

HIGH ORDER FINITE ELEMENTS FOR LAGRANGIAN  
COMPUTATIONAL FLUID DYNAMICS

A Thesis

Presented to

the Faculty of California Polytechnic State University

San Luis Obispo

In Partial Fulfillment

of the Requirements for the Degree

Master of Science in Aerospace Engineering

by

Truman E. Ellis

March 2010

© 2010

Truman E. Ellis

ALL RIGHTS RESERVED

## COMMITTEE MEMBERSHIP

TITLE: High Order Finite Elements for Lagrangian Computational Fluid Dynamics

AUTHOR: Truman E. Ellis

DATE SUBMITTED: March 2010

COMMITTEE CHAIR: Faisal Kolkailah, Ph.D.

COMMITTEE MEMBER: Robert Rieben, Ph.D.

COMMITTEE MEMBER: Tzanio Kolev, Ph.D.

COMMITTEE MEMBER: David Marshall, Ph.D.

COMMITTEE MEMBER: Alberto Jimenez, Ph.D.

## **Abstract**

High Order Finite Elements for Lagrangian Computational Fluid Dynamics

Truman E. Ellis

This is where the abstract goes.

## **Acknowledgements**

Thank you...

# Contents

<b>List of Tables</b>	<b>x</b>
<b>List of Figures</b>	<b>xi</b>
<b>Nomenclature</b>	<b>xvi</b>
<b>1 Introduction</b>	<b>1</b>
<b>2 Theoretical Discussion</b>	<b>4</b>
2.1 Euler Equations in a Lagrangian Frame . . . . .	4
2.2 A Brief Summary of the Finite Element Method . . . . .	6
2.2.1 The Weak Statement of the Problem . . . . .	6
2.2.2 Basis Functions . . . . .	6
2.3 Mesh . . . . .	8
2.3.1 Domain Decomposition . . . . .	8
2.3.2 Mesh Motion . . . . .	10
2.3.3 Jacobian of Transformation . . . . .	11
2.4 Semi-Discrete FEM Approximation . . . . .	15
2.4.1 Momentum Conservation . . . . .	15
2.4.2 Computing FEM Matrices . . . . .	17
2.4.3 Mass Conservation . . . . .	18
2.4.4 Energy Conservation . . . . .	20
2.4.5 Equation of State . . . . .	22
2.5 Fully-Discrete FEM Approximation . . . . .	23
2.5.1 Momentum Conservation . . . . .	23
2.5.2 Mass Conservation . . . . .	24

2.5.3	Energy Conservation . . . . .	24
2.5.4	Equation of State . . . . .	25
2.6	Tensor Artificial Viscosity . . . . .	25
2.7	Fully-Discrete Scheme . . . . .	27
<b>3</b>	<b>Implementation Details</b>	<b>29</b>
3.1	The Drivers . . . . .	32
3.2	Mesh Assembly and Variable Preallocation . . . . .	35
3.3	Initializing the Flow Field . . . . .	36
3.4	Solving the Momentum Equation . . . . .	37
3.5	Mesh Movement and Thermodynamic Update . . . . .	38
3.6	Plotting . . . . .	39
<b>4</b>	<b>Some Elements Considered</b>	<b>41</b>
4.1	Some Guidelines . . . . .	41
4.2	Convergence Tests . . . . .	42
4.2.1	The $\text{grad}(p)$ Test . . . . .	43
4.2.2	The $\text{div}(\text{grad}(V))$ Test . . . . .	44
4.3	The Acoustic Wave Test . . . . .	46
4.4	The $Q_1-Q_0$ Element . . . . .	47
4.4.1	Assets . . . . .	48
4.4.2	Liabilities . . . . .	48
4.4.3	Convergence . . . . .	49
4.4.4	Spurious Modes . . . . .	50
4.5	The $Q_1-P_{0r}$ Element . . . . .	51
4.5.1	Assets . . . . .	52
4.5.2	Liabilities . . . . .	52
4.5.3	Convergence . . . . .	52
4.5.4	Spurious Modes . . . . .	54
4.6	The $Q_1-\hat{P}_1$ Element . . . . .	54
4.7	The $Q_1-\hat{Q}_1$ Element . . . . .	55
4.7.1	Assets . . . . .	56

4.7.2	Liabilities . . . . .	56
4.7.3	Convergence . . . . .	58
4.7.4	Spurious Modes . . . . .	59
4.8	The $Q_1\text{-}\hat{Q}_2$ Element . . . . .	60
4.9	The $Q_2\text{-}Q_0$ Element . . . . .	61
4.9.1	Assets . . . . .	61
4.9.2	Liabilities . . . . .	61
4.9.3	Convergence . . . . .	62
4.10	The $Q_2\text{-}\hat{P}_1$ Element . . . . .	62
4.10.1	Assets . . . . .	62
4.10.2	Liabilities . . . . .	63
4.10.3	Convergence . . . . .	63
4.10.4	Spurious Modes . . . . .	65
4.11	The $Q_2\text{-}\hat{Q}_1$ Element . . . . .	65
4.11.1	Assets . . . . .	66
4.11.2	Liabilities . . . . .	67
4.11.3	Convergence . . . . .	67
4.11.4	Spurious Modes . . . . .	72
4.12	The $Q_2\text{-}\hat{Q}_2$ Element . . . . .	73
4.12.1	Assets . . . . .	74
4.12.2	Liabilities . . . . .	74
4.12.3	Convergence . . . . .	75
4.12.4	Spurious Modes . . . . .	77
4.13	Conclusions . . . . .	78
<b>5</b>	<b>Numerical Results</b>	<b>79</b>
5.1	The Sod Shock Tube . . . . .	79
5.2	The Noh Implosion . . . . .	83
5.2.1	Low Order $Q_1\text{-}Q_0$ . . . . .	85
5.2.2	Low Order $Q_1\text{-}\hat{Q}_1$ . . . . .	88
5.2.3	High Order $Q_2\text{-}\hat{Q}_1$ . . . . .	91
5.2.4	High Order $Q_2\text{-}\hat{Q}_2$ . . . . .	94

5.3	The Saltzman Piston . . . . .	97
5.3.1	Low Order $Q_1$ - $\hat{Q}_0$ . . . . .	98
5.3.2	Low Order $Q_1$ - $\hat{Q}_1$ . . . . .	101
5.3.3	High Order $Q_2$ - $\hat{Q}_1$ . . . . .	103
5.3.4	High Order $Q_2$ - $\hat{Q}_2$ . . . . .	105
5.4	The Sedov Blast Wave . . . . .	108
5.4.1	Low Order $Q_1$ - $\hat{Q}_0$ . . . . .	109
5.4.2	Low Order $Q_1$ - $\hat{Q}_1$ . . . . .	114
5.4.3	High Order $Q_2$ - $\hat{Q}_1$ . . . . .	117
5.4.4	High Order $Q_2$ - $\hat{Q}_2$ . . . . .	123
5.5	Computational Efficiency . . . . .	127
<b>6</b>	<b>Conclusion</b>	<b>129</b>
	<b>Bibliography</b>	<b>136</b>

# List of Tables

3.1	List of switches and knobs . . . . .	34
5.1	Sedov run times compared across the number of kinematic degrees of freedom. Time in seconds. . . . .	127
5.2	Noh run times compared across the number of kinematic degrees of freedom. Time in seconds. . . . .	128

# List of Figures

2.1 Examples of 2D basis functions on a reference zone: a standard $Q_1$ bi-linear function that interpolates at nodes ( <i>left</i> ), a $Q_2$ bi-quadratic function defined at the Gauss-Lobatto quadrature points ( <i>center</i> ) and a $Q_2$ bi-quadratic function defined at the Gauss-Legendre quadrature points ( <i>right</i> ). . . . . .	8
2.2 Domain decomposition for a finite element mesh . . . . .	9
2.3 An initial Cartesian mesh is continuously deformed according to the exact solution of the Sedov blast wave. . . . .	10
2.4 In a semi-discrete setting, a zone $\Omega_z(t)$ is reconstructed from the evolution of only a few of its points (particles) indicated by black dots. Shown are two specific choices corresponding to the traditional $Q_1$ zone ( <i>left</i> ) and a high order $Q_2$ zone with curvilinear boundaries ( <i>right</i> ). . . . . .	11
2.5 Example of a $Q_2$ bi-quadratic mapping from a reference zone ( <i>left</i> ) to a Lagrangian zone ( <i>right</i> ) defined by the locations of the 9 Lagrangian particles (black dots). . . . .	13
3.1 Workflow diagram . . . . .	31
3.2 Options for kinematic basis functions . . . . .	33
3.3 Options for thermodynamic basis functions . . . . .	33
3.4 Velocity node numbering . . . . .	35
4.1 Initial pressure field . . . . .	43
4.2 Series of refined distorted meshes . . . . .	44
4.3 Projected velocity vector field . . . . .	45
4.4 Traditional $Q_1$ - $Q_0$ acoustic wave with hourglass filter . . . . .	47

4.5	The traditional $Q_1 - \hat{Q}_0$ element . . . . .	48
4.6	Traditional $Q_1-Q_0$ acoustic wave without hourglass filter . . . . .	50
4.7	Hourglass/checkerboard modes on a patch of $Q_1-Q_0$ elements . . . . .	51
4.8	The $Q_1-\hat{P}_{0r}$ element . . . . .	52
4.9	Convergence of the $Q_2-\hat{Q}_1$ pressure field . . . . .	53
4.10	Convergence of $\vec{\nabla}p$ acceleration: $Q_1-P_{0r}$ versus $Q_1-Q_0$ on a Cartesian mesh . . . . .	53
4.11	$Q_1-P_{0r}$ acoustic wave without hourglass filter . . . . .	54
4.12	The $Q_1-\hat{P}_1$ element . . . . .	54
4.13	The $Q_1-\hat{Q}_1$ element . . . . .	55
4.14	Average pressure in each cell for $Q_1-\hat{Q}_1$ element for acoustic wave problem . . . . .	56
4.15	Spurious velocity modes present in $Q_1-\hat{Q}_1$ . . . . .	57
4.16	Convergence of $\vec{\nabla}p$ acceleration: $Q_1-\hat{Q}_1$ versus $Q_1-Q_0$ on a Cartesian mesh . . . . .	58
4.17	Convergence of $\vec{\nabla}p$ acceleration: $Q_1-\hat{Q}_1$ versus $Q_1-Q_0$ on a distorted mesh . . . . .	59
4.18	$Q_1-\hat{Q}_1$ acoustic wave without hourglass filter . . . . .	60
4.19	The $Q_1-\hat{Q}_2$ element . . . . .	60
4.20	The $Q_2-Q_0$ element . . . . .	61
4.21	The $Q_2-\hat{P}_1$ element . . . . .	62
4.22	Convergence of $\vec{\nabla}p$ acceleration: $Q_2-\hat{P}_1$ versus $Q_1-Q_0$ on a Cartesian mesh . . . . .	64
4.23	Convergence of $\vec{\nabla}p$ acceleration: $Q_2-\hat{P}_1$ versus $Q_1-Q_0$ on a distorted mesh . . . . .	64
4.24	$Q_2-\hat{P}_1$ acoustic wave without hourglass filter . . . . .	65
4.25	$Q_2-\hat{P}_1$ acoustic wave with hourglass filter . . . . .	65
4.26	The $Q_2-\hat{Q}_1$ element . . . . .	66
4.27	Convergence of the $Q_2-\hat{Q}_1$ pressure field . . . . .	68
4.28	Convergence of $\vec{\nabla}p$ acceleration: $Q_2-\hat{Q}_1$ versus $Q_1-Q_0$ on a Cartesian mesh . . . . .	68
4.29	Convergence of $\vec{\nabla}p$ acceleration: $Q_2-\hat{Q}_1$ versus $Q_1-Q_0$ on a distorted mesh . . . . .	69

4.30	Convergence of $\vec{\nabla}p$ acceleration: $Q_2\hat{Q}_1$ versus $Q_1-Q_1$ on a distorted mesh . . . . .	70
4.31	Convergence of the $Q_2\hat{Q}_1$ velocity projection vs $Q_1-Q_0$ . . . . .	70
4.32	Convergence of $\vec{\nabla} \cdot \vec{\nabla}V$ acceleration: $Q_2\hat{Q}_1$ versus $Q_1-Q_0$ on a Cartesian mesh . . . . .	71
4.33	Convergence of $\vec{\nabla} \cdot \vec{\nabla}V$ acceleration: $Q_2\hat{Q}_1$ versus $Q_1-Q_0$ on a distorted mesh . . . . .	71
4.34	$Q_2\hat{Q}_1$ acoustic wave without hourglass filter . . . . .	72
4.35	Hourglass/checkerboard modes on a patch of $Q_2\hat{Q}_1$ elements . . . . .	73
4.36	$Q_2\hat{Q}_1$ acoustic wave with hourglass filter . . . . .	73
4.37	Average pressure in each cell for $Q_2\hat{Q}_2$ element for acoustic wave problem . . . . .	75
4.38	Convergence of the $Q_2\hat{Q}_2$ pressure field . . . . .	76
4.39	Convergence of $\vec{\nabla}p$ acceleration: $Q_2\hat{Q}_2$ versus $Q_1-Q_1$ on a Cartesian mesh . . . . .	76
4.40	Convergence of $\vec{\nabla}p$ acceleration: $Q_2\hat{Q}_2$ versus $Q_1-Q_1$ on a distorted mesh . . . . .	77
4.41	$Q_2\hat{Q}_2$ acoustic wave without hourglass filter . . . . .	78
5.1	Problem setup for Sod shock tube problem . . . . .	80
5.2	Comparison of $Q_1-Q_0$ and $Q_2\hat{Q}_1$ elements on Sod shock tube and zoomed in on the shock ( <i>right</i> ). Five plot points are considered per zone. . . . .	82
5.3	Density pseudo-color at time $t = 0.6$ for the Noh problem on a 40 by 80 initial grid using the monotonic-scalar artificial viscosity. . . . .	84
5.4	Problem setup for Noh implosion problem . . . . .	85
5.5	Noh explosion problem on $20 \times 20$ mesh using $Q_1-Q_0$ elements without hourglass filtering . . . . .	86
5.6	Noh explosion problem on $40 \times 40$ mesh using $Q_1-Q_0$ elements without hourglass filtering . . . . .	87
5.7	Noh explosion problem on $20 \times 20$ mesh using $Q_1\hat{Q}_1$ elements without hourglass filtering . . . . .	89
5.8	Noh explosion problem on $40 \times 40$ mesh using $Q_1\hat{Q}_1$ elements without hourglass filtering . . . . .	90

5.9 Noh explosion problem on $20 \times 20$ mesh using $Q_2\text{-}\hat{Q}_1$ elements without hourglass filtering . . . . .	92
5.10 Noh explosion problem on $40 \times 40$ mesh using $Q_2\text{-}\hat{Q}_1$ elements without hourglass filtering . . . . .	93
5.11 Noh explosion problem on $20 \times 20$ mesh using $Q_2\text{-}\hat{Q}_2$ elements without hourglass filtering . . . . .	95
5.12 Noh explosion problem on $40 \times 40$ mesh using $Q_2\text{-}\hat{Q}_2$ elements without hourglass filtering . . . . .	96
5.13 Initial perturbed Saltzman piston mesh . . . . .	97
5.14 Saltzman piston problem on $50 \times 10$ mesh using $Q_1\text{-}Q_0$ elements without hourglass filtering at $t=0.925$ . . . . .	99
5.15 Saltzman piston problem on $50 \times 10$ mesh using $Q_1\text{-}Q_0$ elements with hourglass filtering at $t=0.925$ . . . . .	100
5.16 Saltzman piston problem on $50 \times 10$ mesh using $Q_1\text{-}\hat{Q}_1$ elements without hourglass filtering at $t=0.925$ . . . . .	102
5.17 Saltzman piston problem on $50 \times 10$ mesh using $Q_2\text{-}\hat{Q}_1$ elements with hourglass filtering at $t=0.925$ . . . . .	104
5.18 Saltzman piston problem on $50 \times 10$ mesh using $Q_2\text{-}\hat{Q}_2$ elements without hourglass filtering at $t=0.925$ . . . . .	106
5.19 Saltzman piston problem on $50 \times 10$ mesh using $Q_2\text{-}\hat{Q}_2$ elements with hourglass filtering at $t=0.925$ . . . . .	107
5.20 Problem setup for Sedov blast wave problem . . . . .	108
5.21 Exact solution to the Sedov problem applied to an initially Cartesian grid . . . . .	109
5.22 Sedov explosion problem on $10 \times 10$ mesh using $Q_1\text{-}Q_0$ elements with hourglass filtering . . . . .	112
5.23 Sedov explosion problem on $20 \times 20$ mesh using $Q_1\text{-}Q_0$ elements with hourglass filtering . . . . .	113
5.24 Sedov explosion problem on $10 \times 10$ mesh using $Q_1\text{-}\hat{Q}_1$ elements without hourglass filtering . . . . .	115
5.25 Sedov explosion problem on $20 \times 20$ mesh using $Q_1\text{-}\hat{Q}_1$ elements without hourglass filtering . . . . .	116
5.26 Sedov explosion problem on $10 \times 10$ mesh using $Q_2\text{-}\hat{Q}_1$ elements without hourglass filtering . . . . .	118
5.27 Sedov explosion problem on $10 \times 10$ mesh using $Q_2\text{-}\hat{Q}_1$ elements with hourglass filtering . . . . .	119

5.28 Sedov explosion problem on $20 \times 20$ mesh using $Q_2\text{-}\hat{Q}_1$ elements with hourglass filtering . . . . .	120
5.29 Sedov explosion problem on $5 \times 5$ mesh using $Q_2\text{-}\hat{Q}_1$ elements with hourglass filtering . . . . .	121
5.30 Comparison of $Q_1\text{-}Q_0$ ( <i>left</i> ) and $Q_2\text{-}\hat{Q}_1$ ( <i>right</i> ) on a full $21 \times 21$ mesh.	122
5.31 Sedov explosion problem on $10 \times 10$ mesh using $Q_2\text{-}\hat{Q}_2$ elements without hourglass filtering . . . . .	124
5.32 Sedov explosion problem on $20 \times 20$ mesh using $Q_2\text{-}\hat{Q}_2$ elements without hourglass filtering . . . . .	125
5.33 Sedov explosion problem on $20 \times 20$ mesh using $Q_2\text{-}\hat{Q}_2$ elements with hourglass filtering . . . . .	126

# Nomenclature

<b>ALE</b>	Arbitrary Lagrangian-Eulerian
<b>CFD</b>	Computational Fluid Dynamics
<b>DOF</b>	Degree Of Freedom
<b>EOS</b>	Equation Of State
<b>FEM</b>	Finite Element Method
<b>PDE</b>	Partial Differential Equation
<b>SGH</b>	Staggered Grid Hhydrocode
$Q_0$	constant function on quadrilaterals
$Q_1$	continuous bi-linear functions on quadrilaterals
$\hat{Q}_1 = Q_{1d}$	discontinuous bi-linear functions on quadrilaterals
$Q_2$	continuous bi-quadratic functions on quadrilaterals
$\hat{Q}_2 = Q_{2d}$	discontinuous bi-quadratic functions on quadrilaterals

# Chapter 1

## Introduction

Many analytical and empirical relations have been derived to model fluid motion under simplifying assumptions, but there are many applications of great engineering interest that are too complicated to be modeled by these simple relations. The field of computational fluid dynamics (CFD) has arisen to allow engineers to model and understand more complicated systems. The majority of familiar CFD codes describe fluid motion from an Eulerian perspective in which the fluid moves through a stationary grid, and fluid properties such as density, energy, and velocity are calculated based on flux through cell boundaries. Eulerian CFD[34] is most appropriate when the domain of the problem is unchanging. Some common applications of Eulerian CFD include external flow over a wing or whole aircraft[1], pipe and duct flow[11], ground vehicle aerodynamics[27], convective heat transfer for electronics[21], or even process modeling[12], and ship design[29].

A less common, but equally valid description, is the Lagrangian framework of fluid motion. In Lagrangian CFD codes, the mesh moves in time with the material. This is especially appropriate in cases where the problem domain it-

self is changing as is the case with many unsteady simulations. The area where Lagrangian methods particularly shine is in multi-material simulations and interface tracking. Applications include free surface flow[19, 25], detonations[24], astrodynamics[17], fluid-structure interaction[13], and even full simulations of nuclear tests.

A popular flavor of CFD codes solve the Euler equations, often in a Lagrangian frame, and are commonly known as hydrocodes[37]. Lagrangian simulations have been deemed more “delicate” because what may have been an initially sound mesh may become unsound as the solution progresses. Complicated or vortical flows can often cause Langrangian meshes to tangle. In these cases, the fluid moves in such a way that the mesh cannot follow and the mesh vertices start to tangle resulting in non-physical predictions like negative density or pressure. One technique that has arisen to address this issue is the Arbitrary Lagrangian Eulerian[14] (ALE) technique. The Lagrange step[35] is still at the heart of this method, but when mesh tangling is sensed, the fluid motion simulation is paused as the mesh is relaxed to a less tangled state. Then the fluid simulation resumes with this less tangled mesh until the mesh begins to tangle again.

Most ALE CFD codes use a Staggered Grid Hydrodynamics[4] (SGH) scheme in which thermodynamic variables like energy, density, and pressure are defined at cell centers and kinematic variables like position and velocity are defined at cell nodes. Traditionally ALE staggered grid hydrocodes have used low order elements such that for quads, thermodynamic variables are interpolated as constant within each cell but discontinuous between cells, and kinematic variables are interpolated bilinearly within a cell and are continuous between adjacent cells.

While traditional Staggered Grid Hydro codes have been successful in many applications, their predictive capability can be limited by several long-standing

issues. Notable among these are symmetry preservation, energy conservation, the artificial viscosity discretization, and hourglass mode instabilities. The overarching goal of our finite element hydrocode research group at Lawrence Livermore National Laboratory is to address these issues by developing modern finite element discretization schemes using high-order, multi-scale and discontinuous Galerkin methods. The research presented in this thesis considered whether high-order finite element discretizations are even feasible for shock hydrodynamics simulations.

# Chapter 2

## Theoretical Discussion

### 2.1 Euler Equations in a Lagrangian Frame

The Navier-Stokes equations can be simplified for high Reynolds number flows by assuming that inertial forces are much more significant than viscous forces. Thus all viscous terms can be neglected. This simplified set of four equations are known as the Euler equations and must be solved for four unknowns: velocity  $\vec{v}$ , density  $\rho$ , internal energy  $e$ , and pressure  $p$ . The Euler equations in a Lagrangian frame are listed below:

$$\text{Momentum Conservation: } \rho \frac{\partial \vec{v}}{\partial t} = -\vec{\nabla} p \quad (2.1)$$

$$\text{Mass Conservation: } \frac{1}{\rho} \frac{\partial \rho}{\partial t} = -\vec{\nabla} \cdot \vec{v} \quad (2.2)$$

$$\text{Energy Conservation: } \rho \frac{\partial e}{\partial t} = -p \vec{\nabla} \cdot \vec{v} \quad (2.3)$$

$$\text{Equation of State: } p = EOS(\rho, e) \quad (2.4)$$

The Euler equations describe the relationship between fluid motion and the kinematic properties of the fluid such as pressure, energy, and density over a spatial domain  $\Omega$ . The Equation of State (EOS) describes the relationship between the thermodynamic state variables and provides the necessary closure to the set of equations. In a hydrocode the density and energy of a material are fairly straightforward to calculate each time step. The EOS takes these variables as inputs and allows us to calculate the pressure of the material which is necessary for predicting the future movement of the fluid. The total mass in the spatial domain  $\Omega$  is defined to be

$$m \equiv \int_{\Omega} \rho \quad (2.5)$$

Furthermore, the total kinetic energy in the spatial domain  $\Omega$  is defined as

$$KE = \frac{1}{2} \int_{\Omega} \rho \vec{v} \cdot \vec{v} \quad (2.6)$$

The total internal energy in the spatial domain  $\Omega$  is defined as

$$IE = \int_{\Omega} \rho e \quad (2.7)$$

The total energy contained in the spatial domain  $\Omega$  is thus

$$E = KE + IE \quad (2.8)$$

If the spatial domain  $\Omega$  contains no energy sources (or sinks) and there is no flux of energy and/or mass out of the boundary of the domain,  $\partial\Omega$ , then the total energy contained in  $\Omega$  is a constant for all time

$$\frac{\partial E}{\partial t} = 0 \quad (2.9)$$

## 2.2 A Brief Summary of the Finite Element Method

The finite element method (FEM) is a powerful numerical technique for solving partial differential equations (PDE) on complicated domains. In contrast to the finite difference method which finds a solution to a finite difference approximation of the differential equation, the FEM finds an approximate solution to the original differential equation itself.

### 2.2.1 The Weak Statement of the Problem

The first step in the solution to any problem is to first define the problem itself. The FEM requires that the problem be described in a specific format known as the variational or weak statement of the problem[20]. Any differential equation can be reformulated to fit into this required format. In brief, you must multiply the differential equation by a vector valued test function and integrate both sides of the resulting equation over the problem domain. After an integration by parts some of the derivatives are transferred over to the test functions. The resulting system of equations is the variational formulation of the problem. For our purposes, the test functions are chosen to be the same set as the basis functions according to Galerkin's method. We formulate the weak statement of the Euler equations in Section 2.4.1.

### 2.2.2 Basis Functions

The approximation to the solution of the differential equation is built up from a set of analyst-selected *basis functions*. The solution to the PDE is assumed to be the summation of all basis functions multiplied with a set of unknown coef-

ficients known as degrees of freedom (DOF). Generally, the basis functions may be any function at all, but typically they are chosen to be piecewise continuous polynomials defined such that they are valued 1 at their own DOF and zero at all others. Thus, when all basis functions are summed up with their respective DOF coefficients, the overall solution will assume values of a DOF at that DOF location and interpolate polynomially to adjacent DOFs.

As mentioned previously, traditionally staggered grid approaches have associated velocities with zone vertices and thermodynamic variables have been associated with zone centers and defined as piece-wise constant values. Our general FEM approach treats each field -  $\vec{v}$ ,  $\rho$ ,  $e$ , and  $p$  as finite element functions on the computational mesh  $\tilde{\Omega}(t)$  with the following expansions for  $\vec{x} \in \Omega(t)$

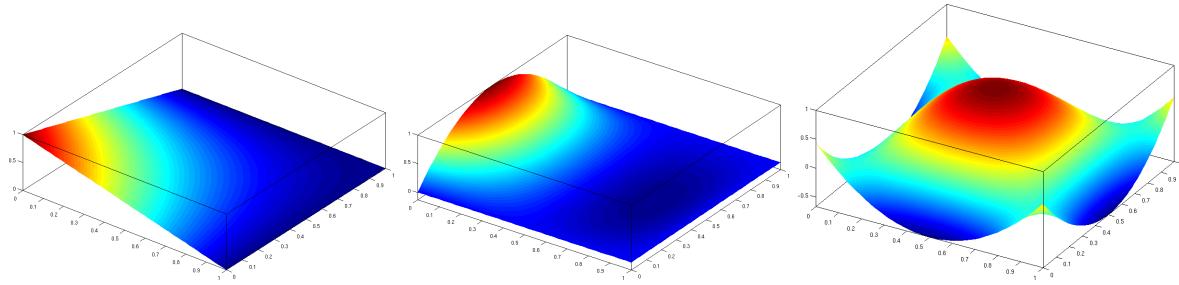
$$\vec{v}(\vec{x}, t) \approx \sum_i^{N_v} \mathbf{v}_i(t) \vec{w}_i(\vec{x}, t), \quad (2.10)$$

$$\rho(\vec{x}, t) \approx \sum_i^{N_r} \mathbf{r}_i(t) \phi_i(\vec{x}, t), \quad (2.11)$$

$$e(\vec{x}, t) \approx \sum_i^{N_e} \mathbf{e}_i(t) \phi_i(\vec{x}, t), \quad (2.12)$$

$$p(\vec{x}, t) \approx \sum_i^{N_p} \mathbf{p}_i(t) \phi_i(\vec{x}, t). \quad (2.13)$$

These basis functions will move with the mesh. Furthermore, for simplicity and ease of computation basis functions will be defined on the reference zone and mapped to physical space via the Jacobian of transformation, see [2.3.3](#).



**Figure 2.1:** Examples of 2D basis functions on a reference zone: a standard  $Q_1$  bi-linear function that interpolates at nodes (*left*), a  $Q_2$  bi-quadratic function defined at the Gauss-Lobatto quadrature points (*center*) and a  $Q_2$  bi-quadratic function defined at the Gauss-Legendre quadrature points (*right*).

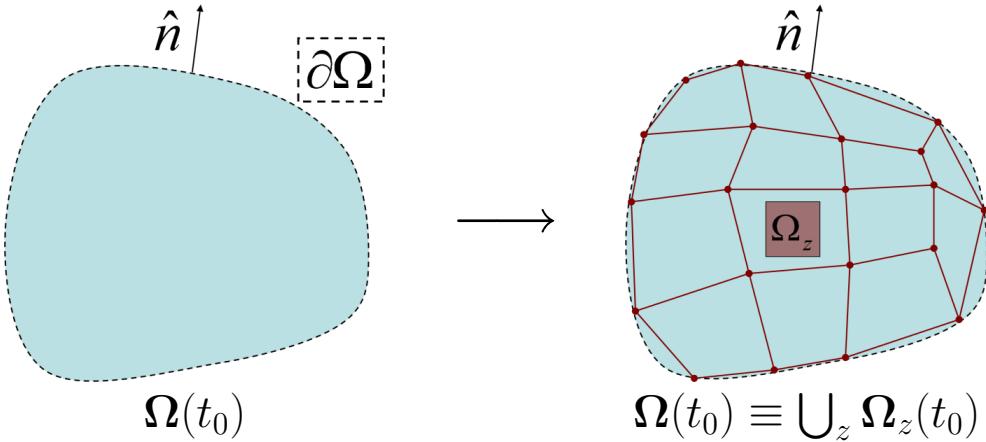
## 2.3 Mesh

The basic premise of the Finite Element Method (FEM) is to divide a large complicated problem into a system of smaller simpler problems. In order to do this, we need to divide the problem domain into a number of non-overlapping volumes or zones (aka elements). The governing equations are then solved on each of these smaller domains and assembled to obtain an overall solution to the problem.

### 2.3.1 Domain Decomposition

The FEM requires us to take many integrals as we set up the system of simple problems. It is difficult to take an integral over an arbitrary region, especially in an automated fashion. Thus, the first step is to divide the arbitrarily complicated problem domain into a number of simpler zones. In 2-D These zones or elements are usually three- or four-sided but can be more complicated. Each side is defined by a polynomial curve. Traditionally, first order curves, i.e. straight lines are used to connect vertices, but quadratic, cubic, or even higher order curves can be used.

Figure 2.2 illustrates this process.



**Figure 2.2: Domain decomposition for a finite element mesh**

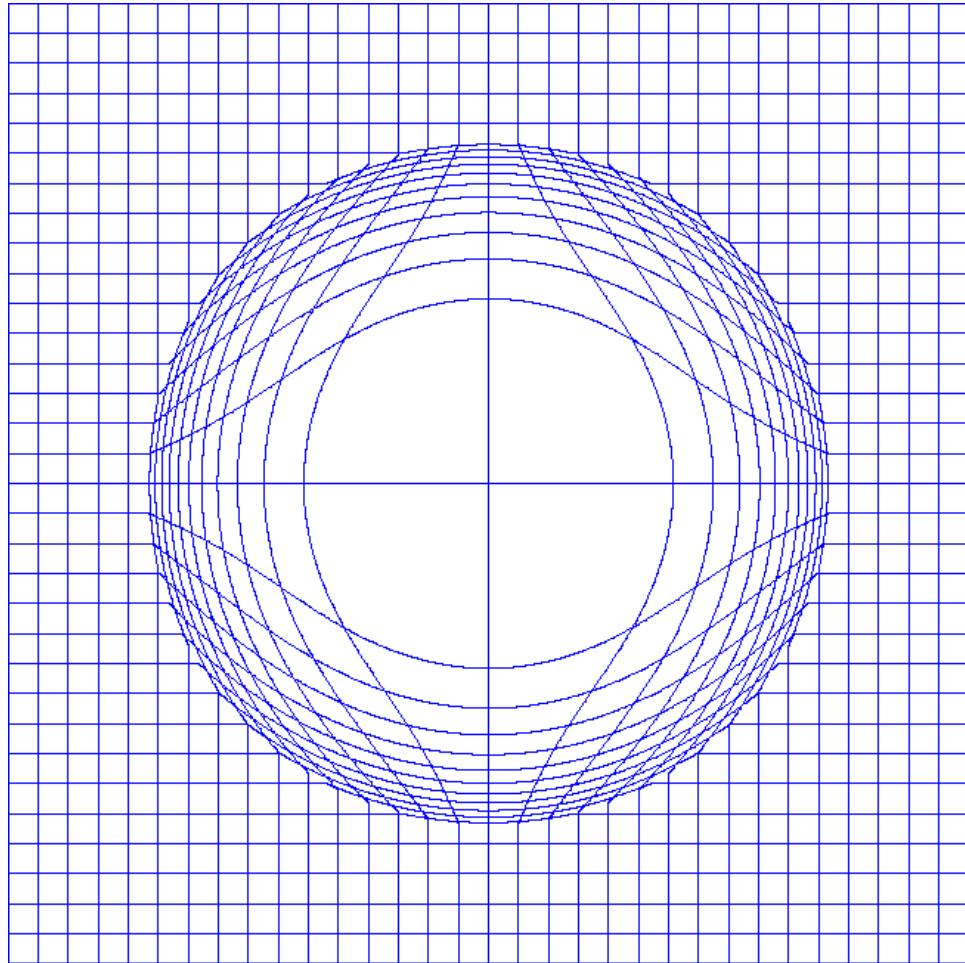
Now that we have divided our problem domain into a number of smaller zones, we can link these together to form the *computational mesh*, denoted by  $\tilde{\Omega}$  and defined as

$$\Omega(t_0) \approx \tilde{\Omega}(t_0) \equiv \bigcup_z \Omega_z(t_0) \quad (2.14)$$

The computational mesh is composed of two sets of information. The *topology* defines the connectivity of the mesh: which nodes belong to which element and which elements share faces, etc. The *geometry* connects nodes to coordinates in physical space. For example, if you wish to compute the volume of element  $n$ , you must first follow the topology to determine which nodes are connected to element  $n$  and the ordering of these nodes. Then you must follow the geometry to find the physical coordinates of these nodes. Once you know the physical locations and ordering of the nodes, then you can perform the desired calculations.

