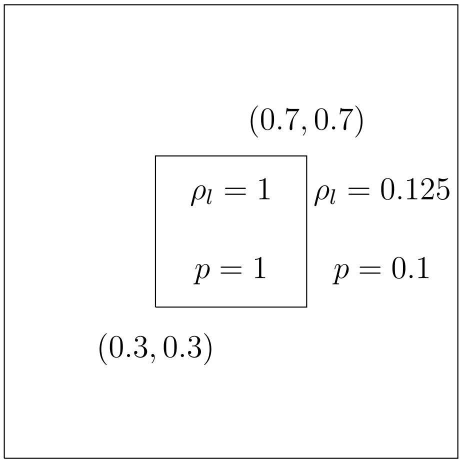


Figure 8.9: Square Sod Initial Data

When using the density gradient as a refinement criteria the entire expansion is refined, see Figure 8.12, which is not necessarily required. The semi-discrete in time error estimate for density detects all areas of interest very well without too much over refinement. The refinement is symmetric and has occurred around the shock and the contact, the corners after the contact are also refined. There is possible over refinement around the shock, since there are no unrefined elements between the shock and the contact. This will add to the overall runtime. The top of the expansion is not detected here, see Figures 8.13 and 8.16, since it is smooth, see Figure 8.10. As we have discussed in Chapter 6, the sharp features are temporal errors, since we use a temporal error estimate here the smooth features are not detected as well. The semi-discrete in space error estimate for pressure also detects the features of interest well. The shock and contact are refined extremely well, without refinement between these two features at the constant state. The top of the expansion is also detected, and the refinement is symmetric, see Figures 8.14 and 8.17. From Figure 8.14 the error estimate is attempting to refine the top of the expansion, which is farely steep, see Figure 8.10. These results are in agreement with the 1D results from Chapter 6, where it was concluded that the shock is a temporal error and the expansion is a spatial error, since we are using a spatial error estimate the smooth features will also be detected. The fully-discrete error estimate in Figure 8.15, detects and refines the shock and contact extremely well. However, further work needs to be carried out on this method since the refinement is not symmetric and there are random areas between the shock and contact which are being refined. Although this method does not perform as well as the others, it is still promising that it refines the shock and contact, with an area of unrefined elements between these two features. The semi-discrete in time error estimate AMR and semi-discrete in space pressure estimate AMR both demonstrate extremely good results as can be seen in the contour plots from Figures 8.21 and 8.22, both are comparable to the uniformly fine results in Figure 8.19. We see the density gradient AMR is also equally comparable to the uniformly fine mesh as illustrated in Figure 8.20. However, the fully-discrete contour plot in Figure 8.23 is least comparable to the fine calculation, particularly in the region between the shock and contact where we saw the patchy refinement in Figure 8.15. The overall run time is reduced by 78% for the semi-discrete in time error estimate for density, whereas the semi-discrete in space error estimate for pressure is only reduced by 42% and the fully-discrete is reduced by 73% compared to the uniformly fine mesh. When the density gradient was applied as a refinement criterion the runtime was 86% less than the fine calculation, see Table 8.3. It is expected that the run time be greater in the estimated error approach since two schemes are implemented in order to obtain the estimated error.

The semi-discrete in time criterion uses 98% fewer elements, the semi-discrete in space uses 73% less and the fully-discrete uses 89% less whereas the density gradient method uses 95% fewer elements than the uniformly fine grid, over the entire evolution as illustrated in Figure 8.24. We see a lot of oscillatory behaviour in Figure 8.24 for the semi-discrete in space and fully-discrete AMR. The semi-discrete in space adds a lot of refinement at the beginning of the problem and then starts to settle, whereas the fully-discrete AMR refines and then derefines most of the refined elements throughout the evolution. This demonstratess the difficulties associated with choosing a beneficial refinement tolerance. However, considering the run times, the semi-discrete in time and fully-discrete error estimations are comparable to the density gradient results, when considering both time and number of elements.

We have seen that the fully-discrete error estimate requires further work, since the refinement is not symmetric, however the shock and contact are captured and refined well, with a 73% saving in the run time compared to the uniformly fine solution, therefore the results are promising. We illustrated in Figure 8.13 and Figure 8.22 how the steep features are detected with temporal errors and the smooth features are detected with spatial errors. When the semi-discrete in time and semi-discrete in space error estimates were implemented for AMR the results are extremely comparable to the uniformly fine solution, with a saving in runtime of 78% and 42% respectively. These results are also comparable to the density gradient AMR. The semi-discrete in time and semi-discrete in space error estimates both used fewer elements over the entire evolution compared to the density gradient AMR. All contour plots in Figures 8.20, 8.21 and 8.22 are extremely comparable, therefore, when considering the balance between run time and number of elements the semi-discrete in time AMR and density gradient AMR produce equally good results.

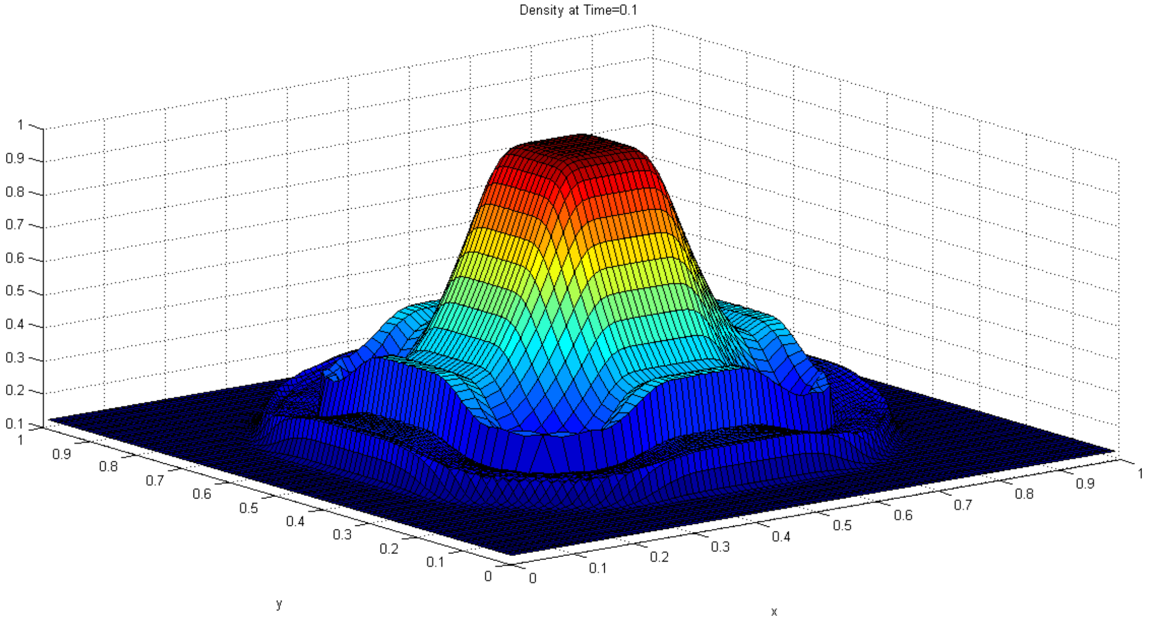


Figure 8.10: Square Sod Problem Density Surface Plot on Fine Mesh at $T = 0.1$.

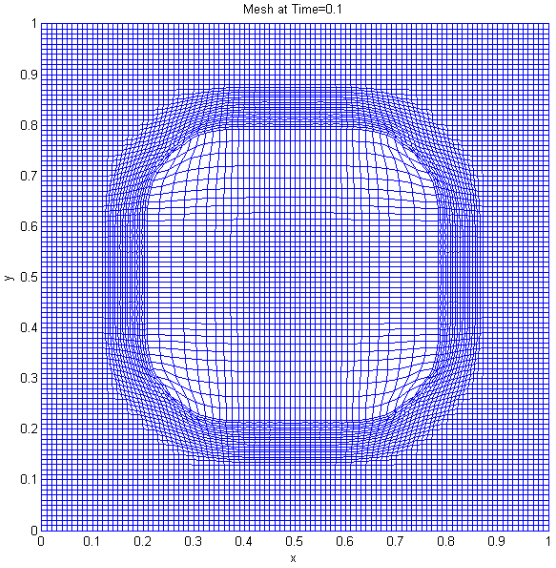


Figure 8.11: Square Sod Problem Lagrangian Fine Mesh at $T = 0.1$

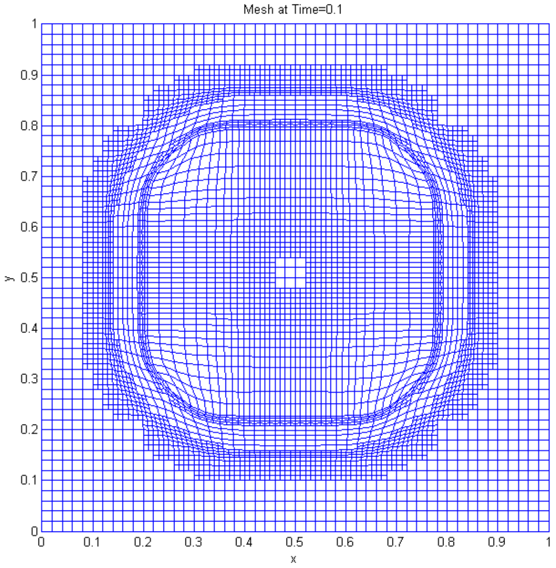


Figure 8.12: Square Sod Problem AMR with Density Gradient Mesh at $T = 0.1$.

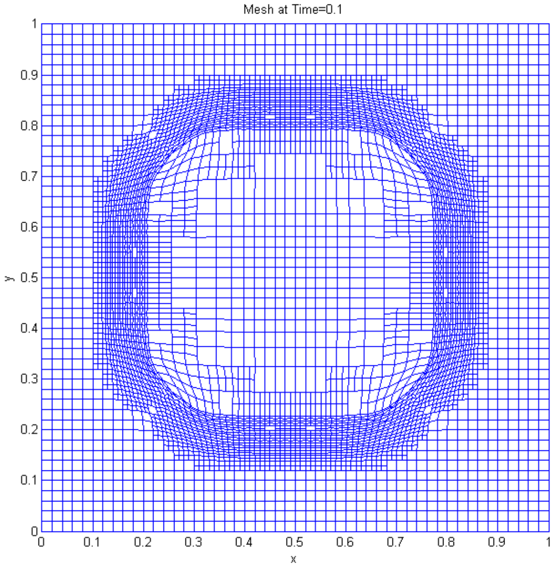


Figure 8.13: Square Sod Problem Semi-Discrete in Time Estimated Error for Density, AMR Mesh at $T = 0.1$.

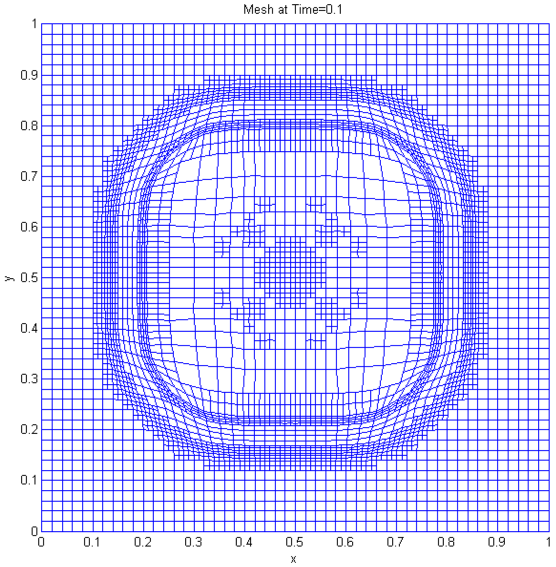


Figure 8.14: Square Sod Problem Semi-Discrete in Space Estimated Error for Pressure, AMR Mesh at $T = 0.1$.

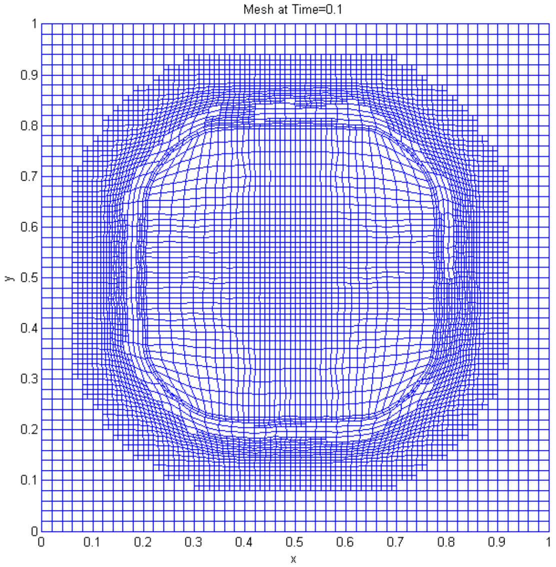


Figure 8.15: Square Sod Problem Fully-Discrete Estimated Error for Density, AMR Mesh at $T = 0.1$.

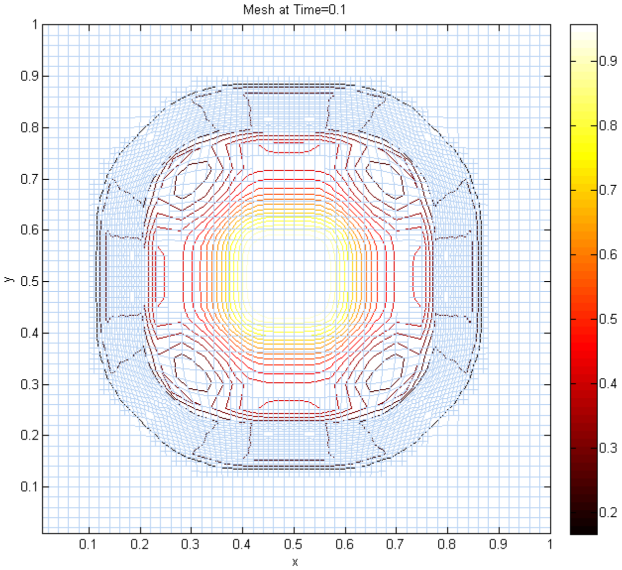


Figure 8.16: Square Sod Problem Semi-Discrete in Time Estimated Error for Density AMR, Mesh and Contour Plot of Density at $T = 0.1$.

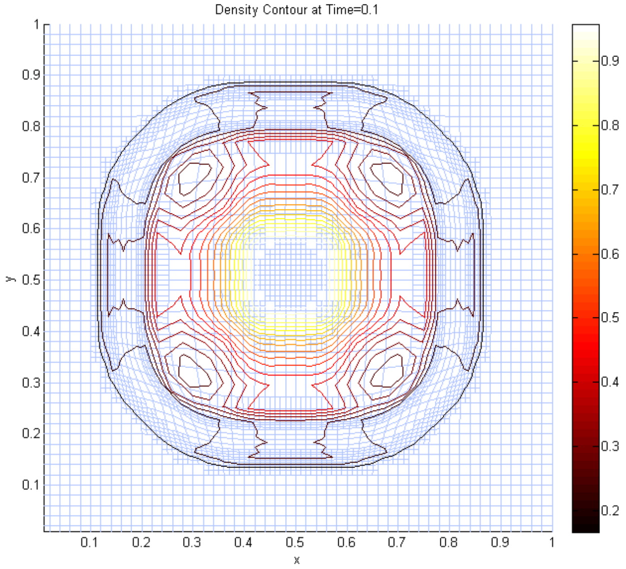


Figure 8.17: Square Sod Problem Semi-Discrete in Space Estimated Error for Pressure AMR, Mesh and Contour Plot of Density at $T = 0.1$.

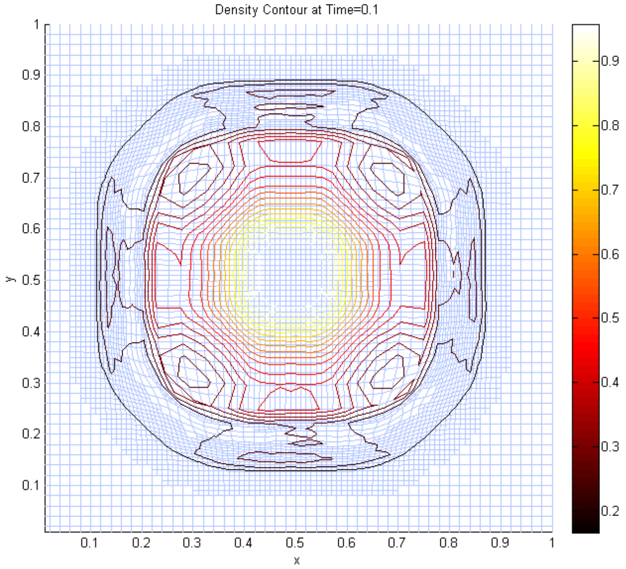


Figure 8.18: Square Sod Problem Fully-Discrete Error for Density AMR, Mesh and Contour Plot of Density at $T = 0.1$.

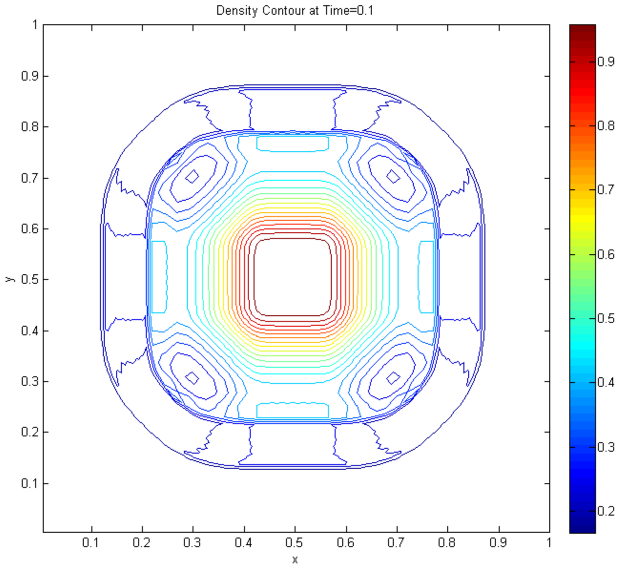


Figure 8.19: Square Sod Problem Density Contour Plot at $T = 0.1$.

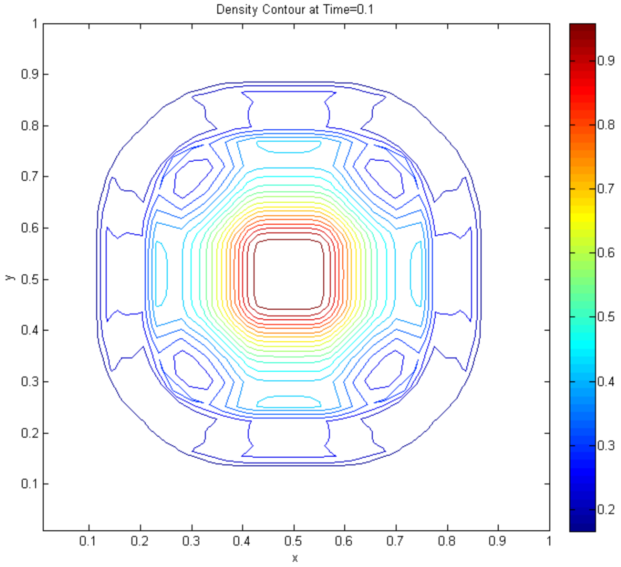


Figure 8.20: Square Sod Problem Density Contour Density Gradient AMR Plot at $T = 0.1$.

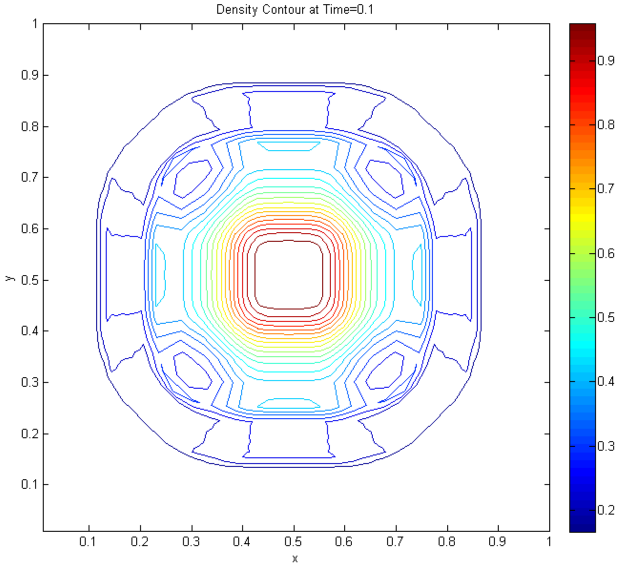


Figure 8.21: Square Sod Problem Density Contour using Semi-Discrete in Time Estimated Error for Density AMR Plot at $T = 0.1$.

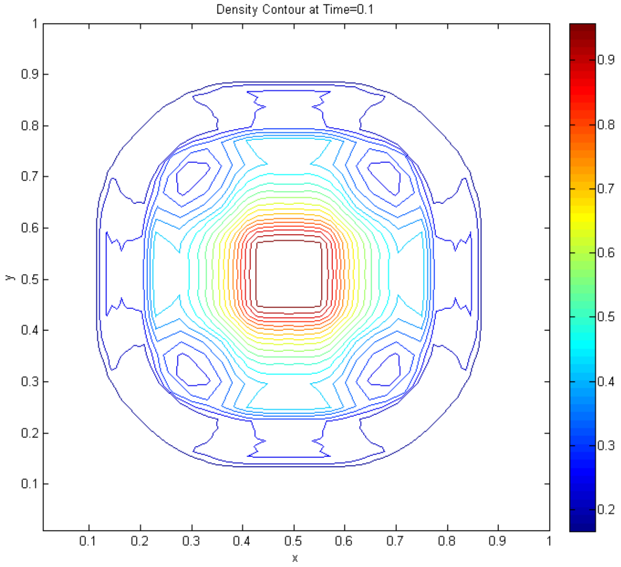


Figure 8.22: Square Sod Problem Semi-Discrete in Space Estimated Error for Pressure for AMR, Density Contour Plot at $T = 0.1$.

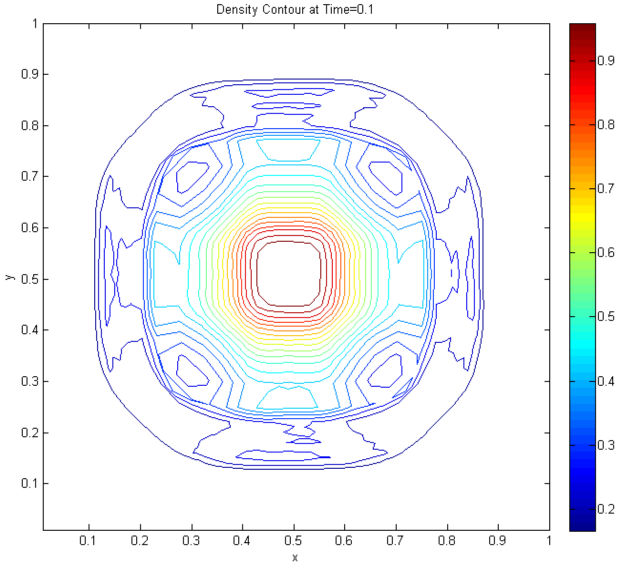


Figure 8.23: Square Sod Problem Fully-Discrete Estimated Error for Density, AMR Mesh at $T = 0.1$.

Table 8.3: Run Times and Total Elements for Square Sod Problem.

| | Fine Mesh | Density Gradient AMR | Semi-Discrete in Time AMR | Semi-Discrete in Space AMR | Fully-Discrete AMR |
|---|---|---|---|---|---|
| Run Times | 3118.600 | 432.279 | 685.900 | 1817.85 | 811.205 |
| Total Elements | 10,310,000 | 477,184 | 216,092 | 2,777,147 | 1,111,404 |

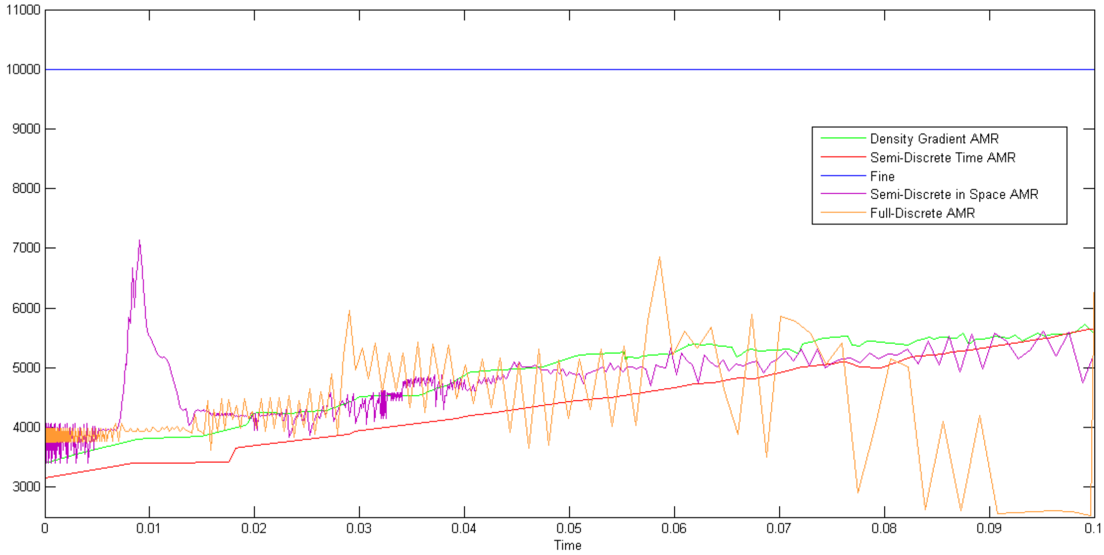


Figure 8.24: Square Sod Problem Number of Elements over Evolution of Problem.