If we take an initially straight Cartesian mesh and continuously apply the exact solution of the Sedov explosion problem to the nodes and edges, we obtain the mesh shown in Figure 2.3. The obviously curved elements, especially

surrounding the origin motivate the need for zones with curved edges.

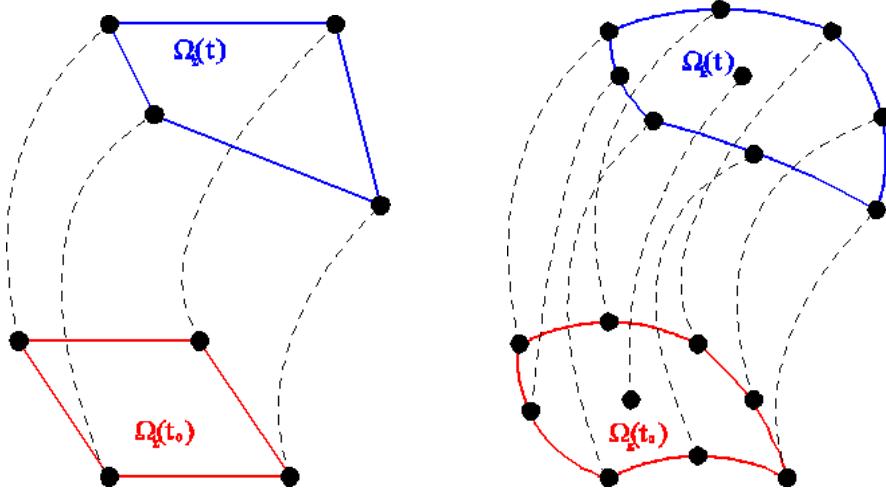


**Figure 2.3:** An initial Cartesian mesh is continuously deformed according to the exact solution of the Sedov blast wave.

### 2.3.2 Mesh Motion

By the definition of the Lagrangian framework, for all interesting flows, the computational mesh will move during the process of the simulation. Each element can be thought of as a small volume of fluid that will move and distort in reaction to pressure fields acting on it. We move each volume by tracking a finite number of particles on the boundaries of (and possibly within) that volume known as

the *position degrees of freedom*. During each time step, each position degree of freedom is moved according its corresponding velocity degree of freedom and the mesh is reconstructed for the next time step, as illustrated by Figure 2.4.



**Figure 2.4:** In a semi-discrete setting, a zone  $\Omega_z(t)$  is reconstructed from the evolution of only a few of its points (particles) indicated by black dots. Shown are two specific choices corresponding to the traditional  $Q_1$  zone (left) and a high order  $Q_2$  zone with curvilinear boundaries (right).

### 2.3.3 Jacobian of Transformation

You may have noticed that even an initially perfectly Cartesian mesh can get rather distorted in the process of a simulation. We mentioned earlier that one of the motivations to divide the problem domain into regular zones with regular topology was to ease the process of integration. But anyone who has taken Calculus II knows that integration over a distorted, much less curved domain is not a completely trivial matter. This is where the Jacobian of transformation comes in.

The Jacobian matrix is used as a mapping from a standard reference zone  $\tilde{\Omega}_z = [0, 1]^2$  to physical space, see Figure 2.5. We can write the mapping in a

functional form as

$$\vec{x} = \Phi(\hat{\vec{x}}, t), \quad (2.15)$$

where  $\vec{x}$  denotes a point in physical space and  $\hat{\vec{x}}$  denotes the corresponding point in the reference zone. This coordinate transformation is referred to as the parametric mapping and is defined by the physical coordinates of the particles associated with each zone. For the case of a traditional  $Q_1$  zone geometry consisting of four vertices connected by straight lines, the parametric mapping is bilinear. This thesis explores some of the benefits of high order mappings such as  $Q_2$  (bi-quadratic) which produce zones with curvilinear geometry as shown in Figure 2.5. These parametric mappings are computed for each zone using an interpolating polynomial expansion of the form

$$\Phi_z(\hat{\vec{x}}, t) = \sum_i \vec{x}_{z,i}(t) \eta_i(\hat{\vec{x}}), \quad (2.16)$$

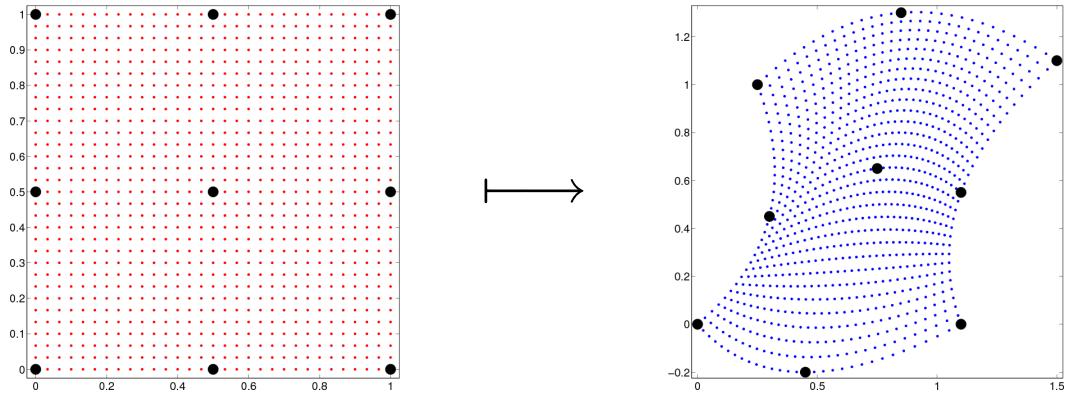
where  $\vec{x}_{z,i}(t)$  denote the physical coordinates of the particles describing the zone  $z$  at time  $t$ , see Figure 2.4, and  $\eta_i$  is the (high order) nodal basis function associated with particle  $i$ . The collection of all particle coordinates listed consecutively in a vector array is denoted by  $\mathbf{x}(t)$ . We define the Jacobian matrix for this mapping as

$$\mathbf{J}_z = \vec{\nabla}_{\hat{\vec{x}}} \Phi_z \quad \text{or} \quad (\mathbf{J}_z)_{i,j} = \frac{\partial x_i}{\partial \hat{x}_j}. \quad (2.17)$$

Note that in general, the Jacobian matrix is a *function* of the reference coordinates  $\hat{\vec{x}}$  and therefore varies in a zone.

For bi-linear elements, the Jacobian is defined according to (2.18), while bi-quadratic elements use the much more complicated entries in (2.19) to assemble the Jacobian matrix.

$$\mathbf{J}_z(\hat{x}, \hat{y}) = \begin{bmatrix} x_2 - x_1 + \hat{y}(x_1 + x_3 - x_2 - x_4) & x_4 - x_1 + \hat{x}(x_1 + x_3 - x_2 - x_4) \\ y_2 - y_1 + \hat{y}(y_1 + y_3 - y_2 - y_4) & y_4 - y_1 + \hat{x}(y_1 + y_3 - y_2 - y_4) \end{bmatrix} \quad (2.18)$$



**Figure 2.5:** Example of a  $Q_2$  bi-quadratic mapping from a reference zone (*left*) to a Lagrangian zone (*right*) defined by the locations of the 9 Lagrangian particles (black dots).

$$\begin{aligned}
J_{1,1} &= (\hat{y} - 1)(2\hat{y} - 1)(4\hat{x} - 3)x_1 + (\hat{y} - 1)(2\hat{y} - 1)(4\hat{x} - 1)x_2 \\
&\quad + \hat{y}(2\hat{y} - 1)(4\hat{x} - 1)x_3 + \hat{y}(2\hat{y} - 1)(4\hat{x} - 3)x_4 \\
&\quad - 4(\hat{y} - 1)(2\hat{y} - 1)(2\hat{x} - 1)x_5 - 4\hat{y}(\hat{y} - 1)(4\hat{x} - 1)x_6 \\
&\quad - 4\hat{y}(2\hat{y} - 1)(2\hat{x} - 1)x_7 - 4\hat{y}(\hat{y} - 1)(4\hat{x} - 3)x_8 \\
&\quad + 16\hat{y}(\hat{y} - 1)(2\hat{x} - 1)x_9 \\
J_{1,2} &= (4\hat{y} - 3)(\hat{x} - 1)(2\hat{x} - 1)x_1 + \hat{x}(4\hat{y} - 3)(2\hat{x} - 1)x_2 \\
&\quad + \hat{x}(4\hat{y} - 1)(2\hat{x} - 1)x_3 + (4\hat{y} - 1)(\hat{x} - 1)(2\hat{x} - 1)x_4 \\
&\quad - 4\hat{x}(4\hat{y} - 3)(\hat{x} - 1)x_5 - 4\hat{x}(2\hat{y} - 1)(2\hat{x} - 1)x_6 \\
&\quad - 4\hat{x}(4\hat{y} - 1)(\hat{x} - 1)x_7 - 4(2\hat{y} - 1)(\hat{x} - 1)(2\hat{x} - 1)x_8 \\
&\quad + 16\hat{x}(2\hat{y} - 1)(\hat{x} - 1)x_9 \\
J_{2,1} &= (\hat{y} - 1)(2\hat{y} - 1)(4\hat{x} - 3)y_1 + (\hat{y} - 1)(2\hat{y} - 1)(4\hat{x} - 1)y_2 \\
&\quad + \hat{y}(2\hat{y} - 1)(4\hat{x} - 1)y_3 + \hat{y}(2\hat{y} - 1)(4\hat{x} - 3)y_4 \\
&\quad - 4(\hat{y} - 1)(2\hat{y} - 1)(2\hat{x} - 1)y_5 - 4\hat{y}(\hat{y} - 1)(4\hat{x} - 1)y_6 \\
&\quad - 4\hat{y}(2\hat{y} - 1)(2\hat{x} - 1)y_7 - 4\hat{y}(\hat{y} - 1)(4\hat{x} - 3)y_8 \\
&\quad + 16\hat{y}(\hat{y} - 1)(2\hat{x} - 1)y_9 \\
J_{2,2} &= (4\hat{y} - 3)(\hat{x} - 1)(2\hat{x} - 1)y_1 + \hat{x}(4\hat{y} - 3)(2\hat{x} - 1)y_2 \\
&\quad + \hat{x}(4\hat{y} - 1)(2\hat{x} - 1)y_3 + (4\hat{y} - 1)(\hat{x} - 1)(2\hat{x} - 1)y_4 \\
&\quad - 4\hat{x}(4\hat{y} - 3)(\hat{x} - 1)y_5 - 4\hat{x}(2\hat{y} - 1)(2\hat{x} - 1)y_6 \\
&\quad - 4\hat{x}(4\hat{y} - 1)(\hat{x} - 1)y_7 - 4(2\hat{y} - 1)(\hat{x} - 1)(2\hat{x} - 1)y_8 \\
&\quad + 16\hat{x}(2\hat{y} - 1)(\hat{x} - 1)y_9
\end{aligned}
\tag{2.19}$$

## 2.4 Semi-Discrete FEM Approximation

In the semi-discrete formulation the problem has been discretized in space, but a specific time discretization has not been selected yet. Time could be discretized in a forward Euler-like scheme, a leap-frog scheme, predictor-corrector, or any other time discretization.

### 2.4.1 Momentum Conservation

In order to derive a semi-discrete momentum conservation law, we start with the variational formulation or the weak statement of the problem. Multiply (2.1) by a vector valued test function  $\vec{w}'$  and integrate over the spatial domain  $\Omega(t)$ :

$$\int_{\Omega(t)} \left( \rho \frac{d\vec{v}}{dt} \right) \cdot \vec{w}' = - \int_{\Omega(t)} (\vec{\nabla} p) \cdot \vec{w}' \quad (2.20)$$

Now, approximating the spatial domain  $\Omega(t)$  with the computational mesh  $\tilde{\Omega}(t)$ , integrate the right hand side of (2.20) and apply the divergence theorem:

$$\int_{\tilde{\Omega}(t)} \left( \rho \frac{d\vec{v}}{dt} \right) \cdot \vec{w}' = \int_{\tilde{\Omega}(t)} p (\vec{\nabla} \cdot \vec{w}') - \int_{\partial\tilde{\Omega}(t)} p (\vec{w}' \cdot \hat{n}) \quad (2.21)$$

where  $\hat{n}$  is the outward pointing unit normal vector of the surface  $\partial\tilde{\Omega}(t)$ .

If we substitute the basis function expansions from (2.10) for  $\vec{v}$  and (2.13) for  $p$ , and for brevity's sake assume boundary conditions that will cause the boundary integral term to vanish, we arrive at

$$\int_{\tilde{\Omega}(t)} \left( \rho \sum_i^{N_v} \frac{d\mathbf{v}_i}{dt} \vec{w}_i \right) \cdot \vec{w}' = \int_{\tilde{\Omega}(t)} \sum_i^{N_p} \mathbf{p}_i \phi_i (\vec{\nabla} \cdot \vec{w}') . \quad (2.22)$$

As mentioned in Section [Section 2.2.1](#), we can apply Galerkin’s method and pick the velocity basis functions  $\vec{w}_j$  as our test functions  $\vec{w}'$ . If we do this we will arrive at the following system of linear ordinary differential equations

$$\sum_i^{N_v} \frac{d\mathbf{v}_i}{dt} \int_{\tilde{\Omega}(t)} \rho(\vec{w}_i \cdot \vec{w}_j) = \sum_i^{N_p} \mathbf{p}_i \int_{\tilde{\Omega}(t)} \phi_i(\vec{\nabla} \cdot \vec{w}_j). \quad (2.23)$$

In matrix form this can be written,

$$\mathbf{M} \frac{d\mathbf{v}}{dt} = \mathbf{D}^T \mathbf{p}$$

(2.24)

where  $\mathbf{M}$ ,  $\mathbf{D}$ ,  $\mathbf{v}$ , and  $\mathbf{p}$  are global matrices or vectors (arrays) that are assembled over the entire computational mesh  $\tilde{\Omega}(t)$  with contributions from DOFs in each individual zone  $z$ . This is known as the assembly operation and can be written as

$$\mathbf{M} = \text{Assemble}(\mathbf{M}_z), \quad \mathbf{D} = \text{Assemble}(\mathbf{D}_z),$$

$$\mathbf{v} = \text{Assemble}(\mathbf{v}_z), \quad \mathbf{p} = \text{Assemble}(\mathbf{p}_z).$$

During this global assembly process the topological (see [2.3.1](#)) information is used to sum to each degree of freedom information from all zones that it is part of. This is analogous to the idea of “nodal accumulation” that is present in traditional SGH codes. Note that the resulting mass matrix  $\mathbf{M}$  is global in scope and, in general, requires a full linear solve at each time step to arrive at the desired accelerations. This may sound computationally daunting, but some simplifications can be made to significantly speed up this process such as mass lumping. Mass lumping sums the each row of the matrix to its diagonal, reducing the accuracy while eliminating a full linear solve. In general the mass matrix is

very sparse and is has not been a major computational bottleneck in the test problems considered.

The global mass matrix must be assembled from local mass matrices  $\mathbf{M}_z$  calculated for each zone. The local mass matrix can be calculated over a zone  $z$

$$(\mathbf{M}_z)_{i,j} \equiv \int_{\Omega_z(t)} \rho(\vec{w}_i \cdot \vec{w}_j) . \quad (2.25)$$

By its construction, the mass matrix is symmetric positive definite with dimension  $ndof_v \times ndof_v$  where  $ndof_v$  is the number of x- or y- velocity degrees of freedom per zone. In theory,  $\mathbf{M}_z$  should be  $(2 \cdot ndof_v) \times (2 \cdot ndof_v)$  to account for both x- and y- velocity degrees of freedom, but in practice they are identical and a single  $ndof_v \times ndof_v$  matrix can be used. For example, a  $Q_1$  element has 4 x- and 4 y- velocity degrees of freedom while a  $Q_2$  element has 9 x- and 9 y- velocity degrees of freedom. Thus the local mass matrix for a  $Q_2$  element would be a  $9 \times 9$  matrix. The local derivative matrix for zone  $z$  is defined as

$$(\mathbf{D}_z)_{i,j} \equiv \int_{\Omega_z(t)} \phi_i(\vec{\nabla} \cdot \vec{w}_j) . \quad (2.26)$$

The derivative matrix is a *rectangular* of dimension  $ndof_p \times ndof_v$  and acts as a map between the two discrete representations of velocity and pressure. It is a discrete version of the *Div* operator. Conveniently, the transpose of the derivative matrix is also a discrete version of the *Grad* operator as seen in (2.24).

## 2.4.2 Computing FEM Matrices

As mentioned previously, numerical integration is at the heart of the finite element method. The mass and derivative matrices mentioned above take the

brunt of this numerical integration effort. In practice, we compute these integrals by transforming them from each Lagrangian zone  $\Omega_z(t)$  to the reference zone  $\hat{\Omega}$  through the parametric mapping of (2.15). For a general integral over a given Lagrangian zone, this transformation gives

$$\int_{\Omega_z(t)} f = \int_{\hat{\Omega}_z} (f \circ \Phi) |\det \mathbf{J}_z|,$$

for some integrand  $f$ , where “ $\circ$ ” denotes composition. Integrals are approximated using quadrature of a user-selected order. This integral over the the Lagrangian zone can therefore be replaced with the following weighted sum

$$\int_{\Omega_z(t)} f \approx \sum_{n=1}^{N_q} \alpha_n \{(f \circ \Phi) |\det \mathbf{J}_z|\}_{\hat{x}=\hat{q}_n}, \quad (2.27)$$

where  $\alpha_n$  are the  $N_q$  quadrature weights and  $\hat{q}_n$  are quadrature points inside of the reference zone where the integrand is sampled at. It is important to realize that depending on the functional form of the integrand and the order of the quadrature rule, numerical quadrature is not always exact. Thus, we have introduced additional, though reasonable, approximation error to the solution of the governing equations. Practically, Gauss-Legendre quadrature is preferred for quadrature on quadrilaterals due to great accuracy for relatively few sample points.

### 2.4.3 Mass Conservation

By the very nature of the Lagrangian description of fluid dynamics, there is no flux across zone boundaries. Hence, the total amount of mass in a zone is

constant for all time.

$$\frac{d}{dt} \int_{\Omega_z(t)} \rho = 0. \quad (2.28)$$

In order to allow for high order density field representations, we need to define high order “mass moments” for a given zone using the density basis functions from (2.11):

$$\mathbf{m}_{z,i} \equiv \int_{\Omega_z(t)} \rho \phi_i. \quad (2.29)$$

We generalize the case of zonal mass conservation to the high order mass moments by postulating that

$$\frac{d\mathbf{m}_{z,i}}{dt} = 0. \quad (2.30)$$

This choice is motivated by the fact that the same equation holds in the continuous case. Furthermore, (2.30) has the same number of conditions as the number of unknown densities and in particular it implies (2.28). Note that for the case of a piece-wise constant density approximation (a single mass moment), we recover the traditional definition of zonal mass conservation. From our basis function representation of the density field (2.11), it follows that

$$\mathbf{m}_{z,i} = \sum_j \mathbf{r}_{z,j} \int_{\Omega_z(t)} \phi_i \phi_j.$$

Written in matrix form, we have

$$\mathbf{m}_z = \mathbf{M}_z^\rho \mathbf{r}_z \quad \text{where} \quad (\mathbf{M}_z^\rho)_{i,j} \equiv \int_{\Omega_z(t)} \phi_i \phi_j.$$

This yields the semi-discrete mass conservation law

$$\frac{d}{dt} (\mathbf{M}_z^\rho \mathbf{r}_z) = 0$$

(2.31)

The above can be viewed as a generalization of the "sub-zonal mass" concept introduced in [6]; it is a statement that no mass enters or leaves a given sub-volume of the zone. If we take the limiting case of this idea and impose mass conservation of the form

$$\frac{d}{dt} \int_{\Omega'(t)} \rho = 0 \quad \text{for any} \quad \Omega'(t) \subseteq \Omega_z(t),$$

then we obtain the *strong mass conservation principle*

$$\rho(t) |\det \mathbf{J}_z(t)| = \rho(t_0) |\det \mathbf{J}_z(t_0)| \quad (2.32)$$

which is a statement of mass conservation for any point in space (not just in a variational sense). Note that the density defined by this equation is not polynomial.

We can write the following relation between the finite element density function defined by (2.31) and the function defined by (2.32) (denoted here by  $\rho_h$  and  $\rho_s$ , respectively)

$$\int_{\Omega_z(t)} \rho_h \phi_i = \int_{\Omega_z(t)} \rho_s \phi_i \quad (2.33)$$

which tells us that  $\rho_h$  is the projection of  $\rho_s$  on the space spanned by  $\{\phi_i\}$ .

#### 2.4.4 Energy Conservation

If we multiply the energy equation, (2.12) by a vector valued test function,  $\phi_j$  and integrate over a local spatial domain,  $\Omega_z(t)$  we arrive at a local formulation

for the energy conservation equation

$$\int_{\Omega_z(t)} \rho \frac{de}{dt} \phi_j = \int_{\Omega_z(t)} p(\vec{\nabla} \cdot \vec{v}) \phi_j .$$

Inserting the basis function expansions for internal energy, pressure and velocity from (2.10 - 2.13) we obtain

$\mathbf{M}_z^e \frac{d\mathbf{e}_z}{dt} = -\mathbf{p}_z \cdot \mathbf{D}_z^e \cdot \mathbf{v}_z$

(2.34)

where

$$(\mathbf{M}_z^e)_{i,j} \equiv \int_{\Omega_z(t)} \rho \phi_i \phi_j \quad \text{and} \quad (\mathbf{D}_z^e)_{i,j,k} \equiv \int_{\Omega_z(t)} \phi_i \phi_j (\vec{\nabla} \cdot \vec{w}_k) .$$

Here  $\mathbf{D}_z^e$  is tensor of rank 3 satisfying  $\mathbf{D}_z = \mathbf{D}_z^e \cdot \mathbf{1}_z^e$ , i.e.,  $(\mathbf{D}_z)_{i,k} = \sum_j (\mathbf{D}_z^e)_{i,j,k} (\mathbf{1}_z^e)_j$ , where  $\mathbf{1}_z^e$  is the zonal representation of the constant 1 in the internal energy space (a vector of ones for nodal finite elements). In this formulation, the matrix  $\mathbf{F}_z = \mathbf{p}_z \cdot \mathbf{D}_z^e$  can be used to generalize the concept of “corner forces” (see below), since  $\mathbf{F}_z^T \cdot \mathbf{1}_z^e$  gives the zonal forces in the momentum equation, while  $\mathbf{F}_z \cdot \mathbf{v}_z$  is the work term due to the pressure gradient forces.

Given the above definitions, we can show that the following semi-discrete energy conservation relation holds:

$$\frac{d\tilde{E}}{dt} = \frac{d}{dt} \left( \frac{1}{2} \mathbf{v} \cdot \mathbf{M} \cdot \mathbf{v} + \sum_z \mathbf{1}_z^e \cdot \mathbf{M}_z^e \cdot \mathbf{e}_z \right) = \frac{1}{2} \mathbf{v} \cdot \frac{d\mathbf{M}}{dt} \cdot \mathbf{v} + \sum_z \mathbf{1}_z^e \cdot \frac{d\mathbf{M}_z^e}{dt} \cdot \mathbf{e}_z , \quad (2.35)$$

where  $\tilde{E}$  denotes the total discrete energy in the computational domain. Note that there is both a kinetic energy term and an internal energy term and that the time rate of change of this sum is equal to zero (implying total energy conservation)

when the time derivatives of the mass matrices  $\mathbf{M}$  and  $\mathbf{M}_z^e$  are zero. For the case where the mass matrices change in time (implying a redistribution of mass within a zone), this change must be taken into account in order to maintain exact energy conservation.

Now consider the special case of piece-wise constant internal energies (i.e. a single constant basis function), and denote the (single) zonal mass by  $\mathbf{m}_z$ . Then, (2.34) reduces to the form

$$\mathbf{m}_z \frac{d\mathbf{e}_z}{dt} = -\mathbf{p}_z \mathbf{D}_z \mathbf{v}_z . \quad (2.36)$$

This can be viewed as a generalization of the so called "compatible hydro" approach of [6] by noting that the term  $\mathbf{p}_z \mathbf{D}_z$  is simply a collection of "corner forces" due to the discrete pressure gradient term and therefore

$$(\mathbf{p}_z \mathbf{D}_z) \mathbf{v}_z = \sum_i \vec{f}_i \cdot \vec{v}_i .$$

#### 2.4.5 Equation of State

We can consider the equation of state (2.4) as either a weak formulation of the equation or simply point by point. As an example, the variational formulation of the gamma-law energy equation  $p = (\gamma - 1)\rho e$  is

$$\int_{\Omega_z(t)} p \phi_j = \int_{\Omega_z(t)} (\gamma - 1) \rho e \phi_j .$$

This can be written in matrix form (for constant  $\gamma$  as

$$\boxed{\mathbf{M}_z^\rho \mathbf{p}_z = (\gamma - 1) \mathbf{M}_z^e \mathbf{e}_z} \quad (2.37)$$

## 2.5 Fully-Discrete FEM Approximation

The fully discrete approximation takes the spatially discretized approximations and applies a time discretization. Any time discretization could be applied, but for the sake of simplicity, and because this research is more concerned with spatial discretization, we consider only the simple case of a forward Euler-like scheme.

### 2.5.1 Momentum Conservation

If we take the semi-discrete momentum conservation equation, (2.24) and apply the forward Euler time integration scheme, we get

$$\boxed{\mathbf{M}^{n+1}\mathbf{v}^{n+1} = \mathbf{M}^n\mathbf{v}^n + \Delta t(\mathbf{D}^n)^T \mathbf{p}^n}$$

If we apply the strong mass conservation principle from (2.32), we can avoid recomputing the mass matrix every cycle because the mass matrix is constant for all time:  $\mathbf{M}^n = \mathbf{M}$ . This provides significant improvements to the computational efficiency of the algorithm because you only have to assemble the mass matrix at the initial time step, never again. This could also make the full linear solve mentioned in Section 2.4.1 less demanding because the constant mass matrix  $\mathbf{M}$  could be decomposed via LU decomposition or a similar technique.

## 2.5.2 Mass Conservation

Because the total mass in each zone remains constant over time, and from equation (2.31), we can update density using

$$\mathbf{M}_z^{\rho,n+1} \mathbf{r}_z^{n+1} = \mathbf{M}_z^{\rho,n} \mathbf{r}_z^n = \dots = \mathbf{M}_z^{\rho,0} \mathbf{r}_z^0 = \mathbf{m}_z^0. \quad (2.38)$$

or,

$$\boxed{\mathbf{M}_z^{\rho,n+1} \mathbf{r}_z^{n+1} = \mathbf{m}_z^0} \quad (2.39)$$

In other words, we can calculate a local mass matrix every cycle for each zone and do a small ( $4 \times 4$ ) for density in the  $\hat{Q}_1$  space) linear solve with the original mass of the zone to calculate the new density degrees of freedom.

## 2.5.3 Energy Conservation

If we apply the forward Euler scheme to the semi-discrete energy conservation equation, the change in kinetic energy over a time step  $\Delta t$  can be calculated from

$$\frac{1}{2}(\mathbf{v}^{n+1})^T \mathbf{M} \mathbf{v}^{n+1} - \frac{1}{2}(\mathbf{v}^n)^T \mathbf{M} \mathbf{v}^n = (\mathbf{v}^{n+\frac{1}{2}})^T \mathbf{M} (\mathbf{v}^{n+1} - \mathbf{v}^n) = \Delta t \mathbf{v}^{n+\frac{1}{2}} (\mathbf{D}^n)^T \mathbf{p}^n,$$

where  $\mathbf{v}^{n+\frac{1}{2}} \equiv (\mathbf{v}^{n+1} + \mathbf{v}^n)/2$ . So in order to preserve total discrete energy exactly from time step  $n$  to  $n + 1$ , we need the energy update to have the form

$$\boxed{\mathbf{M}_z^{e,n+1} \mathbf{e}_z^{n+1} = \mathbf{M}_z^{e,n} \mathbf{e}_z^n - \Delta t \mathbf{p}_z^n \cdot \mathbf{D}_z^{e,n} \cdot \mathbf{v}_z^{n+\frac{1}{2}}} \quad (2.40)$$

which is a discretization of (2.34).

#### 2.5.4 Equation of State

The equation of state discretization does not depend on time at all. Thus, the fully discrete equation of state is just the semi-discrete equation applied at every time step.

$$\mathbf{M}_z^{\rho,n+1} \mathbf{p}_z^{n+1} = (\gamma - 1) \mathbf{M}_z^{e,n+1} \mathbf{e}_z^{n+1}$$

### 2.6 Tensor Artificial Viscosity

One inherent difficulty in solving the Euler equations is that they emit shocks. This can be particularly disturbing in a Lagrangian simulation because the nearly discontinuous nature of shocks can crush a cell to zero or even negative volume. Artificial viscosity has become a popular method for handling shocks in CFD codes. Artificial viscosity smooths a shock over several zones, where physically the shock would have behaved as a discontinuity. Traditionally, this has been added to the molecular viscosity in regions with high gradients, but this method has not performed well in general cases where the shock wave is not aligned with the mesh.

In order to properly handle shocks that are not aligned with the mesh, we use a tensor artificial viscosity to resist cell compression proportionate with the direction of the shock. Specifically we implemented the general finite element based tensor artificial viscosity formulation of Kolev and Rieben[23]. This formulation is general enough to easily allow for high order velocity field representations. In brief, this approach takes the momentum and energy conservation equations of (2.1) and (2.3) and augments them with a generalized viscous force and its

corresponding energy term, respectively.

$$\rho \frac{d\vec{v}}{dt} = -\vec{\nabla} p + \vec{\nabla} \cdot (\mu \vec{\nabla} \vec{v})$$

$$\rho \frac{de}{dt} = -p \vec{\nabla} \cdot \vec{v} + (\mu \vec{\nabla} \vec{v}) : (\vec{\nabla} \vec{v})$$

As usual, we can apply a variational formulation to the momentum equation and use the velocity basis functions to compute local compression-resistive forces for every zone (analogous to corner forces).

$$\mathbf{f}_z = \mathbf{S}_z \mathbf{v}_z, \quad \text{where} \quad (\mathbf{S}_z)_{i,j} = \int_{\Omega_z(t)} (\mu_z \vec{\nabla} \vec{w}_i) : \vec{\nabla} \vec{w}_j. \quad (2.41)$$

Note that, similar to the mass matrix, a global stiffness matrix  $\mathbf{S}$  is assembled from all of the zonal stiffness matrices,  $\mathbf{S} = \text{Assemble}(\mathbf{S}_z)$ . It is also the same size,  $N_v \times N_v$ , where  $N_v$  is the total number of velocity degrees of freedom in the domain, and it is symmetric positive definite. Also note that in general, the zone based artificial viscosity coefficient  $\mu_z$  is a function of space within each zone. Similar to the corner forces, the tensor artificial viscosity forces update the energy equation through a shock/viscosity heating term through an inner product with the zonal velocity.

$$\Delta \mathbf{e}_z = \mathbf{v}_z \cdot \mathbf{S}_z^e \cdot \mathbf{v}_z \quad \text{where}$$

where

$$(\mathbf{S}_z^e)_{i,j,k} = \int_{\Omega_z(t)} (\mu_z \vec{\nabla} \vec{w}_i) : (\vec{\nabla} \vec{w}_k) \phi_j$$

and  $\mathbf{S}_z = \mathbf{S}_z^e \cdot \mathbf{1}_z^e$ . In practice we only care about the viscosity coefficient  $\mu_z$  at the quadrature points of the integral in (2.41):  $\mu_z$  has both a linear and quadratic

diffusion term similar to the form described in [23].

## 2.7 Fully-Discrete Scheme

We now have all of the pieces together to assemble a complete general FEM-based Lagrangian CFD scheme. For computational efficiency, we will use the strong mass conservation principle (2.32) to avoid recomputing the velocity mass matrix  $\mathbf{M}^n = \mathbf{M}$  and the energy matrices  $\mathbf{M}_z^{e,n} = \mathbf{M}_z^e$ . A summary of the details of the full, fully-discrete computational scheme is given below.

$$\begin{aligned}\mathbf{F}^n &= \mathbf{p}^n \cdot \mathbf{D}^{e,n} - \mathbf{v}^n \cdot \mathbf{S}^{e,n} \\ \mathbf{M}\mathbf{v}^{n+1} &= \mathbf{M}\mathbf{v}^n + \Delta t(\mathbf{F}^n)^T \cdot \mathbf{1}^e \\ \mathbf{M}_z^{\rho,n+1} \mathbf{r}_z^{n+1} &= \mathbf{m}_z^0 \\ \mathbf{M}_z^e \mathbf{e}_z^{n+1} &= \mathbf{M}_z^e \mathbf{e}_z^n - \Delta t \mathbf{F}_z^n \cdot \mathbf{v}_z^{n+\frac{1}{2}} \\ \mathbf{M}_z^{\rho,n+1} \mathbf{p}_z^{n+1} &= (\gamma - 1) \mathbf{M}_z^e \mathbf{e}_z^{n+1}\end{aligned}$$

The sequence of events is that at the beginning of each time step we calculate local “corner forces”

$$(\mathbf{F}_z^n)_{i,j} = \int_{\Omega_z(t_n)} \left( p^n I - \mu_z^n \vec{\nabla} \vec{v}^n \right) : \vec{\nabla} \vec{w}_i \phi_j$$

on the current mesh  $\Omega^n$  which we then assemble into a global force vector  $\mathbf{F}^n$ . We then use the momentum conservation equation to solve for accelerations and the new velocity degrees of freedom. After moving the mesh from  $\Omega^n \mapsto \Omega^{n+1}$ , via

$$\mathbf{x}^{n+1} = \mathbf{x}^n + \Delta t \mathbf{v}^{n+1},$$

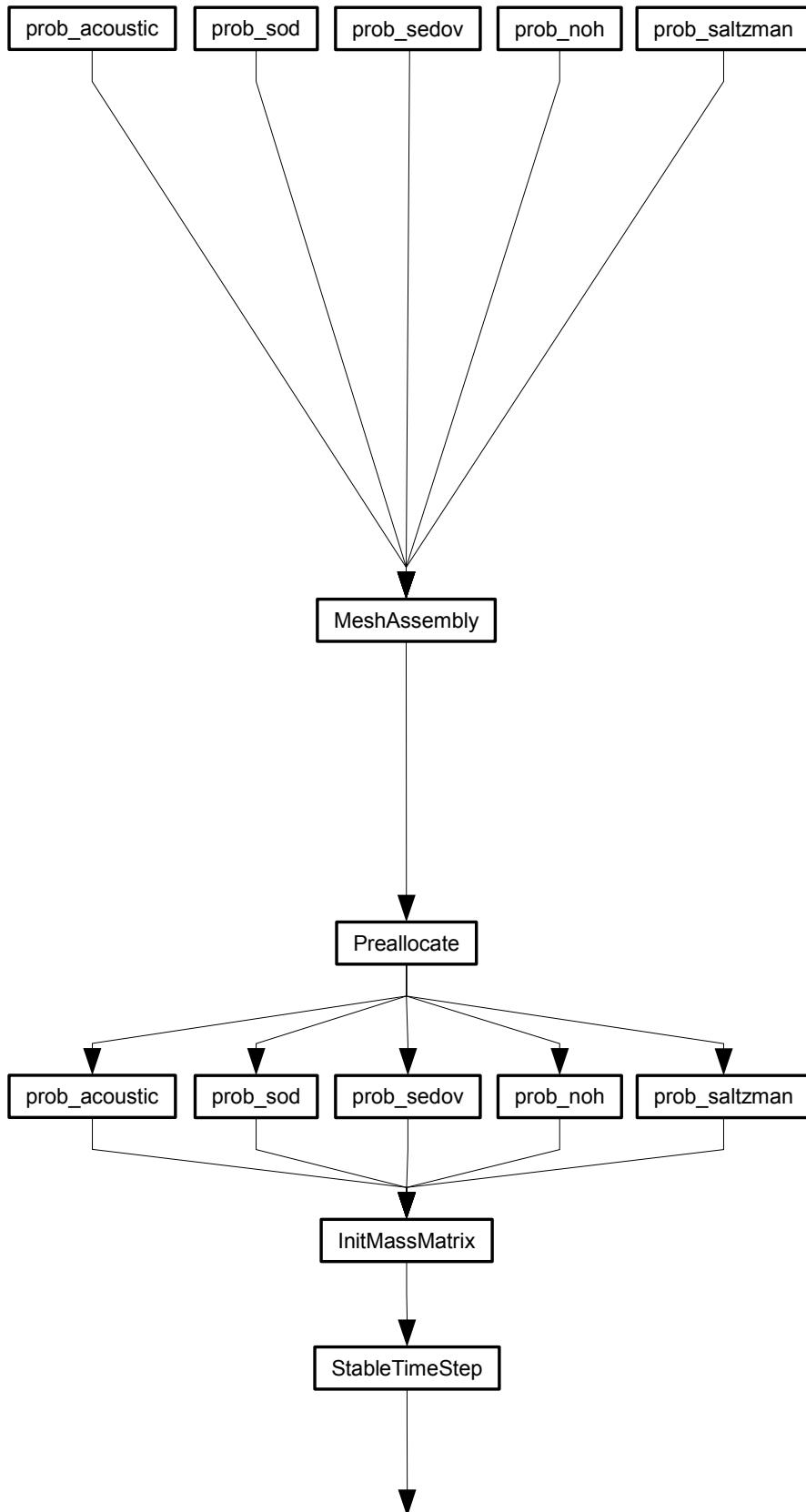
we can update the new density and energy degrees of freedom. Using the equation of state, we can calculate new pressures and move on to the next time step.

# Chapter 3

## Implementation Details

In this chapter we take a walk through the Matlab code to examine various implementation details. Figure 3.1 diagrams an overview of workflow in `Fermium` (Finite Element Research Methods Implemented Using Matlab). This is only a testbed for various high-order finite element methods on a couple of canonical test problems. Therefore no effort has gone toward making a general interface to set up problems of more practical engineering interest such as the ability to work with unstructured grids or define multiple materials, although the FEM theory handles these cases with ease. Also, `Fermium` was developed in Matlab as an initial prototype for a more powerful and general FEM testbed currently being developed in C++.

In order to facilitate debugging and reduce overhead and CPU time, most of the code functions as one giant script file with few actual function calls. This is done via a useful property of Matlab: the ability to embed scripts in other scripts. This allows us to minimize code redundancy while easing the job of debugging and analysis.



## Workflow

- Define Reporting
  - Set plot frequency
  - Set data save frequency
- Define FEM
  - Choose velocity/position basis function
  - Choose pressure/density basis function
  - Toggle low/high order energy
  - Choose quadrature order
- Define Solver Settings
  - Toggle mass lumping
  - Toggle mass update
  - Toggle hourglass filter
  - Adjust artificial viscosity
- Define Discretization Properties
  - Initial domain size
  - Domain refinement
  - Domain transformation
  - Time step / stop time

- Compute  $Q$ , Mesh Topology
- If Higher Order
  - Define edge nodes
  - Define center nodes
  - Assemble high-order topology info
- Define Pressure DOF Mapping
- Assemble List of Boundary DOF
- Perform Mesh Transformation

- Preallocate Memory Space
- Compute Quadrature Values

- Initialize Flow Field

- If Strong Mass Conservation
  - One time assemble mass matrix

- Compute Stable Time Step

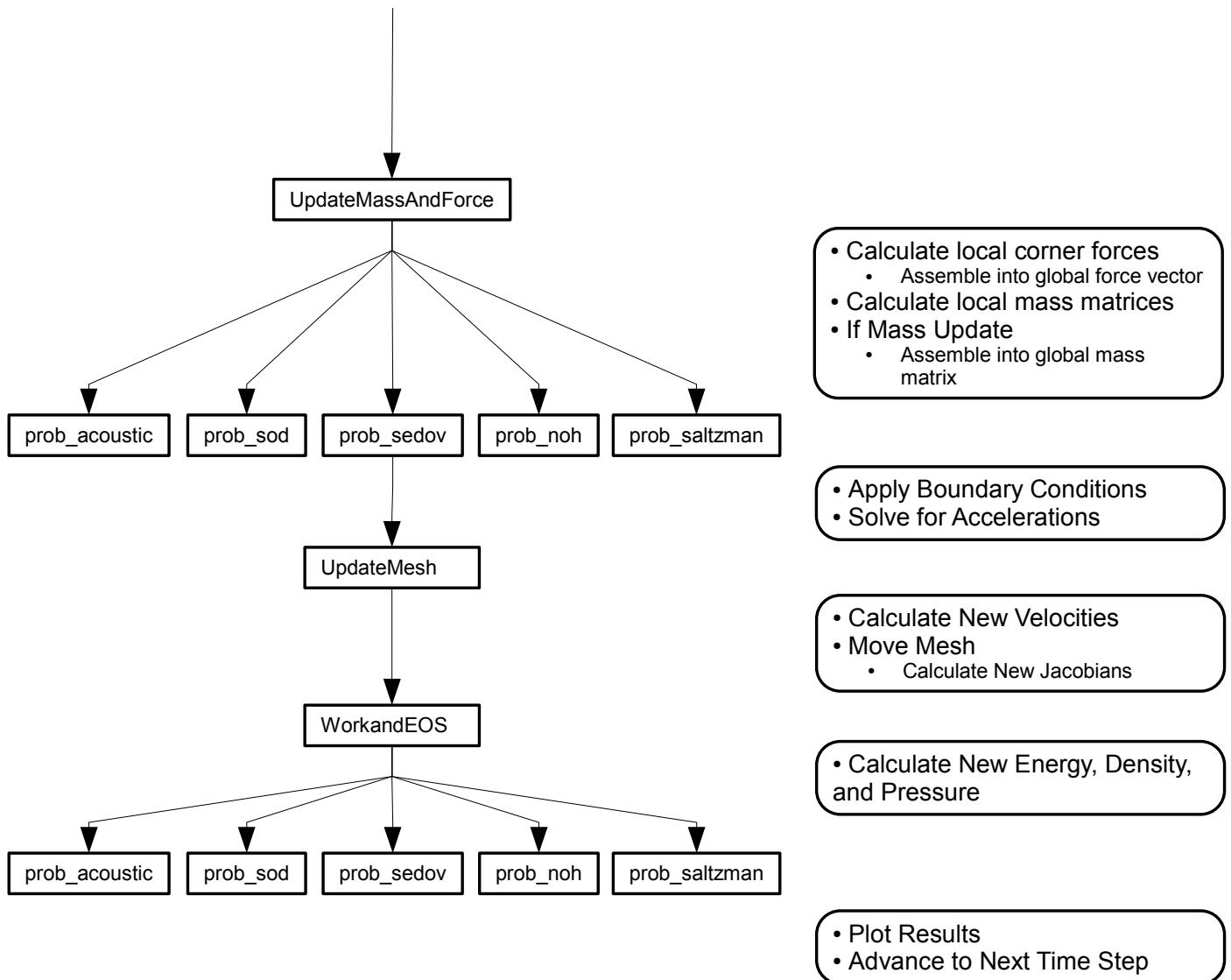


Figure 3.1: Workflow diagram

## 3.1 The Drivers

Starting at the top of the workflow diagram, we see the drivers: scripts which define a problem and connect all of the pieces necessary to solve it. We have implemented several classic test problems as drivers including the acoustic wave problem, the Sod shocktube, the Sedov Explosion, the Noh implosion, the piston driven shock, and the Saltzman piston.

The first thing that we need to set in our code is the method that we want to use to solve the problem. `Fermium` is general enough to allow for the kinematic and thermodynamic basis functions to be chosen independently. Hence, we have the following options available for the velocity/position basis functions: bi-linear, bi-quadratic, and bi-linear with bubble, as illustrated in Figure 3.2. All of these basis functions are defined on quadrilaterals, but there is nothing to prevent us from trying triangles or any other shape (other than the added complexity of going to unstructured grids). We have many more thermodynamic basis functions at our disposal, as depicted in Figure 3.3. At this point higher order energy is still in an experimental stage. From some initial results, it looks like we are able to achieve higher order pressure and density behavior while maintaining piecewise constant energy representations. Therefore, although we maintain a switch to use a higher-order energy representation, we usually represent energy as a  $Q_0$  field.

Furthermore, the driver has switches to toggle several other solver options. We have derived the concept of *strong mass conservation* (see Section 2.4.3), which allows us to avoid recomputing the mass matrix every cycle. The driver script has a switch called `MassUpdate` that allows us to forego recalculating our mass matrix every cycle. Another consequence of strong mass conservation is that  $\rho|J| = \text{const.}$  Therefore, if we use  $\rho|J|$  as our finite element expansion rather

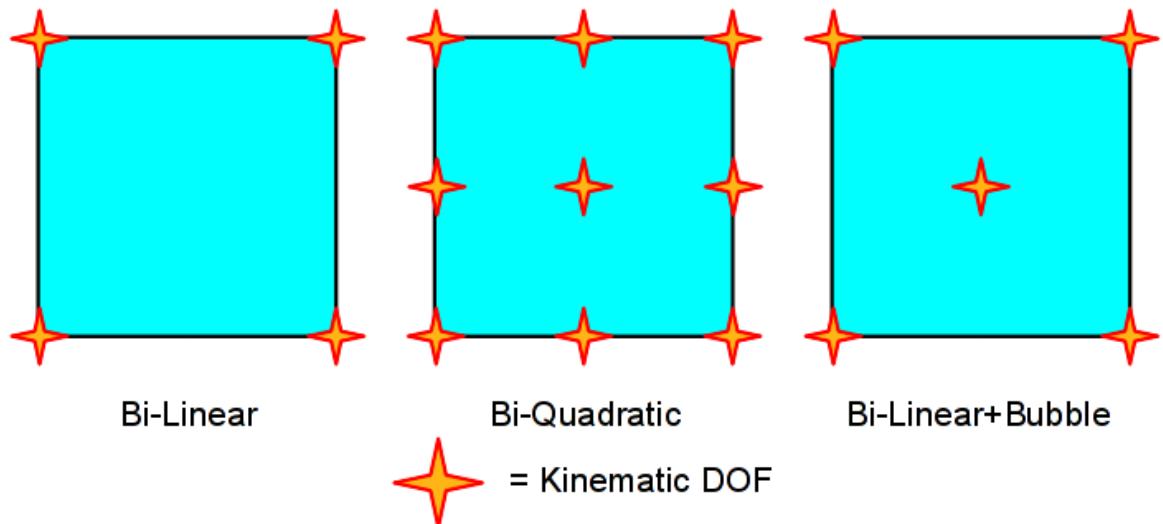


Figure 3.2: Options for kinematic basis functions

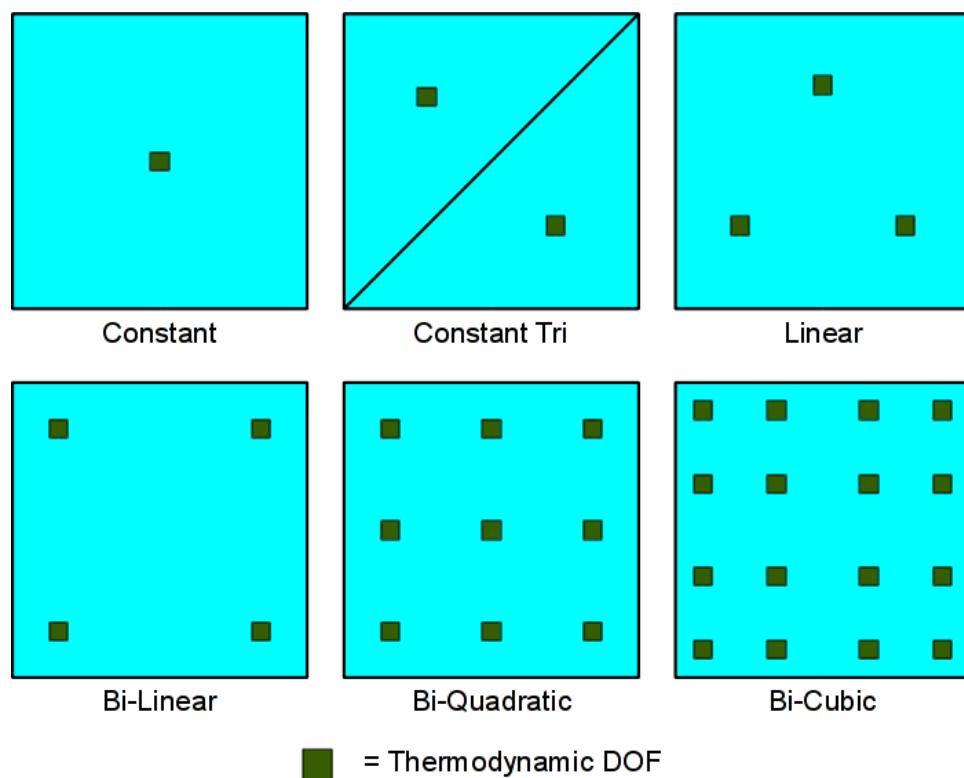


Figure 3.3: Options for thermodynamic basis functions

**Table 3.1: List of switches and knobs**

Switch or Knob	Possible Values	Typical Value
recover	false, path to recover file	
SaveFigures	true, false	
nplots	$N \in [0, 1, 2, \dots]$	20
nsaves	$N \in [0, 1, 2, \dots]$	10
dtInit	$\mathbb{R} \in (0, \infty)$	1e-3
dtMax	$\mathbb{R} \in (0, \infty)$	1e-2
maxcycle	$N \in [1, 2, \dots]$	1e5
VBasis	$\mathbb{C}Q1Basis, \mathbb{C}Q2Basis, \mathbb{C}Q1bBasis\dots$	
PBasis	$\mathbb{C}Q0Basis, \mathbb{C}P1Basis, \mathbb{C}Q1dBasis\dots$	
HighOrderEnergy	true, false	
StrongMass	true, false	
FullMassMatrixSolve	true, false	
MassUpdate	true, false	
QuadOrder	$N \in [1, 2, \dots]$	
hgfrac	$\mathbb{R} \in [0, \infty)$	1
Qfrac	$\mathbb{R} \in [0, \infty)$	1
qquad	$\mathbb{R} \in [0, \infty)$	1
qlin	$\mathbb{R} \in [0, \infty)$	1

than  $\rho$ , we don't have to do a density update. The **StrongMass** switch changes the thermodynamic variables to this representation. This is also somewhat in the experimental stage, so we will mostly consider traditional finite element spaces in this research. There are also knobs to turn on and control hourglass filters and artificial viscosity. In addition to all of this, we use the driver to set the Gauss-Legendre quadrature order and the initial and maximum time steps. The mesh geometry and refinement are also controlled in the header of the driver script. We can also set plot and save-file frequency. Table 3.1 lists the various switches and knobs that are available in the header along with a few typically used values.

## 3.2 Mesh Assembly and Variable Preallocation

In the interests of code re-use, we have moved as much functionality as possible to the common files like `MeshAssembly.m` and `Preallocate.m`. The `MeshAssembly` script fleshes out the details of the FEM chosen: defining the number of spatial and thermodynamic DOFs per zone and their location within the reference element. We also need to generate a mesh and define the connectivity of the elements. The function `ComputeReferenceMeshNodes` creates a Cartesian mesh,  $x_{\min} < x < x_{\max}$  by  $y_{\min} < y < y_{\max}$ , while `ComputeMeshTopology` defines the connectivity: which nodes belong to which quad. If the spatial discretization is higher order, we go on to further refine the mesh. If the element has additional edge nodes we place these and, along with the corner connectivity, add the edge connectivity information to `Quadmap`. Additionally, if the element has center nodes, we add this information to the end of `Quadmap`. Therefore, the velocity nodes within each element are numbered according to Figure 3.4 with the corner nodes listed first and center nodes listed last. For data structure reasons, we have formulated each column of `Quadmap` as an element and each row as a spatial DOF connected to that element.

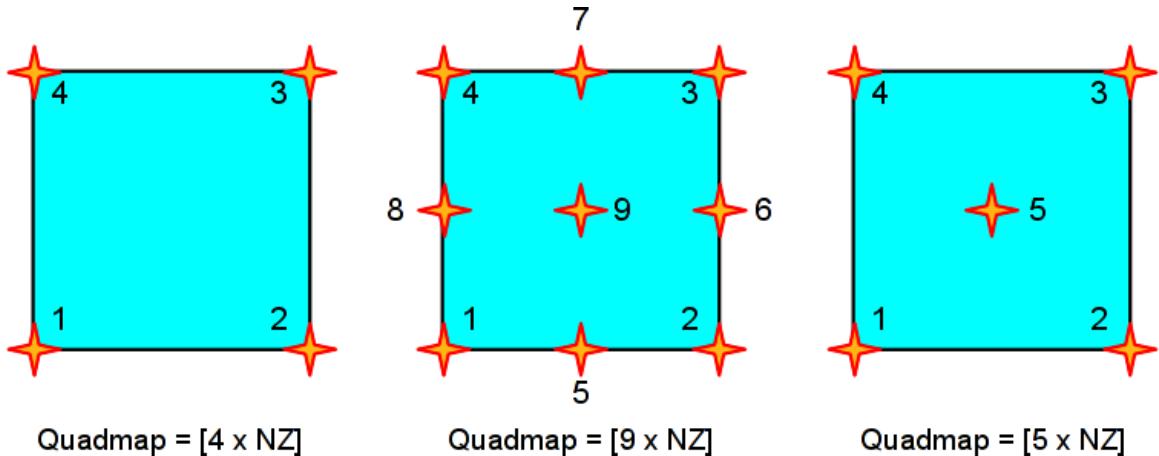


Figure 3.4: Velocity node numbering

The pressure map is a lot simpler to set up because there is no sharing of DOFs between elements. The `PressureMap` array is as simple as (3.1), where each column refers to a different element and each row is a different pressure DOF belonging to that element

$$\text{PressureMap} = \begin{bmatrix} 1 & npdof + 1 & 2 \cdot npdof + 1 & \cdots & (NZ - 1) \cdot npdof + 1 \\ 2 & npdof + 2 & 2 \cdot npdof + 2 & \cdots & (NZ - 1) \cdot npdof + 2 \\ \vdots & \vdots & \vdots & \ddots & \vdots \\ npdof & 2 \cdot npdof & 3 \cdot npdof & \cdots & NZ \cdot npdof \end{bmatrix} \quad (3.1)$$

The next step is to determine which DOFs are on which boundary. This is done with the `ComputeBoundaryDOFQuad` function. `MeshAssembly` also performs any grid transformation called for by the driver, including randomly distorting the mesh, applying the Saltzman mesh pattern, rotating, or refining the mesh.

The `Preallocate` m-file mostly just preallocates empty matrices and vectors for speed (as suggested by Matlab). It also evaluates the basis functions, their derivatives, and the Jacobian at the preset quadrature points.

### 3.3 Initializing the Flow Field

`Preallocate.m` then returns control back to the driver where the flow field is initialized. Here the initial velocity profile and state variables are specified. After this, the `InitMassMatrix` script is called. If we are using strong mass conservation to avoid recalculating the mass matrix every cycle, this script assembles the global mass matrix which will be used for every subsequent time step. Once this

is done, we go into the time stepping loop.

## 3.4 Solving the Momentum Equation

The first thing that we need to do at each step is to calculate a stable time increment using the `StableTimeStep` script. For traditional SGH with an explicit time-marching scheme, this typically involves the CourantFriedrichsLowy[10] condition. We want the time step to be such that a wave will not propagate more than the length of one cell within one time step. This can be written

$$\Delta t < C \frac{h}{V}$$

where  $h$  is a measure of the length across a zone,  $V$  is the maximum velocity in the zone, and  $C$  is typically 1/2. We have not yet been able to perform a full stability analysis of high-order curvilinear finite elements, so we have chosen a very strict definition of a stable time step. We define  $h$  to be the minimum distance between any two velocity DOFs,  $V$  to be the maximum velocity in the domain, and  $C$  to be 1/4. This may be overly harsh, but it ensures a stable solution until a more detailed analysis can be performed.

Once we have calculated a stable time step, we dive into the `UpdateMassAndForce` script file. This script uses the current state of the pressure, velocity, and mesh to calculate forces exerted on each spatial DOF as well as the updated mass matrix (if `MassUpdate` is turned on). Let's assume for a minute that we are calculating both. To do this, we cycle through every element in the mesh and calculate a local mass matrix and ‘corner’ force using either the `elemQ1` or `elemQ2` (this actually works for any higher order velocity basis) function. Hourglass filtering is the

only real difference between these two functions. We are still using a very primitive form of an hourglass filter for higher-order elements, so we are handling the  $Q_1$  elements, with their well-developed hourglass functions separately. The first step in the `elem*` functions is to calculate a local mass matrix from the spatial degrees of freedom. After this, we build up the corner forces with contributions from pressure gradients, artificial viscosity, and hourglass forces as described in Chapter 2.

Once these corner forces and local mass matrices are assembled into the global system via the connectivity information, control is returned to the driver where the boundary conditions are applied and the global systems  $\mathbf{M}\vec{a}_x = \vec{F}_x$  and  $\mathbf{M}\vec{a}_y = \vec{F}_y$  are solved for the accelerations. This can be done via mass lumping, which accumulates all off-diagonal information to the diagonal of the matrix or a full mass matrix solve.

## 3.5 Mesh Movement and Thermodynamic Update

Once the nodal accelerations have been solved for, the driver gives control to the `UpdateMesh` common script. This script first accelerates the velocity DOFs according to (3.2), then it applies any source terms to the velocity before moving the mesh nodes according to (3.3).

$$\text{NEWvelocity} = \text{OLDvelocity} + dt * \text{acceleration} \quad (3.2)$$

$$\text{allnodes} = \text{allnodes} + dt * \text{NEWvelocity} \quad (3.3)$$

`UpdateMesh` then calculates the new Jacobian matrix for every element at every quadrature point for use in the next time step. Finally, it checks for any points where the determinant of the Jacobian is zero or negative, which would indicate that the solution has gone unstable.

Now that we have a new mesh geometry, we can calculate the new thermodynamic properties on this updated mesh using `WorkandEOS`. Please note that the thermodynamic properties are discontinuous between zones, this allows us to loop over ever element and calculate the new state independent of all other zones without assembling any sort of global system. Thus for each zone, we first calculate the change in internal energy with the dot product of the velocity DOFs and corner forces. We then proceed to the density update. As mentioned previously in Section 3.1, if we use the basis functions to represent  $\rho|J|$  rather than just  $\rho$ , this function is constant and no density update is required. However, if we are using a traditional representation of density, then we need to compute new density DOFs. The first step is analogous to assembling a local mass matrix, but instead of using the spatial basis functions, we use the density basis functions. We then solve a small `npdof`  $\times$  `npdof` system of equations for the new density degrees of freedom as explained in Section 2.4.3. Now that we have the energy and density, we can update the pressure pointwise according to the equation of state.

## 3.6 Plotting

Finally, now that all variables have been updated for the next time step, we can move back to the driver for the plotting stage. We first check that this is one of the desired plot or save time steps. We have developed several finite element

specific plotting routines such as `PlotFEMContourf` and `PlotFEMMesh` to accurately represent any arbitrary basis function to any user-specified precision. The driver plots the fields as called for, saves the figures / variable data if requested, then loops back to `StableTimeStep` for the next time step. This whole process repeats until `tstop` is reached or `WorkandEOS` detects that the solution has gone unstable.

# Chapter 4

## Some Elements Considered

We have derived a general finite element scheme for Lagrangian computational fluid dynamics. This general framework has made it very easy to consider a number of different mixed finite element formulations. We have only considered quadrilateral 2-D elements in this research, but the formulation is general enough to allow for triangles or even 3-D tetrahedrals or hexahedrals.

### 4.1 Some Guidelines

In his canonical textbook on the finite element method, Hughes[20] describes some requirements for stable mixed finite element pairs for Stokes flow. The most comprehensive measure of stability is the so-called Ladysenskaja-Babuska-Brezzi (LBB) stability condition. This analysis is far from trivial and several easier estimates of stability have been derived. To our knowledge, no detailed analysis of the LBB condition for Euler's equations has been published. Thus, we used Stokes flow analysis (which has been very thorough), to guide our initial steps in looking for stable basis function pairs.

Constraint counting is a simpler heuristic approach for determining the suitability of a basis function pair for simulating incompressible Stokes flow. In short, this is the ratio of global kinematic to thermodynamic DOFs after boundary conditions have been applied:

$$r = \frac{N_v}{N_p}$$

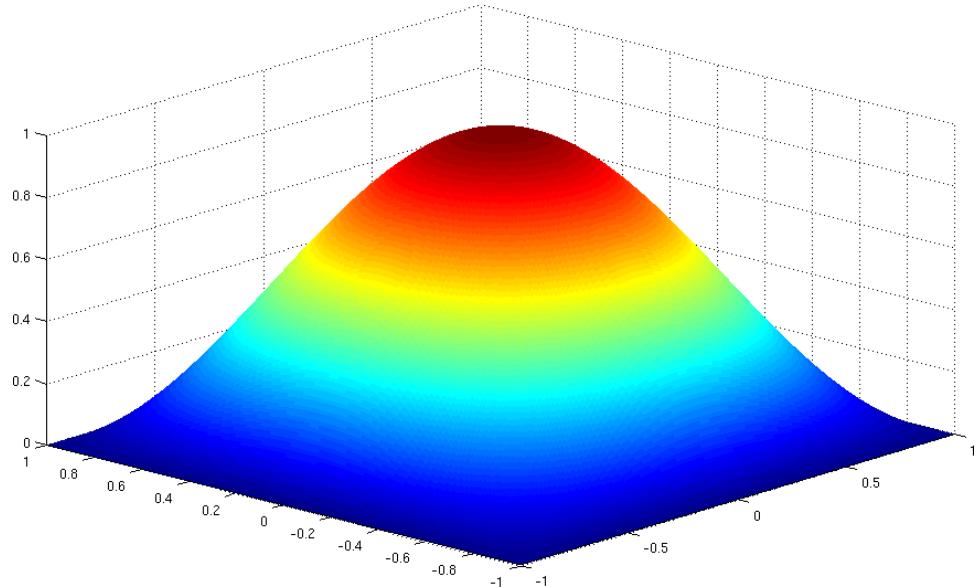
where  $N_v$  is the total number of velocity degrees of freedom and  $N_p$  is the total number of pressure or density degrees of freedom. The ideal value of  $r$  is the number of spatial dimensions,  $nD$ , in the problem (i.e. 2 for two dimensions, 3 for three). A value of less than  $nD$  would indicate a tendency of the elements to lock and a larger value would predict poorly approximated incompressibility conditions. Thus, in two dimensions a traditional  $Q_1-Q_0$  element would have the ideal ratio of 2. We have not yet determined if or how this approach translates to the Euler equations, but it informed our early choices and it appears that pairs that do not satisfy this condition have similar failings.

## 4.2 Convergence Tests

The most important characteristic of a basis function pair is that it is able to converge to the correct solution under refinement. We have developed two simple static problems to test the accuracy and convergence for the two biggest parts of the momentum equation. The  $\text{grad}(p)$  problem tests convergence to the exact solution to the pressure gradient in the momentum equation, while the  $\text{div}(\text{grad}(V))$  problem tests the discrete representation of the  $\text{div}(\text{grad}())$  operator in the artificial viscosity representation.

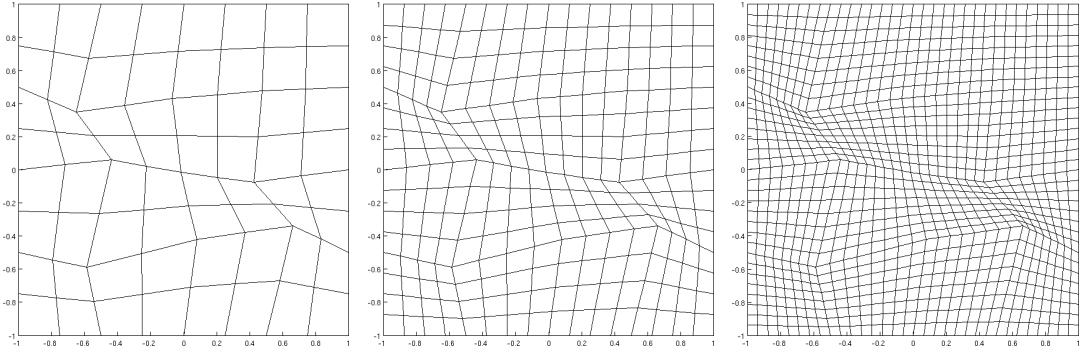
### 4.2.1 The $\text{grad}(p)$ Test

Before progressing to further study, it is essential to test whether a specified mixed finite element pair can solve the simplest of problems. With this in mind, we devised a static pressure gradient test to see whether each method could accurately predict nodal accelerations for a specified pressure field. So the pressure field  $p = \cos(\frac{\pi}{2}x) \cos(\frac{\pi}{2}y)$ , shown in Figure 4.1, is mapped onto two series of meshes on  $x, y \in [-1, 1]$ , one perfectly Cartesian and one distorted, Figure 4.2 . The convergence rate is found by calculating the  $L^2$  error norm on these two sets of grids. We first calculate the error due to the projection of an exact pressure field onto its basis function representation, then we calculate the error of the static momentum solve.



**Figure 4.1: Initial pressure field**

This pressure field was chosen because  $p$  is zero on all boundaries (thus eliminating the surface integral in the finite element formulation), and there is a simple analytical expression for the exact accelerations,  $a = \begin{bmatrix} \frac{\pi}{2} \sin(\frac{\pi}{2}x) \cos(\frac{\pi}{2}y) \\ \frac{\pi}{2} \cos(\frac{\pi}{2}x) \sin(\frac{\pi}{2}y) \end{bmatrix}$ .