### 8.2.3 Two-dimensional Riemann Problem

The 2D Riemann problem consists of a square domain split into four square regions. The quarters are labeled as $1, 2, 3$ and 4 where the respective regions are defined as $x > 0.5$ and $y > 0.5$, $x < 0.5$ and $y > 0.5$, $x < 0.5$ and $y < 0.5$ finally $x > 0.5$ and $y < 0.5$. The problem consists of four shocks, as the shocks interact an oval region of high density is formed. The initial data from [50] for the regions are as follows

$$\begin{array}{llll} p_2 = 0.3500 & \rho_2 = 0.5065 & p_1 = 1.1000 & \rho_1 = 1.1000 \\ u_2 = 0.8939 & v_2 = 0.0000 & u_1 = 1.0000 & v_1 = 0.0000 \\ \\ p_3 = 1.1000 & \rho_3 = 1.1000 & p_4 = 0.3500 & \rho_4 = 0.5065 \\ u_3 = 0.8939 & v_3 = 0.8939 & u_4 = 0.0000 & v_4 = 0.8939. \end{array}$$

The problem is run to $T = 0.2$ on a region $(0, 1) \times (0, 1)$ for a uniformly fine mesh of $100 \times 100$ and when refinement is implemented the initial mesh is $50 \times 50$.
The artificial viscosity coefficients are $c_l = 0.3$ and $c_q = 0.65$ as defined in [64].
We can see from Figure 8.26 that the mesh starts to become tangled within the oval, this

occurs since we are using a Lagrangian method. From Figure 8.29, we see how the tangling of the mesh imprints on the solution, where there is a lot of noise within the oval on a region that is constant (see Figure 8.25). The tangling on the mesh will affect the areas detected for refinement.

From Figure 8.27, where we have used the density gradient for AMR, we see how the oval and the shock is captured, where the distorted mesh at the centre is not refined. For the semi-discrete in time error estimate for density AMR in Figure 8.28 we see that the oval and shocks are captured again, however we now have refinement at the centre of the oval where the distortion occurs. The reason for this detection is likely to be due to the refinement tolerances we have chosen in each case.

From the contour plots on the uniformly fine mesh in Figure 8.29 we can see the shocks at the top left and bottom right of the figure. The distortion of the mesh has imprinted on the solution within the oval, where we can see some noise. Away from the oval we see contours at the boundaries which are left over from the rarefaction. From Figure 8.30, we see the density gradient AMR has a lot more noise within the oval compared to the uniformly fine mesh, and the contours around the shocks and oval are not as close together. The features at the boundaries are only slightly detected for the density gradient AMR. For the semi-discrete in time error estimate for density we see in Figure 8.31 that the contours are in better agreement to the uniformly fine contours. The contours around the shocks and contact are closer to each other than for the density gradient AMR. A promising result from this method is that the noise within the oval is reduced here compared to the uniformly fine mesh. Additionally, the less significant features at the boundaries are also comparable to the uniformly fine mesh. We have a saving of 92% for the elements over the entire evolution for the density gradient AMR, whereas we use 86% fewer elements compared to the uniformly fine mesh for the error estimate AMR. Although the error estimate AMR uses slightly more elements than the density gradient, it is still a significant saving compared to the uniformly fine mesh, where $20,760,00$ elements are used over the entire evolution, see Figure 8.32.

The run time for the semi-discrete in time error estimate for density is 76% less than the uniformly fine grid however, the density gradient is 94% less than the uniformly fine mesh. Although the run time is the smallest for the density gradient AMR , stated in Table 8.4, it is the semi-discrete in time error estimae for density that gives the better results from Figure 8.31, even improving on the uniformly fine grid at the centre of the oval.

Since we have an overall saving of 76% in run time for the semi-discrete in time error estimate for density AMR, and a saving of 20754686 elements for a better result, we conclude the error estimate AMR should be used for this problem.

We have not considered the semi-discrete in space and fully-discrete error estimates here, since these approaches include the spatial derivative estimation the tangling of the mesh has an impact on the error estimate, therefore affecting the run of the problem. However, it is demonstrated in [64] that an AMR approach can be combined with an ALE method to limit the tangling of the mesh. We wish to demonstrate that the current approach can also be incorporated into the ALE method, by testing on the 2D Riemann problem, Radial Sod problem and the Sedov blast wave problem.

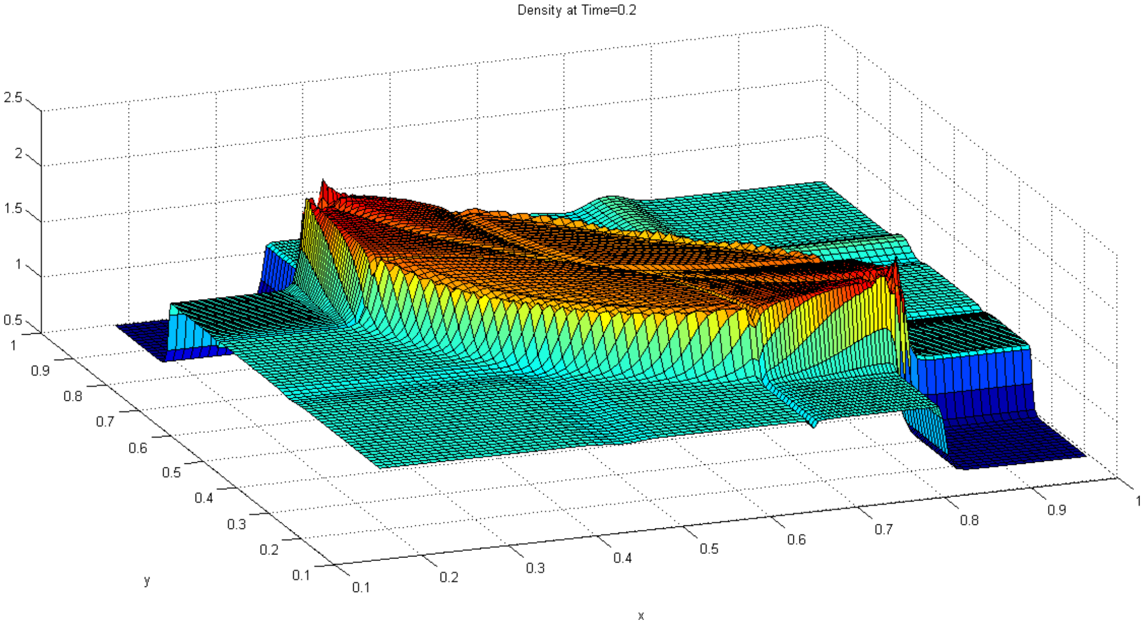


Figure 8.25: 2D Riemann Problem, Surface Plot of density on Fine Mesh at $T = 0.2$.

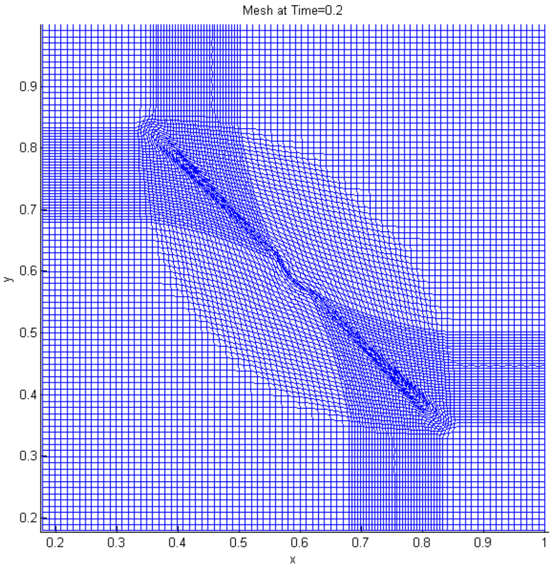


Figure 8.26: 2D Riemann Problem Uniformly Fine Mesh at $T = 0.2$.

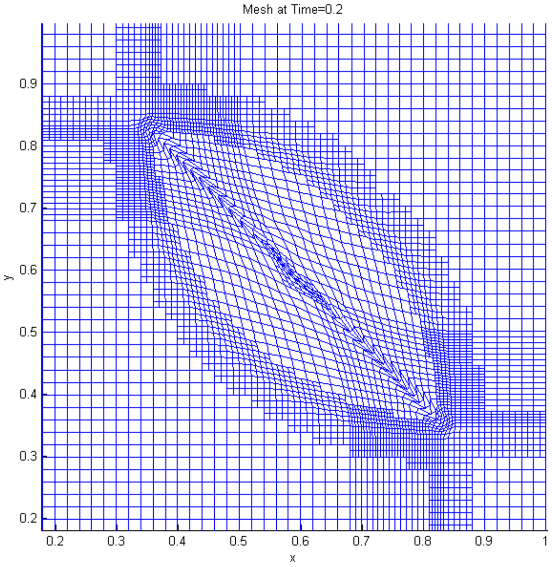


Figure 8.27: 2D Riemann Problem, AMR with Density Gradient at $T = 0.2$.

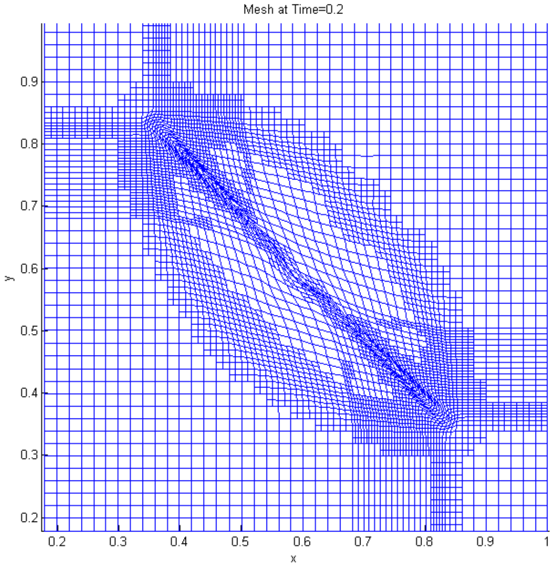


Figure 8.28: 2D Riemann Problem, AMR with Semi-discrete in Time Density Error Estimate at $T = 0.2$.

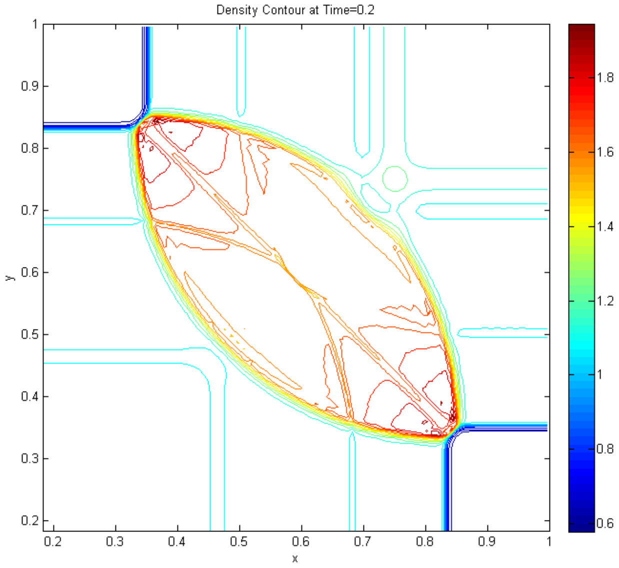


Figure 8.29: 2D Riemann Problem, Contour Plot of Density on Fine Mesh at $T = 0.2$.

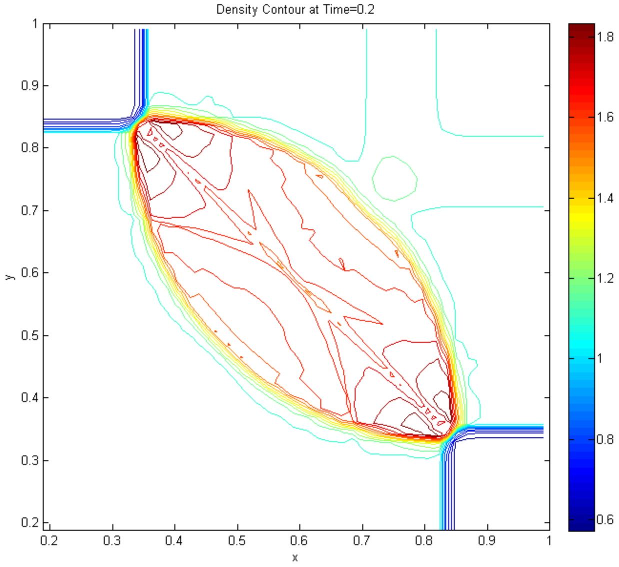


Figure 8.30: 2D Riemann Problem, Contour Plot of Semi-discrete in Time Density with AMR using Density Gradient at $T = 0.2$.

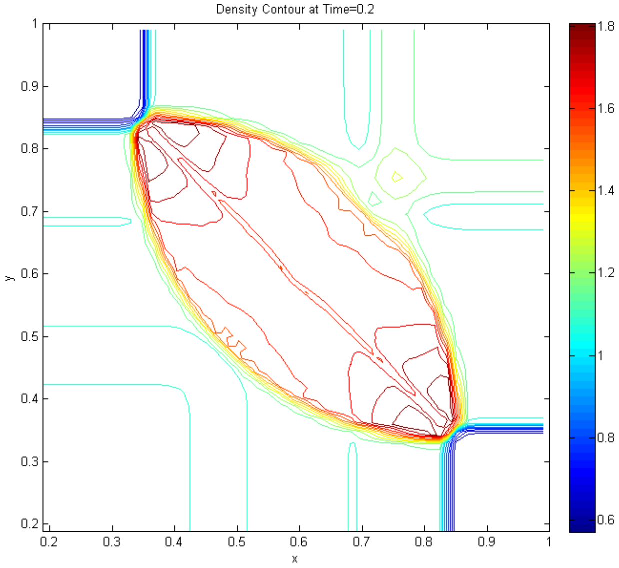


Figure 8.31: 2D Riemann Problem, Contour Plot of Density using AMR with Error Estimate Fine Mesh at $T = 0.2$.

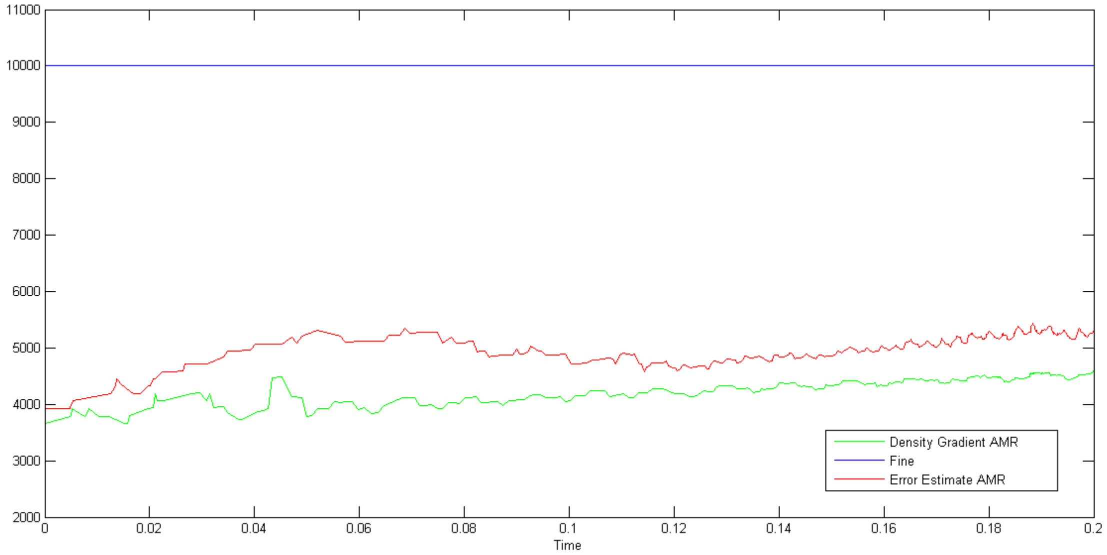


Figure 8.32: 2D Riemann problem total number of elements.

Table 8.4: Run Times and Total Elements for 2D Riemann Problem.

| | Fine Mesh | Density Gradient AMR | Semi-Discrete in Time AMR |
|---|---|---|---|
| Run Times | 11463.3 | 727.994 | 2759.42 |
| Total Elements | 20,760,000 | 1,643,063 | 2,859,724 |

## 8.3 ALE Results

As we have seen in Section 8.2.3, the Lagrangian method can cause the mesh to tangle which in turn imprints on the solution. To overcome this we use an ALE method as described in [64], which uses a relaxation method to prevent the mesh from tangling as much.

We begin by repeating the 2D Riemann problem as this is where we observed the tangling. We then extend this method to the Radial Sod problem and Sedov blast wave problem, since these test problems are known to become tangled with a Lagrangian mesh as mentioned in [64].

### 8.3.1 2D Riemann Problem

We apply the initial data from Section 8.2.3 to the 2D Riemann Problem using an ALE method.

We see from the uniformly fine mesh in Figure 8.34 how the ALE method prevents the mesh from tangling.

From Figure 8.35 we see the density gradient AMR refines around the the oval of high density and shocks. Similarly in Figure 8.36, the semi-discrete in time estimated error for density refines the oval and shocks however, there is extra refinement in the corners of the oval. This refinement is due to the peak in density, see Figure 8.33. The semi-discrete in space error estimate for pressure refines areas which are not significant, see Figure 8.37. The features which are detected and being refined in this case are the contours around the boundaries, along with the contours within the oval, as can be seen in the uniformly fine mesh contour plot, Figure 8.38. The refinement of insignificant features here is due to the refinement tolerance. To loose the additional refinement we would need to increase the refinement tolerance however, by doing so we loose the refinement around the significant features. For the uniformly fine mesh using the ALE method in Figure 8.38, we still see noise at the centre of the oval however the noise does not extend the entire length of the oval, as in Figure 8.29. All contour plots in Figures 8.39-8.41 are in good agreement with the uniformly fine solution in Figure 8.38, where the semi-discrete in space results being the least similar since additional insignificant features at the boundaries are illustrated in Figure 8.41.

The run time for the semi-discrete in space AMR is 4% greater for the uniformly fine results, this is because there is a surge of refinement at the begining of the problem, see Figure 8.42. The total number of elements remains high throughout the evolution. As previously mentioned, if the refinement criteria was increased in order to introduce less refinement around insignificant features, it would also remove refinement around the features we are interested in. The density contours for this problem are the least comparible to the uniformly fine. Therefore, this method is the least suited for this problem.

The run time for the density gradient AMR is 74% less than the uniformly fine mesh whereas, the run time for the semi-discrete in time error estimate for density is 63% less, these methods give extremely similar results. However, the density gradient AMR uses 643680 elements and the error estimate AMR uses 565503 over the entire evolution, where the number of elements in a timestep are illustrated in Figure 8.42. The density gradient AMR and error estimate AMR are both suited to this problem, since they demonstrate similar accuracy in the solution, have similar run times and number of total elements.

The fully-discrete error estimate gave nonsensical results for this problem. Further work needs to be carried out on the fully-discrete error estimate approach, when used with ALE.

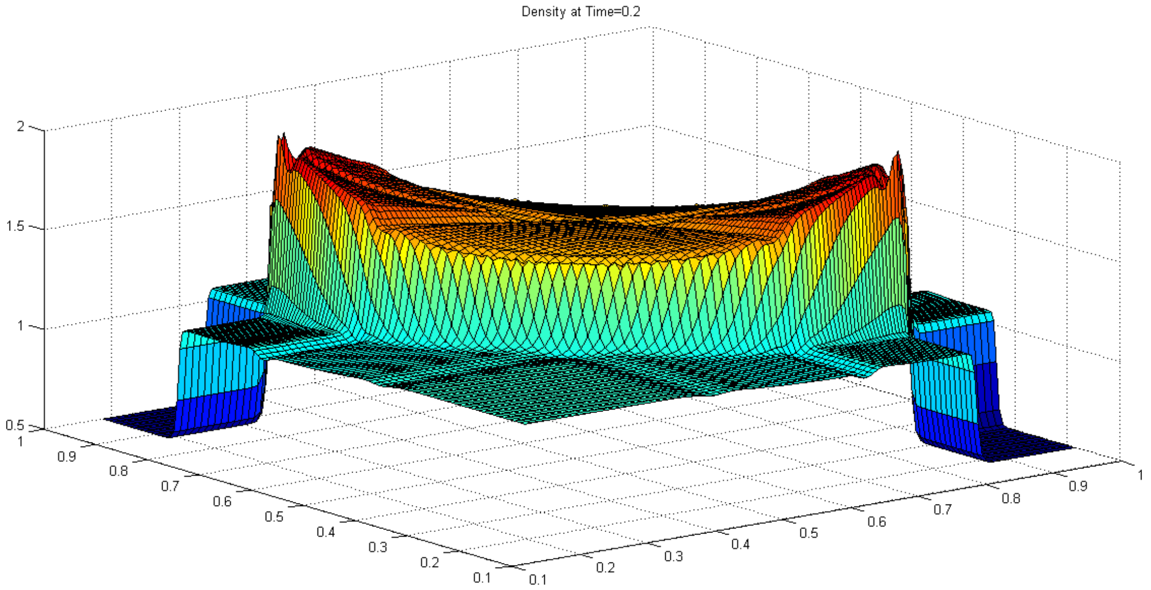


Figure 8.33: 2D Riemann Problem, Surface Plot of density on Fine Mesh at $T = 0.2$.

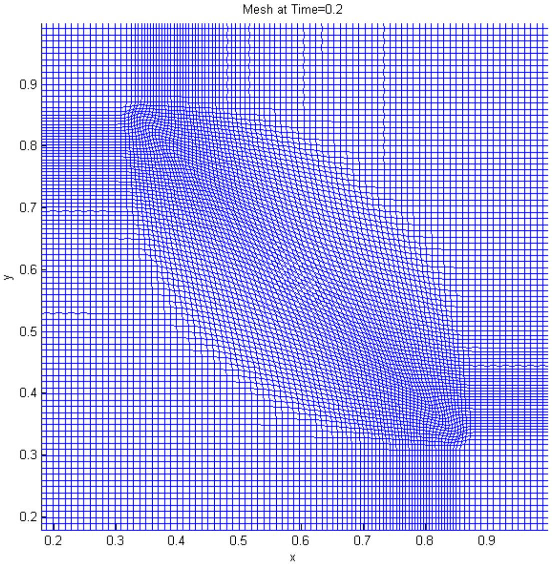


Figure 8.34: 2D Riemann Problem Uniformly Fine Mesh at $T = 0.2$.

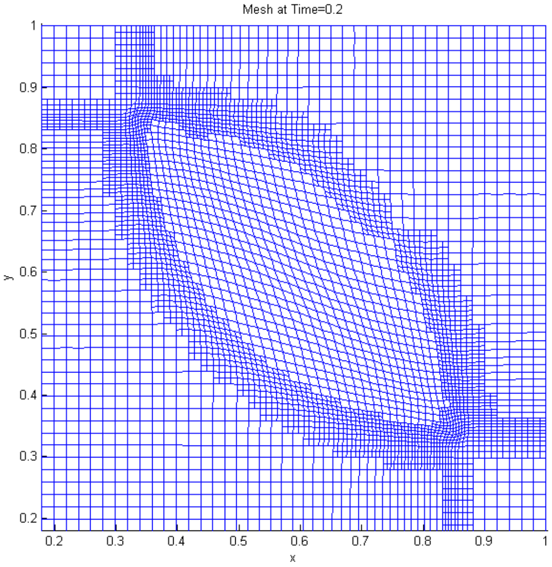


Figure 8.35: 2D Riemann Problem, AMR with Density Gradient at $T = 0.2$.

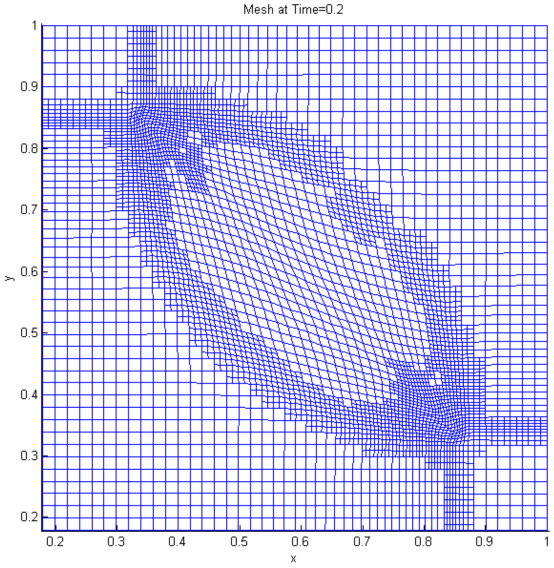


Figure 8.36: 2D Riemann Problem, AMR Mesh with Semi-Discrete in Time Error Estimate for Density at $T = 0.2$.