**Figure 4.2: Series of refined distorted meshes**

#### 4.2.2 The $\text{div}(\text{grad}(V))$ Test

The  $\text{div}(\text{grad}(V))$  test was devised to measure the accuracy and convergence rates of the several methods in approximating the  $\text{div}(\text{grad}())$  operator, a calculation that is imperative for the tensor artificial viscosity term in shock problems. In this problem, we project a velocity field,

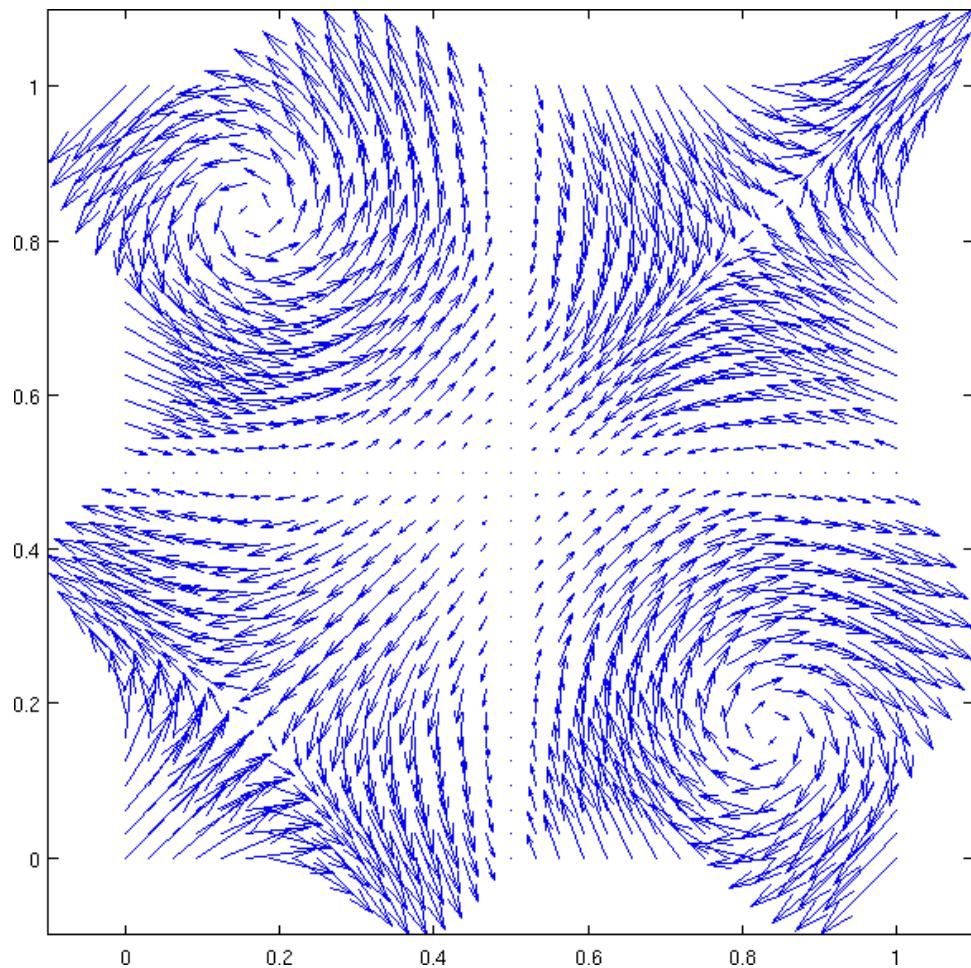
$$V_x = \cos(\pi x) \cos(3\pi y)$$

$$V_y = \cos(3\pi x) \cos(\pi y)$$

onto the mesh,  $x, y \in [0, 1]$ . This velocity field looks like Figure 4.3. We use the same sequence of Cartesian and distorted meshes to establish convergence rates. The first measure of a method is that projection of the velocity field onto the velocity basis functions converges to the exact field under refinement. Finally, we check that the predicted accelerations due to the artificial viscosity converge to the analytically predicted value, (4.1).

$$\begin{aligned} a_x &= -\pi^2 \cos(\pi x) \cos(3\pi y) + 9\pi^2 \cos(\pi x) \cos(3\pi y) \\ a_y &= -9\pi^2 \cos(3\pi x) \cos(\pi y) + \pi^2 \cos(3\pi x) \cos(\pi y) \end{aligned} \tag{4.1}$$

This velocity field was chosen for simplicity's sake in order to eliminate the boundary terms in the weak form description. These boundary terms vanish to zero if  $\vec{\nabla}V \cdot \vec{n} = 0$ , where  $\vec{n}$  is the outward facing unit normal vector of the domain. A simple analysis will confirm that this vector field does indeed satisfy this requirement for this particular domain. We also chose this vector field because it is complicated enough to run our method through the ringer. We have regions of converging, diverging, and swirling flow. Our stiffness matrix needs to be able to capture the details of some complicated flow fields.



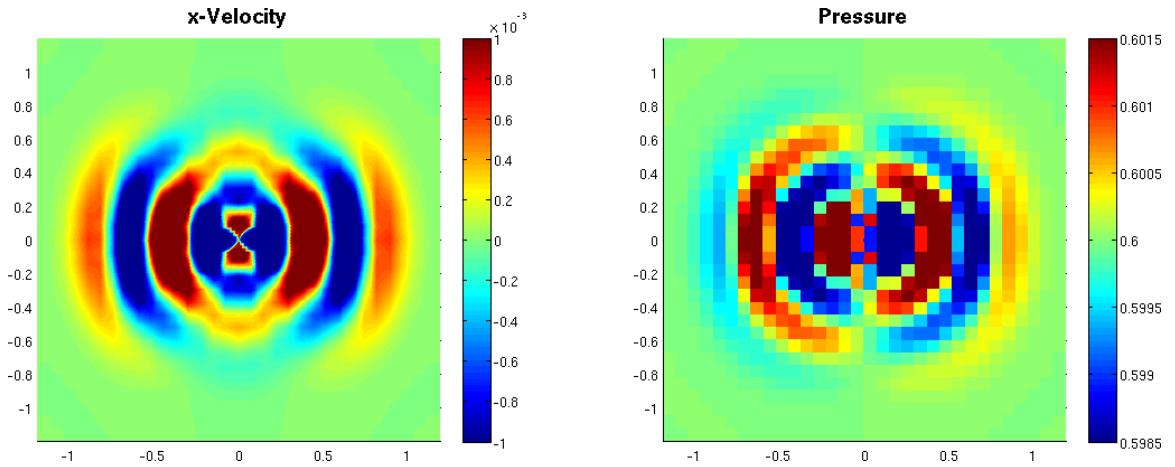
**Figure 4.3: Projected velocity vector field**

## 4.3 The Acoustic Wave Test

We discovered that a delta function disturbance is the surest way of exciting hourglass modes. In light of this, we developed the acoustic wave test problem to reveal whether a mixed finite element pair exhibits tendencies for hourglass mode instabilities. In this test, a single node in an initially Cartesian mesh is oscillated in time with a small amplitude. Because this is supposed to be a smooth problem and no shocks should show up, all artificial viscosity is turned off. Ideally, in an hourglass-free formulation, a simple acoustic wave will propagate through the mesh.

While useful for qualitative analysis, this test problem has several problems that exclude it from a more quantitative analysis. While it may be possible to derive an exact solution to this problem, none has been derived to our knowledge. But this is not the primary concern with this test problem. In fact, there would not be very much benefit in deriving an exact solution because it is impossible to achieve a converging solution under refinement. This is because of an inherent contradiction in the problem definition. In order to produce comparable results between methods and different mesh refinements the acoustic wave should introduce the same amount of energy to the problem. Suppose that we want to introduce a total amount of kinetic energy,  $KE = \int_t \frac{1}{2}mv^2$ . Therefore, in order to achieve a proportionate energy, the driving velocity must be proportional to  $amp_v \propto \sqrt{\frac{KE}{m_{driver}}}$ , where  $amp_v$  is the amplitude of the driving velocity function and  $m_{driver}$  is the mass of the driver node. Now, under refinement,  $\lim_{N_z \rightarrow \infty} m_{driver} = 0$ , therefore  $\lim_{N_z \rightarrow \infty} amp_v = \infty$ . So under refinement, as each zone shrinks in size, the driving amplitude will grow in magnitude, and adjacent cells will be crushed.

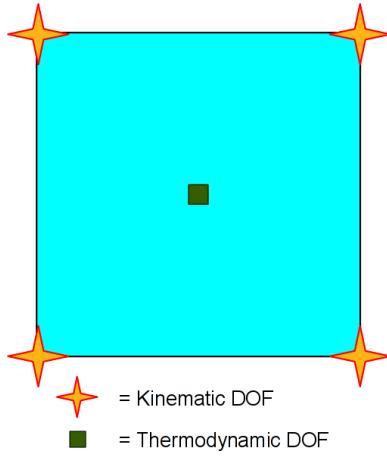
For our purposes, a  $32 \times 32$  mesh provides enough resolution to accurately capture the physics while clearly illustrating hourglass modes and being coarse enough to avoid the trouble mentioned above. Ideally, we should just see the acoustic wave without any hourglass / checkerboard modes showing up. While not perfect, the  $Q_1-Q_0$  element with a traditional hourglass filter, Figure 4.4, gives us a good idea of what we are looking for.



**Figure 4.4:** Traditional  $Q_1-Q_0$  acoustic wave with hourglass filter

## 4.4 The $Q_1-Q_0$ Element

Foremost among our goals is to compare the performance of any new methods with the traditional formulation, the tried and true low order  $Q_1-Q_0$  element pair. This element has a bi-linear kinematic interpolation and a discontinuous-between-cells constant thermodynamic interpolation. A diagram of the master  $Q_1-Q_0$  element is shown in Figure 4.5.



**Figure 4.5: The traditional  $Q_1 - \hat{Q}_0$  element**

#### 4.4.1 Assets

This element has the optimal constrain ratio of  $nD$ . It is simple and the legacy pair for hydrocodes, hence the theory is well developed and the problems are understood and have mostly been addressed in one form or another.

#### 4.4.2 Liabilities

This finite element pair has some long-standing issues because it is fundamentally unstable according to the LBB condition. Hourglass/checkerboard modes can render this method useless without an hourglass filter. Even with an hourglass filter, this method requires custom tuning for different problem types because there is no one ‘safe’ value of the hourglass filter. This method is also low order, so it cannot accurately represent curved boundaries.

### 4.4.3 Convergence

We will be using this method as the ‘control group’ with which we will compare all other methods.

#### The $\text{grad}(p)$ Test on a Cartesian Mesh

On a straight Cartesian mesh the basis function representation of pressure converges linearly with a rate of one. This is what we would expect with piecewise constant pressures.

The final solution converges at a super-linear rate of 1.52. This can be attributed to the well-documented super-convergence of the  $Q_1-Q_0$  element [15].

#### The $\text{grad}(p)$ Test on a Distorted Mesh

The pressure representation converges with a rate of 1.0 on a distorted mesh. We don’t see any sort of super-convergence with the basis function representation. Thus, we don’t see a significant difference between convergence on Cartesian mesh versus a distorted mesh. Henceforth we will report a single value for the convergence of the pressure representation; this will be representative for both meshes.

With a full mass matrix solve, this method converges under refinement at a rate of 1.4, somewhat better than first order but not quite as fast as second order convergence. We will use this as the baseline to compare all other methods against.

### The $\text{div}(\text{grad}(V))$ Test on a Cartesian Mesh

The velocity representation of the continuous vector field converges with a rate of 2.0. This value is representative of both Cartesian and distorted meshes and we will only report one value from now on.

The calculated accelerations converge to the analytical solution at a quadratic rate of 2.0.

### The $\text{div}(\text{grad}(V))$ Test on a Distorted Mesh

On a distorted mesh, the solution converges to a slightly slower rate of 1.52.

#### 4.4.4 Spurious Modes

This finite element pair is notorious for hourglass mode instabilities. In fact, the name “hourglass mode” was coined because of how they manifest themselves on this element. Figure 4.6 illustrates the excited velocity and pressure modes for this basis function pair.

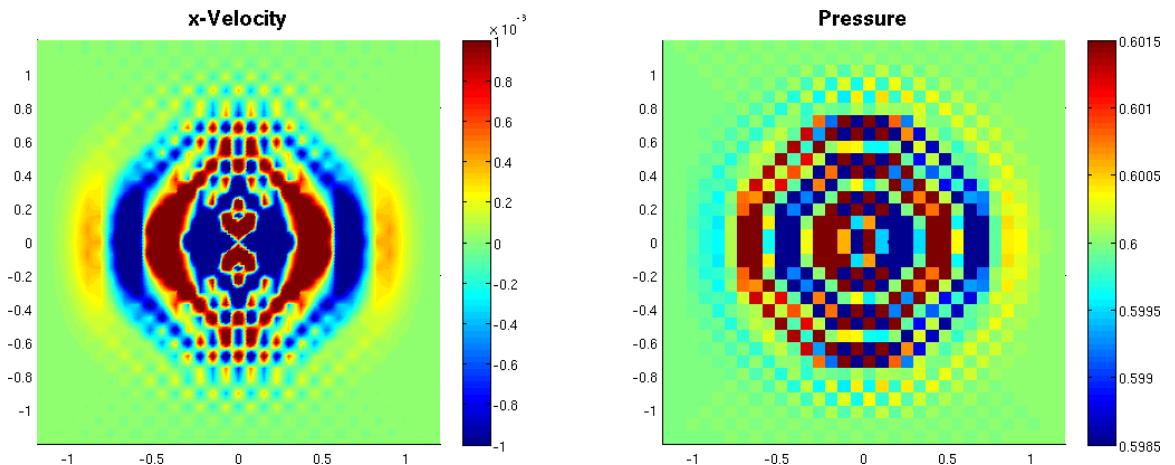
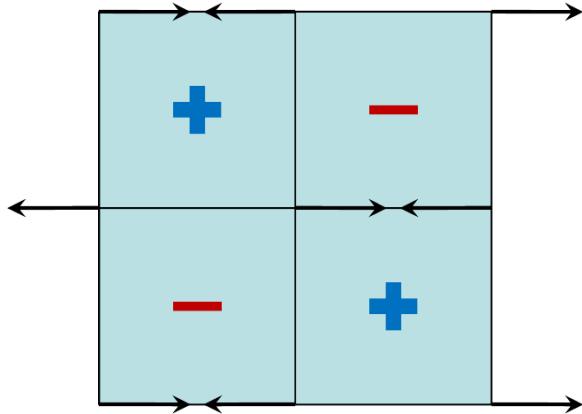


Figure 4.6: Traditional  $Q_1-Q_0$  acoustic wave without hourglass filter

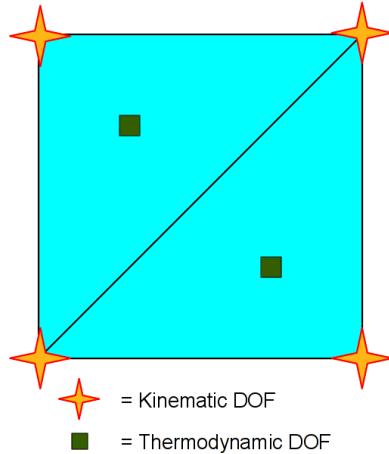
If we take a closer look at the spurious modes, we will see that they take the familiar hourglass/checkerboard form as illustrated on a patch of four elements in Figure 4.7.



**Figure 4.7:** Hourglass/checkerboard modes on a patch of  $Q_1$ - $Q_0$  elements

## 4.5 The $Q_1$ - $P_{0r}$ Element

We include this element in the discussion for the sake of thoroughness, although this element has little potential for a full hydrocode. If we wish to add one more thermodynamic degree of freedom to the  $Q_1$ - $Q_0$  element, we will get Figure 4.8. There are actually several ways that we could have distributed the two pressure DOFs, none of them great. We could have divided the element in half from top to bottom or left to right, but either of these would have doubled the amount of information in one direction versus the other. We have decided to alternatively split the element down one diagonal then the other. This acts to limit the asymmetry of the mesh.



**Figure 4.8:** The  $Q_1-\hat{P}_{0r}$  element

#### 4.5.1 Assets

This element may initially appear to be a straightforward extension of the traditional  $Q_1-Q_0$  element. There are no foreseeable advantages of using this element over any other element. In fact, there are many reasons not to use this element.

#### 4.5.2 Liabilities

In order to limit the inherent asymmetry of this element, it is necessary to alternative placement of the two alternative versions of this element across the mesh. This means changing the software architecture to account for different basis functions depending on whether the element number is even or odd.

#### 4.5.3 Convergence

The pressure field converges to the exact  $\text{grad}(p)$  pressure field at a rate of 1.0, with only slightly less error than  $Q_1-Q_0$ , as shown in Figure 4.9. The calculated

acceleration converges sub-linearly at a rate of 0.54 as shown in Figure 4.10. Apparently, this extra degree of freedom actually slows down the convergence significantly. The  $\text{div}(\text{grad}(V))$  convergence will be the same as  $Q_1-Q_0$ .

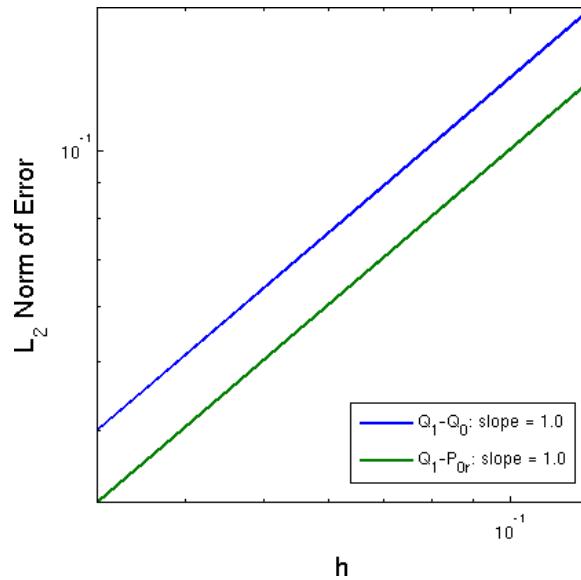


Figure 4.9: Convergence of the  $Q_2-\hat{Q}_1$  pressure field

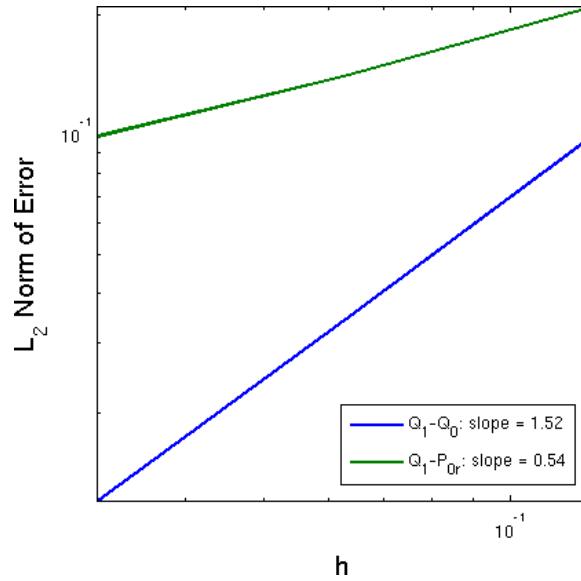


Figure 4.10: Convergence of  $\vec{\nabla}p$  acceleration:  $Q_1-P_{0r}$  versus  $Q_1-Q_0$  on a Cartesian mesh

#### 4.5.4 Spurious Modes

According to Figure 4.11, this element does nothing to eliminate or even reduce spurious velocity and pressure modes.

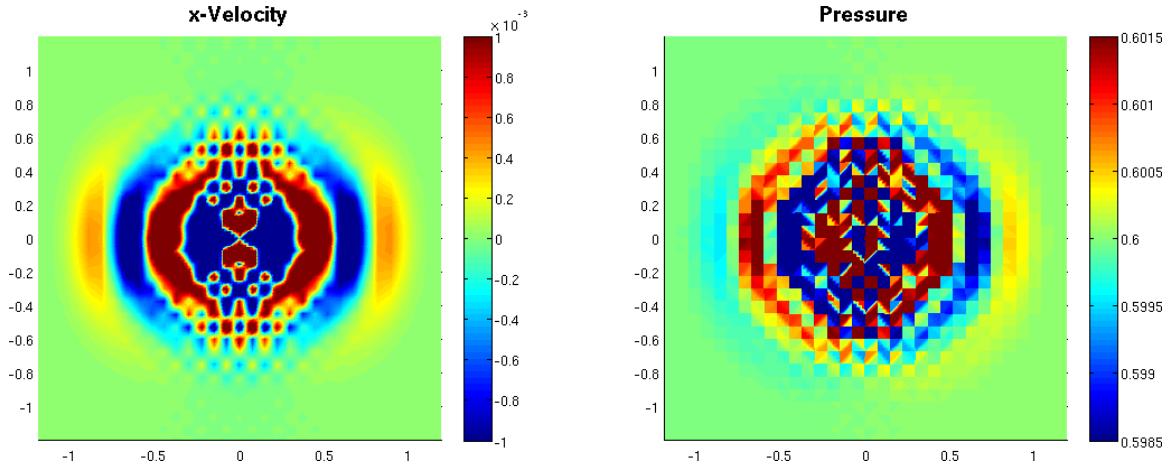


Figure 4.11:  $Q_1-P_0r$  acoustic wave without hourglass filter

#### 4.6 The $Q_1-\hat{P}_1$ Element

We considered the  $Q_1-\hat{P}_1$  and  $Q_2-\hat{P}_1$  elements as a means of breaking up the hourglass modes by breaking the symmetry inherent in quad-quad pairs.

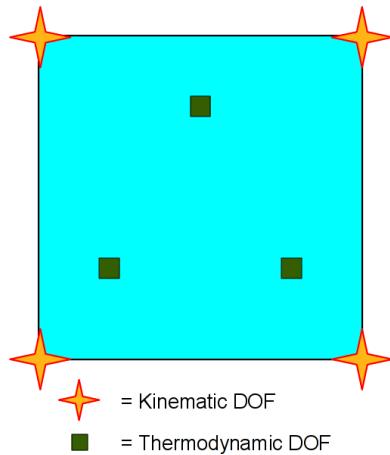
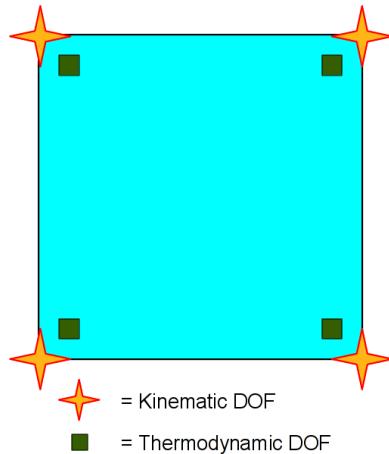


Figure 4.12: The  $Q_1-\hat{P}_1$  element

After some initial research, it appears that this element behaves identically to the  $Q_1-\hat{Q}_1$  element. This actually more of a reflection on  $Q_1-\hat{Q}_1$ ; it reveals that four pressure degrees of freedom are one too many to be paired with  $Q_1$  velocity basis functions. However, because  $Q_1-\hat{Q}_1$  has gained so much recognition because of the method of sub-zonal pressures, we will direct our analysis at that element and limit our discussion of  $Q_1-\hat{P}_1$  to this: all of the assets, liabilities, and convergence properties of  $Q_1-\hat{P}_1$  will be very similar or identical to  $Q_1-\hat{Q}_1$ .

## 4.7 The $Q_1-\hat{Q}_1$ Element

This element pair, which is illustrated in Figure 4.13, is analogous to the method of subzonal pressures described by Caramana and Shashkov[7]. The kinematic interpolation is the same as the traditional  $Q_1-\hat{Q}_0$  element, but four thermodynamic degrees of freedom are used rather than one.



**Figure 4.13:** The  $Q_1-\hat{Q}_1$  element

#### 4.7.1 Assets

On the surface, this method appears to eliminate checkerboard modes from the pressure field. This seems like it could be an attractively easy fix for checkerboard mode instabilities that does not require special tuning between problems. Figure 4.14 shows the resulting average pressures in each cell for the acoustic wave problem.

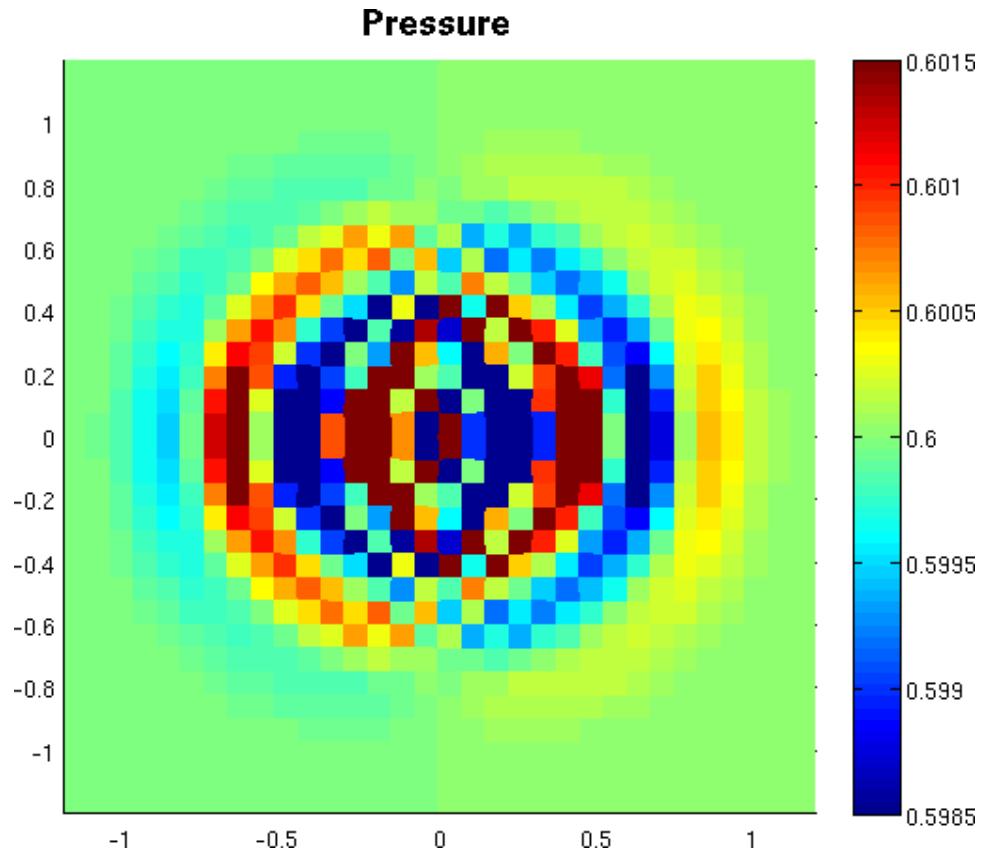
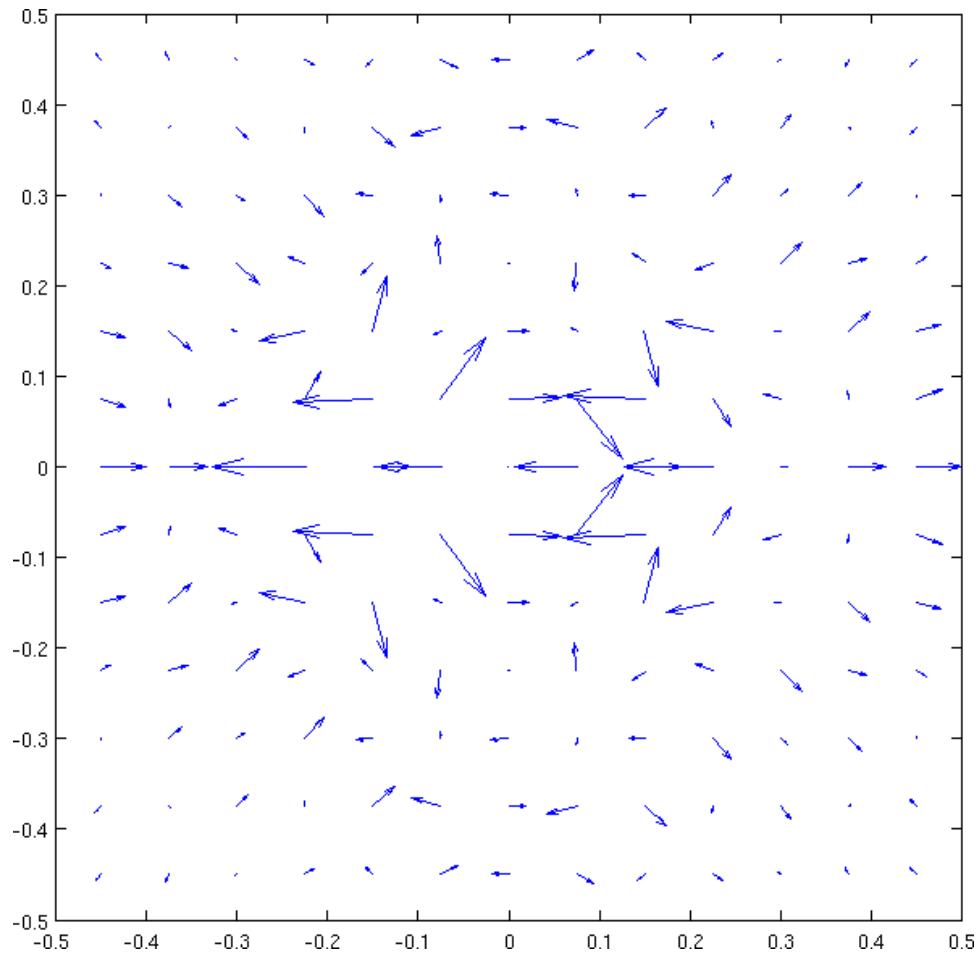


Figure 4.14: Average pressure in each cell for  $Q_1-\hat{Q}_1$  element for acoustic wave problem

#### 4.7.2 Liabilities

The constraint ratio for this element is  $nD/4$ , but let's not let that discourage us. We haven't proven whether this is a viable measure of performance yet.

We have also shown previously in Section 4.6 that this element has one too many thermodynamic degrees of freedom. The major downfall of this element comes when we take a closer look at the resulting velocity field. Apparently, Figure 4.14 just hid the spurious pressure modes by averaging the pressure degrees of freedom because Figure 4.15 shows that spurious velocity modes still exist, although different from the traditional hourglass modes. This may actually be worse because these velocity modes appear to be less structured and possibly harder to control than traditional hourglass modes.



**Figure 4.15:** Spurious velocity modes present in  $Q_1-\hat{Q}_1$

### 4.7.3 Convergence

#### The $\text{grad}(p)$ Test on a Cartesian Mesh

The pressure representation converges to the continuous pressure field at a quadratic rate of 2.0. This element shares the same pressure representation as the  $Q_2-\hat{Q}_1$  element, therefore the pressure field convergence is the same. See Figure 4.27 for the convergence of the pressure field of these two elements.

On a cartesian mesh, the static momentum test converges with a rate of 2.0 as shown in Figure 4.16.

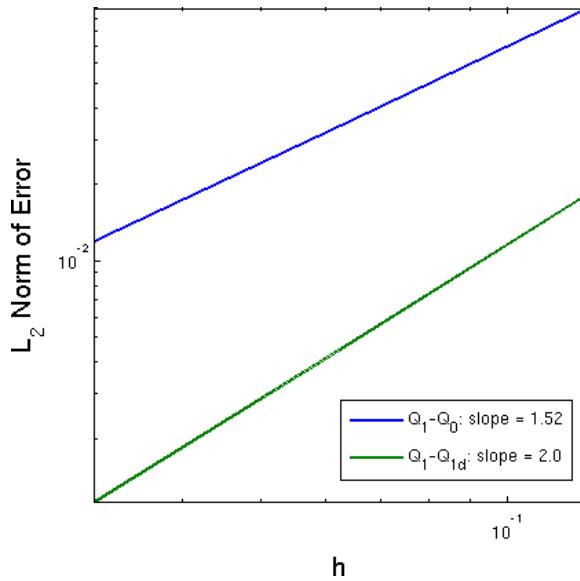
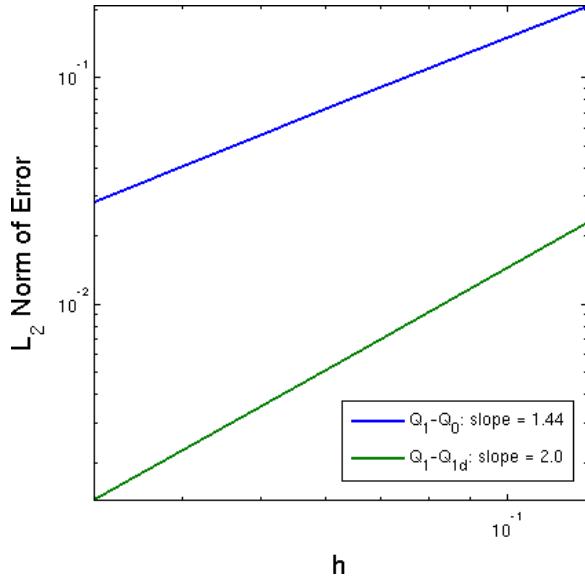


Figure 4.16: Convergence of  $\vec{\nabla}p$  acceleration:  $Q_1-\hat{Q}_1$  versus  $Q_1-Q_0$  on a Cartesian mesh

#### The $\text{grad}(p)$ Test on a Distorted Mesh

Despite its deficiencies, this method does converge to the exact solution on a distorted mesh with quadratic convergence as shown in Figure 4.17.



**Figure 4.17:** Convergence of  $\vec{\nabla}p$  acceleration:  $Q_1-\hat{Q}_1$  versus  $Q_1-Q_0$  on a distorted mesh

#### The $\text{div}(\text{grad}(V))$ Test

It turns out that the stiffness matrix depends entirely on the velocity space. Therefore, the  $Q_1-\hat{Q}_1$  element behaves identically to the  $Q_1-Q_0$  element for  $\text{div}(\text{grad}(V))$  test. The pressure space has nothing to do with the results.

#### 4.7.4 Spurious Modes

If we actually plot the bi-linear interpolation of pressure rather than the average value, we will see that the pressure field is not as pretty as we thought. Figure 4.18 shows the pseudocolor plot for this element on the acoustic wave test.

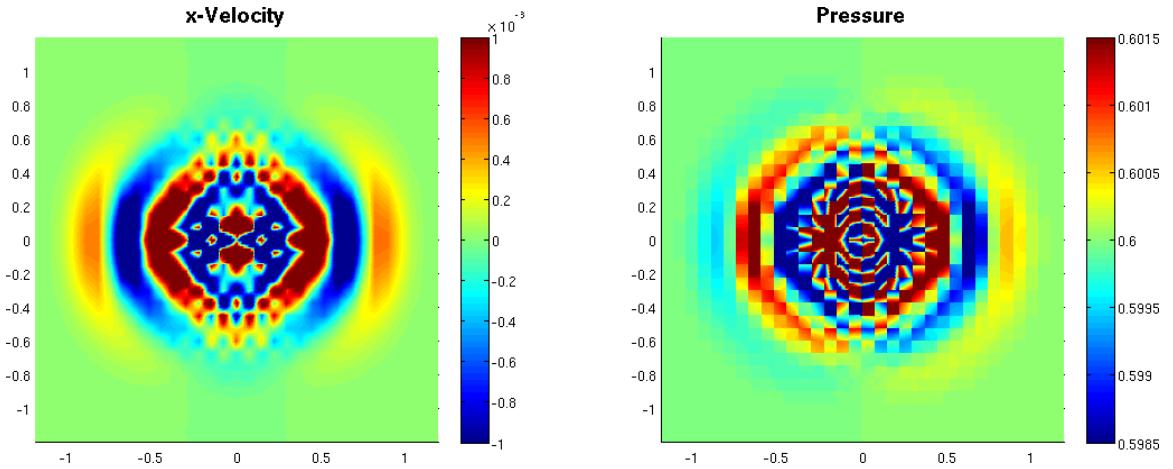


Figure 4.18:  $Q_1-\hat{Q}_1$  acoustic wave without hourglass filter

## 4.8 The $Q_1-\hat{Q}_2$ Element

In order to be completely thorough, we briefly explored further enriching the pressure space, see Figure 4.19. It turns out that there are not enough velocity DOFs to produce a full bi-quadratic pressure field, but unlike the  $Q_2-Q_0$  element which has the reverse problem, this is not a fatal flaw. The extra thermodynamic DOFs end up just taking on a bi-linear shape and thus are superfluous. The same problems plague this element as  $Q_1-\hat{Q}_1$ .