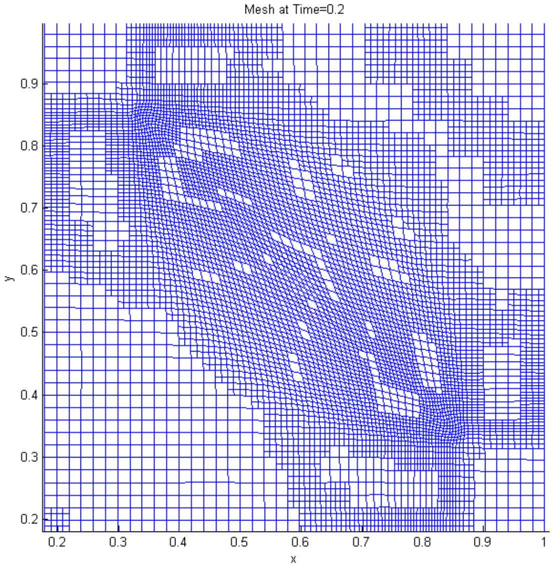


Figure 8.37: 2D Riemann Problem Semi-Discrete in Space Estimated Error for Pressure, AMR Mesh at $T = 0.1$.

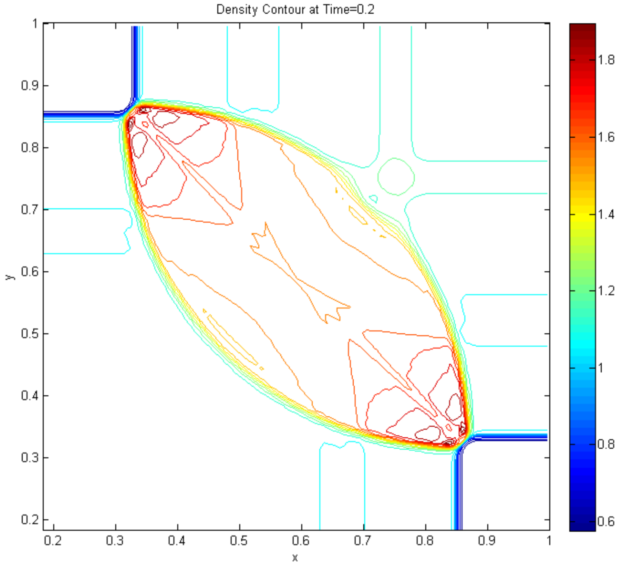


Figure 8.38: 2D Riemann Problem, Contour Plot of Density on Fine Mesh at $T = 0.2$.

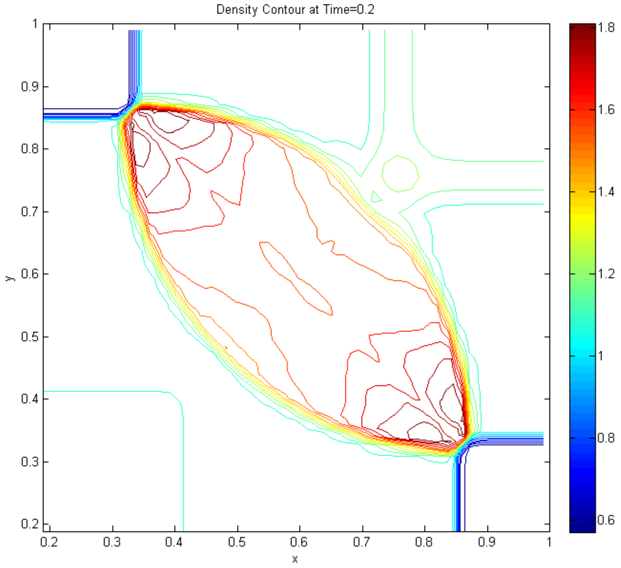


Figure 8.39: 2D Riemann Problem, Contour Plot of Density with AMR using Density Gradient at $T = 0.2$.

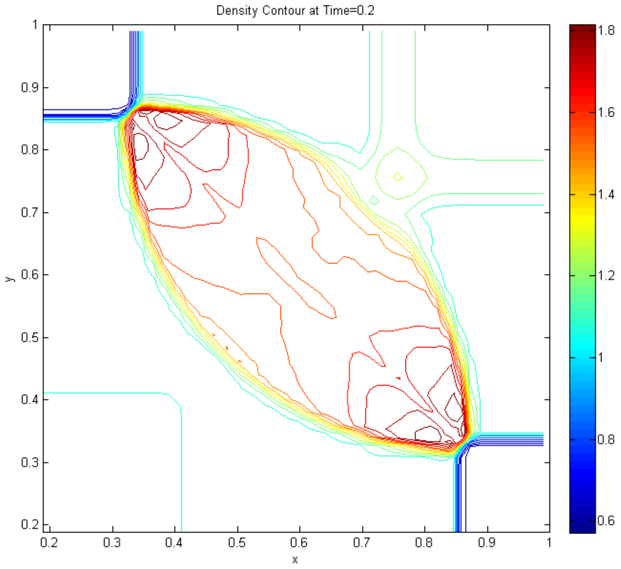


Figure 8.40: 2D Riemann Problem, Contour Plot of Density Using Semi-Discrete in Time Error Estimate for Density AMR at $T = 0.2$.

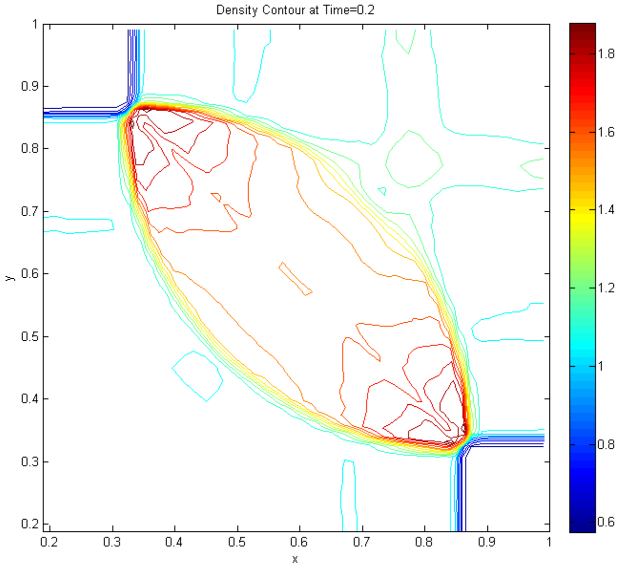


Figure 8.41: 2D Riemann Problem Semi-Discrete in Space Estimated Error for Pressure, Density Contour Plot at $T = 0.1$.

Table 8.5: Run Times and Total Elements for 2D Riemann Problem.

| | Fine Mesh | Density Gradient AMR | Semi-Discrete in Time AMR | Semi-Discrete in Space AMR |
|---|---|---|---|---|
| Run Times | 2488.86 | 656.9 | 920.39 | 2585.4 |
| Total Elements | 3,3500,00 | 643,680 | 565,503 | 273,363 |

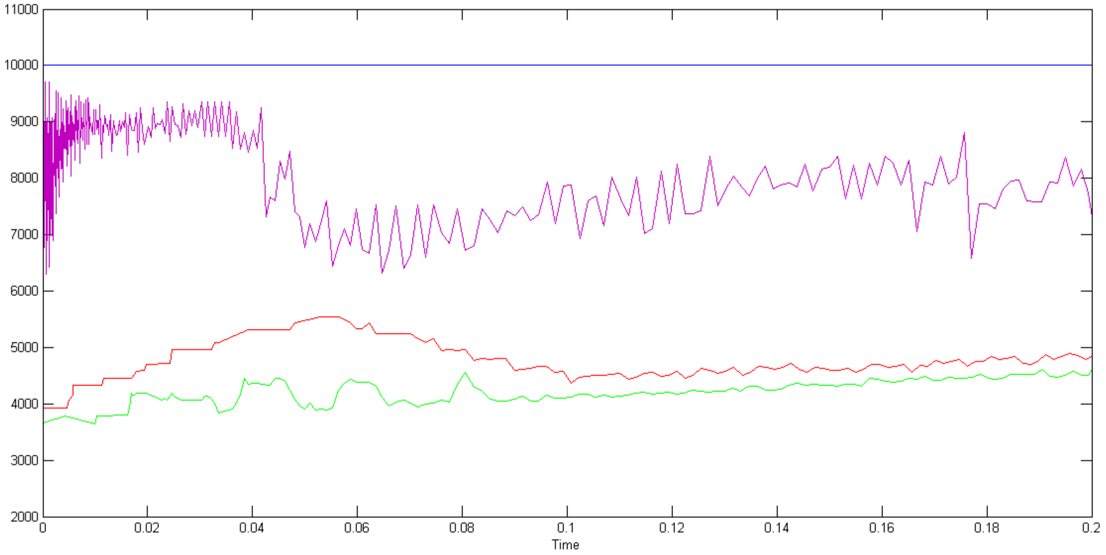


Figure 8.42: Total Number of Elements for the 2D Case.

## 8.3.2 Radial Sod Problem

We run a quarter Radial Sod problem from [90], using an ALE approach, to $T = 0.25$. For the quarter problem where the radius is within 0.4 of $(0, 0)$, the initial conditions are $\rho = 1$, $p = 1$ and zero velocity, elsewhere the conditions are $\rho = 0.125$, $p = 0.1$ and zero velocity. We run the problem on a uniformly fine mesh of size $100 \times 100$, we also implement refinement with density gradients, semi-discrete in time, semi-discrete in space and fully-discrete error estimates initiated from a $50 \times 50$ mesh. The artificial viscosity coefficients are $c_l = 0.5$ and $c_q = 0.75$, see [64].

Since we use an ALE method we do not have any mesh tangling on the fine mesh, see Figure 8.44. When implementing AMR using density gradients we see the expansion, contact and shock are refined, see Figure 8.45. Whereas the results using AMR triggered by a semi-discrete in time error estimate for density, semi-discrete in space error estimate for pressure and fully-discrete error estimate for density refine the shock front, see Figures 8.46, 8.47 and 8.48. However, around the contact the refinement becomes patchy in the latter three. For the semi-discrete in time case, there appears to be less refinement in the constant region between the shock and contact, compared to the semi-discrete in space case and fully-discrete case. However, the semi-discrete in space case, from Figure 8.47, demonstrates more refinement around the expansion. This again agrees with the conclusion in Chapter 6, where the steep solutions are detected by temporal errors and the smooth features are detected by spatial errors. The fully-discrete approach refines most of the areas leading from the shock backwards to the top of the expansion, see Figure 8.48, this is wasteful as several smooth or constant states appear in this region.

The contours for the uniformly fine mesh illustrated in Figure 8.49 are not quite symmetrical, this is because the problem was implemented on a , not polar coordinates. The contours in Figure 8.50 exhibit oscillations around the bottom of the fan also the edges do not reach the boundaries, there are also oscillations around the contact the shock however, is captured well. For the semi-discrete in time error estimate for density there are oscillations around the contact however, they are less severe than the density gradient AMR, see 8.51 and 8.52. Although there are oscillations at the bottom of the fan, the edges do reach the boundaries in this case. The shock is captured well and is comparable to the uniformly fine mesh. The semi-discrete in space estimated error for pressure AMR demonstrates the least amount of oscillatory behaviour around the contact, see Figures 8.53 and 8.54. The base of the fan is more symmetric than the previous two methods. Although there are still oscillation at the base of the fan, the edge is closer to the boundary than in the density gradient case. The shock front is captured least well in the semi-discrete in space AMR since it is slightly irregular. The fully-discrete error estimate exhibits oscillations after the base of the expansion and before the contact, as illustrated in Figures 8.55 and 8.56. The contours around the fan are least symmetric in this case. However, this is the only example where there are not oscillations between the contact and the shock. The shock is slightly irregular, but is generally captured well.

All the methods investigated in this section require significantly less elements than the uniformly fine mesh, which uses 3410000 elements. The total number of elements used over the evolution of the problem for the density gradient AMR is 635399, whereas the semi-discrete in time AMR requires slightly more, 716801. However, the semi-discrete in space and fully-discrete AMR require the most amount of elements, 1011499 and 1311684 elements respectively, see Figure 8.57.

The fully-discrete AMR has the largest run time compared to the other AMR methods, see Table 8.6, it also gives the worst results. We conclude that the fully-discrete AMR is not suitable for this problem. Whereas, there is a significant saving of 85% in the run time for the density gradient, however, although the run time is slightly more for the semi-discrete in time and semi-discrete in space methods, (see Table 8.6), they give better overall results from their contour plots. Where the semi-discrete in time error estimate is best used for sharp features, i.e. the shock and contact. Tthe semi-discrete in space is best applied to smooth features i.e. the fan. This conclusion agrees with the previous results.

We must conclude that further work needs to be done on using error estimate AMR for this problem. The error estimate AMR in Figures 8.46-8.48 may appear to be trying to detect the features of interest, but the refinement is extremely patchy. This will have an effect on the final solution. From Figures 8.52, 8.54 and 8.56, it can be seen that the choice of derefinement tolerance could be causing a problem with the patchy refinement, since the contours are following the areas of derefinement. Whereas in Figure 8.45 we illustrated that the density gradient AMR behaves extremely well. The areas of refinement are clear and capture all desired features.

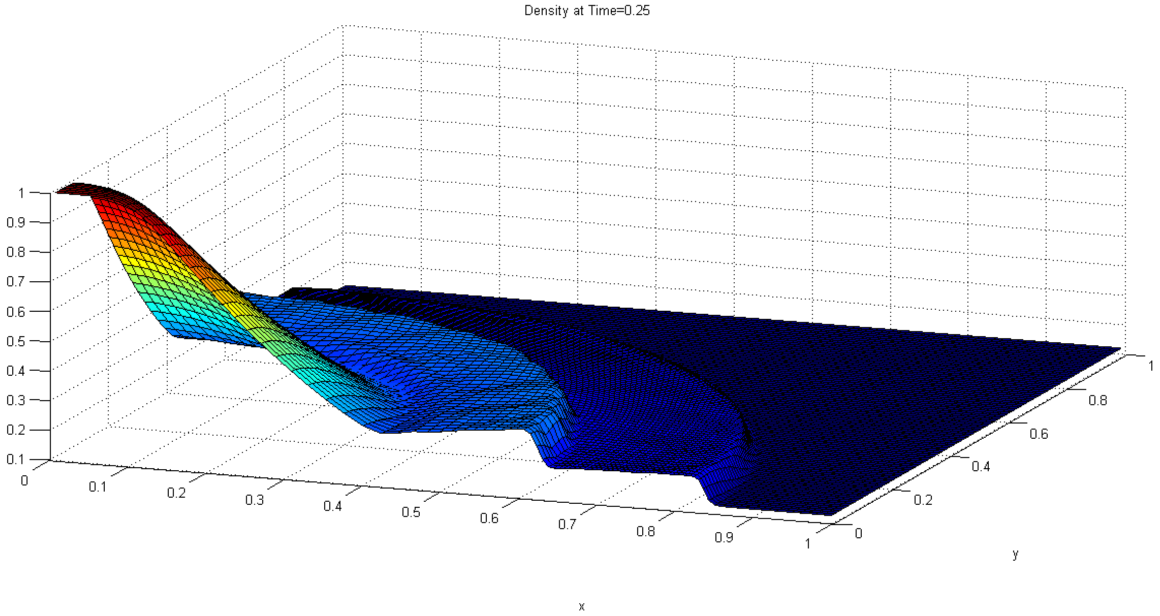


Figure 8.43: Radial Sod Problem, Surface Plot of density on Fine Mesh at $T = 0.25$.

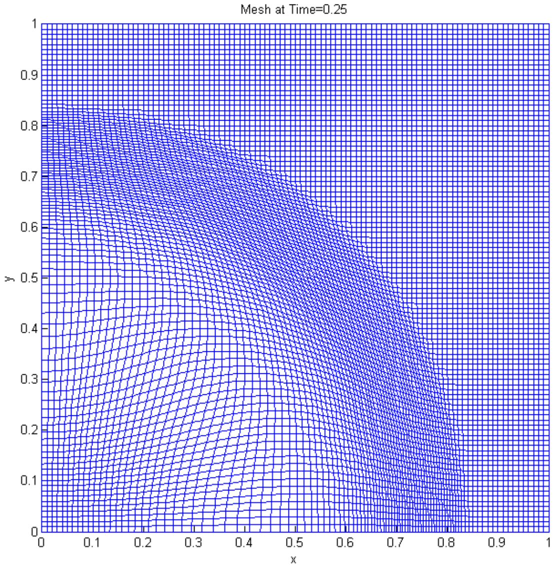


Figure 8.44: Radial Sod Problem Uniformly Fine Mesh at $T = 0.25$.

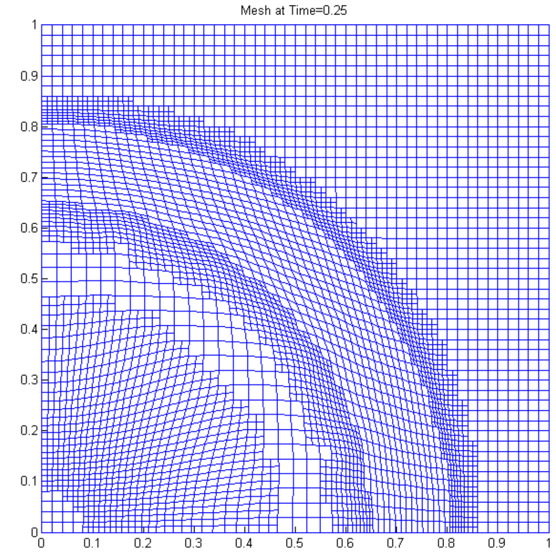


Figure 8.45: Radial Sod Problem, AMR with Density Gradient at $T = 0.25$.

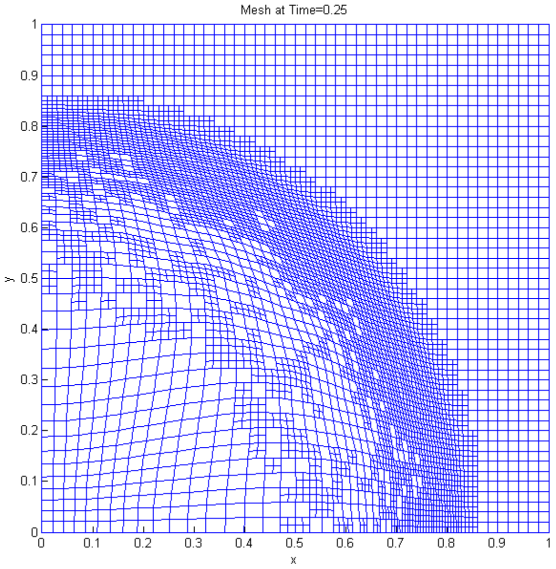


Figure 8.46: Radial Sod Problem with Semi-Discrete in Time Error Estimate for Density AMR at $T = 0.25$.

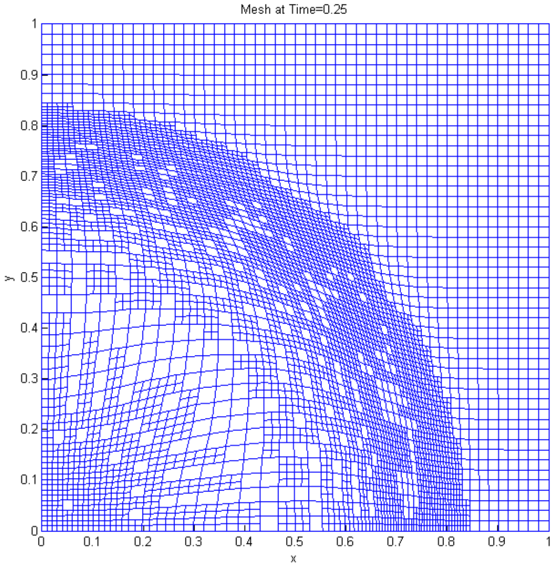


Figure 8.47: Radial Sod Problem with Semi-Discrete in Space Estimated Error for Pressure, AMR Mesh at $T = 0.25$.

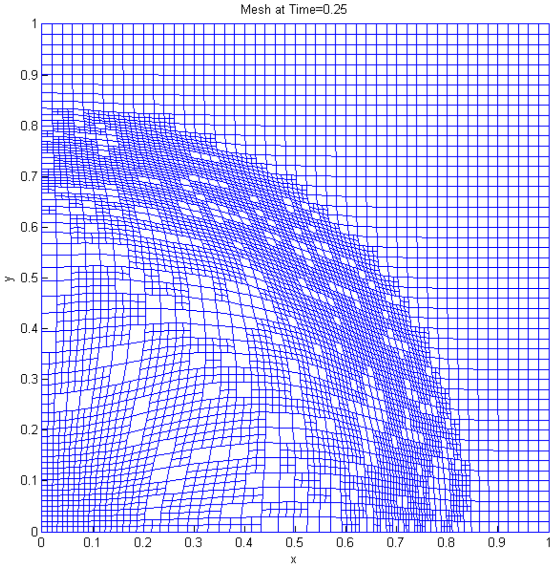


Figure 8.48: Radial Sod Problem with fully-Discrete Estimated Error for Density, AMR Mesh at $T = 0.25$.

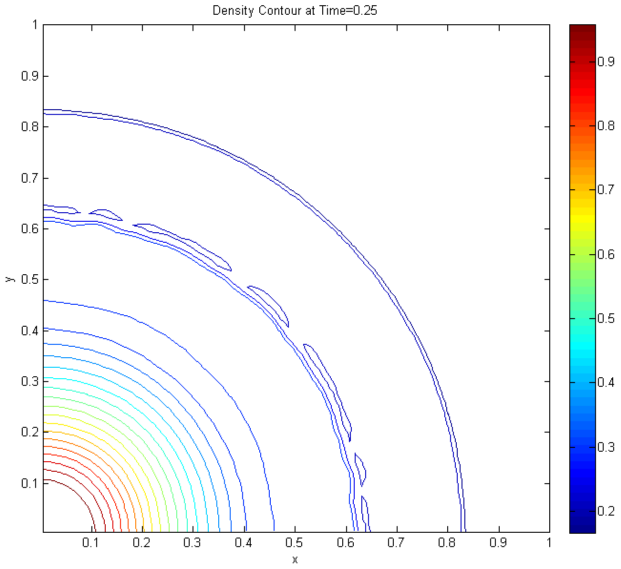


Figure 8.49: Radial Sod Problem, Contour Plot of Density on Fine Mesh at $T = 0.25$.

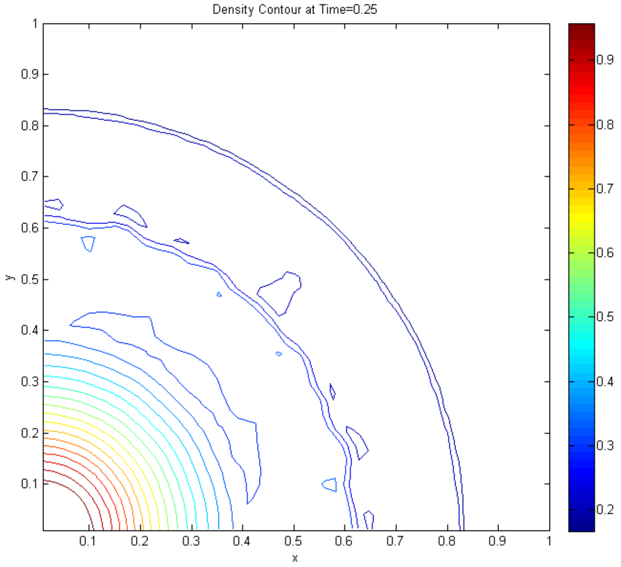


Figure 8.50: Radial Sod Problem, Contour Plot of Density with AMR using Density Gradient at $T = 0.25$.

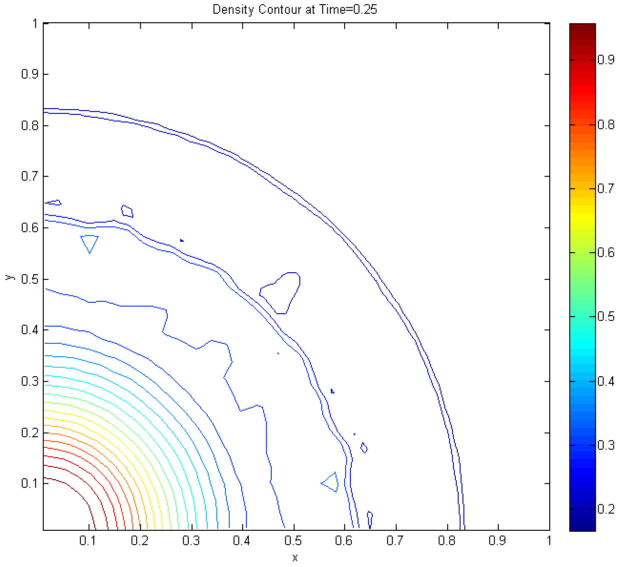


Figure 8.51: Radial Sod Problem, Contour Plot of Density using AMR with Error Estimate at $T = 0.25$.

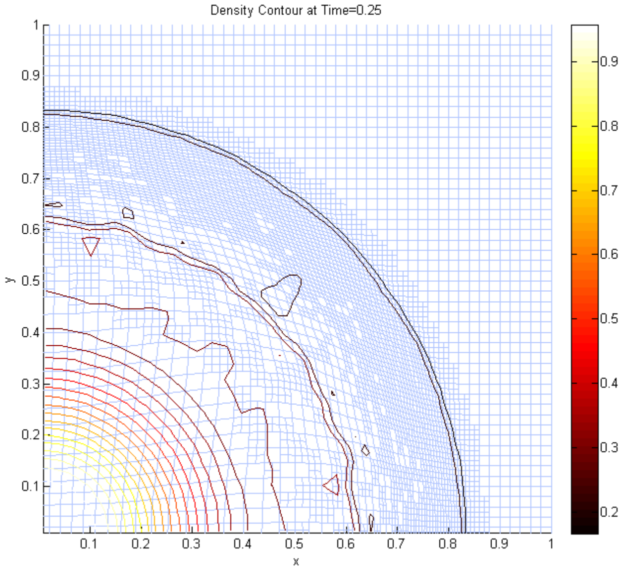


Figure 8.52: Radial Sod Problem, Mesh and Contour Plot of Density using AMR with Error Estimate at $T = 0.25$.

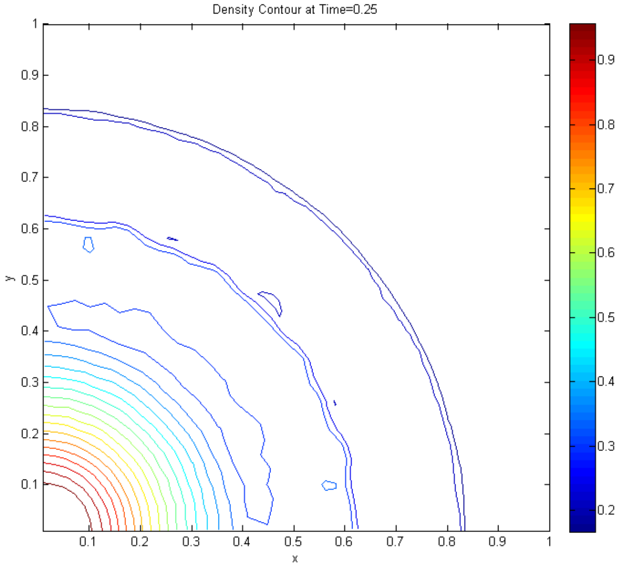


Figure 8.53: Radial Sod Problem with Semi-Discrete in Space Estimated Error for Pressure, Density Contour Plot at $T = 0.25$.

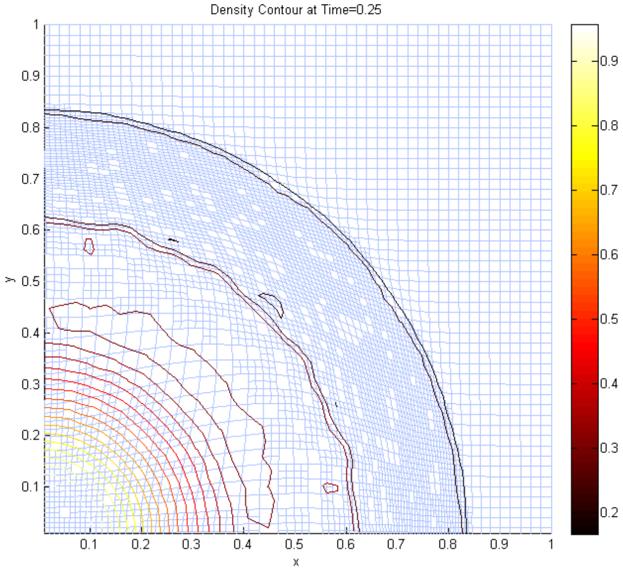


Figure 8.54: Radial Sod Problem with Semi-Discrete in Space Estimated Error for Pressure, Density Contour and Mesh Plot at $T = 0.25$.

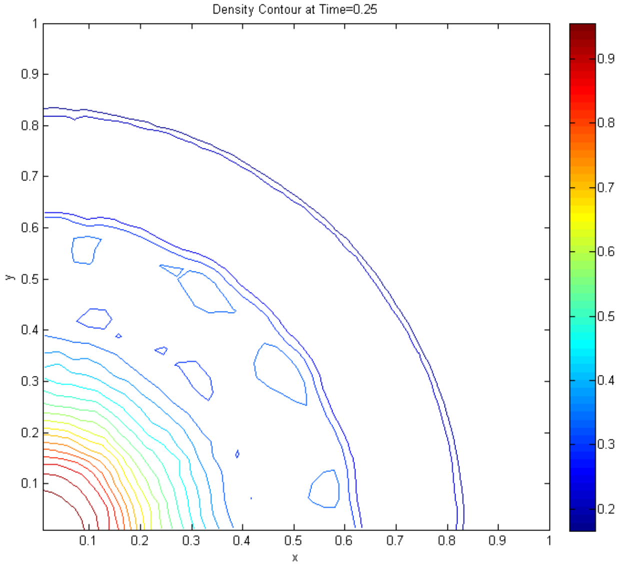


Figure 8.55: Radial Sod Problem with Fully-Discrete Estimated Error for Density, Density Contour Plot at $T = 0.25$.

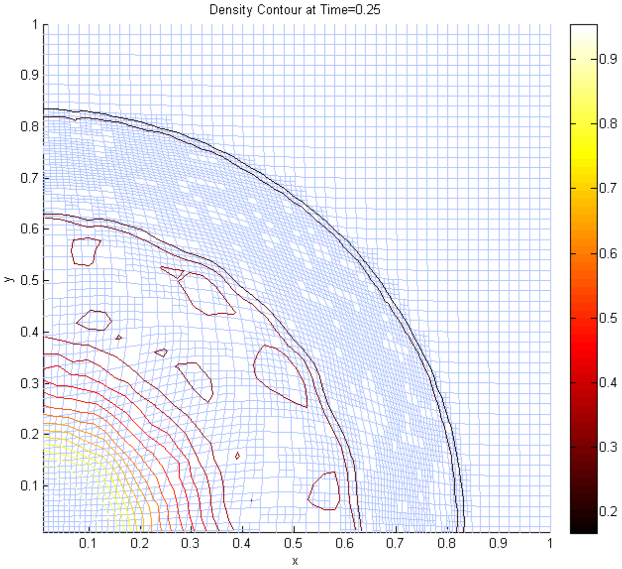


Figure 8.56: Radial Sod Problem with Fully-Discrete Estimated Error for Density, Density Contour and Mesh Plot at $T = 0.25$.

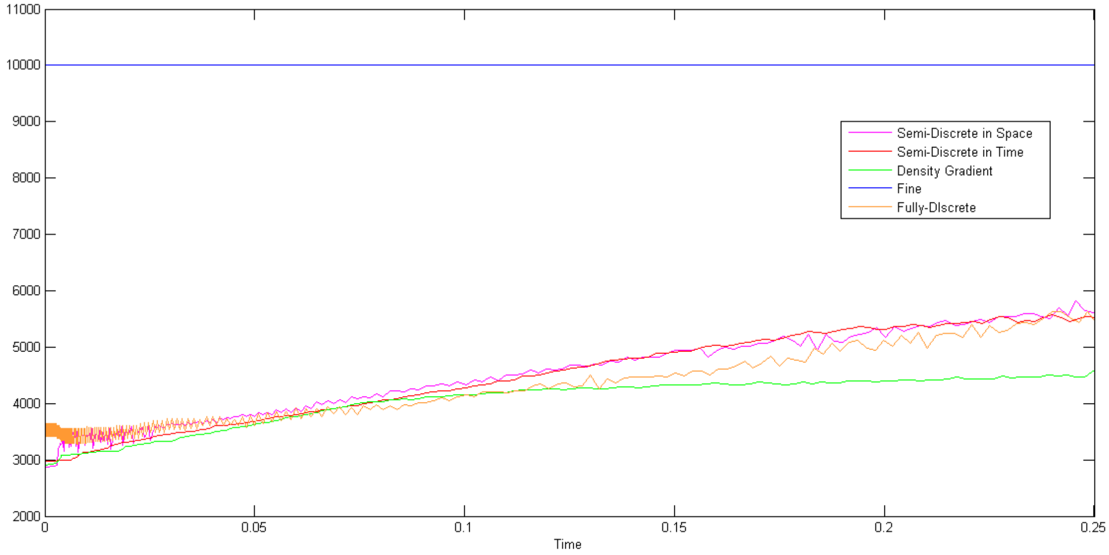


Figure 8.57: Total Number of Elements for the Radial Sod Problem run with ALE.

Table 8.6: Run Times and Total Elements for Radial Sod Problem.

| | Fine Mesh | Density Gradient AMR | Semi-Discrete in Time AMR | Semi-Discrete in Space AMR | Fully Discrete AMR |
|---|---|---|---|---|---|
| Run Times | 2483.720 | 352.200 | 687.650 | 658.590 | 837.663 |
| Total Elements | 3,410,000 | 635,399 | 716,801 | 1,011,499 | 1,311,684 |

### 8.3.3 Taylor-Sedov Blast Wave

The Taylor-Sedov blast wave consists of an intense explosion from a single point, from which a blast wave propagates out. Although this is another radial problem, and as we saw in the previous section the error estimate AMR struggled when applied to the Radial Sod problem, it is much easier to detect the area to refine since the problem only consists of one main feature, rather than three.

The initial conditions for this problem are

$$\begin{aligned} p &= 0 \\ \rho &= 1 \\ \gamma &= \frac{5}{3} \end{aligned}$$

at the origin. The quarter domain is represented by $(0,1) \times (0,1)$. The cell energies are 0 everywhere except the cell at the origin where it is given the value 8 (see [64]). However, we are dealing with a quarter region and to transfer into the internal energy we divide the total energy by four times the cell area.

The calculation is evolved to a final time of $T = 0.1$. We use an ALE calculation on a uniformly fine grid of $64 \times 64$, and apply the density gradient, the semi-discrete in time error estimate, the semi-discrete in space error estimate and the fully-discrete error estimate as refinement criteria, with an initial mesh of $32 \times 32$. The artificial viscosity coefficients are taken as $c_l = 0.5$ and $c_q = 0.75$.

The run time for the density gradient approach is 90% less than the uniformly fine grid, whereas the run time for the semi-discrete in time, semi-discrete in space and fully-discrete AMR is 85%, 88% and 77% less, respectively. All are significant savings on the uniformly fine mesh runtime, see Table 8.7.

The saving for the approximated error comes in the number of elements. Figure 8.58 demonstrates that the semi-discrete in time error estimate uses fewer elements over the evolution than the density gradient approach.

However, it is important that these savings have not compromised the solution. Figure 8.61 illustrates that the density gradient approach refines the front of the shock wave very well, but also refines behind the shock wave. It is stated in [64] that this refinement is due to the density variations which are a result of the energy being initialised as a square rather than a quarter circle as would be more correct due to radial asymmetry. It is only the shock front which is required to be refined, as is excellently done in Figure 8.62, where the semi-discrete in time AMR has been implemented. The semi-discrete in space AMR illustrated in Figure 8.63, refines around the shock very well, with slight refinement at the origin. The fully-discrete AMR also captures the shock front extremely well, illustrated in Figure 8.64. There is slightly extra refinement before the shock front, this is because of the smooth region before the shock front, as illustrated in Figure 8.59, is also being refined, however this refinement is not as symmetric as the previous results.

These results are in agreement with the previous findings. The shock front is detected with temporal errors as seen in Figures 8.62-8.64, whereas the smooth features are detected by spatial errors, where we saw the additional refinement away from the shock front in Figures 8.63 and 8.64.

All contour plots from Figures 8.66-8.69 are in good agreement to the uniformly fine solution in Figure 8.65. However, the fully-discrete error estimate AMR gives the worst results as illustrated in Figure 8.69. The contours are not as symmetric as previous cases, and there is more noise around the shock front.

We have demonstrated that although the density gradient approach has a slightly less overall runtime than the estimated error approach for all cases, the estimated error approach in a semi-discrete in time, semi-discrete in space and fully-discrete manner actually use fewer timesteps and fewer elements over the evolution of the problem, see Figure 8.58. Additionally, the results of the estimated errors are extremely favourable compared to the uniformly fine results.

In conclusion we have demonstrated a method to estimate the error, and shown why this method should be used as a refinement criteria. The investigation has been extended to 2D and refinement has been implemented using this approach. The last chapter discusses the main developments and results of this thesis, with a summary of the main features.

Table 8.7: Run Times and Total Elements for Sedov Blast Wave Problem.

| | Fine Mesh | Density Gradient AMR | Semi-Discrete in Time AMR | Semi-Discrete in Space AMR | Fully-Discrete AMR |
|---|---|---|---|---|---|
| Run Times | 535.830 | 55.350 | 81.850 | 65.500 | 120.901 |
| Total Elements | 1,642,496 | 182,041 | 187,619 | 176,062 | 174,745 |

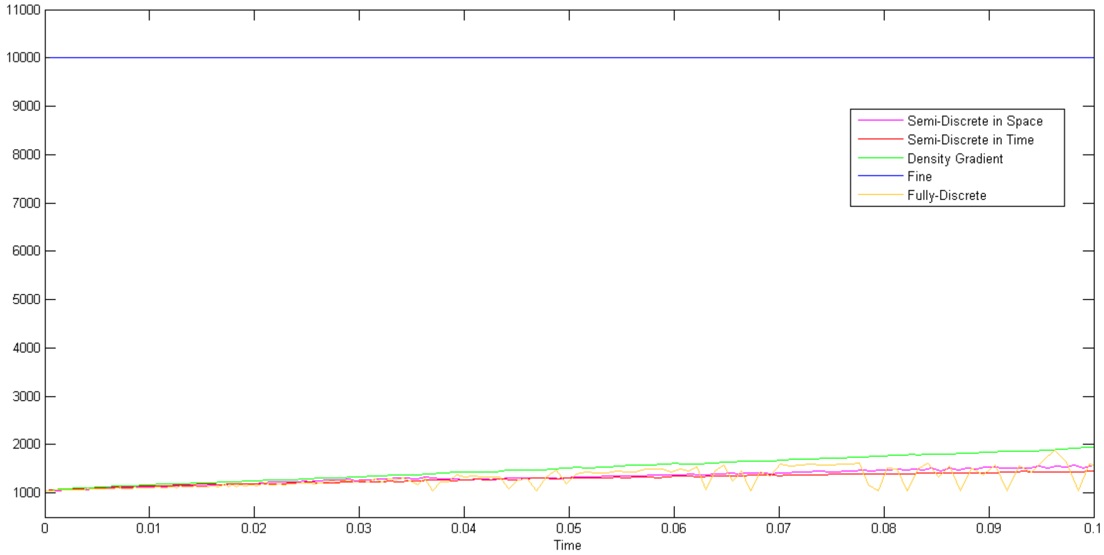


Figure 8.58: Sedov Number of Elements over Evolution of Problem.

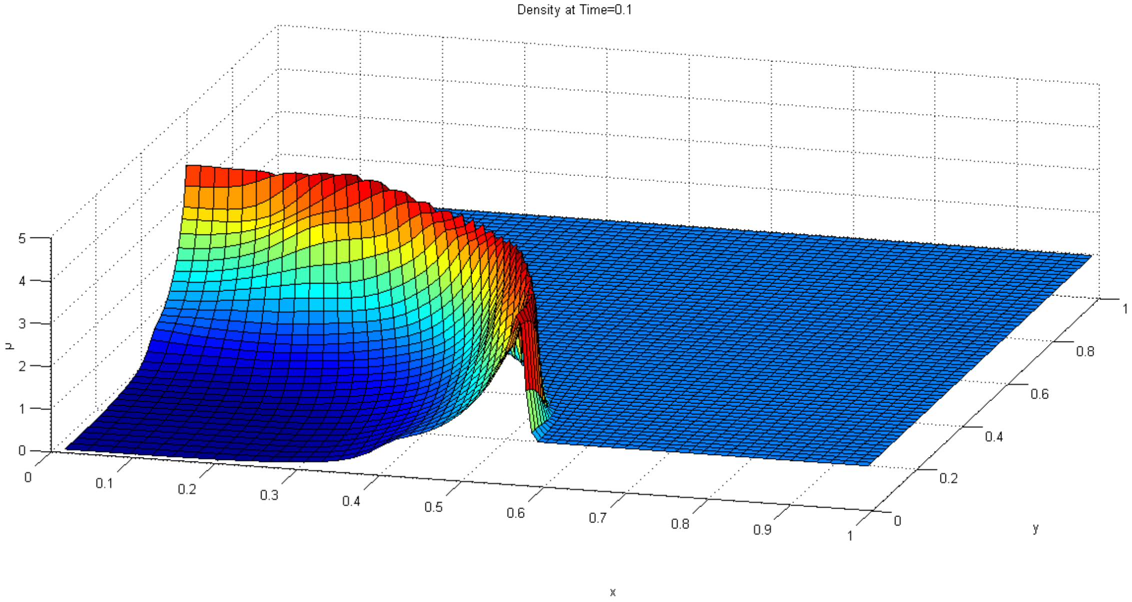


Figure 8.59: Sedov Problem Density Surface Plot on Fine Mesh at $T = 0.1$.

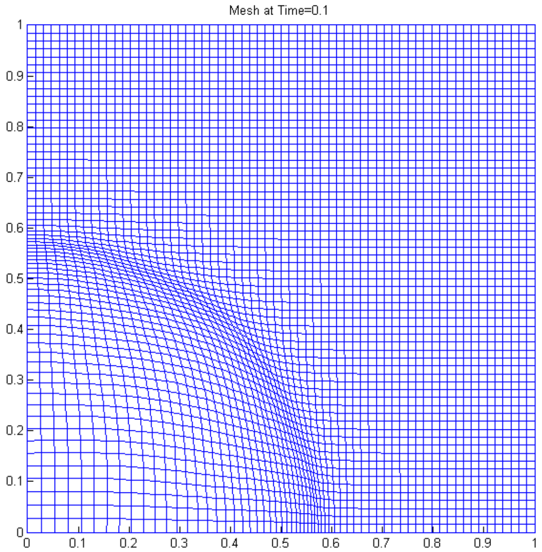


Figure 8.60: Sedov Fine Mesh at $T = 0.1$.

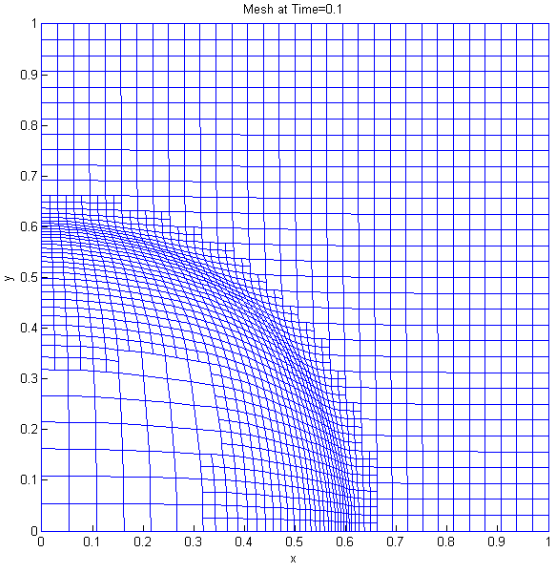


Figure 8.61: Sedov, AMR with Density Gradient Mesh at $T = 0.1$.

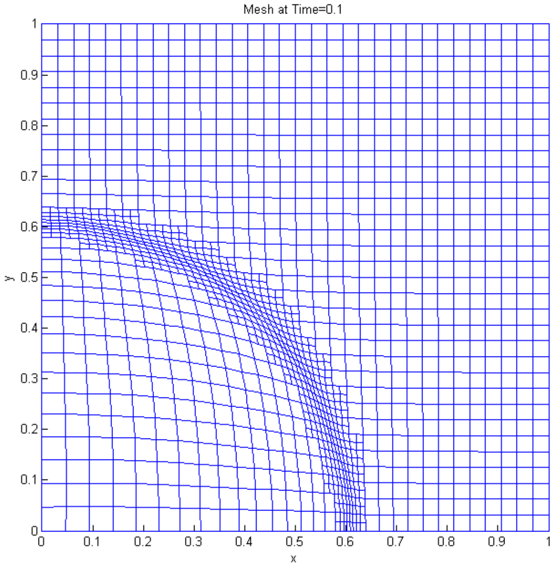


Figure 8.62: Sedov, Using a Semi-Discrete in Time Error Estimate for Density AMR Mesh at $T = 0.1$.

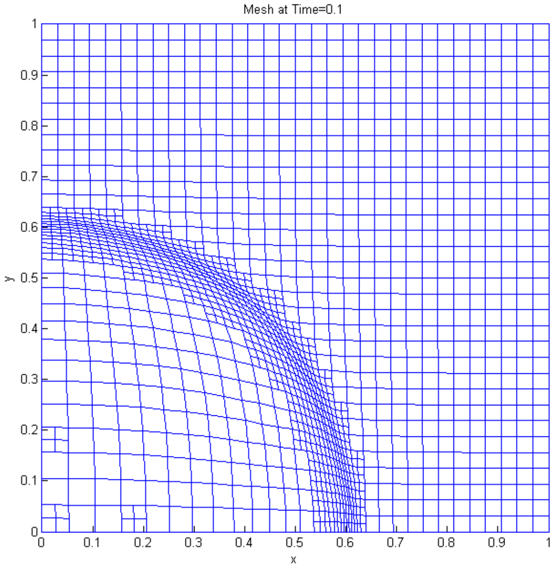


Figure 8.63: Sedov Problem with Semi-Discrete in Space Estimated Error for Pressure AMR Mesh at $T = 0.1$.

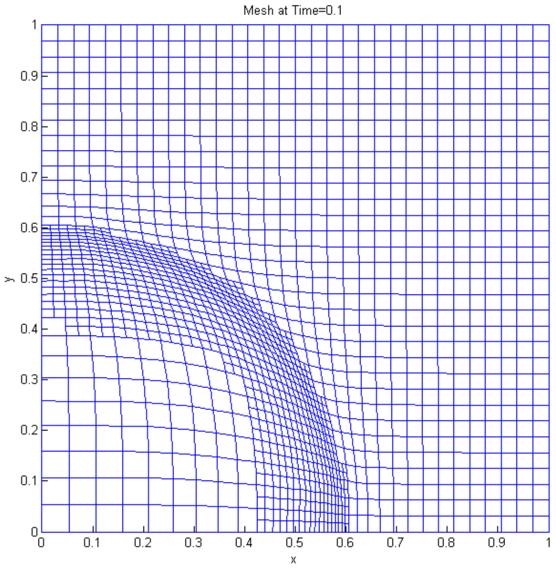


Figure 8.64: Sedov Problem with Fully-Discrete Estimated Error for Density AMR Mesh at $T = 0.1$.

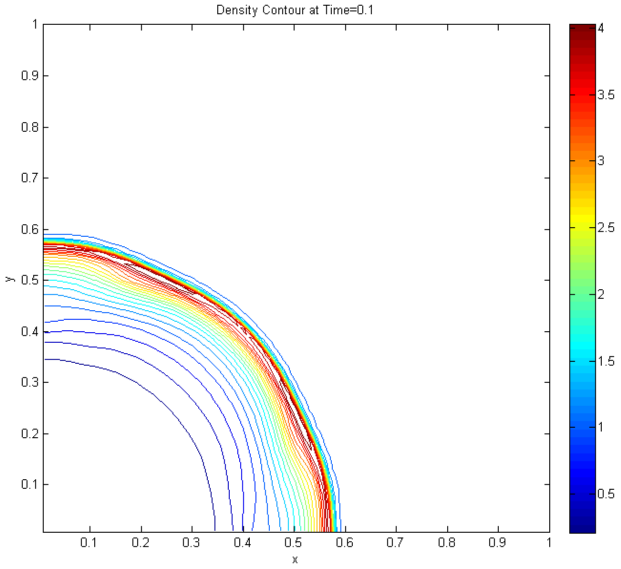


Figure 8.65: Sedov Density Contour Plot at $T = 0.1$.

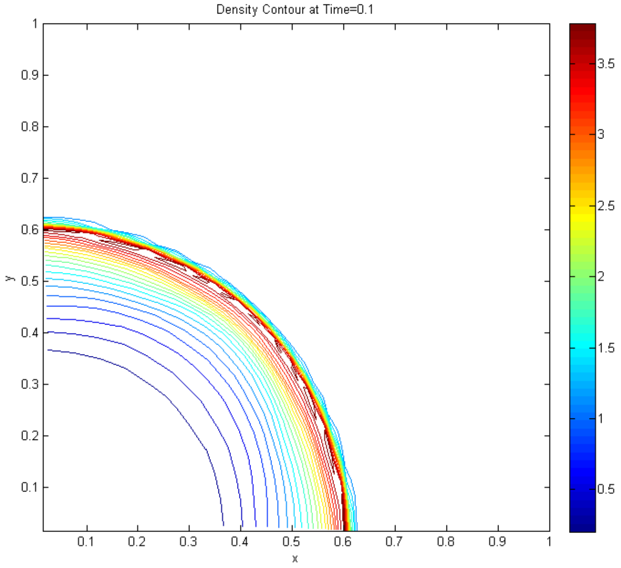


Figure 8.66: Sedov Density Contour Density Gradient AMR Plot at $T = 0.1$.

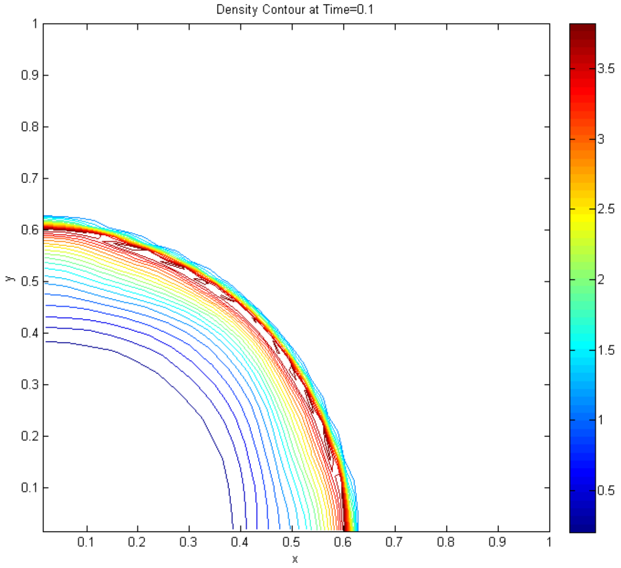


Figure 8.67: Sedov Semi-discrete in time Density Contour Estimated Error AMR Plot at $T = 0.1$.