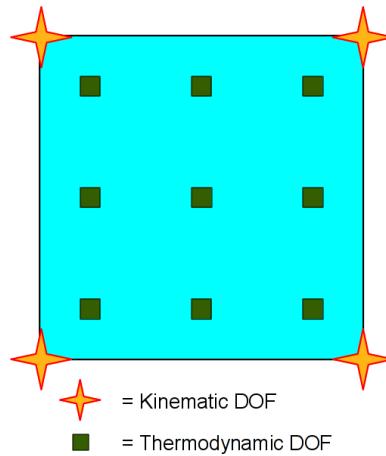
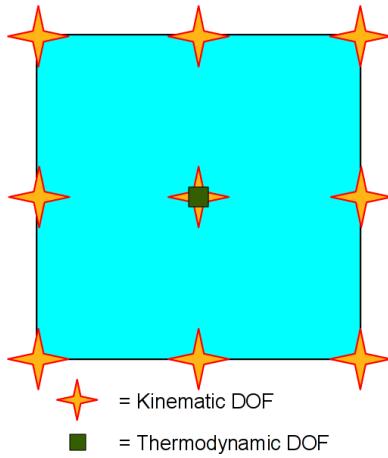


Figure 4.19: The  $Q_1-\hat{Q}_2$  element

## 4.9 The $Q_2-Q_0$ Element

One of our early attempts to correct hourglass mode instabilities was to follow the lead of some research in Stokes flow and enrich the velocity space. This led to the  $Q_2-Q_0$  element and caused us to reconsider the applicability of Stokes flow elements to inviscid flow.



**Figure 4.20:** The  $Q_2-Q_0$  element

### 4.9.1 Assets

Analysis of the inf-sup condition by Stenberg and Suri[33] showed that  $Q_{k+2}-Q_k$  family of elements is stable for Stokes flow. The  $Q_2-Q_0$  element falls into this family, but it is not stable for Euler flow.

### 4.9.2 Liabilities

It turns out that, while this element may work for Stokes flow, there are not enough pressure degrees of freedom to inform the movement of all of the higher order velocity degrees of freedom. The internal degree of freedom, for example,

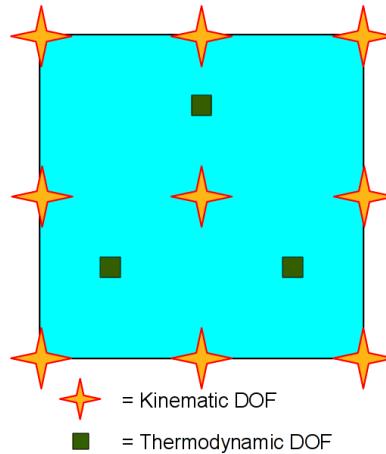
does not ever see any pressure gradient because pressure is constant within each cell. Therefore, the central DOF never moves.

### 4.9.3 Convergence

This finite element pair does not converge at all, which renders it completely unsuitable for a hydrocode.

## 4.10 The $Q_2-\hat{P}_1$ Element

One element that showed some unexpected promise was the  $Q_2-\hat{P}_1$  element. This element has a bi-quadratic kinematic interpolation and a linear thermodynamic interpolation as depicted in Figure 4.21.



**Figure 4.21:** The  $Q_2-\hat{P}_1$  element

### 4.10.1 Assets

With its bi-quadratic spatial interpolation, this element is capable of producing curved geometries. Similar to the other  $Q_2$  elements, this allows us to

calculate second derivatives within a single cell while better representing higher-order phenomena with fewer zones and faster convergence. Three pressure DOFs is also the fewest degrees of freedom needed to fully inform the movement of all 9 higher-order velocity degrees of freedom. This means that we can calculate thermodynamic gradients with ease.

### 4.10.2 Liabilities

Probably the principle oddity of this element lies in the fact that we have “embedded” a 3-point basis function on a quadrilateral. There is no real reason why this shouldn’t be done, at least for the thermodynamic space, but there is an element of uncertainty about this. If we are using quads, shouldn’t we use quad-based basis functions? Where are the most natural locations for these three DOFs? Despite it’s higher order nature, this element appears to have slower convergence characteristics on the  $\text{grad}(p)$  test than traditional methods. Thus it seems that  $Q_2-\hat{Q}_1$  is the preferred higher order method.

### 4.10.3 Convergence

This element has disappointing convergence trends.

#### The $\text{grad}(p)$ Test

The pressure field representation is the same as for  $Q_1-\hat{P}_1$ . The solution converges at a rate of 1.25 which is actually sub-standard compared to  $Q_1-Q_0$ .

When we distort the mesh, we get an even lower convergence rate of 1.20.

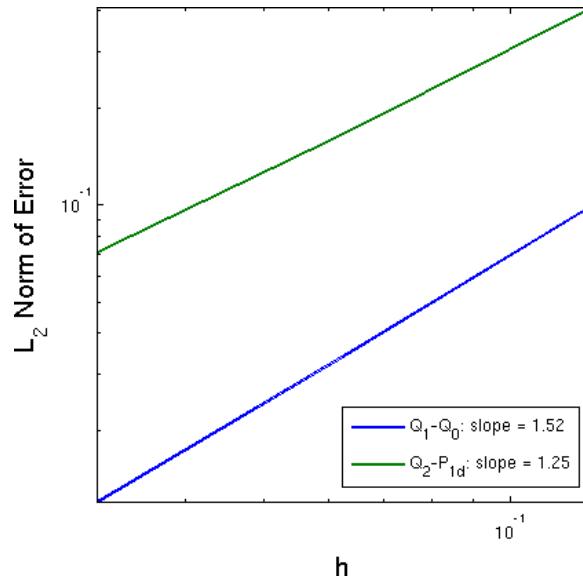


Figure 4.22: Convergence of  $\vec{\nabla}p$  acceleration:  $Q_2 - \hat{P}_1$  versus  $Q_1 - Q_0$  on a Cartesian mesh

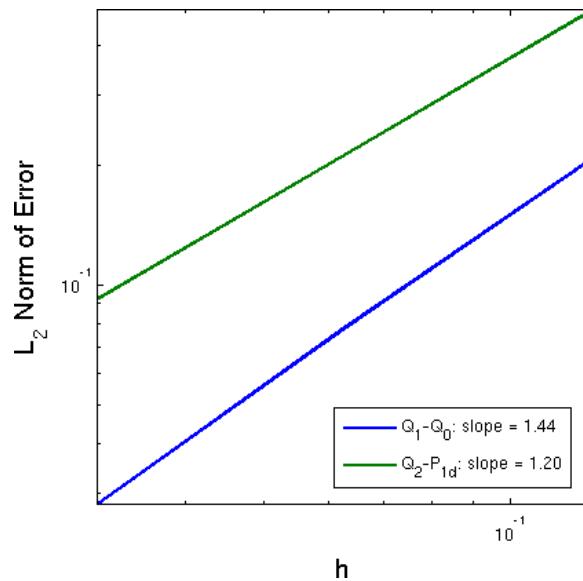


Figure 4.23: Convergence of  $\vec{\nabla}p$  acceleration:  $Q_2 - \hat{P}_1$  versus  $Q_1 - Q_0$  on a distorted mesh

### The $\text{div}(\text{grad}(V))$ Test

Please see Section 4.11.3 for the convergence on the  $\text{div}(\text{grad}(V))$  test.

#### 4.10.4 Spurious Modes

The principle benefit of this element over its  $Q_2-\hat{Q}_1$  cousin is that the spurious modes appear to be much more low key, see Figure 4.34 and especially Figure 4.36.

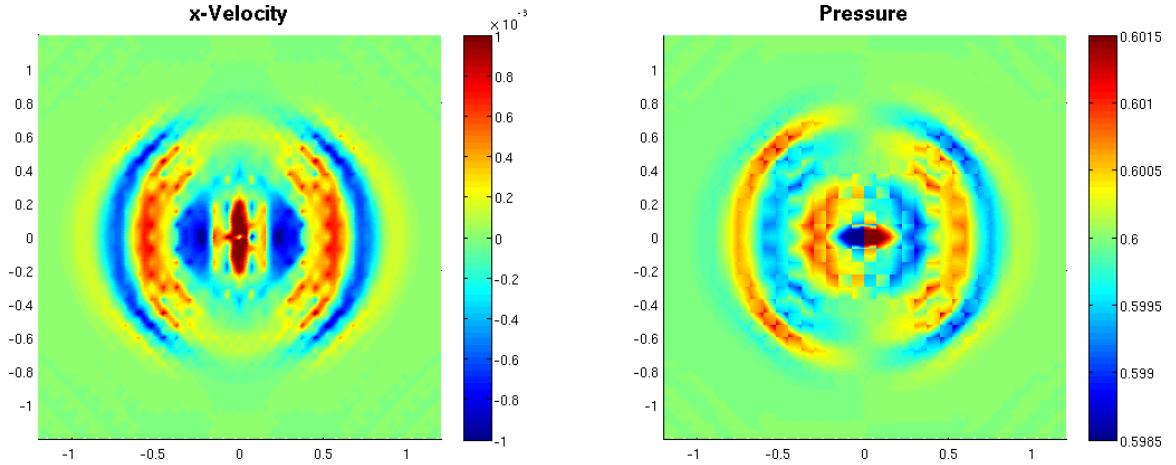


Figure 4.24:  $Q_2-\hat{P}_1$  acoustic wave without hourglass filter

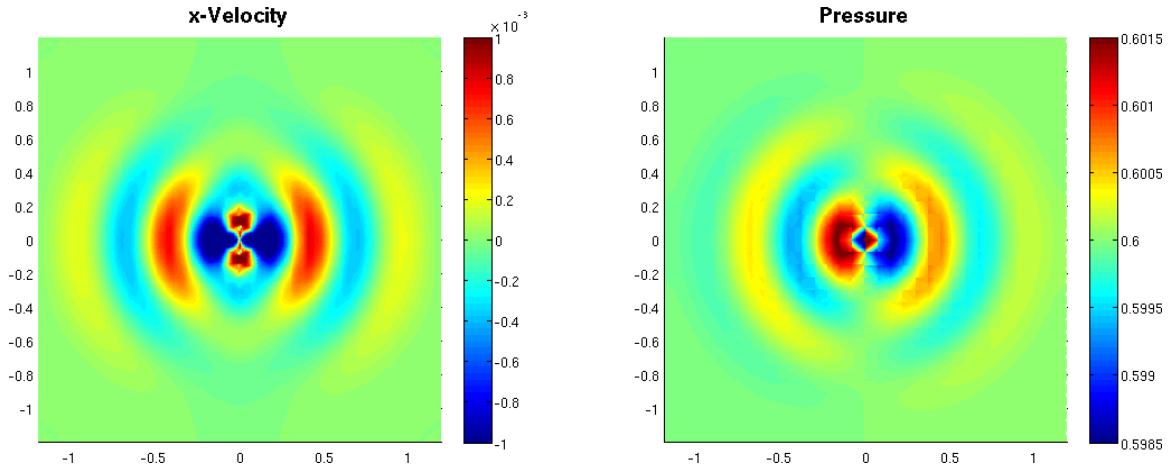
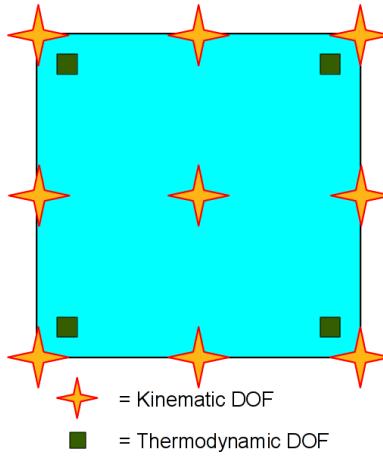


Figure 4.25:  $Q_2-\hat{P}_1$  acoustic wave with hourglass filter

#### 4.11 The $Q_2-\hat{Q}_1$ Element

The logical high-order extension of classical staggered grid hydrodynamics is the  $Q_2-\hat{Q}_1$  mixed finite element pair. If we enrich both the kinematic and thermo-

dynamic basis functions by one order, we arrive at a bi-quadratic interpolation for velocity and position and a bi-linear interpolation for pressure, density, and energy, as illustrated in Figure 4.26. This higher order representation of the kinematic variables allows for curvilinear edges and makes it possible to calculate second derivatives within each cell. The bi-linear representation of thermodynamic variables allows us to calculate pressure, energy, and density gradient within each cell. This property could make some sub-zonal physics / multi-material capabilities more straightforward to implement in a full-scale multi-physics hydrocode.



**Figure 4.26:** The  $Q_2\text{-}\hat{Q}_1$  element

#### 4.11.1 Assets

This basis function pair has the optimal constraint ratio of  $nD$ . This is not a radical departure from traditional SGH; we are only raising the order of each interpolation by one. Therefore this is analogous to a higher order version of traditional SGH, and some of the accumulated knowledge of SGH may carry over to this method. Higher order representations of the variables allow for more accurate calculations and faster convergence while allowing us to model more complicated physics with fewer elements. We will further explore the benefits of

this element in Chapter 5

### 4.11.2 Liabilities

Because this element is analogous to traditional SGH, it shares some of the same pitfalls. For example, hourglass modes take on a very similar, higher order form with  $Q_2-\hat{Q}_1$ . Another possible downside to this element is that there are more possible situations where a zero or negative Jacobian could kill a simulation. For example, if the internal DOF moves more than 25% of the cell length in any direction, the Jacobian will go negative in part of the cell. In practice, for the test problems considered, this has not been a problem, but more study is required. Also, because this is a new method, new analysis must be conducted as to stable time steps, flux between cells during the mesh relaxation stage, etc. We expect a fuller understanding of the pros and cons of this element to come out with time and further study.

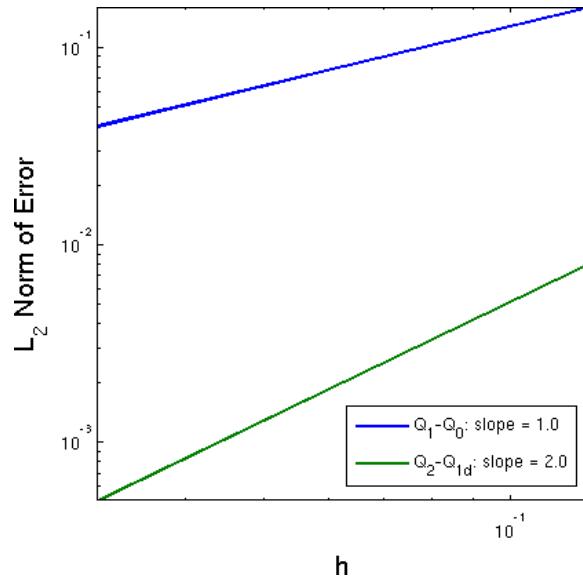
### 4.11.3 Convergence

This element has an attractive set of convergence properties.

#### The $\text{grad}(p)$ Test on a Cartesian Mesh

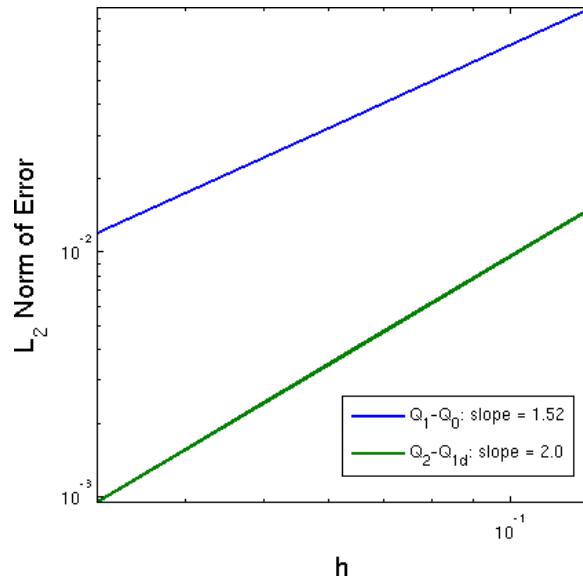
The basis function representation of the continuous pressure field converges quadratically with a rate of 2.0 as shown in Figure 4.27.

On a Cartesian mesh, this element converges quadratically to the analytical solution of the  $\text{grad}(p)$  test as shown in Figure 4.28. This is only slightly better than the  $Q_1-Q_0$  rate of 1.52, but Figure 4.28 demonstrates that the actual absolute



**Figure 4.27:** Convergence of the  $Q_2$ - $\hat{Q}_1$  pressure field

value of error is much lower.



**Figure 4.28:** Convergence of  $\vec{\nabla}p$  acceleration:  $Q_2$ - $\hat{Q}_1$  versus  $Q_1$ - $Q_0$  on a Cartesian mesh

## The $\text{grad}(p)$ Test on a Distorted Mesh

The convergence on a distorted mesh is the real test of an element because we don't expect to see any perfectly Cartesian meshes in any simulations of real engineering interest. On the distorted mesh, the  $Q_1-Q_0$  element drops to a rate of 1.44 while the  $Q_2-\hat{Q}_1$  element drops to a slightly sub-quadratic convergence of 1.88 as shown in Figure 4.29.

Interestingly, even though the  $Q_2-\hat{Q}_1$  element has the same pressure space as  $Q_1-\hat{Q}_1$ , it doesn't quite maintain the same rate of convergence on a distorted mesh. This difference is minor, but a little unexpected. Figure 4.30 compares the convergence of these two elements.

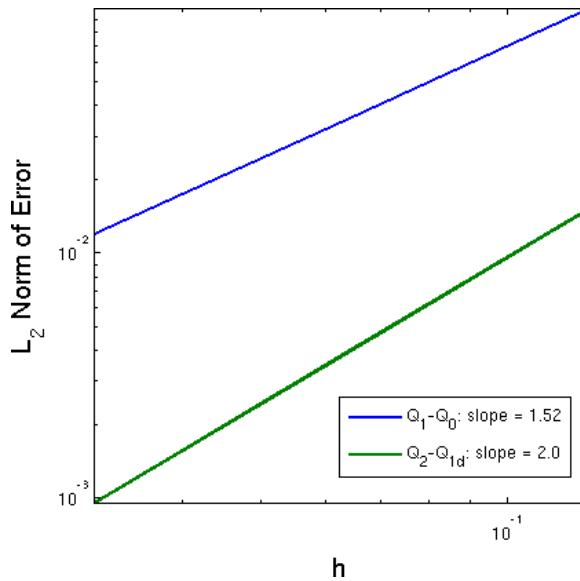
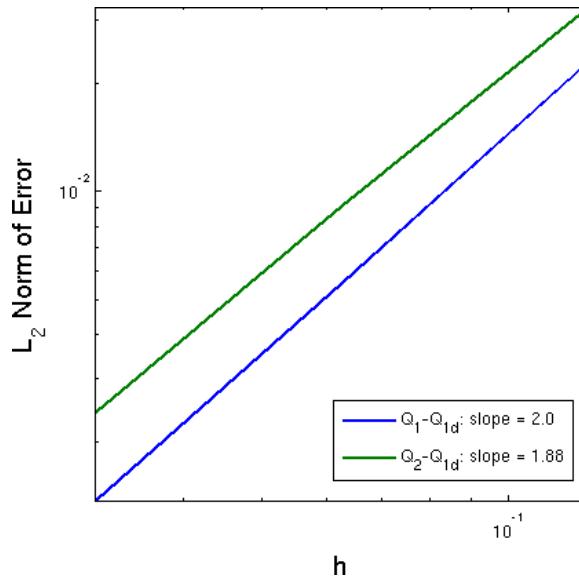


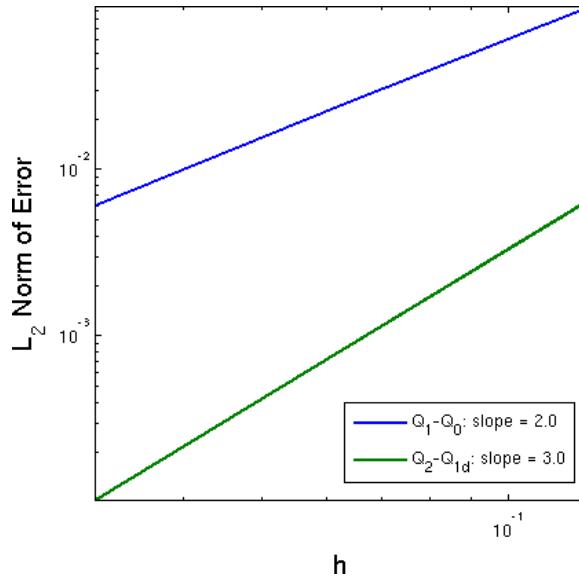
Figure 4.29: Convergence of  $\vec{\nabla}p$  acceleration:  $Q_2-\hat{Q}_1$  versus  $Q_1-Q_0$  on a distorted mesh

## The $\text{div}(\text{grad}(V))$ Test on a Cartesian Mesh

The velocity projection converges cubically at a rate of 3.0 as shown in Figure 4.31.



**Figure 4.30:** Convergence of  $\vec{\nabla}p$  acceleration:  $Q_2-\hat{Q}_1$  versus  $Q_1-Q_1$  on a distorted mesh



**Figure 4.31:** Convergence of the  $Q_2-\hat{Q}_1$  velocity projection vs  $Q_1-Q_0$

The  $Q_2-\hat{Q}_1$  element converges to the exact solution of the  $\text{div}(\text{grad}(V))$  test at a rate of 2.0 on a Cartesian mesh as shown in Figure 4.32. This appears to be somewhat of an anomaly because we expect that the higher order velocity field will result in higher-order convergence.

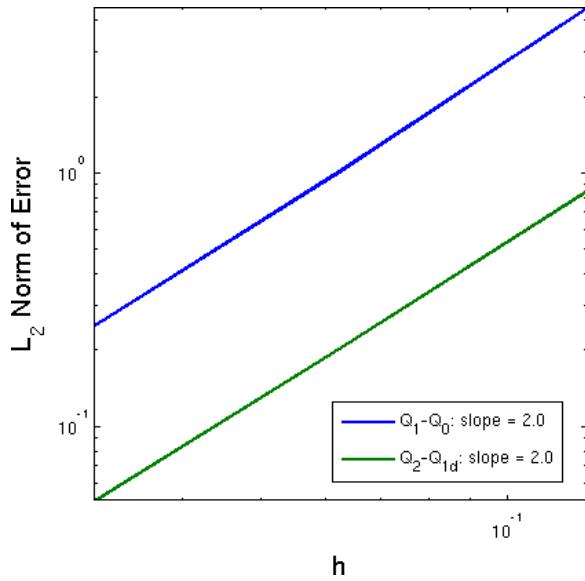


Figure 4.32: Convergence of  $\vec{\nabla} \cdot \vec{\nabla} V$  acceleration:  $Q_2 - \hat{Q}_1$  versus  $Q_1 - Q_0$  on a Cartesian mesh

### The $\text{div}(\text{grad}(V))$ Test on a Distorted Mesh

On a distorted mesh, the convergence of both  $Q_1 - Q_0$  and  $Q_2 - \hat{Q}_1$  drop to sub-quadratic convergence of 1.52 and 1.63, respectively as shown in Figure 4.33.

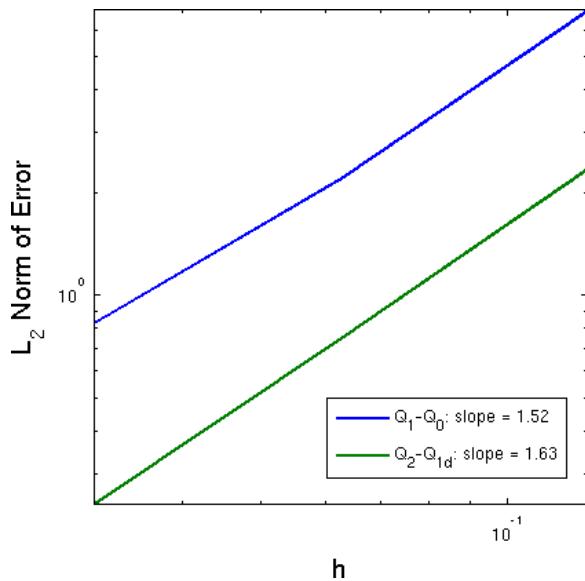
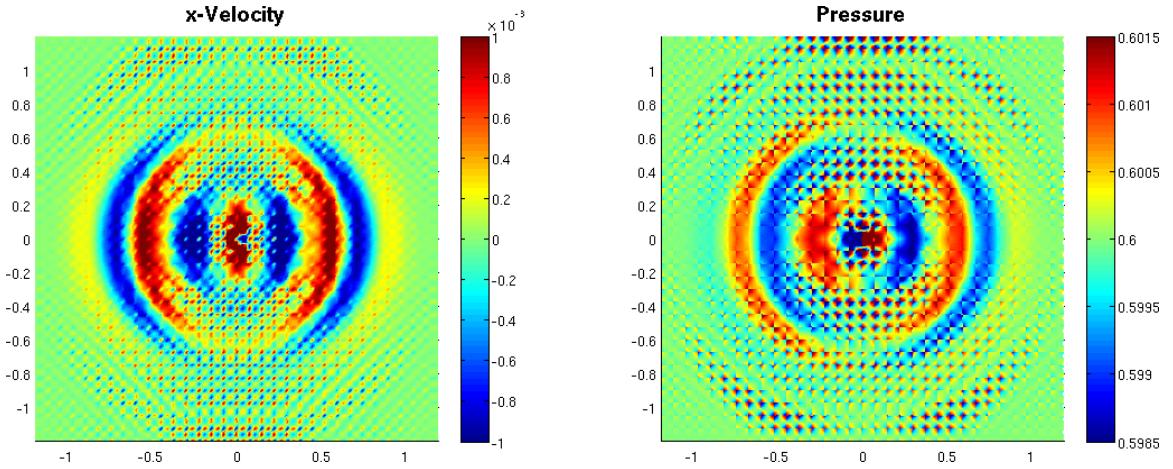


Figure 4.33: Convergence of  $\vec{\nabla} \cdot \vec{\nabla} V$  acceleration:  $Q_2 - \hat{Q}_1$  versus  $Q_1 - Q_0$  on a distorted mesh

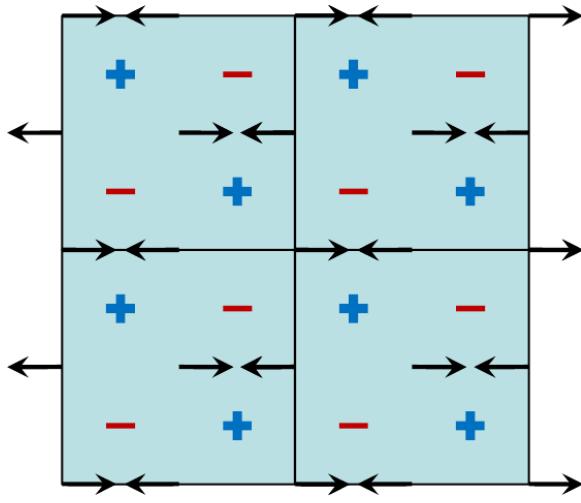
#### 4.11.4 Spurious Modes

To our disappointment, this element did not eliminate hourglass/checkerboard modes. In fact, the spurious modes present in this element bear a striking resemblance to the traditional hourglass modes found in the  $Q_1-Q_0$  element. Figure 4.34 shows the hourglass modes on the acoustic wave test.

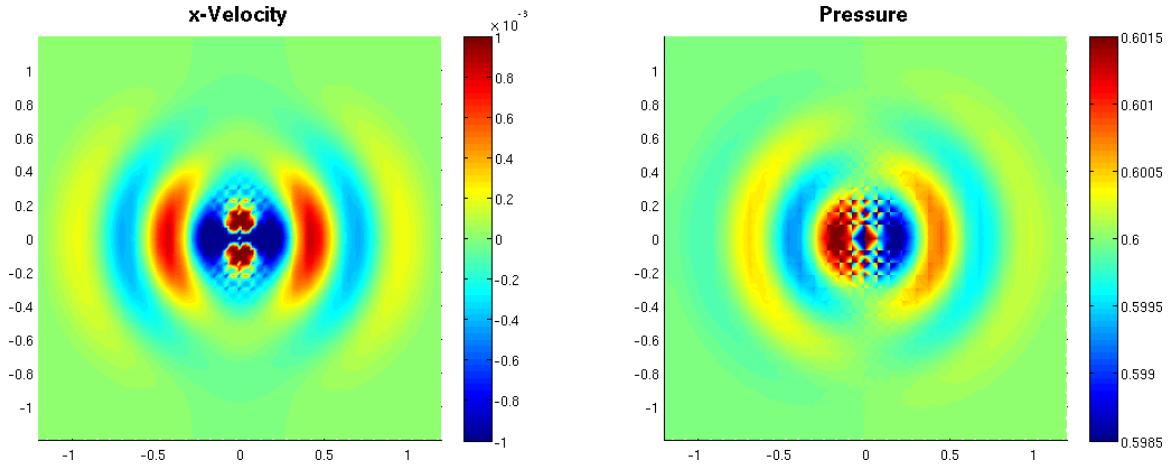


**Figure 4.34:**  $Q_2-\hat{Q}_1$  acoustic wave without hourglass filter

If you look closely, you will see that these spurious modes look exactly like a higher frequency version of hourglass modes. The structure of these hourglass modes is illustrated in Figure 4.35. Compared to some of the other more random spurious modes that we have considered, these more structured modes work to our advantage. They should be easier to analyze and filter, especially considering their similarity to the well-studied  $Q_1-Q_0$  hourglass modes. You may also notice in comparing Figure 4.6 and Figure 4.34, that the higher frequency modes of  $Q_2-\hat{Q}_1$  appear to be of lower magnitude and thus disturb the solution to a smaller degree. Despite the noise from the hourglass modes, we can see a very well defined solution. This will come in handy again as we consider the Sedov explosion test in Chapter 5. If we apply a filter to smooth the hourglass modes, the solution to the acoustic wave problem comes through much clearer in Figure 4.36.



**Figure 4.35:** Hourglass/checkerboard modes on a patch of  $Q_2-\hat{Q}_1$  elements



**Figure 4.36:**  $Q_2-\hat{Q}_1$  acoustic wave with hourglass filter

This filtered solution looks much better even than the filtered  $Q_1-Q_0$  solution in Figure 4.6.

## 4.12 The $Q_2-\hat{Q}_2$ Element

We also tried further refining the thermodynamic space to see what it could buy us. The  $Q_1-\hat{Q}_2$  element is a higher order analog of the  $Q_1-\hat{Q}_1$  element with

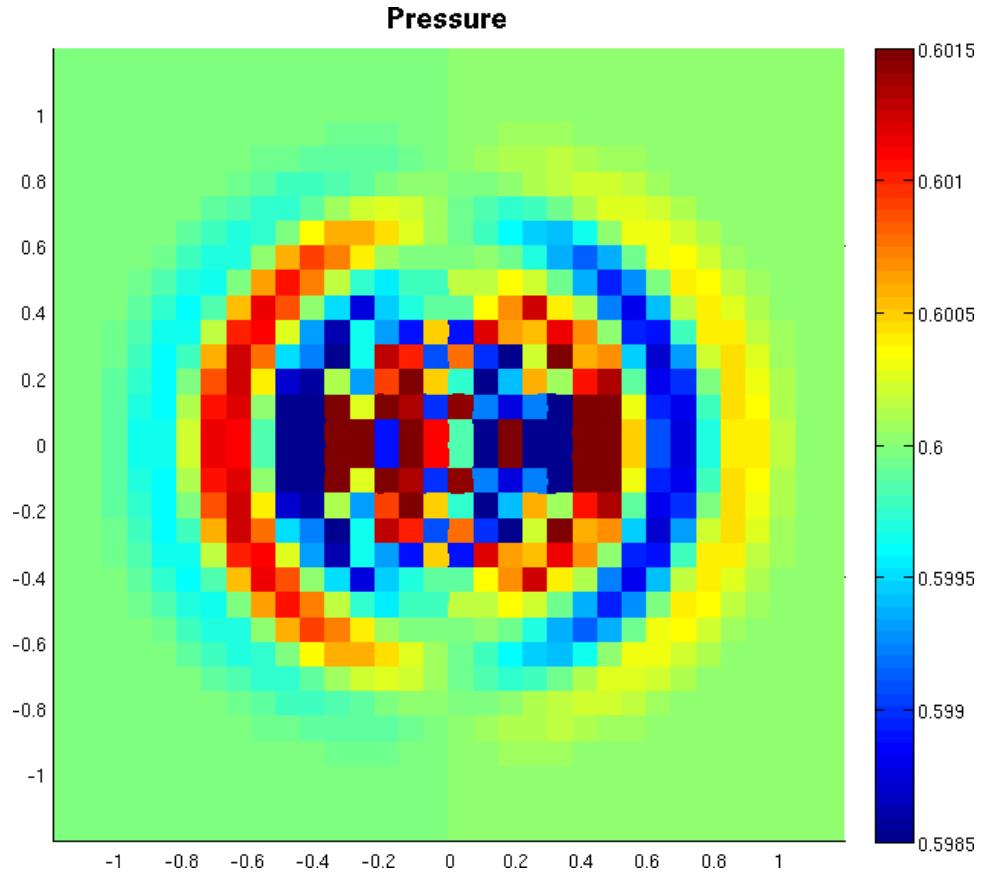
both spatial and state variables using a bi-quadratic interpolation. Similar to the  $Q_1\text{-}\hat{Q}_2$  element, any higher thermodynamic representation like  $Q_2\text{-}\hat{Q}_3$  appears to be redundant.