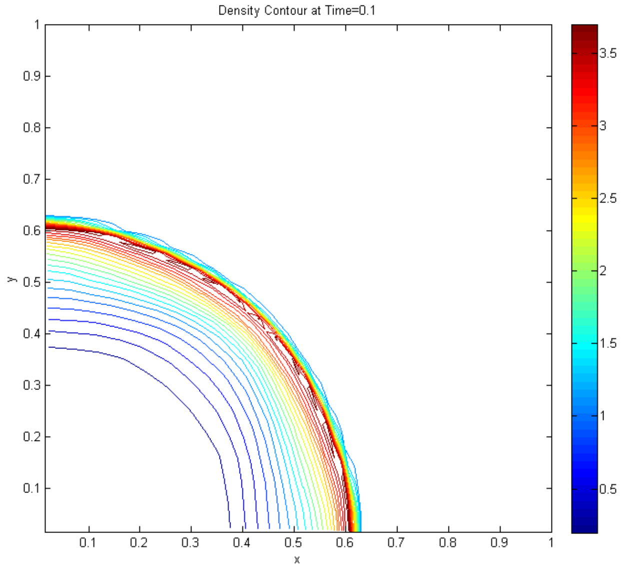


Figure 8.68: Sedov Problem with Semi-Discrete in Space Estimated Error for Pressure, Density Contour Plot at $T = 0.1$.

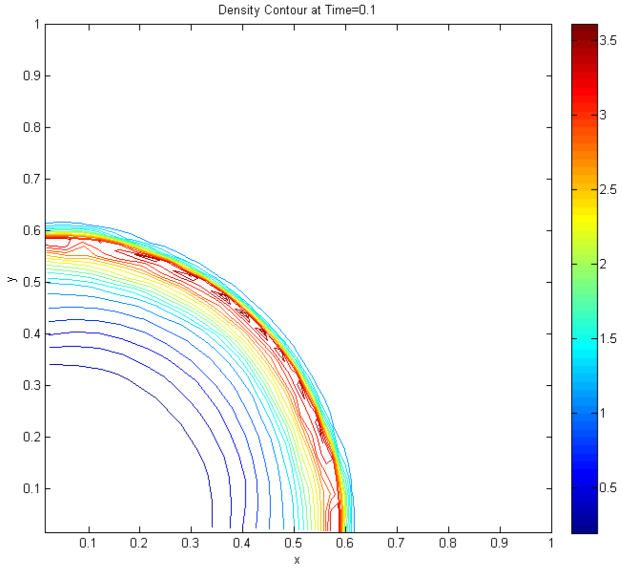


Figure 8.69: Sedov Problem with Fully-Discrete Estimated Error for Density, Density Contour Plot at $T = 0.1$.

# Chapter 9

# Conclusions and Further Work

The main achievement of this thesis has been the development of refinement criterion based on error estimates using truncation errors for the Euler Equations. This is the first time the truncation error has been derived for the Main Scheme, from Sections (3.3.1) and (4.2.1).
When solving fluid flow problems numerically, the accuracy of the solution is dependent on the mesh resolution. Because fluid problems usually involve regions where the solution changes rapidly, it is particularly necessary to refine these areas. In this work, we have developed an alternative refinement criteria to that stated in [64], based on error estimates. Due to the advantage that the Lagrangian mesh follows the solution, we developed the error estimate for this phase.
We investigated a simpler system initially i.e. the Isothermal equations. This was then extended to the system of three equations i.e. the Euler Equations.
In the Introduction, we introduced the main concepts, methods and aims of the thesis.
The ideas introduced in the Introduction were then expanded in Chapter 2. The system of equations which we solve for were stated and described in a Lagrangian reference frame in Chapter 2, since most papers express the Euler equations in a hybrid form using the material derivative to link between the Eulerian and Lagrangian grids. Here it is important to to highlight the equations expressed fully in the Lagrangian reference frame. The main forms of error estimates for refinement were discussed, where the pros and cons of the estimates are outlined in [77]. The Richardson extrapolation method which uses the order of accuracy of truncation errors was highlighted. It was deduced that this method is expensive due to the requirement of a coarse and fine grid. However, the idea of using the truncation errors suggested the development of the idea that we proposed in this thesis.

Once the truncation error was obtained a method was required to approximate the uncomputable terms in the truncation error. This lead to the inclusion of a second scheme of the same order as the Main Scheme, from Sections (3.3.1) and (4.2.1). Using two schemes and an approach similar to Milne's device, see [27], we were able to deduce an error estimate.

Since the system of equations includes spatial and temporal derivatives for simplification we considered the spatial and temporal errors separately.

Chapter 3 began by investigating two schemes used to discretise the temporal derivatives in the system of two equations. One scheme came from [64] and was referred to as the Main Scheme, as seen in Sections (3.3.1) and (4.2.1), where the other came from [28]. The truncation error of $x$ was obtained for each scheme which was then used to obtain an estimate to the leading term of the truncation error for the Main Scheme. Similarly this was done for the Euler equations, again using the Main Scheme from [64], seen in Sections (3.3.1) and (4.2.1), and the second scheme was chosen from [46]. It was then required to obtain an error estimate in terms of the spatial derivative.

In Chapter 4 we looked at comparing two schemes to approximate the spatial derivatives. The Main Scheme from Sections (3.3.1) and (4.2.1), uses a Galerkin finite element approach. The second scheme, which was developed in this thesis, uses a Petrov-Galerkin finite element approach. The truncation errors for each scheme was obtained, and used to obtain an approximate to the leading term of the truncation error, in terms of the two schemes for the spatial derivative.

The results from Chapters 3 and 4 were combined in Chapter 5 to obtain an estimate to the fully discrete form of the error for $x$. Using the estimated error for $x$, we then obtained an error estimate for $\rho$ and $u$ in fully and semi-discrete forms, for the system of two equations and the Euler equations.

Using the results from Chapter 5, we tested the error estimates using a Riemann problem for the Isothermal equations, and Sod's shock tube problem for the Euler equations. These test cases were used since they are simple and an exact solution is available hence, allowing a comparison between the estimated errors and true error. In all cases the location and magnitude of the estimated error for the shock were in extremely good agreement to the true error. The sharp features at the top and bottom of the expansion were detected in all cases. However, the magnitudes of the estimated errors were less than the true error. The rarefaction itself was poorly captured in several cases, the best case was the velocity estimated error for the Isothermal and the pressure estimated error in the Euler Equations. Additionally, a contact exists in the solution for the Euler Equations. Although not obvious, since the error blended in with the oscillatory behaviour before the shock front, the contact was detected.

As for the top and bottom of the fan, the contact was detected with less magnitude for the estimated error case compared to the true error. We investigated the schemes at the contact further in Appendix D, and concluded the schemes were exact at the contact, this confirms why there was such difficulty in detecting the contact throughout. The $\infty$-norms were in good agreement in all cases, since the shock was approximated very well.

Since the main aim of this thesis was to develop refinement criteria for the 2D Euler Equations, we extended the 1D results from Chapters 3 and 4 to the 2D case in Chapter 7. However, we did not extend the error estimate for velocity to 2D, since the velocity is held at a node which has contributions from four surrounding elements, therefore, it is difficult to determine which element would need refining. The semi-discrete in time error estimate was straightforward to obtain, and is just a simple extension of the 1D case. However, there were more difficulties with the semi-discrete in space case. The leading term of the semi-discrete in space case contains several mixed derivatives which can not be simplified to a single term. For this reason we assumed that the mesh remained Cartesian throughout the evolution of the problem, therefore, we disregarded any mixed derivatives. A semi-discrete in time error estimate for $x$ was available, this was then applied to obtain an estimated error for $\rho$, the semi-discrete in space error estimate for pressure was also available. Both semi-discrete errors were combined to obtain a fully-discrete error estimate for density.

The results from Chapter 7 were applied to a variety of test problems including the Sod Shock Tube and the more genuine 2D problems such as the Square Sod, and 2D Riemann problem, which were ran purely Lagrangian. Since mesh tangling was exhibited in the 2D Riemann problem, when ran in a Lagrangian manner we implemented the full ALE method to the 2D Riemann problem. Additionally, we introduced the Radial Sod problem and Sedov Blast Wave, when ALE was applied. By using the ALE method we were able to demonstrate that the error estimates developed in this thesis can be implemented with ALE. When the method was applied to the 1D test problem Sod shock tube, which was run in a 2D manner on a Lagrangian mesh, the areas of refinement were excellent, i.e. only the top and bottom of the fan were detected with the contact and shock also being detected. We demonstrated that the error norms were in better agreement for the error estimate AMR than the density gradient AMR, compared with the uniformly fine mesh, which is an excellent result for the method developed here. However, there were difficulties with the Radial Sod problem. The refinement was patchy and the contour plots were extremely noisy, which suggests this should be an area of further work.

The estimated error behaves excellently when compared against the uniformly fine grid, and proves to be extremely well suited to detecting specific features with a minimal amount of elements compared to the density gradient approach. One of the difficulties discussed in Chapter 2 is how to determine which refinement criteria is best suited to detecting certain flow features. The main conclusion we have from the results in Chapters 6 and 8 is the error estimate is useful in detecting specific flow features, i.e. a temporal error is useful in detecting steep solutions and a spatial error is best suited to detecting smooth features, which is in agreement with Aftosmis' suggestion in [2], that different sensors should be used for different flow features.

From the development of the estimated error approach, we have come across several restrictions to the method. The derivatives of the exact solution in the truncation error need to be simplified so that the two schemes can be compared. Therefore, the leading term of the truncation error for each scheme needs to be of the same form with different coefficients in order to gain an estimate for that term. As we saw in Chapter 7, the leading term of the truncation error contained several derivatives and could not be simplified into a general term. This lead to disregarding most of the terms in the spatial error for the 2D case. When we investigated the 2D results in Chapter 8 it was the semi-discrete in space and fully-discrete estimated errors which under performed or proved difficult to gain any results. This indicates the importance of the terms we had to disregard. However, we were able to obtain an approximate error for density in all cases. This is suggested in [20] to be the best variable to refine upon.

In conclusion we have obtained an error estimate based on truncation errors to the Main Scheme, from Sections (3.3.1) and (4.2.1), for the first time in 1D and 2D. We have investigated the behaviour of the error estimate against a true error. The error estimate has been applied as a refinement criteria to a range of test problems in 2D. The estimated error AMR detects all the specific features of interest extremely well. In the Sod shock tube problem with refinement we see that, although the features are detected extremely well there appears to be over refinement at the shock. As seen in the Square Sod problem there is not an unrefined region between the contact and shock, as would be expected due to the constant state. This is due to the shock being over refined yet again. In some cases this has lead to more elements overall compared to the density gradient approach, which (along with the computational time of the second scheme) contributes to the overall run time being greater in the estimated error case. If the shock was not refined as much the run time would be reduced.

## 9.1 Further Work

The main areas of further work include the automation of setting the refinement tolerance. We have developed an approach which detects specific fluid flow features extremely well. However, it still remains to set the tolerances automatically to determine when refinement should happen. It is particularly difficult to set a suitable refinement tolerance in the 2D Riemann problem, due to the complexity of the features present. From the 1D results, a starting point would be to investigate the magnitude differences between the shock, which is detected extremely well, with the magnitudes of contact, top and bottom of the rarefaction to determine the relationship between these magnitudes in association with the error.

In Chapter 6, we saw that Scheme 1 and Scheme 2 results differed by the overshoots/undershoots being reversed. There was also smoothing around the shock in the Scheme 2, case and evidence of odd-even decoupling. These effects could be a result of the artificial viscosity. By varying the $c_q$ and $c_l$ terms in the artificial viscosity we could determine its effect on the solution from Scheme 2.

As exhibited in many of the 1D results there are oscillations before the shock. The over refinement at the shock could be used to detect and understand these oscillations. As we saw in Chapter 6, many of the results suffered in magnitude around the expansion. We also suggested that the magnitude difference could be due to the oscillations, i.e. since mass is conserved, the result will be compromised elsewhere in the solution due to the oscillations. Since the second scheme was always less than the true error this could suggest that the Main Scheme, from Sections (3.3.1) and (4.2.1), is not behaving as well as would be expected. To understand the oscillations and magnitude differences a deeper understanding of how the schemes behave separately and in relation to each other, i.e. consider the expansion, if one of the schemes is more accurate than the other will the estimated error be dampened in this region?

The approach can be restrictive, if the truncation error does not simplify to a function of derivatives this will cause problems when comparing against the second scheme. In order to determine two schemes which have the same leading term but of different factors a modified equation approach could be imposed. By doing so the two schemes will have the same function in the leading term (with different factors), therefore, there will be no restrictions on obtaining an approximate to this term.

We illustrated in Chapter 7 how the error estimate under performed as a refinement criteria for the Radial Sod problem. All Radial Sod results had islands of derefinement. The contours were extremely noisy, especially the areas where the contours were following the derefinement islands. We have seen in Appendix D how an error estimate is not obtained at the contact, therefore it is difficult to refine this area accurately. Therefore, the contact may be getting refined at some time steps and not being refined at others, hence leaving areas of patchy refinement. The error estimate may be under performing as a refinement criteria since we use Cartesian geometry, but we have a radial solution with a contact. The density gradient performed extremely well on this problem. It is necessary to understand why and how these two methods give such different answers.

We saw in Chapter 7 how we disregarded some of the derivatives in the leading term of the truncation error. This was done under the assumption that the mesh remains relatively Cartesian therefore, mixed derivatives can be disregarded. However, we saw in Chapter 8 the consequence of omitting these terms, where the solutions were the least comparable when the spatial derivative had been estimated. This requires further work to determine how this term should be dealt with. One possible suggestion is to investigate splitting methods, developed by Strang, see [84].

Once the error estimation for the spatial derivatives has been improved, this error could be used to trigger anisotropic refinement. This could be implemented by doing a sweep in the $x$ direction followed by a sweep in the $y$ direction and calculate a refinement ratio from this. The refinement ratio would determine whether isotropic or anisotropic refinement is required. Further investigations could be carried out to determine the effect of the error for different variables has on the solution. As we saw in the 1D test case the rarefaction was not detected very well by the density error, although the velocity detected it better. This could have an effect on the magnitude differences in the overall errors. Following from this a combination of these variables could be used in such a way that different variables trigger refinement for specific flow features.

Further investigation needs to be done on reducing the computational run time. Although in some cases we reduced the overall number of elements the run time was still higher in the estimated error AMR approach, due to the additional scheme introduced to estimate the error. Since there is over refinement at the shock, less buffering cells could be used, therefore this would contribute to lowering the overall runtime.

Along with the above suggestions, other interesting areas of further work include; how to implement refinement when using an error estimate of a nodal value, multiple levels of refinement, understanding how to use this method to trigger an anisotropic refinement and fully implement this method theoretically with the ALE approach, by extending the analysis rigorously.

# Appendix A

# Preliminaries

## A.1 Artificial Viscosity

### A.1.1 Artificial Viscosity 1D

The 1D monotonic limiting artificial viscosity is developed in [16]. The monotonic velocity gradients, $\Delta u$, are calculated for the element under consideration and its neighbouring elements, as seen in Figure A.1.

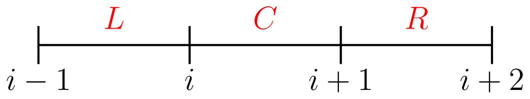


Figure A.1: 1D Artificial Viscosity Stencil

$$\left(\frac{\partial u}{\partial x}\right)_L = \frac{u_i - u_{i-1}}{x_i - x_{i-1}} \tag{A.1}$$

$$\left(\frac{\partial u}{\partial x}\right)_C = \frac{u_{i+1} - u_i}{x_{i+1} - x_i} \tag{A.2}$$

$$\left(\frac{\partial u}{\partial x}\right)_R = \frac{u_{i+2} - u_{i+1}}{x_{i+2} - x_{i+1}} \tag{A.3}$$

The left and right ratios of the velocity gradient are,

$$R_L = \frac{\left(\frac{\partial u}{\partial x}\right)_L}{\left(\frac{\partial u}{\partial x}\right)_C} \quad (A.4)$$

$$R_R = \frac{\left(\frac{\partial u}{\partial x}\right)_R}{\left(\frac{\partial u}{\partial x}\right)_C} \quad (A.5)$$

the limiter is

$$\Phi = max(0, min(\frac{1}{2}(R_L + R_R), 2R_L, 2R_R, 1)), \quad (A.6)$$

the 1D monotonic artificial viscosity for element $C$ is

$$q = c_q\rho|\Delta u|^2(1 - \Phi)^2 + c_l\rho c_s|\Delta u|(1 - \Phi) \quad (A.7)$$

by default the linear term $c_l$ is set to $\frac{1}{2}$ and, as explained in [16], the value $c_q = \frac{3}{4}$ is valid for most materials.

### A.1.2 Artificial Viscosity 2D

To extend the above to 2D, it is not obvious, as explained in [10]. However, a method for 2D is given in [89], by using an operator splitting technique. The 2D artificial viscosity looks at the four edge artificial viscosities, surrounding an element $C$, see Figure A.2.

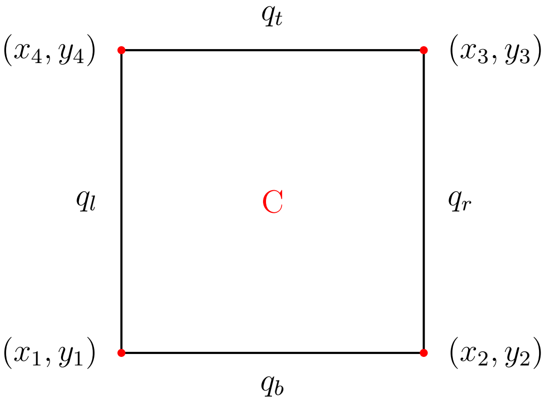


Figure A.2: 2D Artificial Viscosity in an Element C

From Figure A.2, $q_t$ and $q_b$ are associated with compressions in the logical mesh direction and $q_l$ and $q_r$ are orthogonal to $q_t$ and $q_b$.

In order to calculate the edge artificial viscosities, we define the appropriate direction vector for he horizontal and vertical logical mesh, known as mesh legs. Each edge leg has a length proportional to their perpendicular sides, such that,

$$\begin{aligned} Lhor_y &= -(x_4 + x_3 - x_2 - x_1) & \text{(A.8)} \\ Lhor_x &= (y_4 + y_3 - y_2 - y_1) & \text{(A.9)} \\ Lver_y &= (x_3 + x_2 - x_4 - x_1) & \text{(A.10)} \\ Lver_x &= -(y_3 + y_2 - y_4 - y_1), & \text{(A.11)} \end{aligned}$$

the area weighted length scales are then calculated as

$$\begin{aligned} \Delta x_{tb} &= \frac{area}{|\overline{Lhor}|} & \text{(A.12)} \\ \Delta x_{lr} &= \frac{area}{|\overline{Lver}|}, & \text{(A.13)} \end{aligned}$$

where, $\overline{Lhor} = (Lhor_x, Lhor_y)$ and $\overline{Lver} = (Lver_x, Lver_y)$. The velocity gradient for a cell edge is calculated by projecting the difference between nodal velocities of that edge onto the mesh leg, such that,

$$\begin{aligned} \frac{\Delta u_b}{\Delta x_b} &= \frac{\overline{Lhor} \cdot (\overline{u}_2 - \overline{u}_1)}{area} & \text{(A.14)} \\ \frac{\Delta u_t}{\Delta x_t} &= \frac{\overline{Lver} \cdot (\overline{u}_4 - \overline{u}_1)}{area} & \text{(A.15)} \\ \frac{\Delta u_l}{\Delta x_l} &= \frac{\overline{Lhor} \cdot (\overline{u}_3 - \overline{u}_4)}{area} & \text{(A.16)} \\ \frac{\Delta u_r}{\Delta x_r} &= \frac{\overline{Lver} \cdot (\overline{u}_3 - \overline{u}_2)}{area}, & \text{(A.17)} \end{aligned}$$

where $\overline{u}_n$ is the velocity vector at each element node, $n = 1, 2, 3, 4$, in the horizontal and vertical logical mesh directions, seen in Figure A.3.

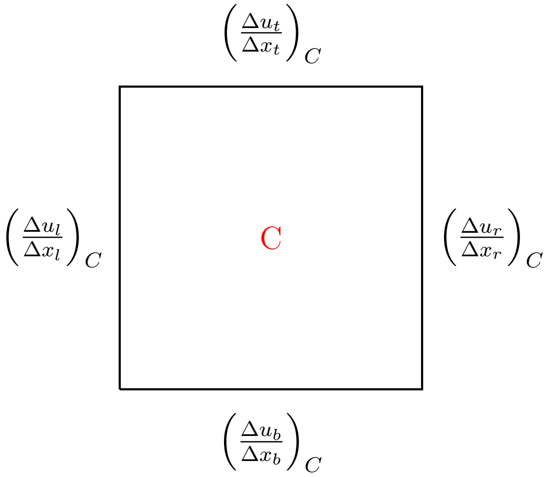

$\left(\frac{\Delta u_t}{\Delta x_t}\right)_C$
$\left(\frac{\Delta u_l}{\Delta x_l}\right)_C$
C
$\left(\frac{\Delta u_r}{\Delta x_r}\right)_C$
$\left(\frac{\Delta u_b}{\Delta x_b}\right)_C$

Figure A.3: 2D Velocity Gradients for Element C used in Velocity Gradient Ratios

Velocity gradient ratios are calculated, see in eqn. (A.4) and eqn. (A.5) for the 2D case, the limiter is used from eqn. (A.6). Second order accuracy can usually be achieved by only limiting the linear term. However, Christensen [16], suggests that it's beneficial to also limit the quadratic term, such that

$$|\Delta u| \rightarrow |\Delta u|(1-\Phi) \qquad \text{for the linear term} \tag{A.18}$$

$$|\Delta u|^2 \rightarrow |\Delta u|^2(1-\Phi^2) \qquad \text{for the quadratic term.} \tag{A.19}$$

Before the edge artificial viscosities are calculated it is necessary to check if there is an indication of expansion, since artificial viscosity only acts in the direction of compression. During an expansion, the nodes move away from each other, this would result in a positive velocity gradient, during compression the nodes move in towards each other resulting in a negative velocity gradient. Therefore, we have

$$\text{if} \qquad \frac{\Delta u}{\Delta x} > 0 \qquad \text{then set} \qquad \frac{\Delta u}{\Delta x} = 0$$

this gives a non-zero value only when we are in the presence of shocks.

The edge artificial viscosities are given as a linear combination of the linear and quadratic terms where,

$$q_b = c_q\rho\left(\frac{\Delta u_b}{\Delta x_b}\Delta x_{tb}\right)^2(1-\Phi)^2 + c_l\rho c_s\left|\frac{\Delta u_b}{\Delta x_b}\Delta u_{tb}\right|(1-\Phi) \tag{A.20}$$

$$q_t = c_q\rho\left(\frac{\Delta u_t}{\Delta x_t}\Delta x_{tb}\right)^2(1-\Phi)^2 + c_l\rho c_s\left|\frac{\Delta u_b}{\Delta x_t}\Delta x_{tb}\right|(1-\Phi) \tag{A.21}$$

$$q_l = c_q\rho\left(\frac{\Delta u_l}{\Delta x_l}\Delta x_{lr}\right)^2(1-\Phi)^2 + c_l\rho c_s\left|\frac{\Delta u_l}{\Delta x_l}\Delta x_{lr}\right|(1-\Phi) \tag{A.22}$$

$$q_r = c_q\rho\left(\frac{\Delta u_r}{\Delta x_r}\Delta x_{lr}\right)^2(1-\Phi)^2 + c_l\rho c_s\left|\frac{\Delta u_r}{\Delta x_r}\Delta x_{lr}\right|(1-\Phi), \tag{A.23}$$

the gradient of $u$ is calculated from the required velocity gradient multiplied by the appropriate edge distance.

The edge artificial viscosities are taken as the average of the artificial viscosities in the same logical directions, where

$$q_{hor} = \frac{1}{2}(q_t + q_b) \tag{A.24}$$

$$q_{ver} = \frac{1}{2}(q_l + q_r), \tag{A.25}$$

the edge artificial are then summed over an element to give a scalar result for the element artificial viscosity, resulting in

$$q = q_{hor} + q_{ver}. \tag{A.26}$$

The artificial viscosity now has all the required desirable features, it is staggered, monotonic, optimally reduces dissipation and improves stability.

## A.2 Time Stepping Control

During the Lagrangian step we advance the solution by one time step. The time step must be set in accordance to some tolerance for stability. The Courant-Friedrichs-Lewy (CFL) condition is used from [52], where

$$\Delta t = \frac{Cl}{c}; \qquad C \leq 1 \tag{A.27}$$

$C$ is the Courant number, usually taken to be $\frac{1}{3}$ or $\frac{1}{2}$, see [64], $l$ is the minimum distance in which a wave has to propagate until it crosses an element and $c$ is the sound speed. Therefore, the CFL restricts the timestep so that it is not greater than the time it takes a sound wave to cross any cell.

Since we use artificial viscosity, $q$, we must modify eqn. (A.27), see [37], where the sound speed is

$$c = \sqrt{c_s^2 + \frac{2q}{\rho}}, \tag{A.28}$$

$c_s$ is the element sound speed, therefore the time step becomes

$$\Delta t = \frac{Cl}{\sqrt{c_s^2 + \frac{2q}{\rho}}}. \tag{A.29}$$

The information required is the local sound speed and the minimum distance across a cell. In 1D, the distance is the length of the cell. However, it is slightly more difficult to obtain the minimum distance in 2D. For this project we take the minimum of four perpendicular projections from the mid-point along each side to the intersection with another face, as is done in [10]. This gives a good balance of stability and computational efficiency.

We usually favour $C = \frac{1}{2}$, since we use half of the perpendicular projection sides and the actual minimum distance for a wave to propagate across the cell may be less than this.

Since we have the local sound speed, $c_s$, and minimum distance across a cell, $l$, the timestep is calculated for each cell, and it is the minimum time step across the region that is used. The timestep is calculated at the beginning of each step.

# Appendix B

# Odd-Even Decoupling

To demonstrate the odd even decoupling ‘chequerboard’ phenomenon in 1D, consider the velocity $u(x)$ to be zero everywhere and the density $\rho$ and the pressure $p$ to have a ‘sawtooth’ type profile where

$$\rho_i = \begin{cases} 1 & \text{if } i = 1, 3, 5.. \\ 2 & \text{if } i = 2, 4, 6.. \end{cases} \tag{B.1}$$

as shown in [22] and demonstrated in Figure B.1.

It is observed that the even nodes have high pressures compared to the odd nodes. Physically it would be expected that the areas of higher pressure will smooth out the oscillations by moving to the regions of lower pressure. However, if we look at the Lagrangian momentum equation in 1D,

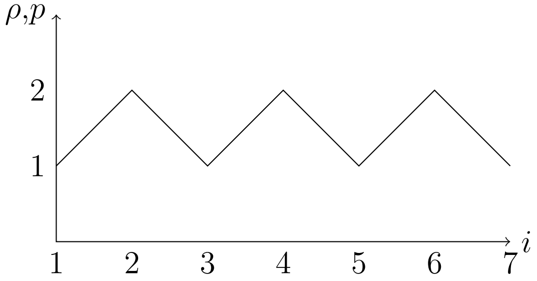


Figure B.1: Sawtooth $\rho$ and $p$

$$m\frac{\partial u}{\partial t} = -p_\xi,$$

when discretised in space using cell centered methods, we obtain Figure B.2. The discretisation becomes

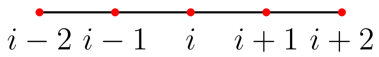


Figure B.2: Nodal positions of $\rho$, $p$ and $u$

$$m_i \frac{du_i}{dt} = -\frac{p_{i+1} - p_i + (p_i - p_{i-1})}{2\Delta\xi} = -\frac{p_{i+1} - p_{i-1}}{2\Delta\xi} = 0$$

since $p_{i+1} = p_{i-1}$. This is what is known as *odd-even decoupling*. Since we are solving for node $i$ but the derivative only consists of information from the neighbouring nodes $i-1$ and $i+1$. The update of the momentum is zero, we are left with an unphysical effect where the oscillations in the pressure are not smoothed out.

However, with a staggered mesh where the velocities and pressures are stored at different locations, see Figure B.3

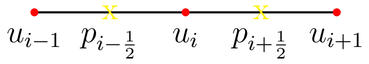


Figure B.3: Nodal positions of $u$ and $p$

the momentum equation now results in

$$m_i \frac{du_i}{dt} = -\frac{p_{i+\frac{1}{2}} - p_{i-\frac{1}{2}}}{\Delta\xi},$$

where the pressure smoothes out.

This argument can also be applied to $\rho$, $\varepsilon$, $m$ and $V$, where these variables are held at the cell centre.

# Appendix C

# Verification of the Error Estimate

Since we are using Taylor expansions for the error estimate, the data is required to be smooth to enable us to use order arguments. We must verify the estimation holds for data with steep gradients, as mentioned in the Introduction and Chapter 2.

To do so, we reduce the system of two equations to the wave equation. We can then choose the initial data to represent a smooth curve which can be modified for curves with different gradients. By doing this we can determine how steep our data can be before our approximated error is not valid.

Using the idea from [70], we reduce the system of two equations to the wave equation, if we say

$$f = x_{\xi\xi}. \tag{C.1}$$

Using the system of two equations in Lagrangian form we have

$$\frac{\partial x}{\partial t} = u \tag{C.2}$$

$$\frac{\partial u}{\partial t} = f. \tag{C.3}$$

We now use eqn. (C.1), and reduce the system of equations to

$$u_t = f = x_{\xi\xi}$$

and since

$$x_t = u$$

it follows that

$$x_{tt} = u_t = f = x_{\xi\xi},$$

so $x$ satisfies the general wave equation.
For the momentum equation in eqn. (C.3), using eqn. (C.1) we have

$$\begin{aligned} u_t &= x_{\xi\xi} && \text{(C.4)} \\ \Rightarrow u_{tt} &= x_{\xi\xi t}, && \text{(C.5)} \end{aligned}$$

thus using eqn. (C.2),

$$\begin{aligned} x_{t\xi} &= u_{\xi} && \text{(C.6)} \\ \Rightarrow x_{t\xi\xi} &= u_{\xi\xi}, && \text{(C.7)} \end{aligned}$$

therefore,

$$u_{tt} = u_{\xi\xi}.$$

Hence, $u$ also satisfies the standard wave equation.
D'Alembert's formula expresses the general solution to the wave equation i.e.

$$x(\xi) = F(\xi - t) + G(\xi + t), \tag{C.8}$$

where $F$ and $G$ are determined from initial data.
As a simple test, we restrict this verification to the semi-discrete in time error estimate for density only. Hence, we can determine the exact solution for $\rho$ for the wave equation. We have

$$\rho = \frac{m}{x_{\xi}},$$

where

$$x_{\xi} = F_{\xi} + G_{\xi}.$$

The density becomes

$$\rho = \frac{m}{F_{\xi} + G_{\xi}}. \tag{C.9}$$

We wish to choose initial data which exhibits steep solutions. Let

$$F(\nu) = \nu + \frac{e^{\beta\nu}}{\beta} = G(\nu). \tag{C.10}$$

When using $\xi - t$ and $\xi + t$ for the functions $F$ and $G$ respectively, this becomes

$$\begin{aligned} F(\xi - t) &= \xi - t + \frac{e^{\beta(\xi - t)}}{\beta} && \text{(C.11)} \\ G(\xi + t) &= \xi + t + \frac{e^{\beta(\xi + t)}}{\beta}, && \text{(C.12)} \end{aligned}$$

where $\beta$ is an arbitrary constant which can be altered to vary the gradient of the wave. We have the solutions

$$\begin{aligned} x(\xi, t) &= 2\xi + \frac{e^{\beta(\xi - t)}}{\beta} + \frac{e^{\beta(\xi + t)}}{\beta} && \text{(C.13)} \\ u(\xi, t) = x_t &= e^{\beta(\xi + t)} - e^{\beta(\xi - t)} && \text{(C.14)} \\ \rho(\xi, t) &= \frac{c(\xi)}{2 + e^{\beta(\xi - t)} + e^{\beta(\xi + t)}}. && \text{(C.15)} \end{aligned}$$

The initial data can be obtained by substituting $t = 0$ into the above solutions

$$\begin{aligned} u(\xi) &= 0 && \text{(C.16)} \\ \rho(\xi) &= \frac{c(\xi)}{2 + 2e^{\beta\xi}}. && \text{(C.17)} \end{aligned}$$

The initial data can now be used in testing how well the error estimate works when applied to steep solutions.
We demonstrate how the semi-discrete in time error estimate for density, derived in Chapter 3, can be applied for varying gradients of a wave at the same time, $t = 0.2$, and for the same initial mesh of 200 points.

The results verify the approximated error can be applied to shock like waves. Note that as the gradient of the wave gets steeper, the error and estimated error converge at the shock. We have demonstrated that the estimated error is large at the shock, therefore, proves to be a suitable method to apply as a refinement criteria.

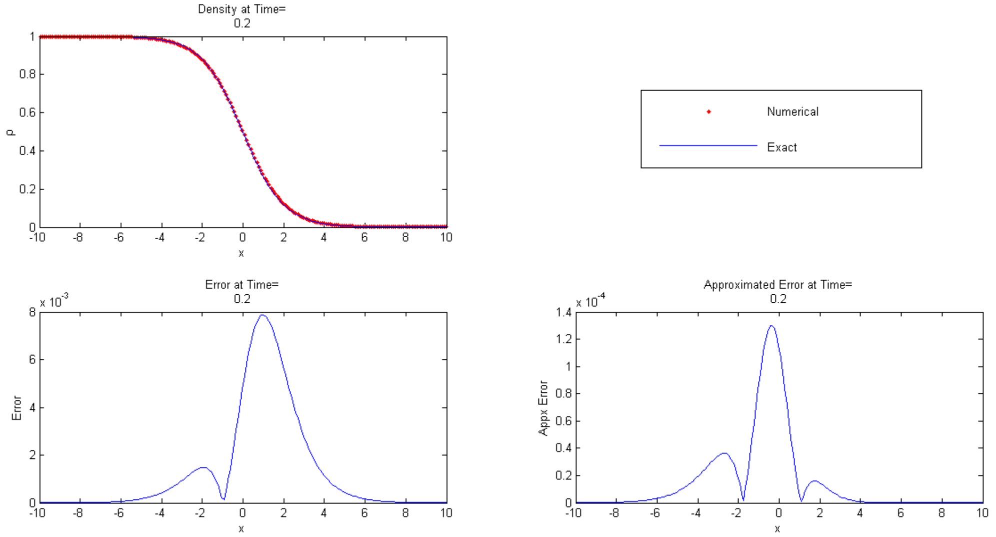


Figure C.1: $\beta = 1$

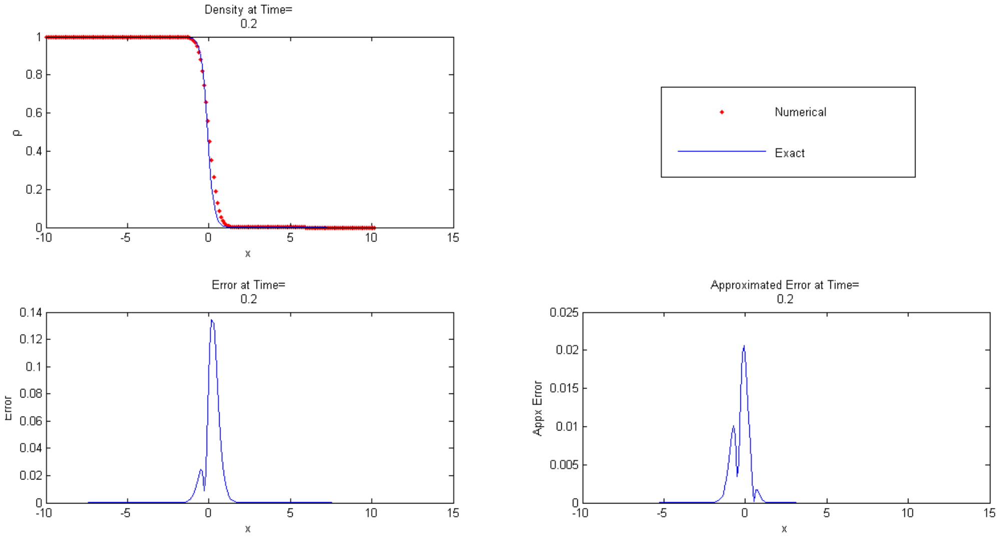


Figure C.2: $\beta = 5$

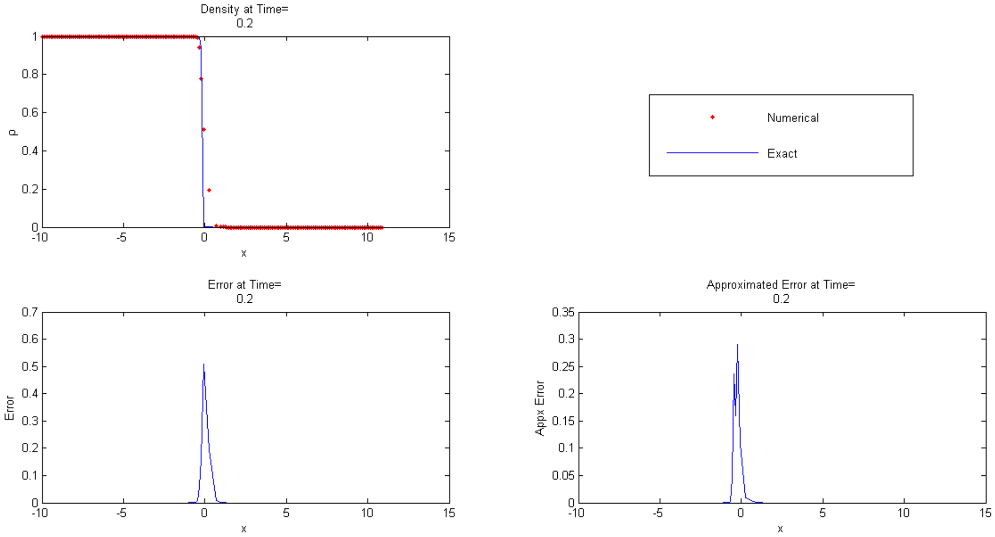


Figure C.3: $\beta = 15$

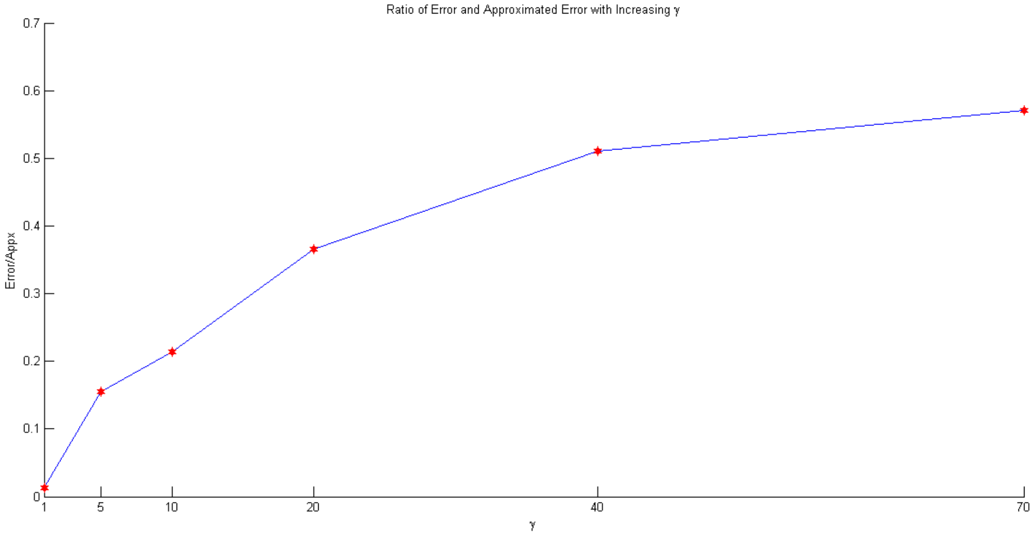


Figure C.4: Ratio of Error and Approximated Error

# Appendix D

# Difficulties at the Contact

From Chapter 6 it was not clear if we were detecting the contact in the error estimate. To analyse this we look at both temporal schemes from Sections 3.7.1 and 3.7.2, using a jump in density $\rho$ and energy $\epsilon$, with a constant velocity, $u$, and pressure, $p$.
We have

$$\varepsilon_l = 1 \qquad , \qquad \varepsilon_r = 0 \tag{D.1}$$

$$\rho_l = 1 \qquad , \qquad \rho_r = 0 \tag{D.2}$$

$$x_\xi = x_r - x_l \tag{D.3}$$

where $c_1$ and $p$ are constants.

## D.1 Scheme 1

Using Scheme 1 from Section 3.7.1 for the Euler equations we have

$$(x_l)^{n+\frac{1}{2}} = (x_l)^n + \frac{k}{2}u \qquad , \qquad (x_r)^{n+\frac{1}{2}} = (x_r)^n + \frac{k}{2}u \tag{D.4}$$

$$(x_\xi)^{n+\frac{1}{2}} = (x_r)^n + \frac{k}{2}u - \left((x_l)^n + \frac{k}{2}u\right) = (x_r)^n - (x_l)^n = (x_\xi)^n \,. \tag{D.5}$$

Since $(x_\xi)^{n+\frac{1}{2}} = (x_\xi)^n$ we have

$$(\rho_l)^{n+\frac{1}{2}} = \frac{m^n}{(\left((x_l)_\xi\right)^{n+\frac{1}{2}}} = \frac{m^n}{\left((x_l)_\xi\right)^n} = (\rho_l)^n \tag{D.6}$$

$$(\rho_r)^{n+\frac{1}{2}} = \frac{m^n}{\left((x_r)_\xi\right)^{n+\frac{1}{2}}} = \frac{m^n}{\left((x_r)_\xi\right)^n} = (\rho_r)^n\,. \tag{D.7}$$

Since $u$ is constant, $u_x = 0$, therefore,

$$\varepsilon^{n+\frac{1}{2}} = \varepsilon^n - \frac{kp^n\,(u_\xi)^n}{2\rho^n} \tag{D.8}$$

$$\text{therefore} \qquad (\varepsilon_l)^{n+\frac{1}{2}} = (\varepsilon_l)^n \qquad \text{and} \qquad (\varepsilon_r)^{n+\frac{1}{2}} = (\varepsilon_r)^n\,. \tag{D.9}$$

Now using $\rho^{n+\frac{1}{2}} = \rho^n$ and $\varepsilon^{n+\frac{1}{2}} = \varepsilon^n$ we obtain

$$p^{n+\frac{1}{2}} = (\gamma-1)\rho^{n+\frac{1}{2}}\varepsilon^{n+\frac{1}{2}} = (\gamma-1)\rho^n e^n = p^n = p. \tag{D.10}$$

We use

$$\begin{aligned}\bar{u} &= \frac{1}{2}(u^{n+1} + u^n)\\ &= \frac{1}{2}(u+u)\\ &= u.\end{aligned} \tag{D.11}$$

For the predictor step

$$(x_l)^{n+1} = (x_l)^n + ku \quad , \quad (x_r)^{n+1} = (x_r)^n + ku \tag{D.12}$$

$$(x_\xi)^{n+1} = (x_r)^n + ku - ((x_l)^n + ku) = (x_r)^n - (x_l)^n = (x_\xi)^n \tag{D.13}$$

since $(x_\xi)^{n+1} = (x_\xi)^n$ we have

$$(\rho_l)^{n+1} = \frac{m^n}{\left((x_l)_\xi\right)^{n+1}} = \frac{m^n}{((x_l)\,\xi)^n} = (\rho_l)^n \tag{D.14}$$

$$(\rho_r)^{n+1} = \frac{m^n}{\left((x_r)_\xi\right)^{n+1}} = \frac{m^n}{((x_r)\,\xi)^n} = (\rho_r)^n\,. \tag{D.15}$$

Since $\bar{u} = u$ is a constant, from eqn. (D.11), therefore, $\bar{u}_\xi = 0$, which is used in

$$\varepsilon^{n+1} = \varepsilon^n - \frac{kp^{n+\frac{1}{2}}\bar{u}_\xi}{\rho^{n+\frac{1}{2}}} \tag{D.16}$$

$$\text{to obtain} \qquad (\varepsilon_l)^{n+1} = (\varepsilon_l)^n, \qquad \text{and}\, (\varepsilon_r)^{n+\frac{1}{2}} = (\varepsilon_r)^n. \tag{D.17}$$

We now have $\rho^{n+1} = \rho^n$ and $\varepsilon^{n+1} = \varepsilon^n$

$$p^{n+1} = (\gamma - 1)\rho^{n+1}\varepsilon^{n+1} = (\gamma - 1)\rho^n e^n = p^n = p. \tag{D.18}$$

## D.2 Scheme 2

Using Scheme 2 from Section 3.7.2 we have

$$u_*^{n+1} = u^n - k\frac{(p_\xi)^n}{m^n}, \tag{D.19}$$

since $p$ is a constant, then $p_\xi = 0$, therefore

$$u_*^{n+1} = u^n = u \tag{D.20}$$

and

$$\begin{aligned} \bar{u}_* &= \frac{1}{2}\left(u_*^{n+1} + u^n\right) \\ &= \frac{1}{2}(u + u) \\ &= u. \end{aligned} \tag{D.21}$$

Using

$$((x_*)_l)^{n+1} = (x_l)^n + ku \qquad \text{and} \qquad ((x_*)_r)^{n+1} = (x_r)^n + ku$$

we obtain

$$\left((x_*)_\xi\right)^{n+1} = (x_l)^n + ku - ((x_r)^n + ku) = (x_r)^n - (x_l)^n = (x_\xi)^n. \tag{D.22}$$

Since $(x_\xi)^{n+1} = (x_\xi)^n$, we have

$$((\rho_*)_l)^{n+1} = \frac{m^n}{\left((x_l)_\xi\right)^{n+1}} = \frac{m^n}{\left((x_l)_\xi\right)^n} = (\rho_l)^n \tag{D.23}$$

$$((\rho_*)_r)^{n+1} = \frac{m^n}{\left((x_r)_\xi\right)^{n+1}} = \frac{m^n}{\left((x_r)_\xi\right)^n} = (\rho_r)^n. \tag{D.24}$$

Using $\bar{u} = u$, which is a constant, hence, $\bar{u}_\xi = 0$, along with using

$$(\varepsilon_*)^{n+1} = \varepsilon^n - \frac{kp^n (\bar{u}_*)_\xi}{(\rho_*)^{n+1}} \tag{D.25}$$

$$\text{we obtain} \qquad ((\varepsilon_*)_l)^{n+1} = (\varepsilon_l)^n, \qquad \text{and}\, ((\varepsilon_*)_r)^{n+1} = (\varepsilon_r)^n. \tag{D.26}$$

Since $(\rho_*)^{n+1} = \rho^n$ and $(\varepsilon_*)^{n+1} = \varepsilon^n$, we obtain

$$(p_*)^{n+1} = (\gamma - 1)(\rho_*)^{n+1}(\varepsilon_*)^{n+1} = (\gamma - 1)\rho^n e^n = p^n = p. \tag{D.27}$$

For the predictor step we have

$$u^{n+1} = u^n - k\frac{\left((p_*)_\xi\right)^{n+1}}{m^n}. \tag{D.28}$$

Since $(p_*)^{n+1} = p^n = p$ is a constant then $\left((p_*)_\xi\right)^{n+1} = 0$. Therefore,

$$u^{n+1} = u^n = u, \tag{D.29}$$

and

$$\begin{aligned} \bar{u} &= \frac{1}{2}\left(u^{n+1} + u^n\right) \\ &= \frac{1}{2}(u + u) \\ &= u. \end{aligned} \tag{D.30}$$

We have

$$(x_l)^{n+1} = x_l^n + ku \qquad \text{and} \qquad (x_r)^{n+1} = x_r^n + ku$$

$$\text{resulting in} \qquad (x_\xi)^{n+1} = x_l^n + ku - (x_r^n + ku) = x_r^n - x_l^n = (x_\xi)^n. \tag{D.31}$$

Since $(x_\xi)^{n+1} = (x_\xi)^n$ we have

$$\begin{aligned} ((\rho_l))^{n+1} &= \frac{m^n}{\Big((x_l)_\xi\Big)^{n+1}} = \frac{m^n}{\Big((x_l)_\xi\Big)^n} = (\rho)_l^n && \text{(D.32)} \\ ((\rho_r))^{n+1} &= \frac{m^n}{\Big((x_r)_\xi\Big)^{n+1}} = \frac{m^n}{\big((x_\xi)_r\big)^n} = (\rho_r)^n \,. && \text{(D.33)} \end{aligned}$$

We have $\bar{u} = u$, a constant, therefore, $\bar{u}_\xi = 0$, and

$$\begin{aligned} \varepsilon^{n+1} &= \varepsilon^n - \frac{kp^n \bar{u}_\xi}{\rho^{n+1}} && \text{(D.34)} \\ \text{therefore} \qquad \varepsilon_l^{n+1} &= \varepsilon_l^n, \qquad \text{and} \qquad \varepsilon_r^{n+1} = \varepsilon_r^n. && \text{(D.35)} \end{aligned}$$

Finally, using $\rho^{n+1} = \rho^n$ and $\varepsilon^{n+1} = \varepsilon^n$ we obtain

$$p^{n+1} = (\gamma - 1)\rho^{n+1}\varepsilon^{n+1} = (\gamma - 1)\rho^n e^n = p^n = p. \qquad \text{(D.36)}$$

We see that with the stated initial data, both schemes are exact at the contact, meaning that an error based on the difference between these two schemes will not be available here.

# Appendix E

# Shape Function Integrals

In Chapter 7, we have the weak form of the conservation equations in terms of integrals. The integrals are taken over the element after a mapping to the computational mesh, in $(\xi, \eta)$ space. The evaluation of integrals can be seen in [10, 38], and are reproduced here.