### 4.12.1 Assets

This element shares a lot of the same benefits and handicaps of the  $Q_1\text{-}\hat{Q}_1$  element. At first glance, if you average the pressure DOFs in each zone, it appears to eliminate checkerboard modes as shown in Figure 4.37, similar to  $Q_1\text{-}\hat{Q}_1$ . But if we look at the whole picture in Figure 4.41, the spurious modes are actually very ugly for this element. We will see in Section 5.2.4 that this element appears to arrest density undershoots in the Noh implosion problem. This element also shares the same spatial interpolation with  $Q_2\text{-}\hat{Q}_1$  which lets this element take on curvilinear geometries. As with  $Q_2\text{-}\hat{Q}_1$ , we can take second derivatives in space and compute gradients of the thermodynamic fields in a very straightforward way.

### 4.12.2 Liabilities

This element shares many of the failures of  $Q_1\text{-}\hat{Q}_1$ , including some serious velocity and pressure modes in the acoustic wave problem as shown in Figure 4.34. Similar to  $Q_2\text{-}\hat{Q}_1$  the Jacobian may be more sensitive to going negative for very deformed meshes.



**Figure 4.37:** Average pressure in each cell for  $Q_2-\hat{Q}_2$  element for acoustic wave problem

#### 4.12.3 Convergence

The high order interpolations for spatial and kinematic variables give this element fast convergence rates.

##### The $\text{grad}(p)$ Test on a Cartesian Mesh

The bi-quadratic pressure interpolation converges to the exact pressure field cubically, as demonstrated in Figure 4.38. On a Cartesian mesh,  $Q_2-\hat{Q}_2$  converges to the analytical acceleration at a rate of 3.0, as shown in Figure 4.39.

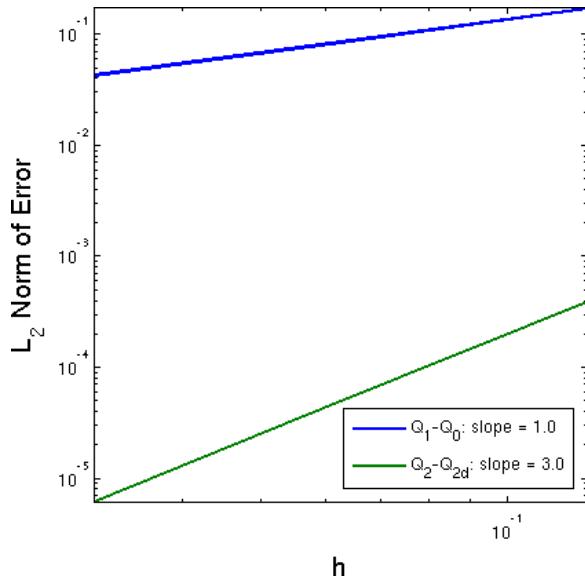


Figure 4.38: Convergence of the  $Q_2 - \hat{Q}_2$  pressure field

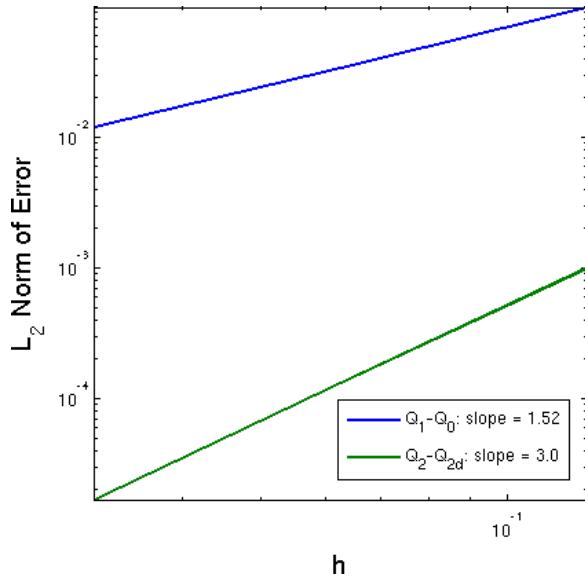
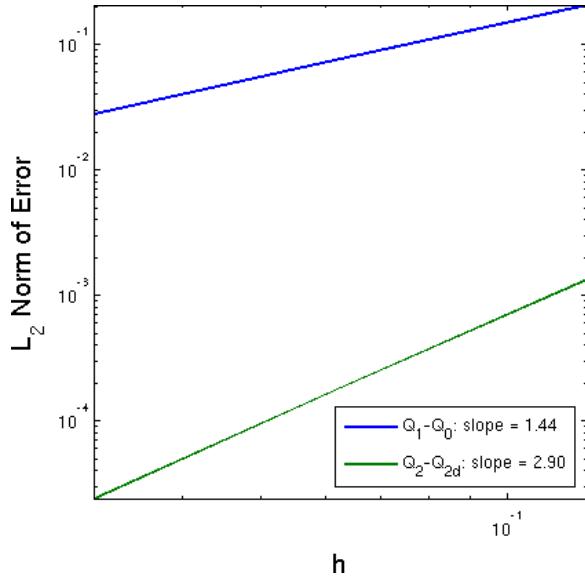


Figure 4.39: Convergence of  $\vec{\nabla}p$  acceleration:  $Q_2 - \hat{Q}_2$  versus  $Q_1 - Q_1$  on a Cartesian mesh

### The $\text{grad}(p)$ Test on a Distorted Mesh

As we can see from Figure 4.40, when we distort the mesh, the convergence rate drops down to 2.9.



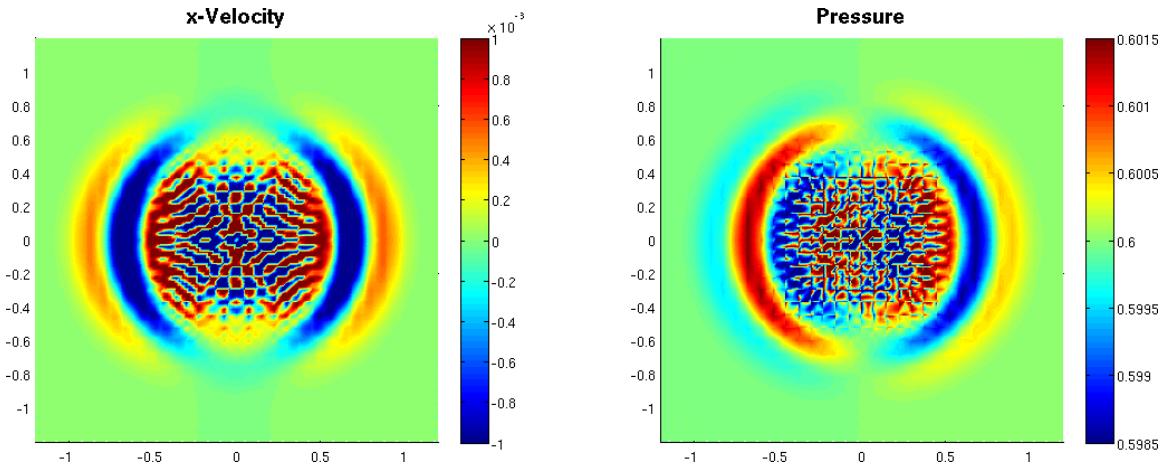
**Figure 4.40: Convergence of  $\vec{\nabla}p$  acceleration:  $Q_2-\hat{Q}_2$  versus  $Q_1-Q_1$  on a distorted mesh**

#### The $\text{div}(\text{grad}(V))$ Test

This element has the same velocity space as  $Q_2-\hat{Q}_1$ , therefore the convergence rates will be identical for the  $\text{div}(\text{grad}(V))$  test.

#### 4.12.4 Spurious Modes

This element has some very ugly spurious modes as we see in Figure 4.41. During the process of the solution, it appears that the very noisy interior travels outward from the center at half of the wave speed. Thus as the solution progresses, more and more of it will be overcome by the spurious modes.



**Figure 4.41:**  $Q_2-\hat{Q}_2$  acoustic wave without hourglass filter

## 4.13 Conclusions

We have considered a large array of different mixed finite element methods in the course of this research, but it would be prohibitive to attempt to do an in-depth study of every one of these elements for every canonical test problem that we will consider. Thus, we need to pair down the search space to a few mixed finite element pairs that showed the most promise. The  $Q_1-\hat{Q}_1$  element has been considered in detail in [7], so we will carry the results around for comparison. We would like to focus our study on curvilinear elements because they present a new frontier in the field of Lagrangian hydrodynamics. Therefore we will focus primarily on the  $Q_2-\hat{Q}_1$  element, which showed the most promise for a full hydrocode. The  $Q_2-\hat{P}_1$  element behaves very similarly, so we can assume that most of the discussion concerning  $Q_2-\hat{Q}_1$  will apply to this element as well. We will also consider the  $Q_2-\hat{Q}_2$  because it represents the higher order analogue of the method of subzonal pressures, namely  $Q_1-\hat{Q}_1$ . Of course, we need a control to compare against. With that in mind, we will continue to compare against results from traditional  $Q_1-Q_0$  SGH with and without hourglass filters.

# Chapter 5

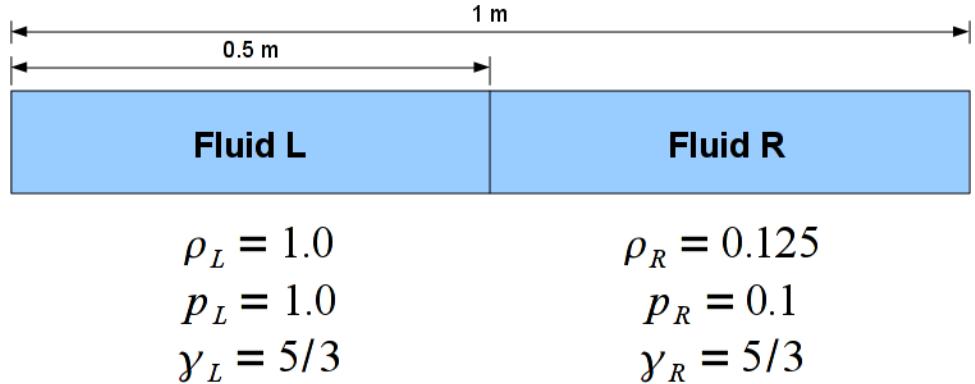
## Numerical Results

In this chapter, we will apply a few selected mixed finite element pairs to several canonical test problems in order to learn how they might perform on real problems of engineering interest.

### 5.1 The Sod Shock Tube

The Sod shock tube problem[32] is a good starting point to check the ability of high order finite element pairs to model shock problems. The Sod shock tube is especially practical because it is one-dimensional, contains both shock waves and expansion fans, and can be solved exactly by means of a Riemann solver[36]. The problem definition is of two initially static fluids of disparate pressures and densities in a shock tube. The diaphragm is burst, and the higher pressure gas on the left is allowed to expand into the lower pressure gas on the right. This forms two shock waves: one at the interface and the other within the lower density fluid, and one expansion fan in the high density fluid. Solid walls exist at  $x = 0$  and  $x = 1$  and the upper and lower boundaries are symmetry plains. The problem is

allowed to run until  $t = 0.6$ . The initial problem setup is illustrated in Figure 5.1.



**Figure 5.1: Problem setup for Sod shock tube problem**

For this one-dimensional problem, all higher order methods had nearly identical performance. Thus all of our comparisons will be between a single high-order method ( $Q_2-\hat{Q}_1$ ) and low order  $Q_1-Q_0$  and assume that  $Q_2-\hat{Q}_2$  would have performed similarly. Just to be absolutely clear, there are several little tweaks to  $Q_1-Q_0$  that we could make to improve the solution, such as implementing a monotonic slope limiter[3] or tuning of the `qquad` and `qlin` artificial viscosity coefficients. For our comparisons, we have put both elements on an equal footing by setting identical values of `qquad = 1` and `qlin = 1`. You will notice that both methods exactly capture the contact discontinuity. This is because the discontinuous nature of the thermodynamic basis functions makes multi-material physics natural and straightforward.

We can learn the most from a set of three plots. It is educational to consider  $Q_1-Q_0$  on a refined mesh together with  $Q_2-\hat{Q}_1$  on a coarse mesh. The higher-order  $Q_2-\hat{Q}_1$  has twice as many DOFs in any direction as  $Q_1-Q_0$ , so if we use 60 zones for  $Q_2-\hat{Q}_1$  and 120 zones for  $Q_1-Q_0$ , we will have the same total number of DOFs in the  $x$ -direction. Indeed, if we compare the blue and green lineouts in Figure 5.2, we see similar behavior despite the disparity in the mesh refinement. Thus we

can understand that in general, for  $Q_1$ - $Q_0$  to achieve comparable accuracy to  $Q_2$ - $\hat{Q}_1$  we need twice as many zones in one dimension, four times as many in two dimensions, and eight times as many in three dimensions. This will be an important point when we consider the computational efficiency of each method. Furthermore, if we now consider  $Q_2$ - $\hat{Q}_1$  on the same refined mesh as  $Q_1$ - $Q_0$ , we see a significant improvement in accuracy. Please note the discontinuous behavior of the density plots. The  $Q_0$  density climbs the shock in a series of flat steps while  $\hat{Q}_1$  scales it with a smaller series of discontinuous line segments.

These plots tell us some even more fundamental information about high order mixed finite elements than just accuracy. This first test problem demonstrates the practicality of using high order elements for shock problems. Our greatest fear was that Gibbs phenomenon[16] would cause high frequency ringing around shock discontinuities for high order elements. We will see later that Gibbs phenomenon still occurs for high order elements, but this problem indicates that it may not be as significant an obstacle to overcome as we initially guessed.

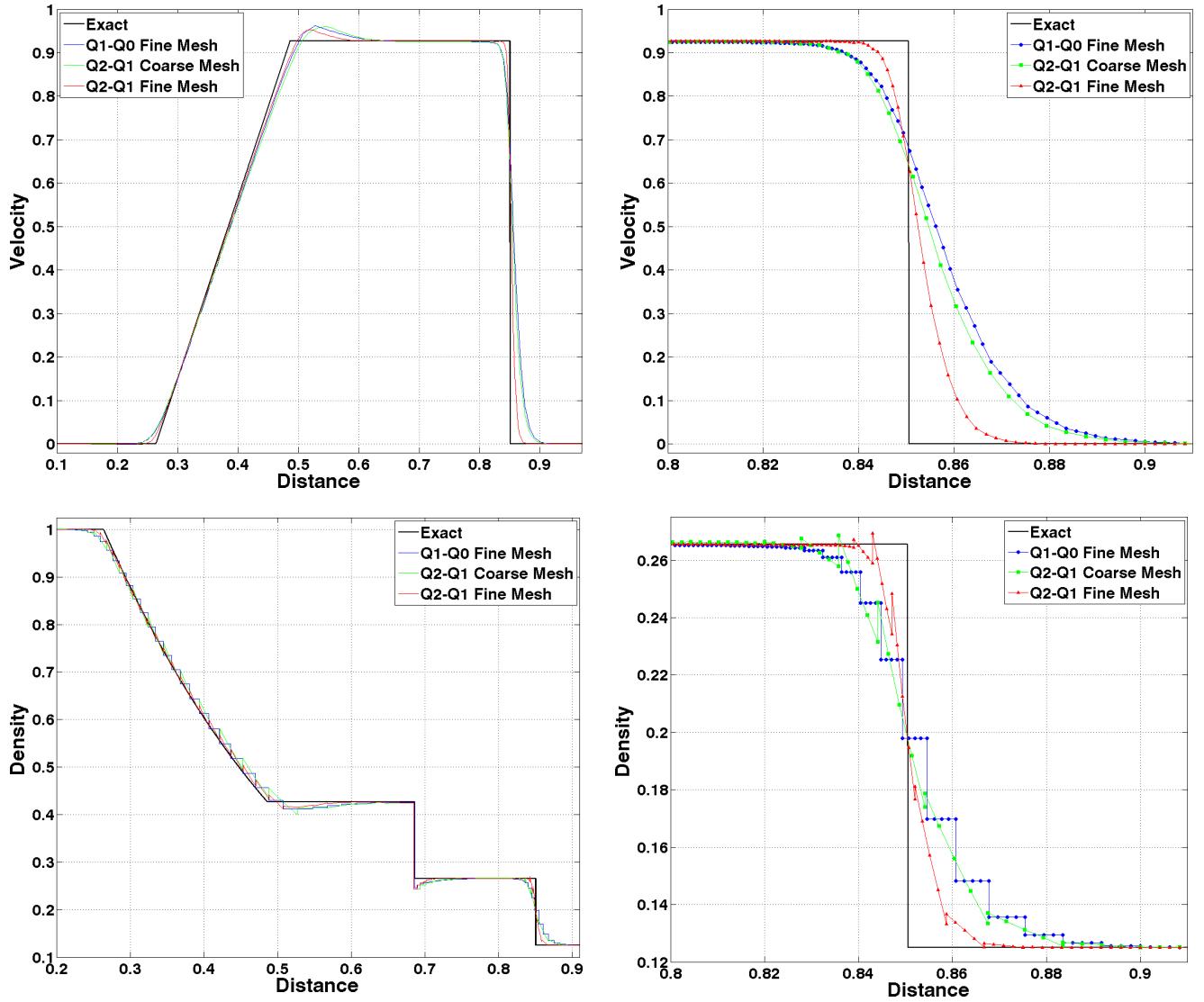


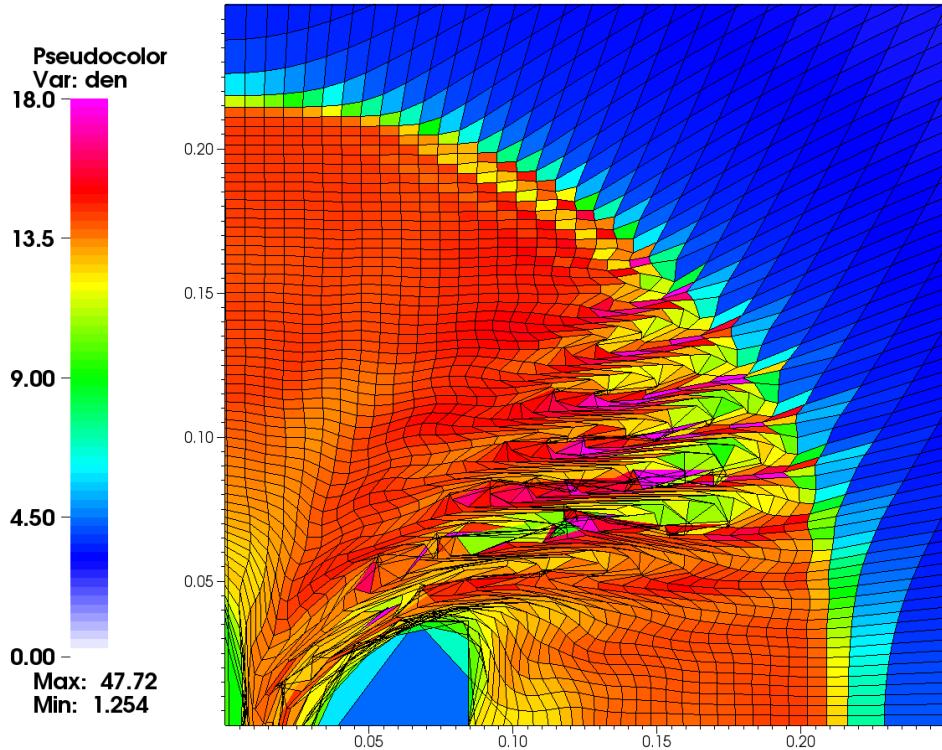
Figure 5.2: Comparison of  $Q_1\text{-}Q_0$  and  $Q_2\text{-}\hat{Q}_1$  elements on Sod shock tube and zoomed in on the shock (*right*). Five plot points are considered per zone.

## 5.2 The Noh Implosion

The Noh implosion problem was first proposed by W. F. Noh[26]. This problem teaches us a lot about symmetry preservation of the method as well as accurate prediction of the shock speed and performance for a very strong shock (shock that decelerates gas to zero velocity). Historically, this problem has posed a significant challenge to bulk artificial viscosity formulations, as demonstrated by Figure 5.3, but we have built in the tensor artificial viscosity formulation of Campbell and Shashkov[5] and formulated in a finite element sense by Kolev and Rieben[23]. Thus, we should have little problem with symmetry breaking. Hourglass mode interference is minor or negligible for this problem, which allows us to experiment with a 2D shock problem without worrying too much about what effect the hourglass filter is having. In fact, we will be running all of the following cases without any hourglass filtering. Another useful characteristic of the Noh problem is that it has a simple analytical solution for all time, see below.

$$\begin{aligned}\vec{V} &= \begin{cases} \vec{0}, & \text{for } r \leq \frac{t}{3}, \\ -\hat{r}, & \text{for } r > \frac{t}{3}. \end{cases} \\ \rho &= \begin{cases} 16, & \text{for } r \leq \frac{t}{3}, \\ 1 + \frac{t}{r}, & \text{for } r > \frac{t}{3}. \end{cases} \\ p &= \begin{cases} 16/3, & \text{for } r \leq \frac{t}{3}, \\ 0, & \text{for } r > \frac{t}{3}. \end{cases}\end{aligned}$$

The Noh problem is simple to set up. A uniform gas at zero pressure, with

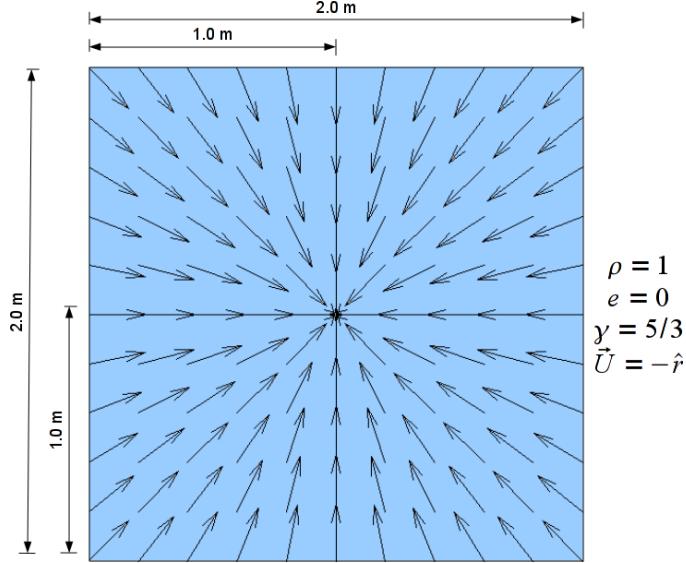


**Figure 5.3:** Density pseudo-color at time  $t = 0.6$  for the Noh problem on a 40 by 80 initial grid using the monotonic-scalar artificial viscosity.

a density of 1.0 and  $\gamma = 5/3$  is initialized with a unit radially inward velocity at every point in the domain, as illustrated in Figure 5.4. As the gas collides at the origin, the density builds up, and a shock wave expands radially outward at a speed of  $1/3$ . There is one particular feature of this test problem that has plagued Lagrangian codes since its inception. “Wall heating,” which is caused by numerical shocks repeatedly reflecting through the zone at the origin causes that first cell to have a drastically lower density than it should[28]. This is unavoidable for a simple research code such as Fermium, and is not a function of the finite element pair chosen, but purely of the framework (Eulerian or Lagrangian). We will just need to acknowledge its presence and then look past it.

It should be immediately obvious that we don’t actually have to model this entire domain. One quadrant should be sufficient to obtain all the information

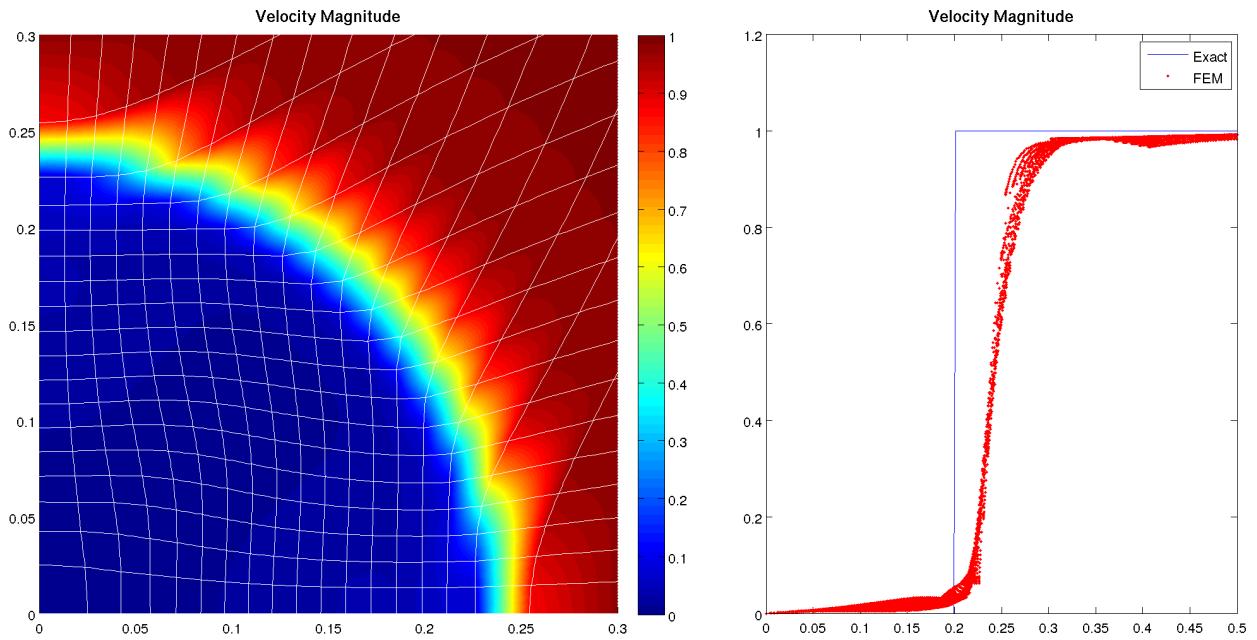
we need. Therefore for this and the Sedov problem, we will only model the first quadrant and define the bottom and left boundaries to be symmetry plains.



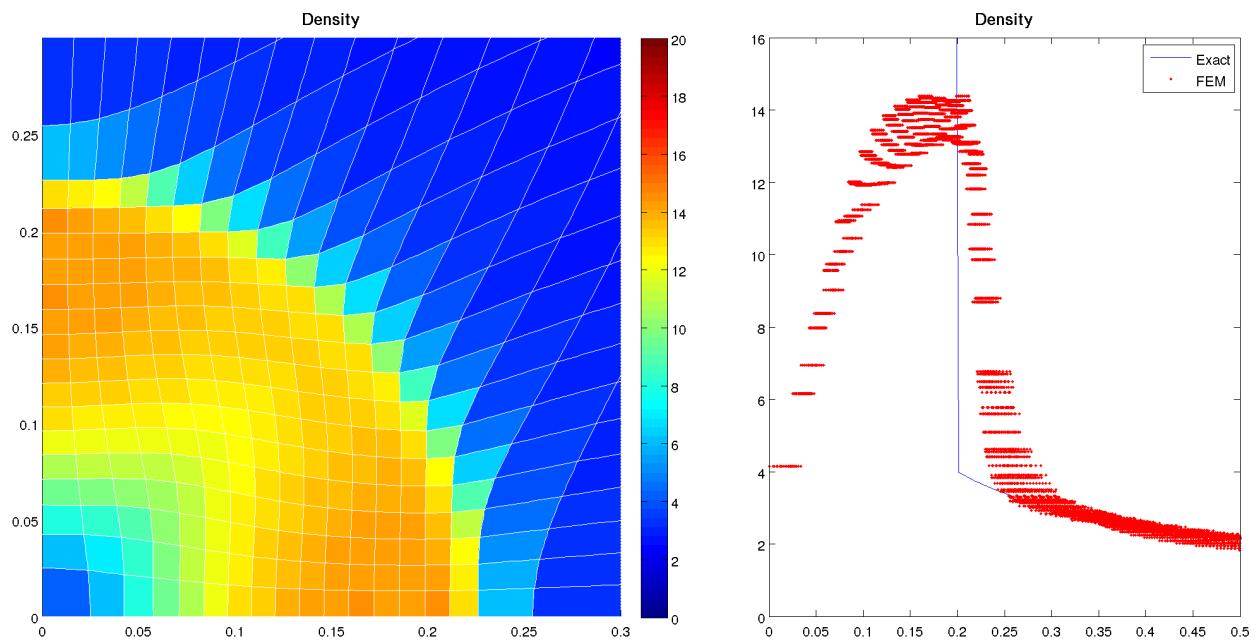
**Figure 5.4: Problem setup for Noh implosion problem**

### 5.2.1 Low Order $Q_1$ - $Q_0$

We require a huge number of low order elements to get anywhere close to the exact solution of the Noh problem. The original shock heating appears to irreparably damage the solution, dragging the predicted density way below the correct value and offsetting the shock past its correct position, see Figure 5.5. The  $Q_1$ - $Q_0$  solution does have this going for it however, it does not undershoot or overshoot the solution at the shock front. This lends favorably to the stability of the method because we never see negative pressures or densities show up in the solution. We also consider a refined  $40 \times 40$  solution in Figure 5.6 in order to observe the convergence to the analytical solution.

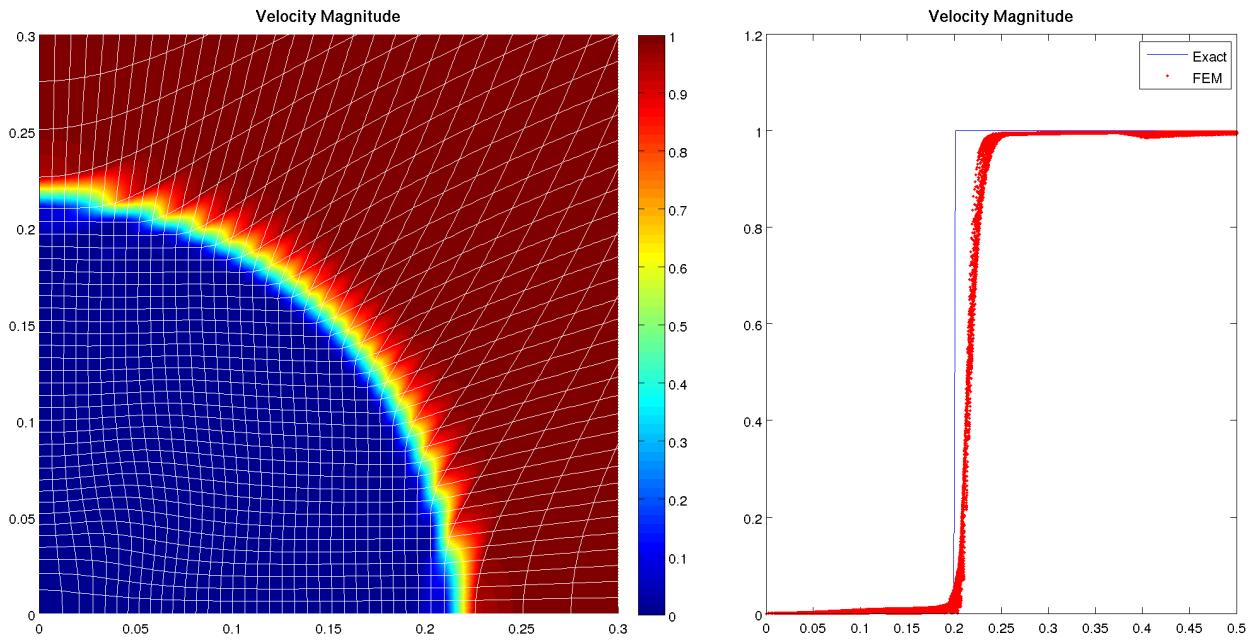


(a) Velocity magnitude pseudocolor plot (*left*). Velocity magnitude scatter plot compared to exact solution (*right*).

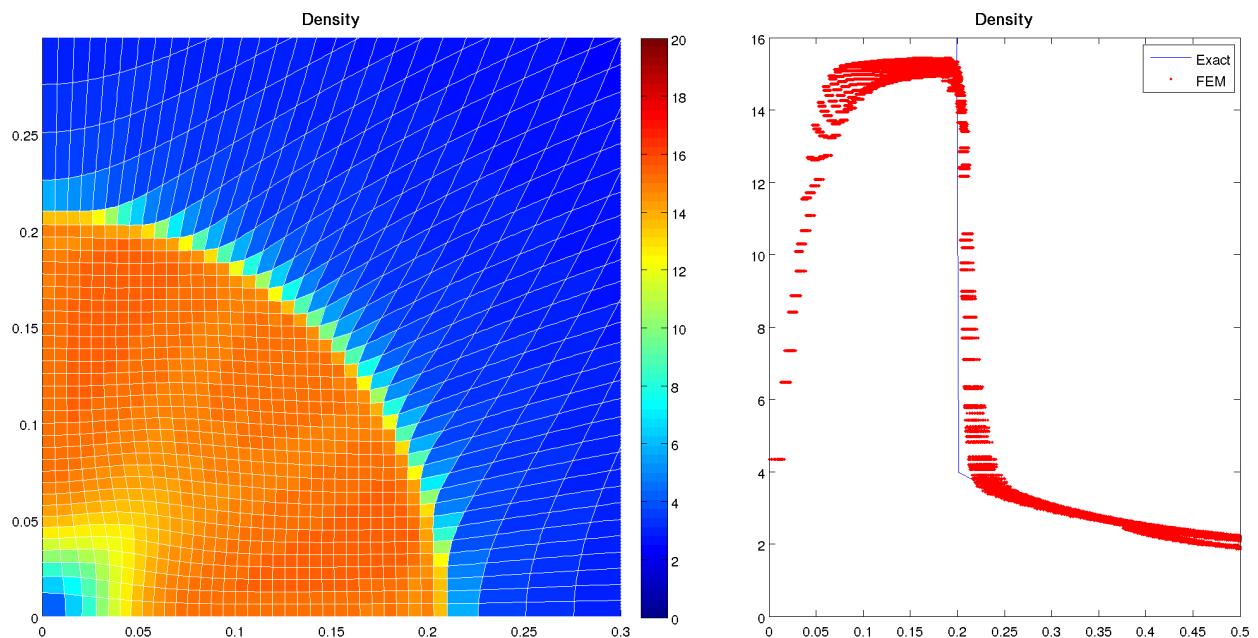


(b) Density pseudocolor plot (*left*). Density scatter plot compared to exact solution (*right*).