## E.1 Galerkin method

For the Galerkin Method in Chapter 7, we use basis functions which vanish on the boundaries. These bilinear functions are expressed as

$$N_1(\xi, \eta) = \frac{1}{4}(1-\xi)(1-\eta), \tag{E.1}$$

$$N_2(\xi, \eta) = \frac{1}{4}(1+\xi)(1-\eta), \tag{E.2}$$

$$N_3(\xi, \eta) = \frac{1}{4}(1+\xi)(1+\eta), \tag{E.3}$$

$$N_4(\xi, \eta) = \frac{1}{4}(1-\xi)(1+\eta). \tag{E.4}$$

In order to compute the integrals with respect to the computational mesh, in $(\xi, \eta)$ space, we have mapped from the physical mesh, in $(x, y)$ space. The mapping introduces a Jacobian matrix. To deduce the Jacobian we have the the quadrilateral vertices expressed as $\mathbf{r}_i = (x_i, y_i)(i = 1, 2, 3, 4)$, such that

$$\mathbf{r}(\xi, \eta) = N_1(\xi, \eta)\mathbf{r}_1 + N_2(\xi, \eta)\mathbf{r}_2 + N_3(\xi, \eta)\mathbf{r}_3 + N_4(\xi, \eta)\mathbf{r}_4. \tag{E.5}$$

This gives

$$\begin{aligned} x(\xi,\eta) &= \frac{1}{4}((x_1+x_2+x_3+x_4)+\xi(-x_1+x_2+x_3-x_4) \\ &+ \eta(-x_1-x_2+x_3+x_4)+\xi\eta(x_1-x_2+x_3-x_4)), \qquad \text{(E.6)} \\ y(\xi,\eta) &= \frac{1}{4}((y_1+y_2+y_3+y_4)+\xi(-y_1+y_2+y_3-y_4) \\ &+ \eta(-y_1-y_2+y_3+y_4)+\xi\eta(y_1-y_2+y_3-y_4)). \qquad \text{(E.7)} \end{aligned}$$

Define

$$\begin{aligned} A_1 &= \frac{1}{4}(-x_1+x_2+x_3-x_4), \qquad \text{(E.8)} \\ A_2 &= \frac{1}{4}(x_1-x_2+x_3-x_4), \qquad \text{(E.9)} \\ A_3 &= \frac{1}{4}(-x_1-x_2+x_3+x_4), \qquad \text{(E.10)} \\ A_4 &= \frac{1}{4}(x_1+x_2+x_3+x_4), \qquad \text{(E.11)} \\ B_1 &= \frac{1}{4}(-y_1+y_2+y_3-y_4), \qquad \text{(E.12)} \\ B_2 &= \frac{1}{4}(y_1-y_2+y_3-y_4), \qquad \text{(E.13)} \\ B_3 &= \frac{1}{4}(-y_1-y_2+y_3+y_4), \qquad \text{(E.14)} \\ B_4 &= \frac{1}{4}(y_1+y_2+y_3+y_4). \qquad \text{(E.15)} \end{aligned}$$

The Jacobian matrix is given by

$$\mathbf{J} = \begin{bmatrix} \frac{\partial x}{\partial \xi} & \frac{\partial y}{\partial \xi} \\ \frac{\partial x}{\partial \eta} & \frac{\partial y}{\partial \eta} \end{bmatrix}. \qquad \text{(E.16)}$$

Using eqn. (E.6) and eqn. (E.7) we obtain

$$\begin{aligned}
\frac{\partial x}{\partial \xi} &= \frac{1}{4}(-x_1+x_2+x_3-x_4)+\frac{\eta}{4}(x_1-x_2+x_3-x_4) && \text{(E.17)}\\
&= A_1+A_2\eta, && \text{(E.18)}\\
\frac{\partial x}{\partial \eta} &= \frac{1}{4}(-x_1-x_2+x_3+x_4)+\frac{\xi}{4}(x_1-x_2+x_3-x_4) && \text{(E.19)}\\
&= A_3+A_2\xi, && \text{(E.20)}\\
\frac{\partial y}{\partial \xi} &= \frac{1}{4}(-y_1+y_2+y_3-y_4)+\frac{\eta}{4}(y_1-y_2+y_3-y_4) && \text{(E.21)}\\
&= B_1+B_2\eta, && \text{(E.22)}\\
\frac{\partial y}{\partial \eta} &= \frac{1}{4}(-y_1-y_2+y_3+y_4)+\frac{\xi}{4}(y_1-y_2+y_3-y_4) && \text{(E.23)}\\
&= B_3+B_2\xi, && \text{(E.24)}
\end{aligned}$$

so the Jacobian becomes

$$\mathbf{J}=\begin{bmatrix} A_1+A_2\eta & B_1+B_2\eta \\ A_3+A_2\xi & B_3+B_2\xi \end{bmatrix}, \tag{E.25}$$

where

$$det\mathbf{J}=(A_1+A_2\eta)(B_3+B_2\xi)-(B_1+B_2\eta)(A_3+A_2\xi).$$

The inverse Jacobian now becomes

$$\mathbf{J^{-1}}=\begin{bmatrix} \frac{\partial\xi}{\partial x} & \frac{\partial\eta}{\partial x} \\ \frac{\partial\xi}{\partial y} & \frac{\partial\eta}{\partial y} \end{bmatrix}=\frac{1}{|\boldsymbol{J}|}\begin{bmatrix} \frac{\partial y}{\partial\eta} & \frac{-\partial y}{\partial\xi} \\ \frac{-\partial x}{\partial\eta} & \frac{\partial x}{\partial\xi} \end{bmatrix}=\frac{1}{|\boldsymbol{J}|}\begin{bmatrix} B_3+B_2\xi & -(B_1+B_2\eta) \\ -(A_3+A_2\xi) & A_1+A_2\eta \end{bmatrix}=\frac{\boldsymbol{C}}{|\boldsymbol{J}|}. \tag{E.26}$$

where $\boldsymbol{C}$ is the adjoint matrix.
From eqn. (7.41) we have

$$\boldsymbol{C}\nabla N=\begin{pmatrix} C_{11} & C_{12} \\ C_{21} & C_{22} \end{pmatrix}\begin{pmatrix} N_\xi \\ N_\eta \end{pmatrix}=\begin{pmatrix} C_{11}\frac{\partial N}{\partial\xi}+C_{12}\frac{\partial N}{\partial\eta} \\ C_{21}\frac{\partial N}{\partial\xi}+C_{22}\frac{\partial N}{\partial\eta} \end{pmatrix}. \tag{E.27}$$

The form of $C$ is demonstrated in eqn. (E.26).

We also have

$$\begin{aligned}
\frac{\partial N_1}{\partial \xi} &= \frac{1}{4}(\eta - 1) \quad , \quad \frac{\partial N_1}{\partial \eta} = \frac{1}{4}(\xi - 1), \\
\frac{\partial N_2}{\partial \xi} &= \frac{1}{4}(1 - \eta) \quad , \quad \frac{\partial N_2}{\partial \eta} = \frac{1}{4}(-\xi - 1), \\
\frac{\partial N_3}{\partial \xi} &= \frac{1}{4}(\eta + 1) \quad , \quad \frac{\partial N_3}{\partial \eta} = \frac{1}{4}(\xi + 1), \\
\frac{\partial N_4}{\partial \xi} &= \frac{1}{4}(-\eta - 1) \quad , \quad \frac{\partial N_4}{\partial \eta} = \frac{1}{4}(1 - \xi).
\end{aligned}$$

This leads to

$$\begin{aligned}
C_{11}\frac{\partial N_1}{\partial \xi} + C_{12}\frac{\partial N_1}{\partial \eta} &= \frac{1}{4}\left((\eta - 1)(B_3 + B_2\xi) - (\xi - 1)(B_1 + B_2\eta)\right), \\
C_{21}\frac{\partial N_1}{\partial \xi} + C_{22}\frac{\partial N_1}{\partial \eta} &= \frac{1}{4}\left((\eta - 1)(A_3 + A_2\xi) + (\xi - 1)(A_1 + A_2\eta)\right), \\
C_{11}\frac{\partial N_2}{\partial \xi} + C_{12}\frac{\partial N_2}{\partial \eta} &= \frac{1}{4}\left((1 - \eta)(B_3 + B_2\xi) + (\xi + 1)(B_1 + B_2\eta)\right), \\
C_{21}\frac{\partial N_2}{\partial \xi} + C_{22}\frac{\partial N_2}{\partial \eta} &= \frac{-1}{4}\left((1 - \eta)(A_3 + A_2\xi) + (\xi + 1)(A_1 + A_2\eta)\right), \\
C_{11}\frac{\partial N_3}{\partial \xi} + C_{12}\frac{\partial N_3}{\partial \eta} &= \frac{1}{4}\left((\eta + 1)(B_3 + B_2\xi) - (\xi + 1)(B_1 + B_2\eta)\right), \\
C_{21}\frac{\partial N_3}{\partial \xi} + C_{22}\frac{\partial N_3}{\partial \eta} &= \frac{-1}{4}\left((\eta + 1)(A_3 + A_2\xi) - (\xi + 1)(A_1 + A_2\eta)\right), \\
C_{11}\frac{\partial N_4}{\partial \xi} + C_{12}\frac{\partial N_4}{\partial \eta} &= \frac{-1}{4}\left((\eta + 1)(B_3 + B_2\xi) + (1 - \xi)(B_1 + B_2\eta)\right), \\
C_{21}\frac{\partial N_4}{\partial \xi} + C_{22}\frac{\partial N_4}{\partial \eta} &= \frac{1}{4}\left((\eta + 1)(A_3 + A_2\xi) + (1 - \xi)(A_1 + A_2\eta)\right).
\end{aligned}$$

The information is now available to evaluate the integrals from eqn. (7.41), which is

$$\int_{-1}^{1}\int_{-1}^{1} N_i|\boldsymbol{J}|d\xi d\eta.$$

For the nodes $i = 1, 2, 3, 4$ surrounding the element we have

$$\begin{aligned}
i &= 1 \qquad \int_{-1}^{1}\int_{-1}^{1} N_1|\boldsymbol{J}|d\xi d\eta = \frac{1}{9}[(3B_3 - B_2)(3A_1 - A_2) - (3A_3 - A_2)(3B_1 - B_2)],\\
i &= 2 \qquad \int_{-1}^{1}\int_{-1}^{1} N_2|\boldsymbol{J}|d\xi d\eta = \frac{1}{9}[(3B_3 + B_2)(3A_1 - A_2) - (3A_3 + A_2)(3B_1 - B_2)],\\
i &= 3 \qquad \int_{-1}^{1}\int_{-1}^{1} N_3|\boldsymbol{J}|d\xi d\eta = \frac{1}{9}[(3B_3 + B_2)(3A_1 + A_2) - (3A_3 + A_2)(3B_1 + B_2)],\\
i &= 4 \qquad \int_{-1}^{1}\int_{-1}^{1} N_4|\boldsymbol{J}|d\xi d\eta = \frac{1}{9}[(3B_3 - B_2)(3A_1 + A_2) - (3A_3 - A_2)(3B_1 + B_2)].
\end{aligned}$$

For the right hand side of equation (7.41), we evaluate the integral of the basis function at each node surrounding the element, which is differentiated with respect to the coordinate direction. We have,

$$\begin{aligned}
\int_{-1}^{1}\int_{-1}^{1}\left(C_{11}\frac{\partial N_1}{\partial\xi} + C_{12}\frac{\partial N_1}{\partial\eta}\right)d\xi d\eta &= (B_1 - B_3),\\
\int_{-1}^{1}\int_{-1}^{1}\left(C_{21}\frac{\partial N_1}{\partial\xi} + C_{22}\frac{\partial N_1}{\partial\eta}\right)d\xi d\eta &= (A_3 - A_1),\\
\int_{-1}^{1}\int_{-1}^{1}\left(C_{11}\frac{\partial N_2}{\partial\xi} + C_{12}\frac{\partial N_2}{\partial\eta}\right)d\xi d\eta &= (B_1 + B_3),\\
\int_{-1}^{1}\int_{-1}^{1}\left(C_{21}\frac{\partial N_2}{\partial\xi} + C_{22}\frac{\partial N_2}{\partial\eta}\right)d\xi d\eta &= -(A_3 + A_1),\\
\int_{-1}^{1}\int_{-1}^{1}\left(C_{11}\frac{\partial N_3}{\partial\xi} + C_{12}\frac{\partial N_3}{\partial\eta}\right)d\xi d\eta &= (B_3 - B_1),\\
\int_{-1}^{1}\int_{-1}^{1}\left(C_{21}\frac{\partial N_3}{\partial\xi} + C_{22}\frac{\partial N_3}{\partial\eta}\right)d\xi d\eta &= (A_1 - A_3),\\
\int_{-1}^{1}\int_{-1}^{1}\left(C_{11}\frac{\partial N_4}{\partial\xi} + C_{12}\frac{\partial N_4}{\partial\eta}\right)d\xi d\eta &= -(B_1 + B_3),\\
\int_{-1}^{1}\int_{-1}^{1}\left(C_{21}\frac{\partial N_4}{\partial\xi} + C_{22}\frac{\partial N_4}{\partial\eta}\right)d\xi d\eta &= (A_3 + A_1).
\end{aligned}$$

## E.2 Petrov Galerkin Method

We now evaluate the shape integral functions from Section 7.5.
The orthogonal basis functions here are,

$$\begin{aligned}
\psi_1 &= 4N_1 - 2N_2 + N_3 - 2N_4 = \frac{1}{4}(1-3\xi)(1-3\eta), \\
\psi_2 &= -2N_1 + 4N_2 - 2N_3 + N_4 = \frac{1}{4}(1+3\xi)(1-3\eta), \\
\psi_3 &= N_1 - 2N_2 + 4N_3 - 2N_4 = \frac{1}{4}(1+3\xi)(1+3\eta), \\
\psi_4 &= -2N_1 + N_2 - 2N_3 + 4N_4 = \frac{1}{4}(1-3\xi)(1+3\eta),
\end{aligned} \tag{E.28}$$

in terms of the bilinear basis functions $N_i$, which have previously been used. Now we have (with $\mathbf{r}_i = (x_i, y_i)(i = 1, 2, 3, 4)$)

$$\mathbf{r}(\xi,\eta) = \psi_1(\xi,\eta)\mathbf{r}_1 + \psi_2(\xi,\eta)\mathbf{r}_2 + \psi_3(\xi,\eta)\mathbf{r}_3 + \psi_4(\xi,\eta)\mathbf{r}_4, \tag{E.29}$$

this gives

$$\begin{aligned}
x(\xi,\eta) &= \frac{1}{4}((x_1+x_2+x_3+x_4) + 3\xi(-x_1+x_2+x_3-x_4) \\
&+ 3\eta(-x_1-x_2+x_3+x_4) + 9\xi\eta(x_1-x_2+x_3-x_4)) \\
y(\xi,\eta) &= \frac{1}{4}((y_1+y_2+y_3+y_4) + 3\xi(-y_1+y_2+y_3-y_4) \\
&+ 3\eta(-y_1-y_2+y_3+y_4) + 9\xi\eta(y_1-y_2+y_3-y_4)).
\end{aligned}$$

The Jacobian for the orthogonal basis function becomes

$$\mathbf{J} = \begin{bmatrix} \frac{\partial x}{\partial \xi} & \frac{\partial y}{\partial \xi} \\ \frac{\partial x}{\partial \eta} & \frac{\partial y}{\partial \eta} \end{bmatrix} = \frac{1}{|\boldsymbol{J}|}\begin{bmatrix} \frac{\partial y}{\partial \eta} & \frac{-\partial y}{\partial \xi} \\ \frac{-\partial x}{\partial \eta} & \frac{\partial x}{\partial \xi} \end{bmatrix} = \begin{bmatrix} 3A_1 + 9A_2\eta & 3B_1 + 9B_2\eta \\ 3A_3 + 9A_2\xi & 3B_3 + 9B_2\xi \end{bmatrix}, \tag{E.30}$$

and the required determinant is

$$|\boldsymbol{J}| = (3A_1 + 9A_2\eta)(3B_3 + 9B_2\xi) - (3B_1 + 9B_2\eta)(3A_3 + 9A_2\xi).$$

The inverse Jacobian now becomes

$$\mathbf{J^{-1}} = \begin{bmatrix} \frac{\partial\xi}{\partial x} & \frac{\partial\eta}{\partial x} \\ \frac{\partial\xi}{\partial y} & \frac{\partial\eta}{\partial y} \end{bmatrix} = \frac{1}{|\boldsymbol{J}|}\begin{bmatrix} 3B_3+9B_2\xi & -(3B_1+9B_2\eta) \\ -(3A_3+9A_2\xi) & 3A_1+9A_2\eta \end{bmatrix} = \frac{\boldsymbol{C}}{|\boldsymbol{J}|}. \tag{E.31}$$

We also have

$$\begin{aligned}
\frac{\partial\psi_1}{\partial\xi} &= \frac{1}{4}(9\eta-3) \quad , \quad \frac{\partial\psi_1}{\partial\eta} = \frac{1}{4}(9\xi-3), \\
\frac{\partial\psi_2}{\partial\xi} &= \frac{1}{4}(3-9\eta) \quad , \quad \frac{\partial\psi_2}{\partial\eta} = \frac{1}{4}(-9\xi-3), \\
\frac{\partial\psi_3}{\partial\xi} &= \frac{1}{4}(9\eta+3) \quad , \quad \frac{\partial\psi_3}{\partial\eta} = \frac{1}{4}(9\xi+3), \\
\frac{\partial\psi_4}{\partial\xi} &= \frac{1}{4}(-9\eta-3) \quad , \quad \frac{\partial\psi_4}{\partial\eta} = \frac{1}{4}(3-9\xi).
\end{aligned}$$

This leads to

$$\begin{aligned}
C_{11}\frac{\partial\psi_1}{\partial\xi}+C_{12}\frac{\partial\psi_1}{\partial\eta} &= \frac{1}{4}\left((9\eta-3)(3B_3+9B_2\xi)-(9\xi-3)(3B_1+9B_2\eta)\right), \\
C_{21}\frac{\partial\psi_1}{\partial\xi}+C_{22}\frac{\partial\psi_1}{\partial\eta} &= \frac{1}{4}\left((9\eta-3)(3A_3+9A_2\xi)+(9\xi-3)(3A_1+9A_2\eta)\right), \\
C_{11}\frac{\partial\psi_2}{\partial\xi}+C_{12}\frac{\partial\psi_2}{\partial\eta} &= \frac{1}{4}\left((3-9\eta)(3B_3+9B_2\xi)+(9\xi+3)(3B_1+9B_2\eta)\right), \\
C_{21}\frac{\partial\psi_2}{\partial\xi}+C_{22}\frac{\partial\psi_2}{\partial\eta} &= \frac{-1}{4}\left((3-9\eta)(3A_3+9A_2\xi)+(9\xi+3)(3A_1+9A_2\eta)\right), \\
C_{11}\frac{\partial\psi_3}{\partial\xi}+C_{12}\frac{\partial\psi_3}{\partial\eta} &= \frac{1}{4}\left((9\eta+3)(3B_3+9B_2\xi)-(9\xi+3)(3B_1+9B_2\eta)\right), \\
C_{21}\frac{\partial\psi_3}{\partial\xi}+C_{22}\frac{\partial\psi_3}{\partial\eta} &= \frac{-1}{4}\left((9\eta+3)(3A_3+9A_2\xi)-(9\xi+3)(3A_1+9A_2\eta)\right), \\
C_{11}\frac{\partial\psi_4}{\partial\xi}+C_{12}\frac{\partial\psi_4}{\partial\eta} &= \frac{-1}{4}\left((9\eta+3)(3B_3+9B_2\xi)+(3-9\xi)(3B_1+9B_2\eta)\right), \\
C_{21}\frac{\partial\psi_4}{\partial\xi}+C_{22}\frac{\partial\psi_4}{\partial\eta} &= \frac{1}{4}\left((9\eta+3)(3A_3+9A_2\xi)+(3-9\xi)(3A_1+9A_2\eta)\right).
\end{aligned}$$

We are now able to evaluate the integrals in eqn. (7.55). This is done by looking at one element, which surrounds node $i$.

We seek the solution to the integral on the left hand side in equation (4.14), which is

$$\int_{-1}^{1}\int_{-1}^{1}\psi_j N_i|\boldsymbol{J}|d\xi d\eta.$$

For the nodes $i = 1, 2, 3, 4$ surrounding the element we have

$$i = 1 \qquad \int_{-1}^{1}\int_{-1}^{1} \psi_1 N_1 |\boldsymbol{J}| d\xi d\eta = 18A_2B_1 - 18A_1B_2 + 9A_1B_3 - 9A_3B_1 - 18A_2B_3 + 18A_3B_2,$$

$$i = 2 \qquad \int_{-1}^{1}\int_{-1}^{1} \psi_2 N_2 |\boldsymbol{J}| d\xi d\eta = 18A_1B_2 - 18A_2B_1 + 9A_1B_3 - 9A_3B_1 - 18A_2B_3 + 18A_3B_2,$$

$$i = 3 \qquad \int_{-1}^{1}\int_{-1}^{1} \psi_3 N_3 |\boldsymbol{J}| d\xi d\eta = 18A_1B_2 - 18A_2B_1 + 9A_1B_3 - 9A_3B_1 + 18A_2B_3 - 18A_3B_2,$$

$$i = 4 \qquad \int_{-1}^{1}\int_{-1}^{1} \psi_4 N_4 |\boldsymbol{J}| d\xi d\eta = 18A_2B_1 - 18A_1B_2 + 9A_1B_3 - 9A_3B_1 + 18A_2B_3 - 18A_3B_2.$$

We now look at the boundary integral, $-p_{side} \oint \psi_i d\boldsymbol{S}$. We have the side differential vector $d\boldsymbol{S}$,

$$d\boldsymbol{S} = (d\xi, d\eta, -d\xi, -d\eta),$$

which is illustrated in Figure E.1. We integrate each nodal basis function $\psi_i$ over each side of the element, consider the element in Figure E.2.

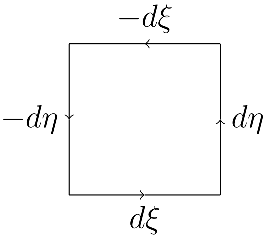


Figure E.1: Direction Vector

To evaluate the integral in the correct direction we have,

$$d\boldsymbol{S} = (cos\theta, sin\theta)dS,$$

where $S$ is the side we are solving for.

To determine $(cos\theta, sin\theta)$, consider the right angled triangle, below in Figure E.3. The hypotenuse of the right angled triangle which has been constructed, as the side length

$$L_1 = \sqrt{(x_2 - x_1)^2 + (y_2 - y_1)^2}.$$

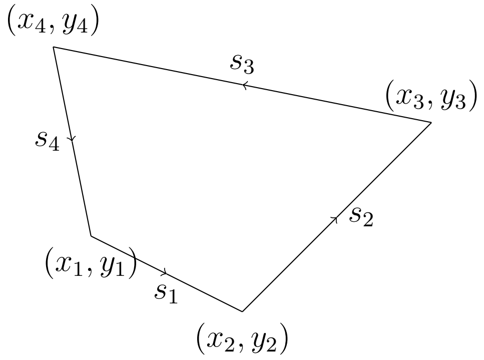


Figure E.2: Sides

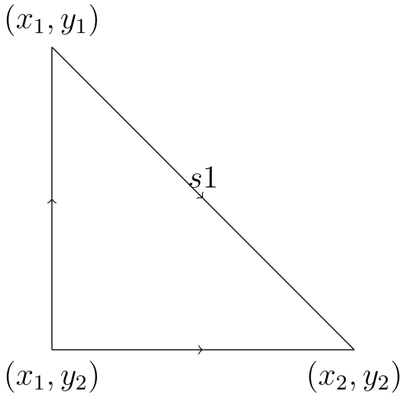


Figure E.3: Side Calculation

The adjacent and opposite lengths to $\theta$ are $y_1 - y_2$ and $x_2 - x_1$ respectively, resulting in

$$cos\theta = \frac{y_1 - y_2}{L_1} \qquad \text{and} \qquad sin\theta = \frac{x_2 - x_1}{L_1}. \tag{E.32}$$

Focusing on the component in the $x$ direction for the following sides, $cos\theta$ becomes $S_1 = \frac{y_1 - y_2}{L_1}$, $S_2 = \frac{y_2 - y_3}{L_2}$, $S_3 = \frac{y_3 - y_4}{L_3}$ and $S_4 = \frac{y_4 - y_1}{L_4}$.

We now transform our irregular quadrilateral onto a canonical square, as illustrated in Figure E.4. The integral dealing with the component in the $x$ direction is

$$-p_{side} \int_S \psi_i cos\theta dS$$

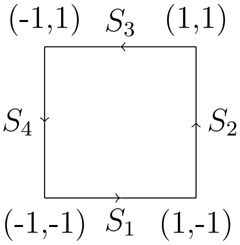


Figure E.4: Canonical Square

for one side, where $S$ is now a function of $\xi$ or $\eta$ therefore, for $\xi = 1$ or $\xi = -1$ we have

$$dS = \frac{dS}{d\eta}d\eta = \frac{L_i}{2}d\eta.$$

Where $\eta = 1$ or $\eta = -1$

$$dS = \frac{d}{d\xi}d\xi = \frac{L_i}{2}d\xi,$$

where $dS$ is the untransformed side length, $d\eta$ or $d\xi$ is the transformed side length and $L_i$ is 2 in this case. The integral becomes

$$-p_{side}cos\theta\frac{L_i}{2}\int_S \psi_i dr,$$

since $cos\theta$ and $L$ are not functions of $\xi$ and $\eta$ they come outside the integral, and $r$ is $\eta$ or $\xi$. We evaluate the integral for the sides of each element. The side $p$ terms are taken as a straight average from neighbouring elements. This results in a different discretisation to the Galerkin method as we now have a larger spread of pressure values.
For a node $i$, where the nodes are labelled in an anticlockwise manner, we require the nodes as illustrated in Figure E.5.