**Figure 5.5:** Noh explosion problem on  $20 \times 20$  mesh using  $Q_1-Q_0$  elements without hourglass filtering



(a) Velocity magnitude pseudocolor plot (*left*). Velocity magnitude scatter plot compared to exact solution (*right*).

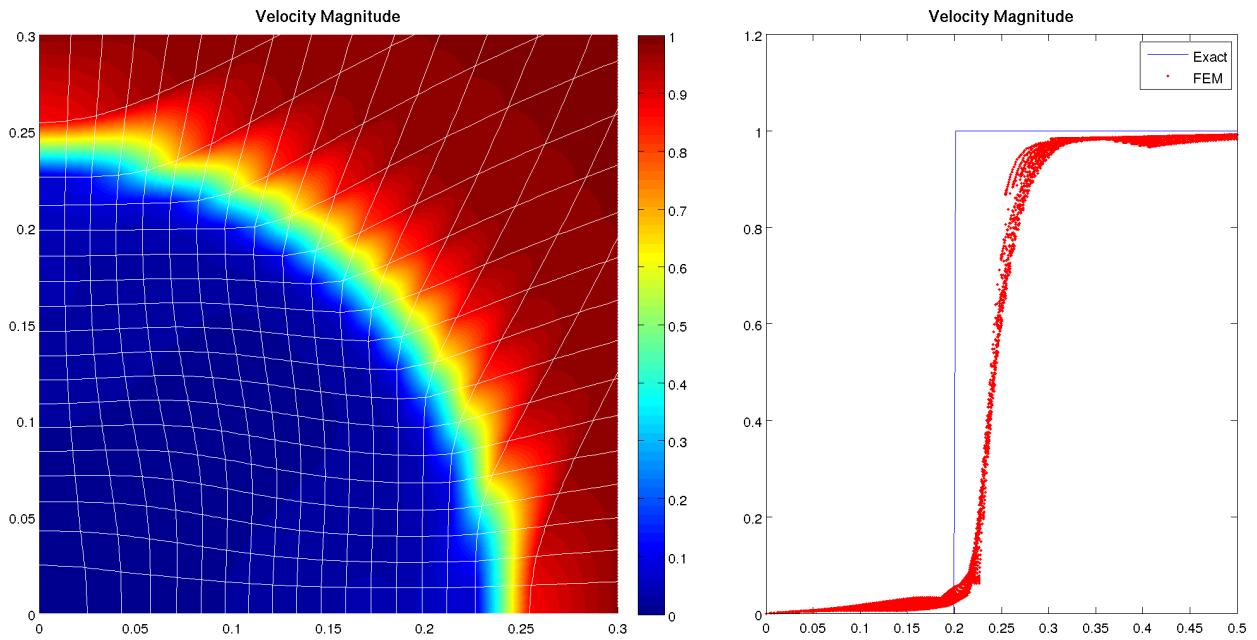


(b) Density pseudocolor plot (*left*). Density scatter plot compared to exact solution (*right*).

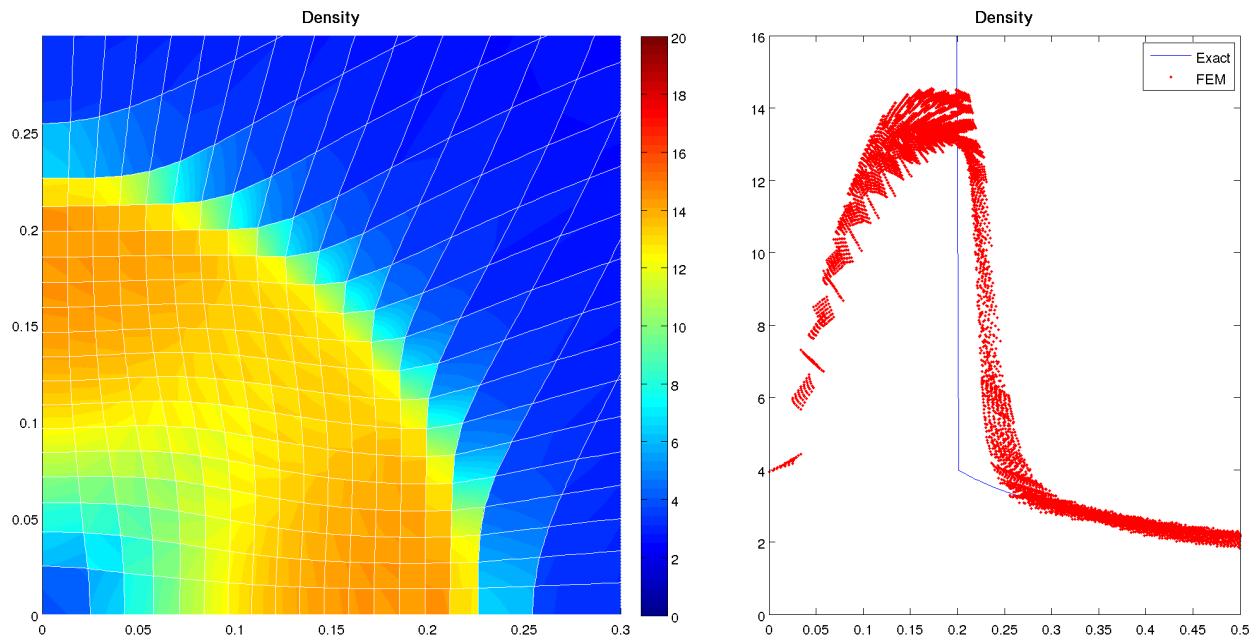
**Figure 5.6:** Noh explosion problem on  $40 \times 40$  mesh using  $Q_1-Q_0$  elements without hourglass filtering

### 5.2.2 Low Order $Q_1$ - $\hat{Q}_1$

Now turning to the  $Q_1$ - $\hat{Q}_1$  element, we can see some promising results from the Noh problem, as illustrated by Figure 5.7. We still don't achieve the desired post-shock density, but the solution looks very reasonable and stable overall. The post-shock density field looks especially uniform compared to  $Q_2$ - $\hat{Q}_1$  and  $Q_2$ - $\hat{Q}_2$ . Even with the refined  $40 \times 40$  mesh, the density doesn't quite reach the correct value, but the field is pretty good overall. Sharing the same kinematic basis functions, the velocity field is very similar to  $Q_1$ - $Q_0$ .

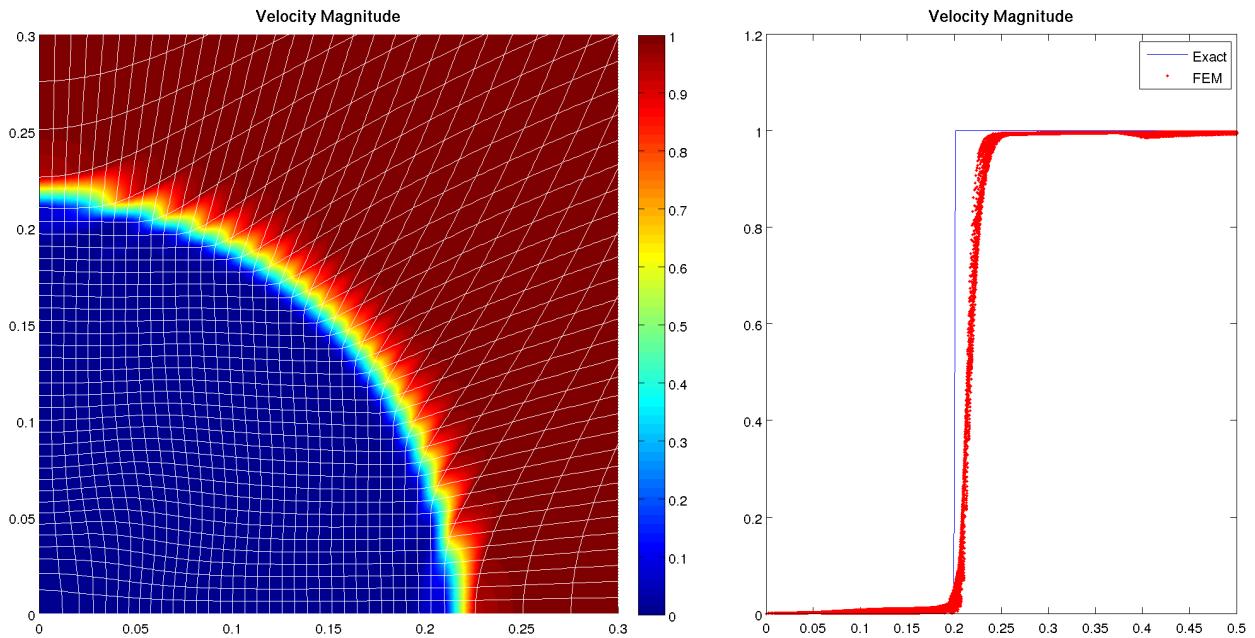


(a) Velocity magnitude pseudocolor plot (*left*). Velocity magnitude scatter plot compared to exact solution (*right*).

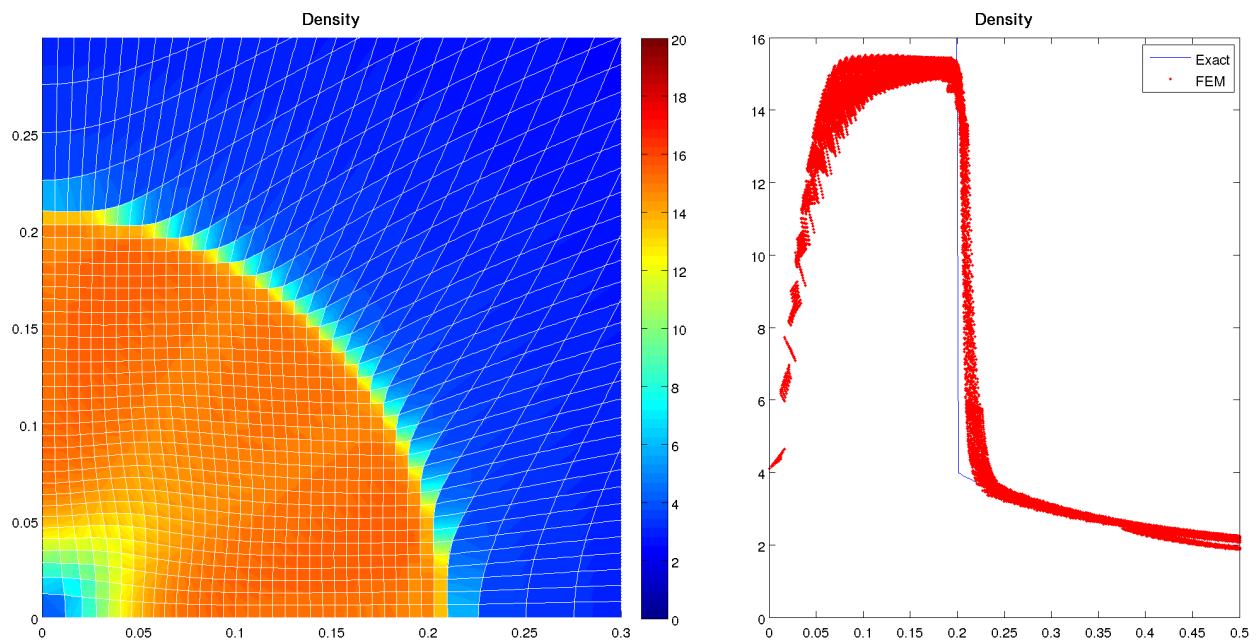


(b) Density pseudocolor plot (*left*). Density scatter plot compared to exact solution (*right*).

**Figure 5.7:** Noh explosion problem on  $20 \times 20$  mesh using  $Q_1-\hat{Q}_1$  elements without hourglass filtering



(a) Velocity magnitude pseudocolor plot (*left*). Velocity magnitude scatter plot compared to exact solution (*right*).

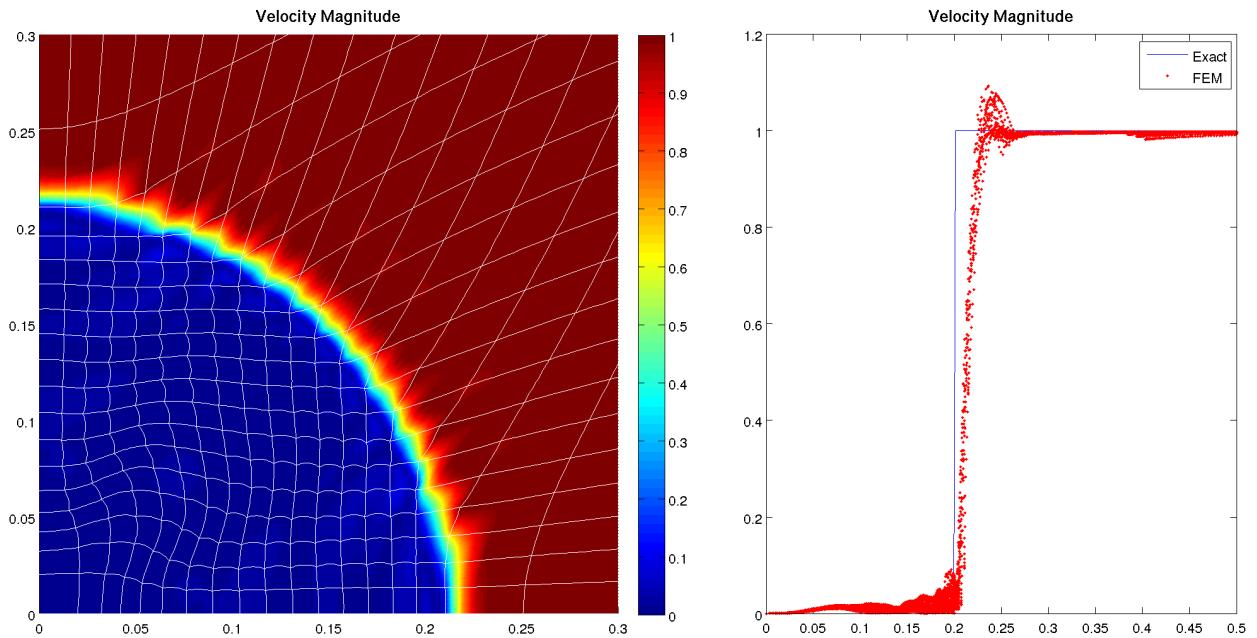


(b) Density pseudocolor plot (*left*). Density scatter plot compared to exact solution (*right*).

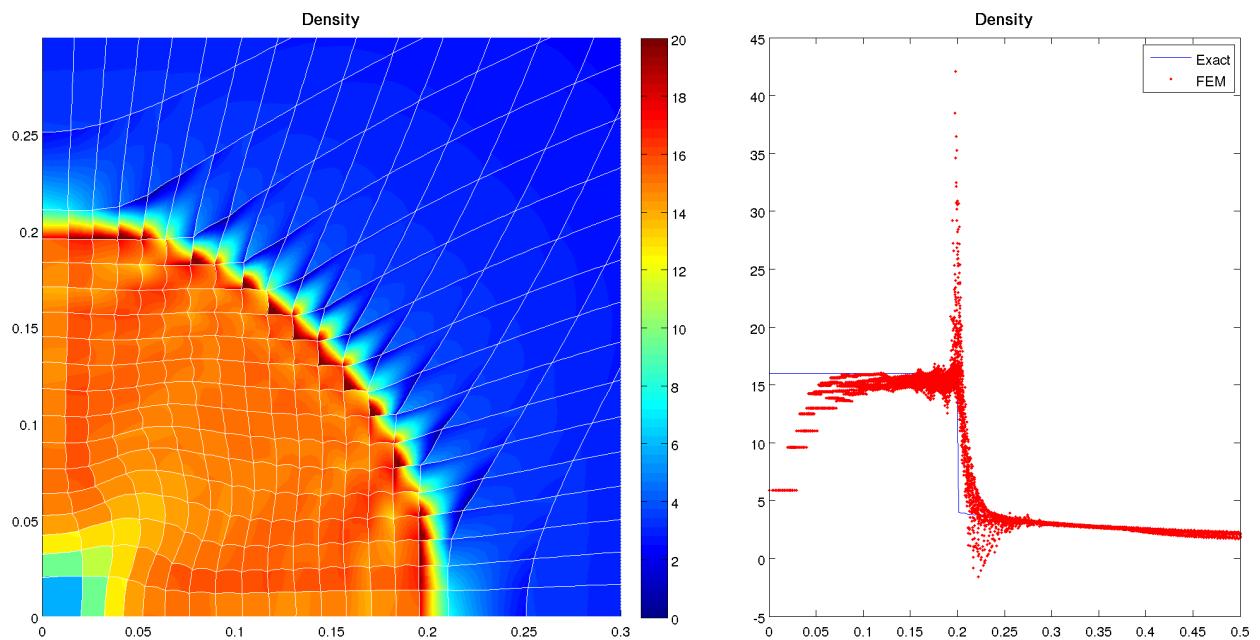
**Figure 5.8:** Noh explosion problem on  $40 \times 40$  mesh using  $Q_1-\hat{Q}_1$  elements without hourglass filtering

### 5.2.3 High Order $Q_2$ - $\hat{Q}_1$

We see a marked improvement when we switch to higher order, as illustrated by Figure 5.9. Both the velocity and density plots get a lot closer to the exact solution, but we also start to see some troubling features. The density scatter plot shows some huge overshoots and undershoots oscillating around the shock front, we also get some overshoots in the velocity plot. This is very troubling because we actually see density dip below the x-axis to some negative values. The average density in each cell would still be positive, but even a negative interpolation is alarming. Indeed, it is very important that we address this if we ever wish to implement a high-order Lagrangian hydro code. We have observed that this wild oscillation can be controlled through adjustment of the artificial viscosity coefficients, but this also significantly smears out the shock. The concept of hyperviscosity, as postulated by Cook and Cabot[8, 9] shows particular promise at eliminating shock oscillations for higher order methods. This method uses multiplications of the stiffness matrix to filter out non-physical oscillatory behavior with a minimal number of problem dependent constants. Please note the significant improvement in the matching capability of the density plot. While a  $40 \times$  by  $Q_1$ - $Q_0$  mesh fails to accurately predict the post-shock density, a mere  $20 \times 20$   $Q_2$ - $\hat{Q}_1$  hits it right on the head. Although the oscillations get worse under refinement, the overall solution gets much closer to the analytic prediction under refinement as we can see in Figure 5.10. The velocity matching is the closest of anything that we have seen so far.

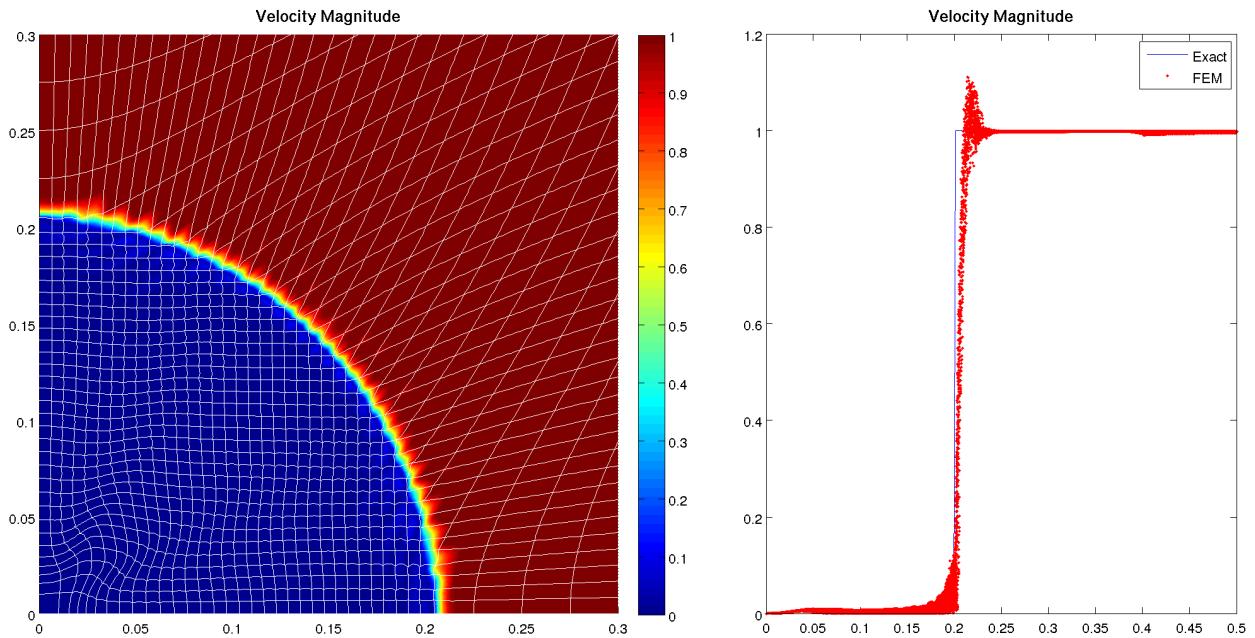


(a) Velocity magnitude pseudocolor plot (*left*). Velocity magnitude scatter plot compared to exact solution (*right*).

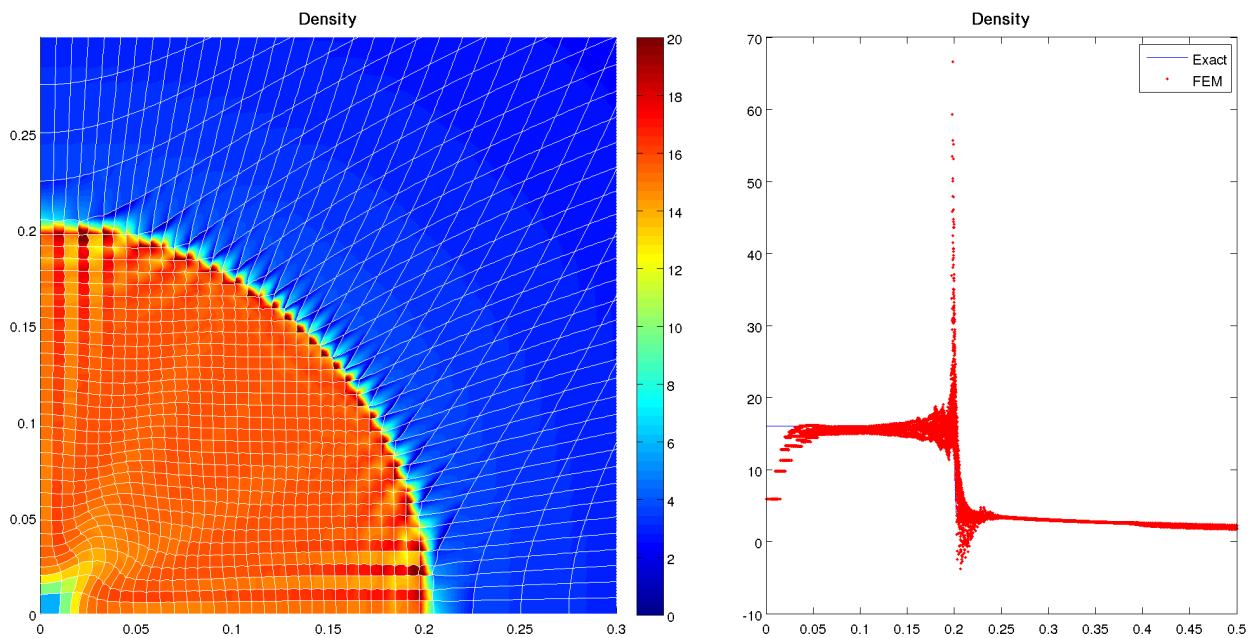


(b) Density pseudocolor plot (*left*). Density scatter plot compared to exact solution (*right*).

**Figure 5.9:** Noh explosion problem on  $20 \times 20$  mesh using  $Q_2-\hat{Q}_1$  elements without hourglass filtering



(a) Velocity magnitude pseudocolor plot (*left*). Velocity magnitude scatter plot compared to exact solution (*right*).

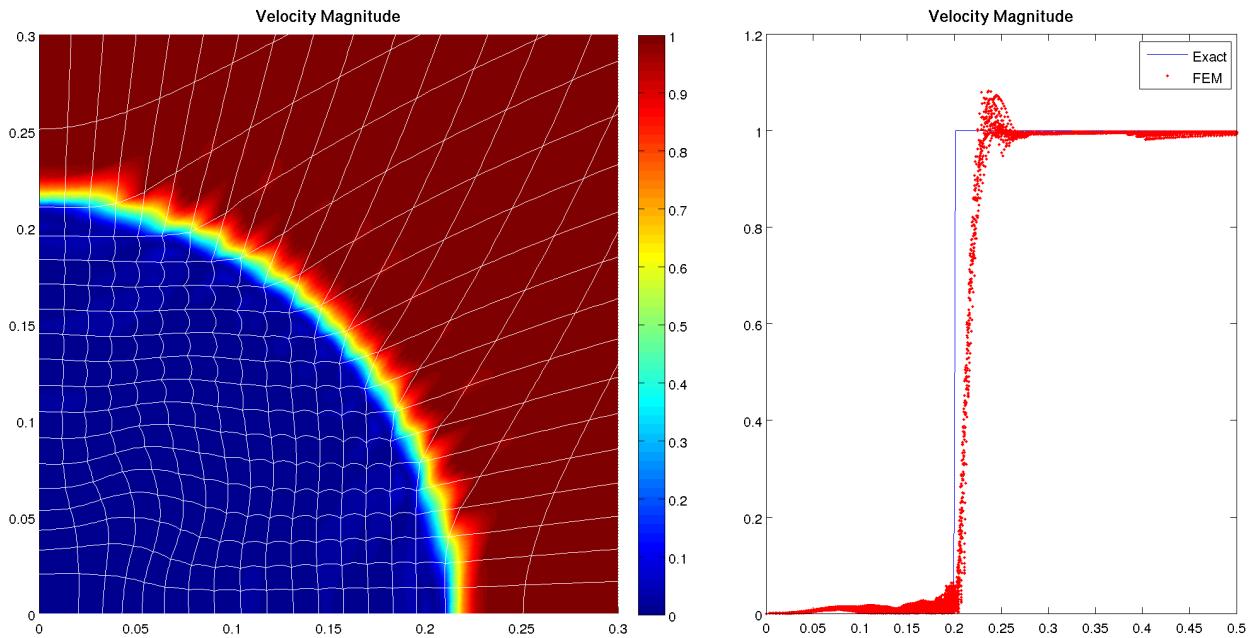


(b) Density pseudocolor plot (*left*). Density scatter plot compared to exact solution (*right*).

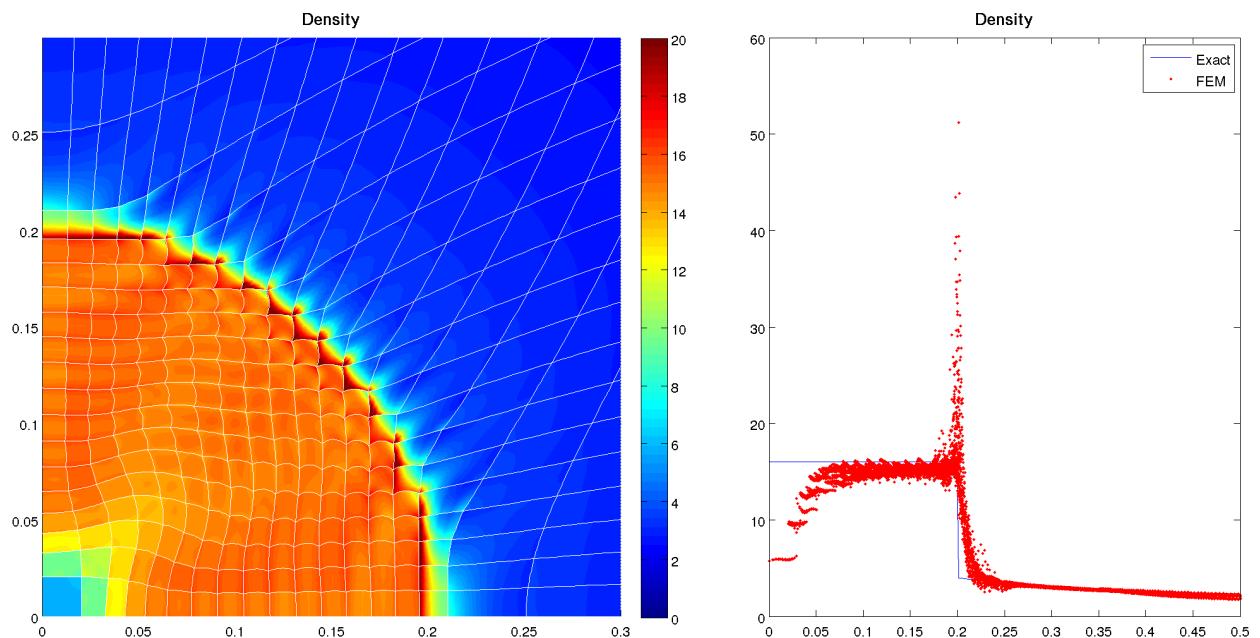
**Figure 5.10:** Noh explosion problem on  $40 \times 40$  mesh using  $Q_2-\hat{Q}_1$  elements without hourglass filtering

### 5.2.4 High Order $Q_2$ - $\hat{Q}_2$

The benefit of this element chiefly lies in its ability to avoid density undershoots at the shock front. The bi-quadratic density and pressure fields must give a single element the flexibility to jump from a low to a high density without overshooting both directions while still maintaining the correct mass within that zone. The density overshoots on the other hand, are more severe than the  $Q_2$ - $\hat{Q}_1$  element. The velocity field is much the same as its  $Q_2$ - $\hat{Q}_1$  cousin.

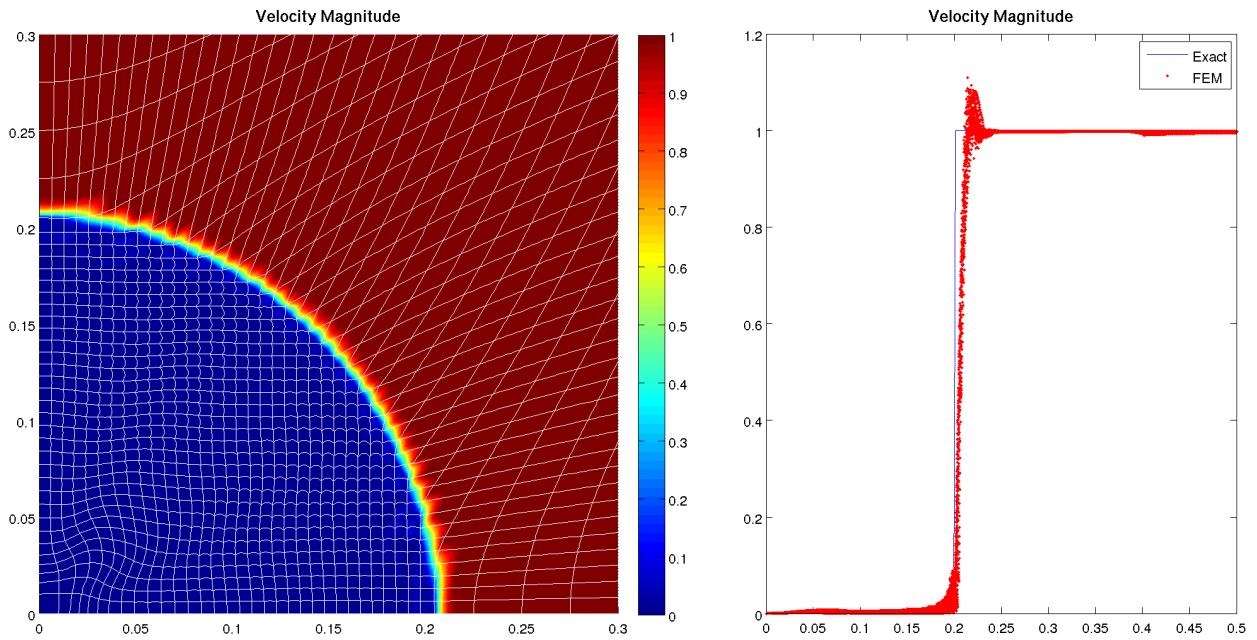


(a) Velocity magnitude pseudocolor plot (*left*). Velocity magnitude scatter plot compared to exact solution (*right*).