The element numbering is illustrated in Figure E.6.
On the sides $S_1$, $S_2$, $S_3$ and $S_3$, we have $\eta = -1$, $\xi = 1$, $\eta = 1$ and $\xi = -1$ as constants on the sides respectively. For an element the respective side intervals are as follows:

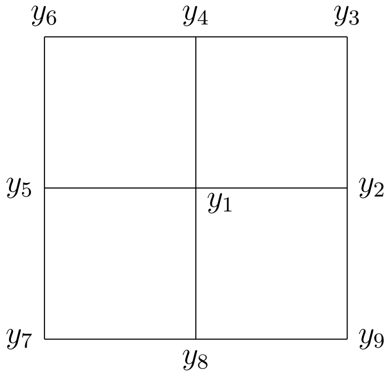


Figure E.5: Nodal labels about a point labelled 1

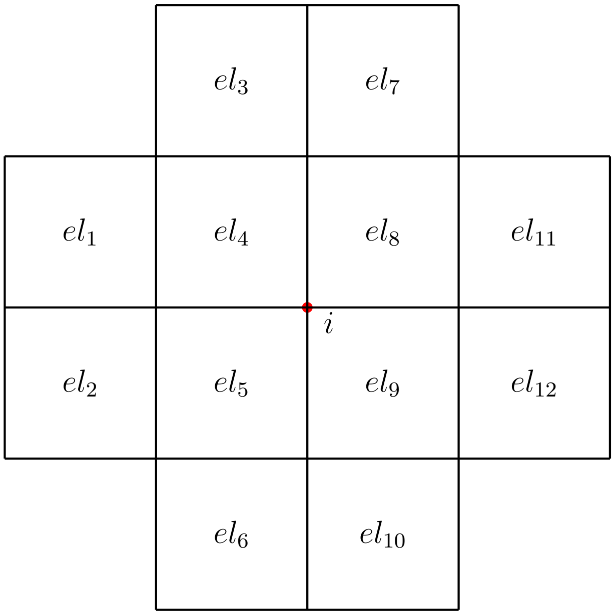


Figure E.6: Elements used to evaluate the side term at a node $i$

<u>Element 1 where $\psi_i = \psi_1$</u>

$$\begin{aligned}
\frac{(y1-y2)}{2}\frac{(p8+p9)}{2}\int_{-1}^{1}\psi_1(\xi,-1)d\xi &= \frac{1}{2}(y1-y2)(p8+p9),\\
\frac{(y2-y3)}{2}\frac{(p8+p11)}{2}\int_{-1}^{1}\psi_1(1,\eta)d\eta &= \frac{1}{4}(y3-y2)(p8+p11),\\
\frac{(y3-y4)}{2}\frac{(p8+p7)}{2}\int_{-1}^{1}\psi_1(\xi,1)d\xi &= \frac{1}{4}(y4-y3)(p8+p7),\\
\frac{(y4-y1)}{2}\frac{(p8+p4)}{2}\int_{-1}^{1}\psi_1(-1,\eta)d\eta &= \frac{1}{2}(y4-y1)(p8+p4).
\end{aligned}$$

Element 2 where $\psi_i = \psi_2$

$$\begin{aligned}
\frac{(y5-y1)}{2}\frac{(p4+p5)}{2}\int_{-1}^{1}\psi_2(\xi,-1)d\xi &= \frac{1}{2}(y5-y1)(p4+p5),\\
\frac{(y1-y4)}{2}\frac{(p8+p4)}{2}\int_{-1}^{1}\psi_2(1,\eta)d\eta &= \frac{1}{2}(y1-y4)(p8+p4),\\
\frac{(y4-y6)}{2}\frac{(p4+p3)}{2}\int_{-1}^{1}\psi_2(\xi,1)d\xi &= \frac{1}{4}(y6-y4)(p4+p3),\\
\frac{(y6-y5)}{2}\frac{(p1+p4)}{2}\int_{-1}^{1}\psi_2(-1,\eta)d\eta &= \frac{1}{4}(y5-y6)(p1+p4).
\end{aligned}$$

Element 3 where $\psi_i = \psi_3$

$$\begin{aligned}
\frac{(y7-y7)}{2}\frac{(p6+p5)}{2}\int_{-1}^{1}\psi_3(\xi,-1)d\xi &= \frac{1}{4}(y8-y7)(p6+p5),\\
\frac{(y8-y1)}{2}\frac{(p5+p9)}{2}\int_{-1}^{1}\psi_3(1,\eta)d\eta &= \frac{1}{2}(y8-y1)(p5+p9),\\
\frac{(y1-y5)}{2}\frac{(p5+p4)}{2}\int_{-1}^{1}\psi_3(\xi,1)d\xi &= \frac{1}{2}(y1-y5)(p5+p4),\\
\frac{(y5-y7)}{2}\frac{(p5+p2)}{2}\int_{-1}^{1}\psi_3(-1,\eta)d\eta &= \frac{1}{4}(y7-y5)(p5+p2).
\end{aligned}$$

Element 4 where $\psi_i = \psi_4$

$$\begin{aligned}
\frac{(y8-y9)}{2}\frac{(p10+p9)}{2}\int_{-1}^{1}\psi_4(\xi,-1)d\xi &= \frac{1}{4}(y9-y8)(p10+p9),\\
\frac{(y9-y2)}{2}\frac{(p9+p12)}{2}\int_{-1}^{1}\psi_4(1,\eta)d\eta &= \frac{1}{4}(y2-y9)(p9+p12),\\
\frac{(y2-y1)}{2}\frac{(p8+p9)}{2}\int_{-1}^{1}\psi_4(\xi,1)d\xi &= \frac{1}{2}(y2-y1)(p8+p9),\\
\frac{(y1-y8)}{2}\frac{(p9+p5)}{2}\int_{-1}^{1}\psi_4(-1,\eta)d\eta &= \frac{1}{2}(y1-y8)(p9+p5).
\end{aligned}$$

For the right hand side of eqn. (7.55), we evaluate the integral of the basis function at each node surrounding the element, which is then differentiated with respect to the coordinate direction, resulting in the following

$$
\begin{aligned}
\int_{-1}^{1}\int_{-1}^{1}\left(C_{11}\frac{\partial\psi_1}{\partial\xi}+C_{12}\frac{\partial\psi_1}{\partial\eta}\right)d\xi d\eta &= 9(B_1-B_3),\\
\int_{-1}^{1}\int_{-1}^{1}\left(C_{21}\frac{\partial\psi_1}{\partial\xi}+C_{22}\frac{\partial\psi_1}{\partial\eta}\right)d\xi d\eta &= 9(A_3-A_1),\\
\int_{-1}^{1}\int_{-1}^{1}\left(C_{11}\frac{\partial\psi_2}{\partial\xi}+C_{12}\frac{\partial\psi_2}{\partial\eta}\right)d\xi d\eta &= 9(B_1+B_3),\\
\int_{-1}^{1}\int_{-1}^{1}\left(C_{21}\frac{\partial\psi_2}{\partial\xi}+C_{22}\frac{\partial\psi_2}{\partial\eta}\right)d\xi d\eta &= -9(A_3+A_1),\\
\int_{-1}^{1}\int_{-1}^{1}\left(C_{11}\frac{\partial\psi_3}{\partial\xi}+C_{12}\frac{\partial\psi_3}{\partial\eta}\right)d\xi d\eta &= 9(B_3-B_1),\\
\int_{-1}^{1}\int_{-1}^{1}\left(C_{21}\frac{\partial\psi_3}{\partial\xi}+C_{22}\frac{\partial\psi_3}{\partial\eta}\right)d\xi d\eta &= 9(A_1-A_3),\\
\int_{-1}^{1}\int_{-1}^{1}\left(C_{11}\frac{\partial\psi_4}{\partial\xi}+C_{12}\frac{\partial\psi_4}{\partial\eta}\right)d\xi d\eta &= -9(B_1+B_3),\\
\int_{-1}^{1}\int_{-1}^{1}\left(C_{21}\frac{\partial\psi_4}{\partial\xi}+C_{22}\frac{\partial\psi_4}{\partial\eta}\right)d\xi d\eta &= 9(A_3+A_1).
\end{aligned}
$$

# Bibliography

[1] F. Addessio, J. Baumgardner, and J. Dukowicz et al. *CAVEAT: A computer code for uid dynamics problems with large distortion and internal split*, (1992). Los Alamos Technical Report.

[2] M. Aftosmis. Upwind method for simulation of viscous flow on adaptively refined meshes. *AIAA Journal*, **32**(2):268–277, (1994).

[3] M. J Aftosmis and M. J Berger. Multilevel error estimation and adaptive $h$-refinement for cartesian meshes with embedded boundaries. *AIAA Aerospace Sciences Meeting 2002-0863*, (2002).

[4] A. A. Amsden, H. M. Ruppel, and C. W. Hirt. *SALE: A Simplified ALE Computer Program for Fluid Flow at All Speeds*, (1980). Los Alamos Scientific Laboratory 8095.

[5] A.A. Amsden and H. Ruppel. *SALE-3D, A Simplified ALE Computer Program for Calculating Three-dimensional Fluid Flow.* The Commission, (1981). U.S. Nuclear Regulatory Commission. Division of Reactor Safety Research and Los Alamos National Laboratory.

[6] R. W. Anderson and N.S. Elliott *et al.* An Arbitrary Lagrangian-Eulerian method with adaptive mesh refinement for the solution of the Euler equations. *Journal of Computational Physics*, **199**:598–617, (2004).

[7] T. Apel, S. Groman, P. K. Jimack, and A. Meyer. A new methodology for anisotropic refinement based upon error gradients. *Applied Numerical Mathematics*, **50**:329–341, (2004).

[8] I. Babuška and C. Rheinboldt. A posteriori error estimates for the finite element method. *Int. J. Numer. Methods Engrg*, **12**:1597–1615, (1978).

[9] I. Babuška and C. Rheinboldt. Adaptive approaches and reliability estimates in finite element analysis. *Comput. Methods Appl. Mech. Engrg*, **17-18**(3):519–540, (1979).

[10] A. Barlow. *An adaptive multi-material Arbitrary Lagrange Eulerian algorithm for computational shock hydrodynamics*, (2002). Ph.D. Thesis. Swansea University. U.K.

[11] M. J. Berger and P. Colella. Local adaptive mesh refinement for shock hydrodynamics. *Journal of Computational Physics*, **82**:64–84, (1989).

[12] J. Breil, T. Harribey, P-H. Maire, and M. Shashkov. A multi-material ReALE method with mof interface reconstruction. *Computers Fluids*, **83**:115 – 125, (2013).

[13] J. E. Castillo, J. M. Hyman, and M. J. Shashkov. The sensitivity and accuracy of fourth order finite-difference schemes on nonuniform grids in one dimension. *Computers math. Applic.*, **30**:41–55, (1995).

[14] J. E. Castillo, J. M. Hyman, S. Steinberg, and M. J. Shashkov. *High-Order Mimetic Finite-Difference Schemes on Nonuniform Grids*, (1996). Proceedings of the third International Conference on Spectral and High Order Methods, Houston.

[15] J. Cheng, C. W. Shu, and Q. Zeng. *A conservative Lagrangian scheme for solving compressible fluid flows with multiple internal energy equations*, (2011). Scientific Computing Group, Brown University 2011-5.

[16] R. B. Christensen. Godunov methods on a staggered mesh. An improved artificial viscosity, (1990). Lawrence Livermore National Laboratory. UCRL-JC-105269 Preprint.

[17] C. Cole and J.E. Jones. On the stability of concentric spherical clouds of bipolar gaseous ions. *Journal of Electrostatics*, **56**(2):155 – 182, (2002).

[18] P. W. Cooper. *Explosives Engineering.* Wiley VCH, (1996). ISBN: 978-0-471-18636-6.

[19] R. Courant. Variational methods for the solution of problems of equilibrium and vibrations. *Bull. Amer. Math. Soc.*, **49**(1):1–23.

[20] J. Dannenhoffer and J. Baron. Grid adaption for the 2D Euler equations. *AIAA Paper 85-0484l*, pages 1–11, (1985).

[21] A. Dobre and D. Ramtal. *Physics for Flash Games, Animation, and Simulations.* Friends of Ed Series. Apress, (2011). ISBN: 9781430236740.

[22] C. P. Dullemond. *Lecture on: Numerical Fluid Dynamics*, (2008). URL: http://www.mpia.de/homes/dullemon/lectures/fluiddynamics08/chap_5_numhyd_classic.pdf [27 July 2013].

[23] K. Duraisamy, J.J. Alonso, and P. Chandrasekhar. Error estimation for high speed flows using continuous and discrete adjoints. *AIAA Paper 2010*, page 128, (2010).

[24] F. Ercolessi. *A Molecular Dynamics Primer* , (1997). URL: http://www.fisica.uniud.it/ ercolessi/md/md/node21.html [27 July 2013].

[25] M. D. Gee. Linear multistep methods for functional differential equations. *Mathematics of Computations*, **48**:633–649, (1987).

[26] S. K. Godunov. A difference scheme for numerical computation of discontinuous solution of hyperbolic equations. *Math. Sbornik*, **47**:271–306, (1959).

[27] D. F. Griffiths and D. J. Highman. *Numerical methods for Ordinary differential equations: Initial value problems*. Springer, 2010. ISBN: 978-0-85729-148-6.

[28] E. Hairer, C. Lubich, and G. Wanner. Geometric numerical integration illustrated by the Störmer/Verlet method. *Acta Numerica*, **12**:399–450, (2003).

[29] J. O. Hallquist. *DYNA 3D Course Notes*, (1987). Lawrence Livermore National Laboratory. Technical Report UCID-19899.

[30] F. Ham, F. S. Lien, and A. B. Strong. A cartesian grid method with transient anisotropic adaption. *Journal of Computational Physics*, **179**:469–494, (2002).

[31] F. Harlow and J. Welch. Numerical calculation of time-dependent viscous incompressible flow of fluid with free surface. *The physics of fluids.*, **8**:2182–2189, (1965).

[32] A. Harten. High resolution schemes for hyperbolic conservation laws. *J. Comput. Phys*, **49**:357–293, (1983).

[33] A. Harten, B. Engquist, S. Osher, and S. R. Chakravarthy. Uniformly high order accurate essentially non-oscillatory schemes III. *J. Comput. Phys*, **71**:231–303, (1987).

[34] A. Harten and J. M. Hyman. Self adjusting grid methods for one-dimensional hyperbolic conservation laws. *J. Comput. Phys.*, **50**:235–269, (1983).

[35] R. Hartmann, J. Held, and T. Leicht. Adjoint-based error estimation and adaptive mesh refinement for the RANS and $k\omega$ turbulence model equations. *Journal of Computational Physics*, **230**:4268–4284, (2011).

[36] P. Henrici. *Discrete Variable Methods in Ordinary Differential Equations.* Wiley, (1962).

[37] D. L. Hicks. Stability analysis of WONDY for a special case of Maxwell's law. *Mech. Comps*, **32**:1123, (1978).

[38] E. Hinton and D. R. Owen. *An Introduction to Finite Element Computation.* Pineridge Press, (1979).

[39] C. Hirt and A. Amsden *et al.* An Arbitrary Lagrange-Eulerian computing method for all flow speeds. *Journal of Computational Physics*, **14**(3):227–253.

[40] J. K. Hunter. *UC Davis Mathematics: John K. Hunter*, (2006). URL: https://www.math.ucdavis.edu/ hunter/notes/euler.pdf [27 July 2013].

[41] D. L. Illman. *Pathbreakers*, (1996). URL: http://www.washington.edu/research/pathbreakers/1950b.html [27 July 2013].

[42] C. E. Anderson Jr. An overview of the theory of hydrocodes. *International Journal of Impact Engineering*, **5**(1-4):33 – 59, (1987).

[43] Y. Kallinderis and J. Baron. Adaption methods for a new Navier-Stokes algorithm. *AIAA Journal*, **27**(1):37–43, (1989).

[44] S. J. Kamkar, A. M. Wissink, V. Sankaran, and A. Jameson. Combined feature-driven Richardson based Adaptive Mesh Refinement for unsteady vortical flows. *AIAA*, **50**(12), (2012).

[45] J.R. Kamm and M.J. Shashkov. A pressure relaxation closure model for one-dimensional, two-material Lagrangian hydrodynamics based on the Riemann problem. *Communications in Computational Physics*, **7**:927–976, (2010).

[46] J.R. Kamm, M.J. Shashkov, J. Fung, A.K. Harrison, and T.R. Canfield. A comparative study of various pressure relaxation closure models for one-dimensional, two-material lagrangian hydrodynamics. *Int. J. Numer. Meth. Fluids*, **65**(11-12):1311–1324, (2010).

[47] S. Karni and A. Kurganov. Local error analysis for approximate solutions of hyperbolic conservation laws. *Advances in Computational mathematics*, **22**:79–99, (2005).

[48] W. Keats and F. Lien. Two-dimensional anisotropic cartesian mesh adaptation for the compressible Euler equations. *International Journal for Numerical Methods in Fluids*, **46**:1099–1125, (2004).

[49] A. Khokhlov. Fully threaded tree algorithms for adaptive refinement fluid dynamics. *Journal of Computational Physics*, **143**:519–543, (1998).

[50] A. Kurganov and E. Tadmoor. Solution of two-dimensional Riemann problems for gas dynamics without Riemann problem solvers. *Numerical Methods Partial Differential Equations*, **18**:584–604, (2002).

[51] R. Landshoff. *A numerical method for treating fluid flow in the presence of shocks*, (1955). Los Alamos Scientific Laboratory. LA-1930.

[52] R. J. LeVeque. *Numerical Methods for Conservation Laws.* Birkhauser Verlag, (1990).

[53] R. J. LeVeque. *Finite Volume Methods For Hyperbolic Problems.* Cambridge University Press, (2002).

[54] L. Yi Li, Y. Allaneau, and A. Jameson. Continuous adjoint approach for adaptive mesh refinement. *AIAA Computational Fluid Dynamics Conference 2011-3982*, (2011).

[55] S. Li and L. Petzold. Adjoint sensitivity analysis for time-dependent partial differential equations with adaptive mesh refinement. *Journal of Computational Physics*, **198**:310–325, (2004).

[56] F. Lien. A pressure based unstructured grid method for all speed flows. *International Journal for Numerical Methods in Fluids*, **33**:355–374, (2000).

[57] R. Liska and B. Wendroff. Comparison of several difference schemes in 1D and 2D test problems for the Euler equations. *SIAM J. Sci. Comput.*, **25**:995–1017, (2003).

[58] R. W. MacCormack. The effect of viscosity in hypervelocity impact cratering. *Journal of Spacecraft and Rockets*, **40**(5):757–763, (2003).

[59] P. H. Maire. A high-order cell-centered Lagrangian scheme for two-dimensional compressible fluid flows on unstructured mesh. *J. Comp. Phys*, **228**(7):2391–2425, (2009).

[60] L. E. Malvern. *Introduction to the Mechanics of a Continuous Medium.* Prentice-Hall, Englewood Cliffs, (1969).

[61] H. C. Martin. *Introduction to Matrix Methods of Structural Analysis.* McGraw-Hill, (1966).

[62] H. C. Martin and G. F. Carey. *Introduction to Finite Element Analysis.* McGraw-Hill, (1973).

[63] Y. Morinishi, T. S. Lund, O. V. Vasilyev, and P. Moin. Fully conservative higher order finite difference schemes for incompressible flow. *Journal of Computational Physics*, **143**:90–124, 1998.

[64] J. Morrell. A cell by cell anisotropic adaptive mesh ALE scheme for the numerical solution of the Euler equations, 2007. Ph.D. Thesis. University of Reading. U.K.

[65] J. Morrell, P. K. Sweby, and A. Barlow. A cell by cell anisotropic adaptive mesh ALE scheme for the numerical solution of the Euler equations. *J. Comp. Phys*, **226**, (2007).

[66] M. Nemec, M. J. Aftosmis, and M. Wintzer. Adjoint-based Adaptive Mesh Refinement for complex geometries. *$46^{th}$AIAA Aerospace Sciences Meeting*, (2008).

[67] S. V. Patankar and D. B. Spalding. A calculation procedure for heat, mass and momentum transfer in three-dimensional parabolic flows. *Int. J. Heat Mass Transfer*, **15**:1787–1806, (1972).

[68] R. B. Pember and R. W. Anderson. *A comparison of staggered-mesh Lagrange plus remap and cell-centred direct Eulerian Godunov schemes for Eulerian shock hydrodynamics*, (2000). Lawrence Livermore National Laboratory. Preprint UCRL-JC-139820.

[69] B. Perot. Conservation properties of unstructured staggered mesh schemes. *Journal of Computational Physics*, **159**:58–89, (2000).

[70] J. F. Price. *Lagrangian and Eulerian Representations of Fluid Flow: Kinematics and the Equations of Motion*, (2006). Woods Hole Oceanographic Institution.

[71] T. Richter. A *posteriori* error estimation and anisotropy detection with the dual-weighted residual method. *Int. J. Numer. Meth. Fluids*, **62**(1), (2009).

[72] R. D. Richtmyer and K. W. Morton. *Difference Methods for Initial-value Problems.* Wiley-Interscience, (1967).

[73] William J. Rider. Revisiting wall heating. *Journal of Computational Physics*, **162**(2):395 – 410, (2000).

[74] R. C. Ripley. *Unstructured-grid methods for numerical simulation of shock wave interactions with bodies and boundary layers*, (2002). University of Waterloo.

[75] J. Robinson. *Early FEM Pioneers.* Pitman Press, (1985).

[76] P. L. Roe. Approximate Riemann solvers, parameter vectors and difference schemes. *J. Comput. Phys.*, **43**:357–372, (1981).

[77] C. J. Roy. Review of discretization error estimators in scientific computing, (2010). *48th AIAA Aerospace Sciences Meeting Including the New Horizons Forum and Aerospace Exposition.*

[78] T. Schlick. *Molecular Modeling and Simulation: An Interdisciplinary Guide.* Springer, (2010).

[79] L. I. Sedov. *Similarity and dimensional methods in mechanics.* CPC, (1993).

[80] L. F. Shampine. Local extrapolation in solution of ordinary differential equations. *Mathematics of Computations*, **27**:91–97, (1973).

[81] C. W. Shu and S. Osher. Efficient implementation of essentially non-oscillatory shock capturing schemes. *J. Comput. Phys.*, **77**:439–471, (1988).

[82] J. Smoller. *Shock waves and reaction-diffusion equations.* Springer Verlag, (1983).

[83] G. Sod. A survey of several finite difference methods for systems of nonlinear hyperbolic conservation laws. *Journal of Computational Physics*, **27**:1–31, (1978).

[84] G. Strang. On the construction and comparison of difference schemes. *SIAM Journal on Numerical Analysis*, **5**(3):506–517, (1968).

[85] M. Sun. *Numerical and experimental studies of shock wave interaction with bodies*, (1998). Tohoku University.

[86] P. Sun, R. D. Russell, and J. Xu. A new adaptive local mesh refinement algorithm and its application on fourth order thin film flow problem. *Journal of Computational Physics*, **224**(2):1021 – 1048, (2007).

[87] E. Tadmor. Local error estimates for discontinuous solutions of nonlinear hyperbolic equations. *SIAM J. Numer. Anal.*, **28**:891–906, (1991).

[88] J. F. Thompson, Z. U. A. Warsi, and C. W. Mastin. *Numerical Grid Generation, Foundations and Applications*. North Holland, (1985). ISBN: 044400985X.

[89] R.E. Tipton. *CALE Lagrange Step*, (1990). Lawrence Livermore National Laboratory.

[90] F. Toro. *Riemann Solvers and Numerical Methods for Fluid Dynamics*. Springer, 1997.

[91] M. J. Turner, R. W. Clough, H. C. Martin, and L. J. Topp. Stiffness and deflection analysis of complex structures. *J. of Aero. Sci.*, **23**(9).

[92] J. van der Vegt and H. van der Ven. Discontinuous Galerkin finite element method with anisotropic local grid refinement for inviscid compressible flows. *Journal of Computational Physics*, **141**:46–77, (1998).

[93] D. A. Venditti and D.L. Darmofal. A multilevel error estimation and grid adaptive strategy for improving the accuracy of integral outputs, (2008). *14th Computational Fluid Dynamics Conference*. 3292. 28 June - 1 July, 1999. Norfolk, VA.

[94] L. Verlet. Computer 'experiments' on classical fluids. I. Thermodynamical properties of LennardJones molecules. *Physical review*, **159**:98–103, (1967).

[95] J. von Neumann and R. D Richtmyer. A method for the numerical calculations of hydrodynamical shocks. *Journal of Applied Physics*, **21**(232), (1950).

[96] E. Wendland and H. E. Schulz. Numerical experiments on mass lumping for the advection-diffusion equation. *Minerva*, **2**:227–233, (2002).

[97] K. S. Yee. Numerical solution of initial boundary value problems involving Maxwells equations in isotropic media. *IEE Trans. Antennas Propag*, **14**(3), (1966).

[98] Y. Q. Zeng and Q. H. Liu. A staggered grid nite difference method with perfectly matched layers for poroelastic wave equations. *J. Acoust. Soc. Am*, **109**(6):2571–2580, (2001).

[99] O. C. Zienkiewickz. The background of error estimation and adaptivity in finite element computations. *Comput. Methods Appl. Mech. Engrg*, **195**:207–213, (2006).

[100] O. C. Zienkiewicz and R. L. Taylor. *The Finite Element Method.* Butterworth-Heinemann, (2005).