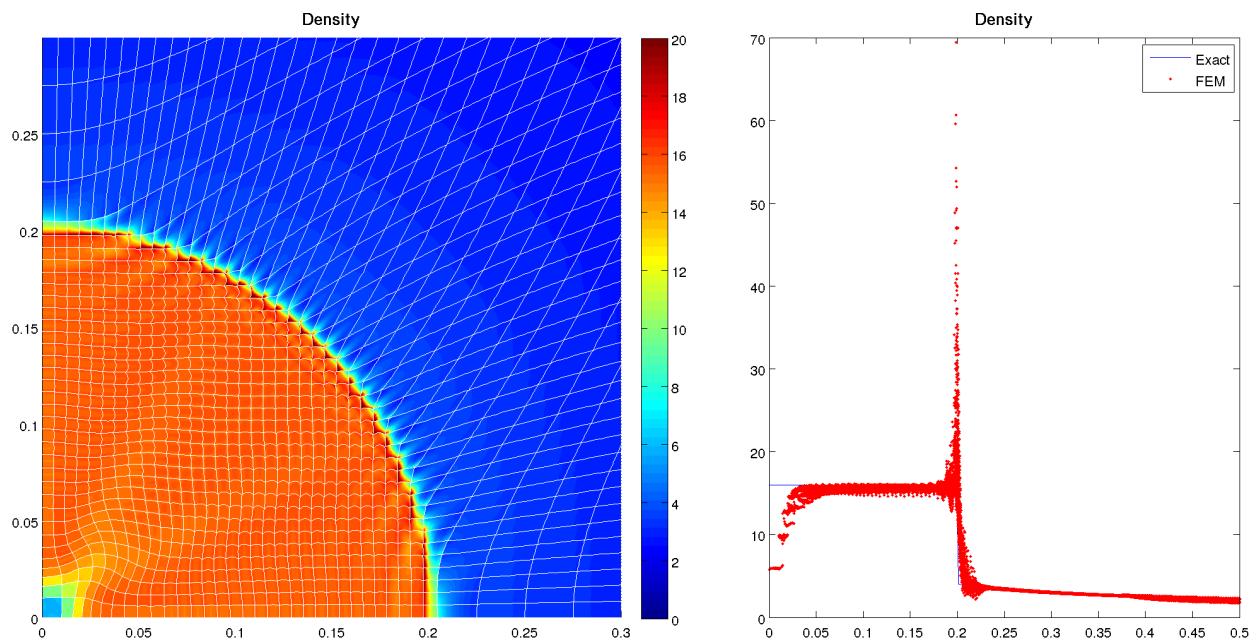


(b) Density pseudocolor plot (*left*). Density scatter plot compared to exact solution (*right*).

**Figure 5.11:** Noh explosion problem on  $20 \times 20$  mesh using  $Q_2-\hat{Q}_2$  elements without hourglass filtering



(a) Velocity magnitude pseudocolor plot (*left*). Velocity magnitude scatter plot compared to exact solution (*right*).



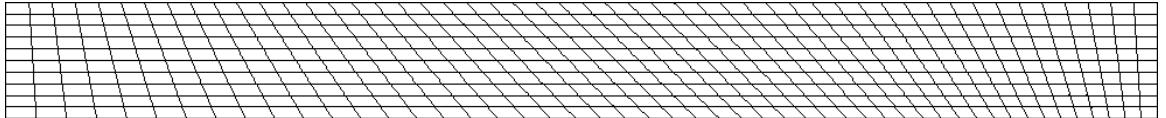
(b) Density pseudocolor plot (*left*). Density scatter plot compared to exact solution (*right*).

**Figure 5.12:** Noh explosion problem on  $40 \times 40$  mesh using  $Q_2-\hat{Q}_2$  elements without hourglass filtering

### 5.3 The Saltzman Piston

The Saltzman piston problem[18, 30] is commonly used to test the effectiveness of new artificial viscosity schemes. In this problem a one-dimensional piston driven shock propagates through a two-dimensional mesh that has been perturbed according to (5.1). We applied this transformation to a  $50 \times 10$  mesh with original width and height of 1.0 and 0.1, as shown in Figure 5.13. This tests the ability of a method to model shock waves that are not aligned with the mesh. The tensor artificial viscosity formulation has already been tested on the Saltzman piston by [5], [23], and [22], but we aim to demonstrate the effectiveness of high order elements on this problem.

$$x(i, j) = (i - 1) \frac{x_{max} - x_{min}}{NZ_x} + (NZ_y + 1 - j) \sin \left( \frac{\pi(i - 1)}{NZ_x} \right) \frac{y_{max} - y_{min}}{NZ_y} \quad (5.1)$$

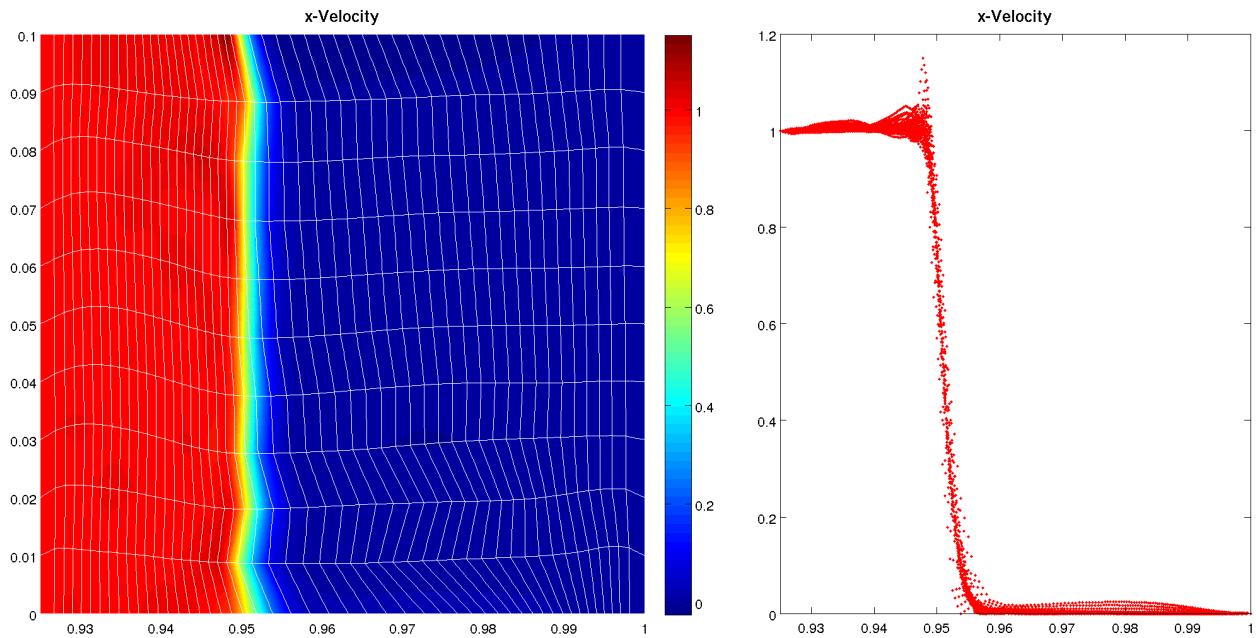


**Figure 5.13: Initial perturbed Saltzman piston mesh**

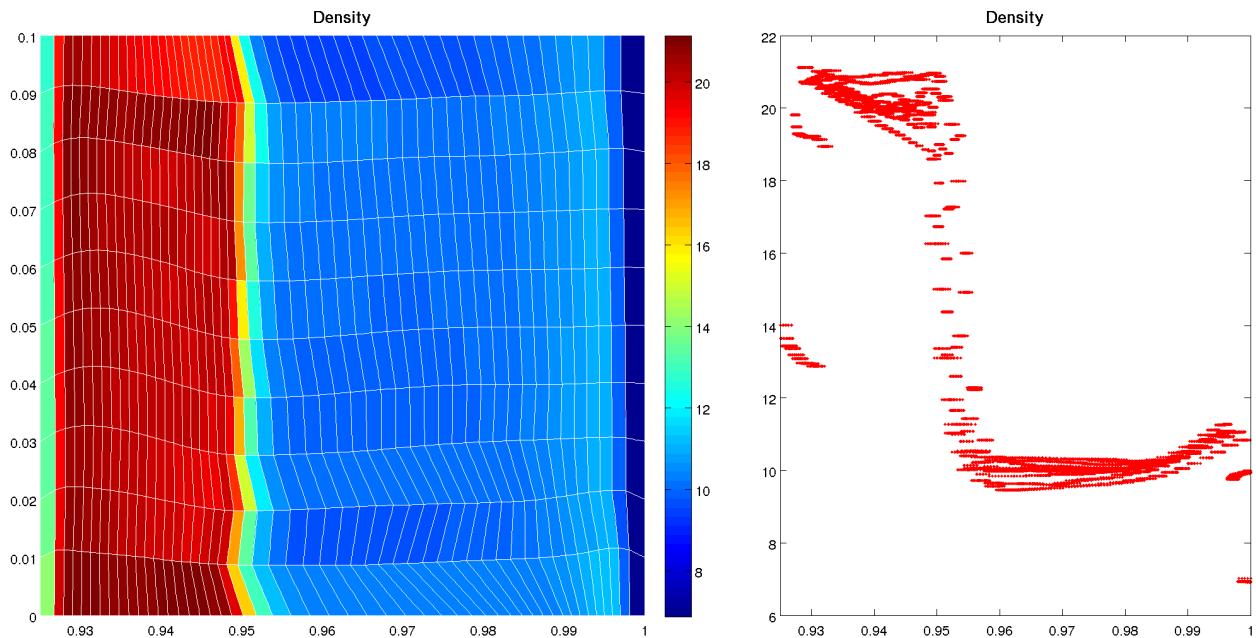
The domain is initially filled with an ideal gas ( $\gamma = 5/3$ ) of density 1 and zero pressure. A piston enters from the left side at a constant speed of 1.0 which generates a shock that reaches the right wall at  $t = 0.8$  and reflects off the left wall again at  $t = 0.9$ . We run the simulation to a time of 0.925 which stops the simulation shortly before the shock reaches the right wall a second time. The analytical pre- and post-shock densities are 10 and 20, respectively. Similar to Noh, this problem also suffers from “wall heating” and the low right and left densities should be overlooked.

### 5.3.1 Low Order $Q_1$ - $Q_0$

The tensor artificial viscosity allows this solution to progress smoothly without the mesh tangling that we see with bulk artificial viscosity formulations[2]. The shock front is a little jagged because the shock must tackle a whole piece-wise constant density cell at a time rather than smoothly traversing it with a bilinear or bi-quadratic thermodynamic interpolation. Thus the shock gets ahead of itself at points and behind in others. When we look at the hourglass filtered solution, we see our first evidence that hourglass filters should not be arbitrarily applied with equal magnitude to every problem. They must be tuned for each simulation, and herein lies their Achilles heel. The simulationist must invest a significant amount of time running and rerunning any new problem to determine the optimal hourglass filter setting. We seek either a filter that does not require tuning or preferably an element pair that will be naturally hourglass free. We have not yet arrived at a suitable solution. Figure 5.15 demonstrates that an hourglass filter can do more to harm a solution than to fix it. Even without an hourglass filter, we see some post-shock waves in the mesh where we should see horizontal lines.

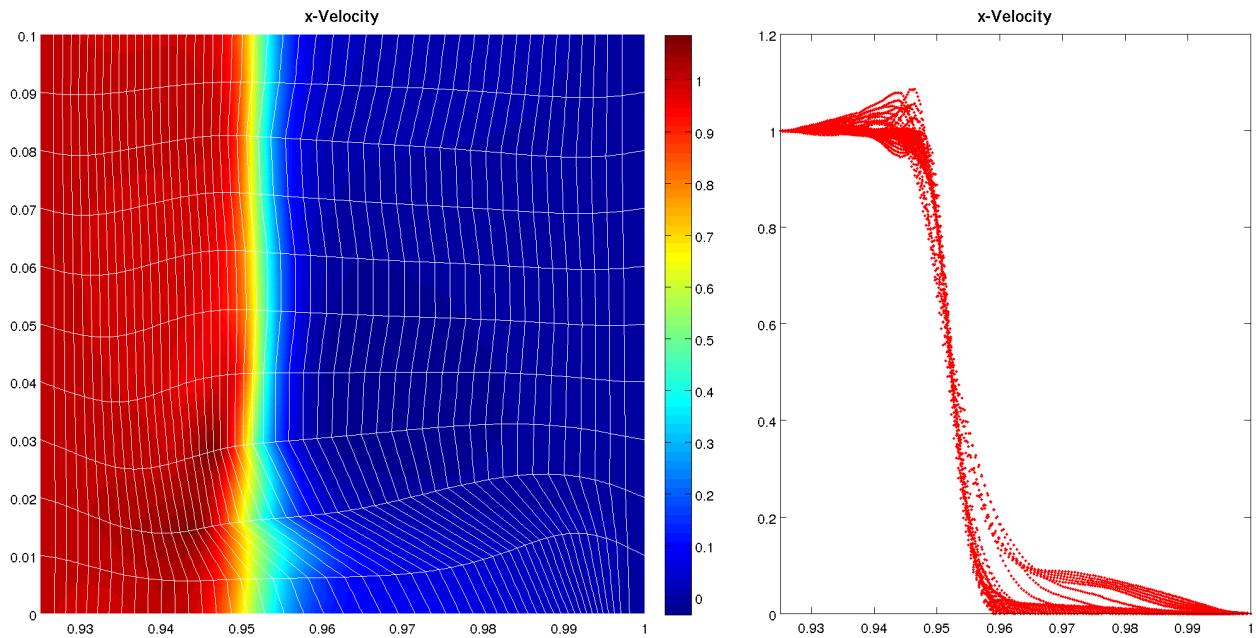


(a) Pseudocolor plot of  $x$ -velocity (*left*). Scatter plot of  $x$ -velocity (*right*).

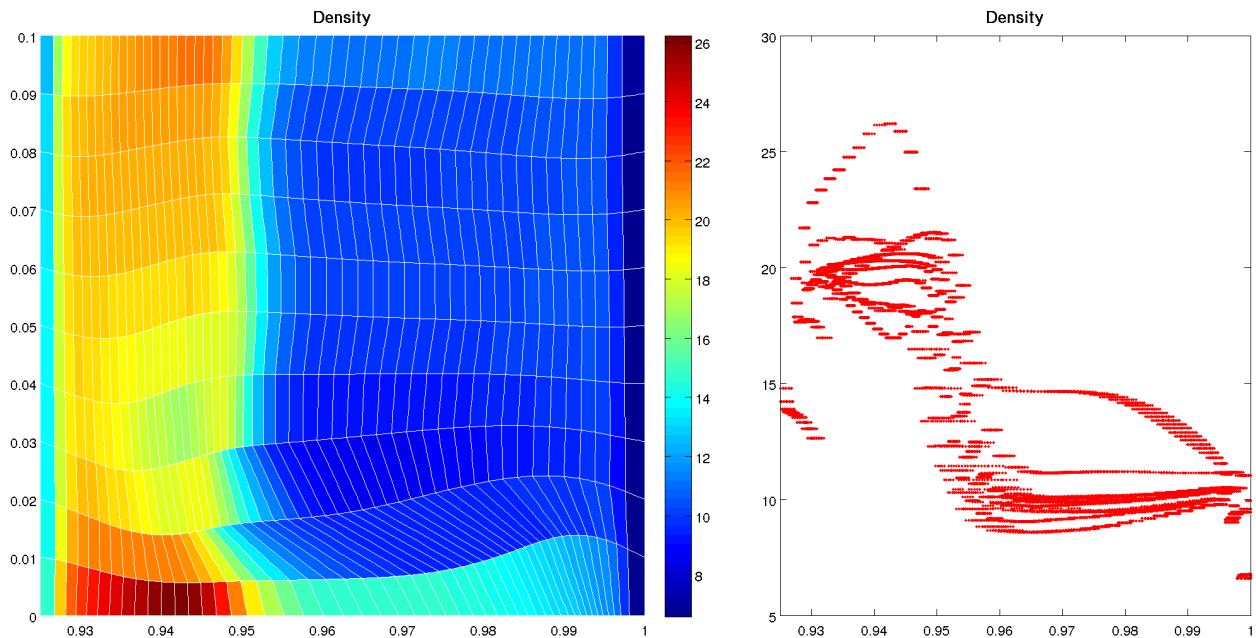


(b) Density pseudocolor plot (*left*). Density scatter plot (*right*).

**Figure 5.14:** Saltzman piston problem on  $50 \times 10$  mesh using  $Q_1-Q_0$  elements without hourglass filtering at  $t=0.925$



(a) Pseudocolor plot of  $x$ -velocity (*left*). Scatter plot of  $x$ -velocity (*right*).

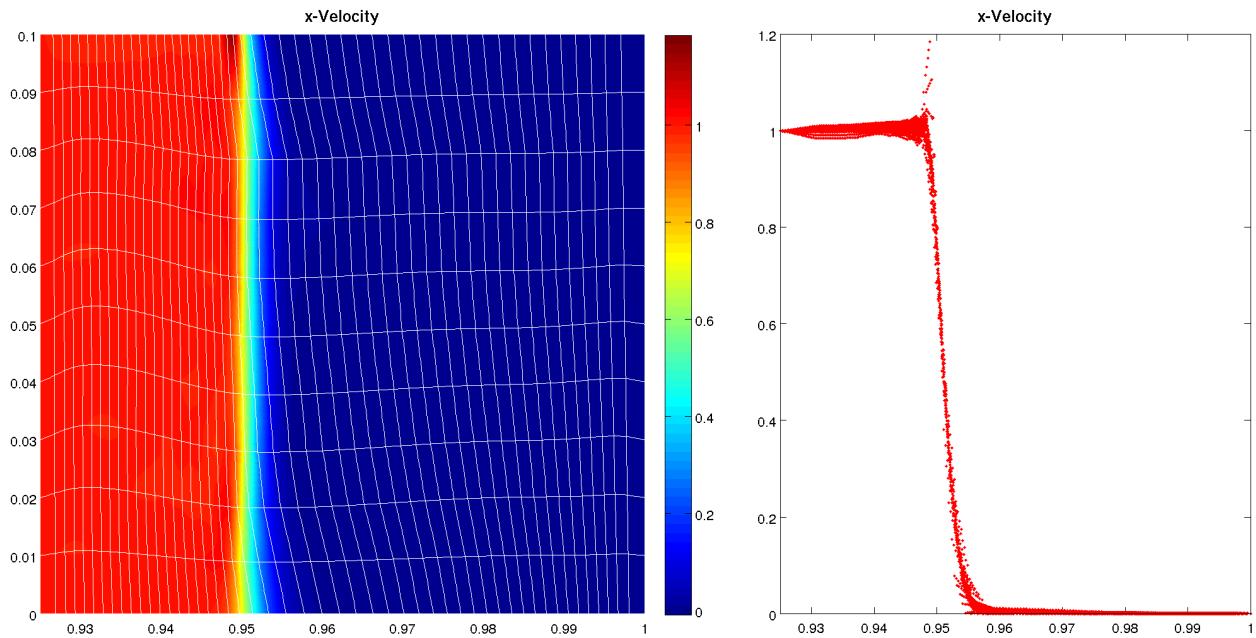


(b) Density pseudocolor plot (*left*). Density scatter plot (*right*).

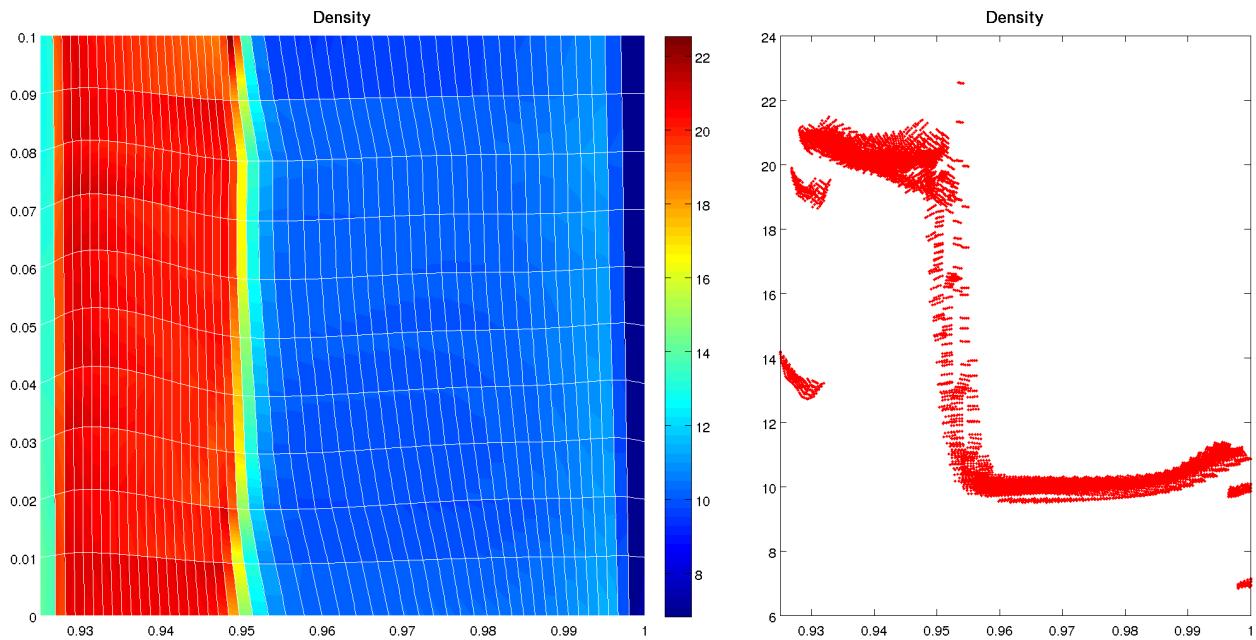
**Figure 5.15: Saltzman piston problem on  $50 \times 10$  mesh using  $Q_1-Q_0$  elements with hour-glass filtering at  $t=0.925$**

### 5.3.2 Low Order $Q_1$ - $\hat{Q}_1$

We get some very positive results when we update to a bi-linear thermodynamic space. The additional pressure and density information allows the shock to move through the domain with less interference to the mesh. In Figure 5.16(b) we can see certain zones partly shocked while one corner remains in its pre-shock condition. Similar to  $Q_1$ - $Q_0$ , we also see some incorrect post-shock waves in the mesh. Overall, however, this is a very solid solution.



(a) Pseudocolor plot of  $x$ -velocity (*left*). Scatter plot of  $x$ -velocity (*right*).

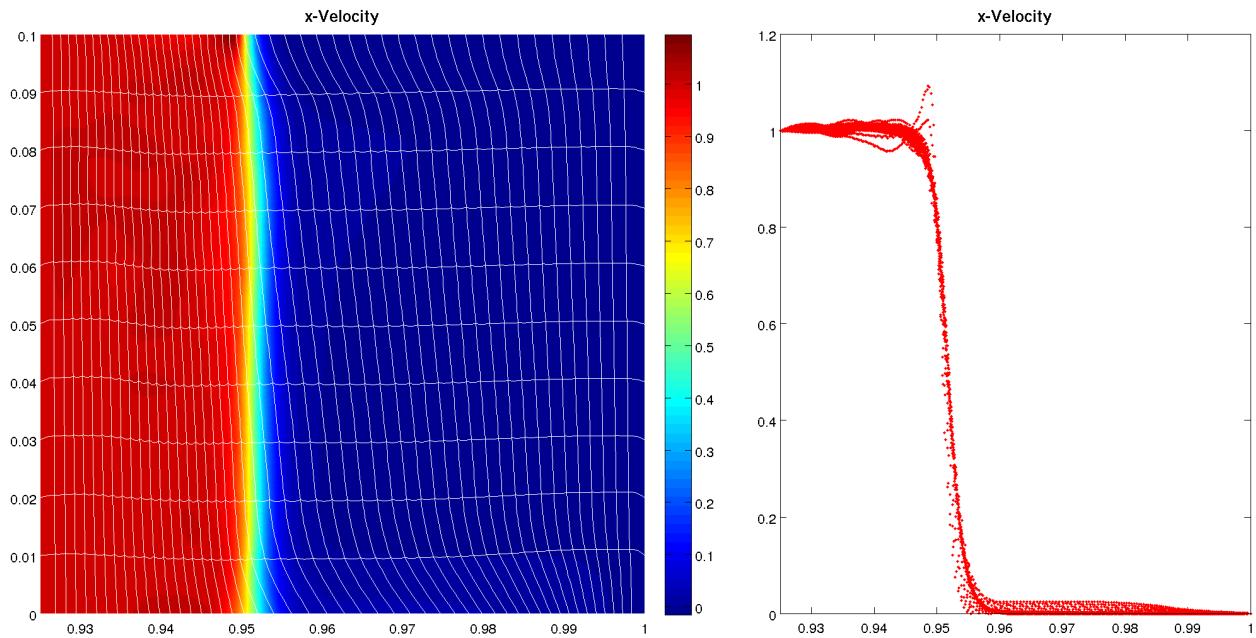


(b) Density pseudocolor plot (*left*). Density scatter plot (*right*).

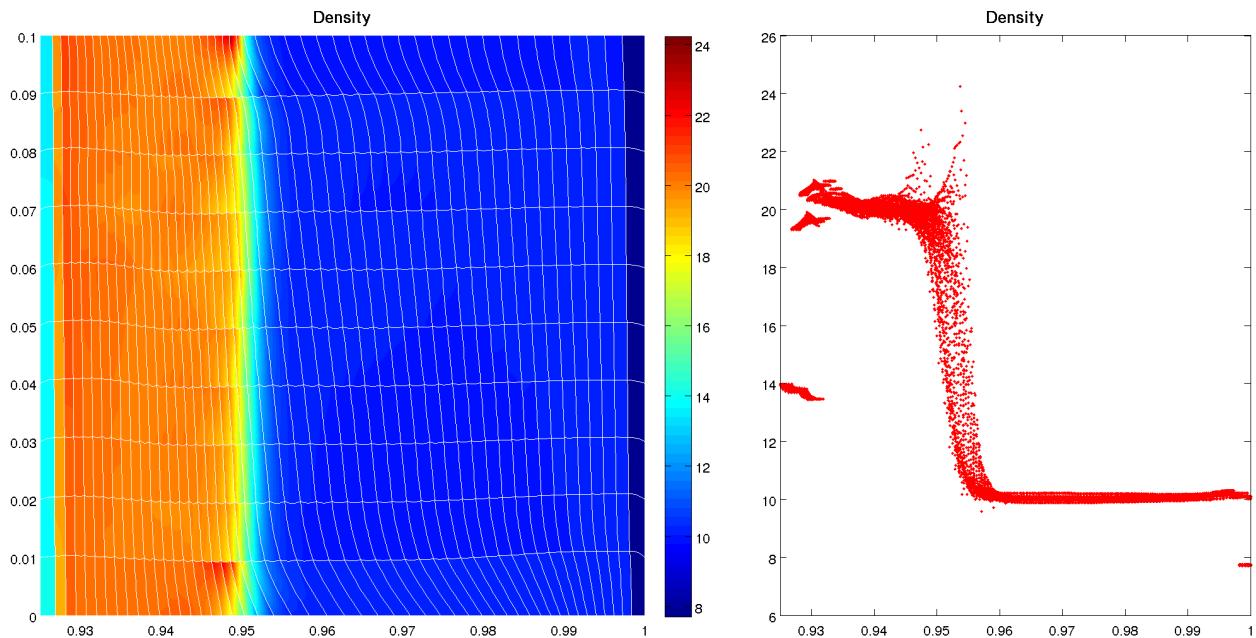
**Figure 5.16: Saltzman piston problem on  $50 \times 10$  mesh using  $Q_1-\hat{Q}_1$  elements without hourglass filtering at  $t=0.925$**

### 5.3.3 High Order $Q_2$ - $\hat{Q}_1$

Looking primarily at the mesh,  $Q_2$ - $\hat{Q}_1$  produces the best solution so far. The horizontal mesh lines are straighter than and the shock more uniform than any of the lower order results. The shock has a lot more flexibility to travel unimpeded across each cell layer because of high order information within each zone. The shock itself is very sharp with some velocity overshoots and significant density overshoots. The chief failing of this method is in the manifestation of the hourglass modes. Indeed, this simulation will not run to completion without an hourglass filter. We see the opposite scenario with the Sedov problem -  $Q_1$ - $Q_0$  will not run without and hourglass filter, but  $Q_2$ - $\hat{Q}_1$  will. This further illustrates the unpredictable nature of hourglass modes. It is very hard to tell if and when they will throw a wrench in a simulation.



(a) Pseudocolor plot of  $x$ -velocity (*left*). Scatter plot of  $x$ -velocity (*right*).

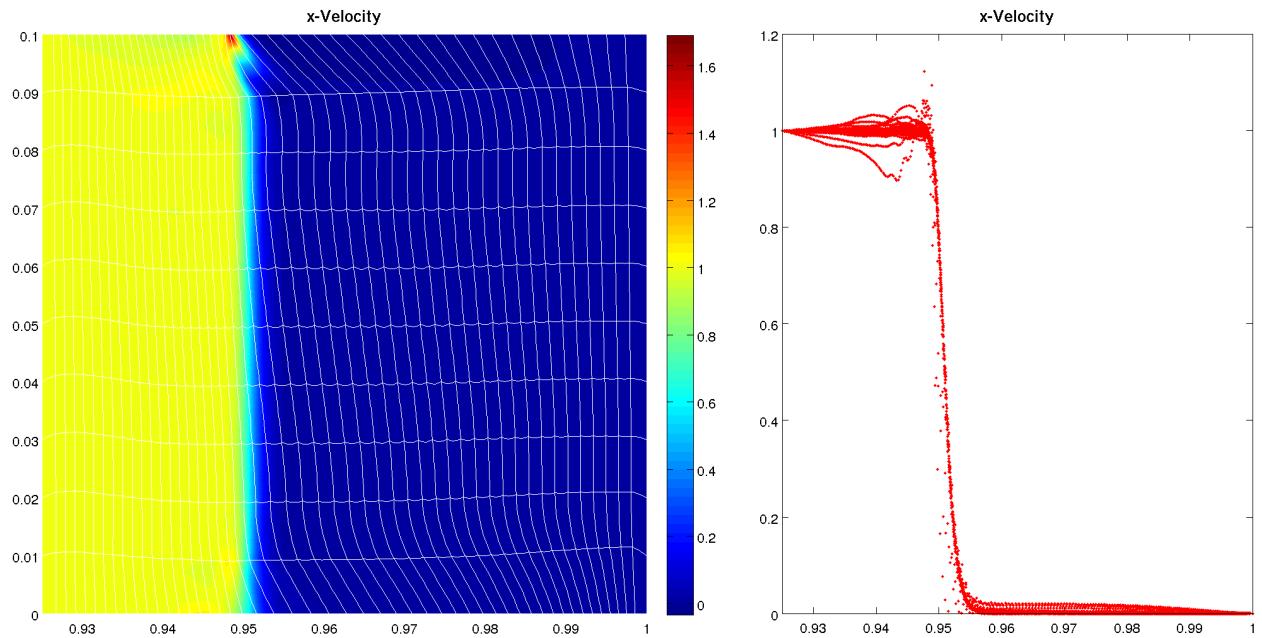


(b) Density pseudocolor plot (*left*). Density scatter plot (*right*).

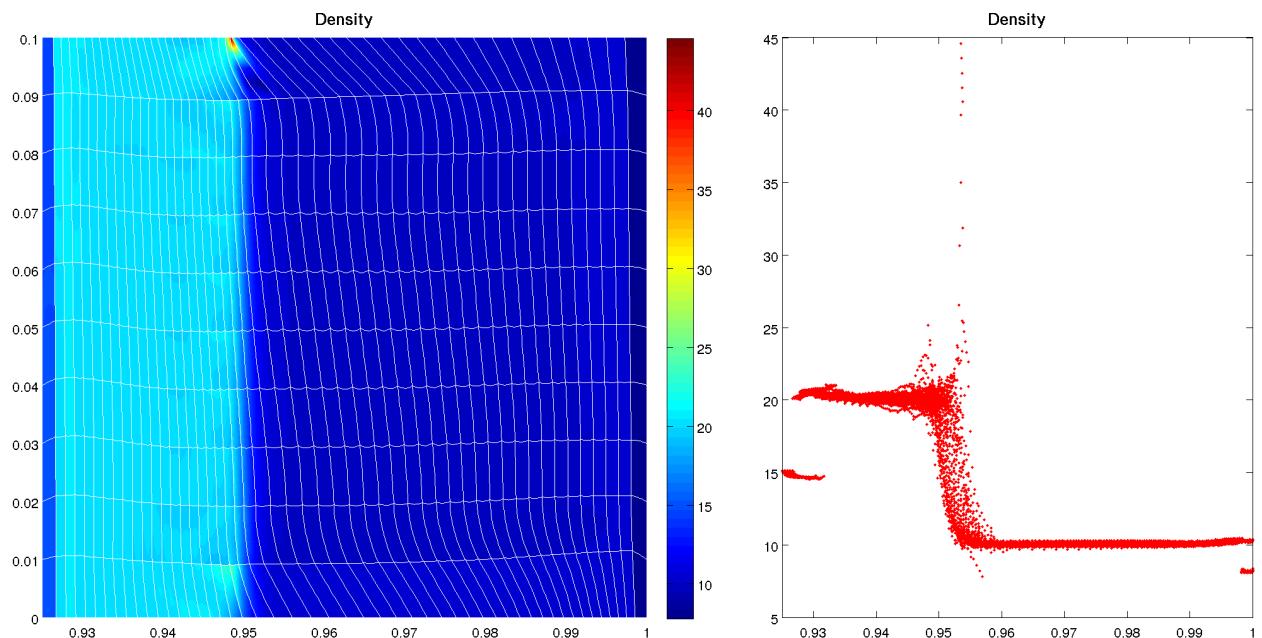
**Figure 5.17:** Saltzman piston problem on  $50 \times 10$  mesh using  $Q_2-\hat{Q}_1$  elements with hour-glass filtering at  $t=0.925$

### 5.3.4 High Order $Q_2$ - $\hat{Q}_2$

A higher order thermodynamic field allows us to run the simulation to completion without an hourglass mode, but we still see evidence of near failure at the shock front near the top surface. A near mesh-collapse has produced a “hot spot” of very high density and pressure. Aside from this, the mesh looks very good with relatively straight horizontal mesh lines. We get a more stable solution with an hourglass filter except for an anomalous “cold spot” in the center of the post-shock velocity field. With an hourglass filter, the overall solution appears to be superior to the low order methods but inferior to the  $Q_2$ - $\hat{Q}_1$  results with an hourglass filter.

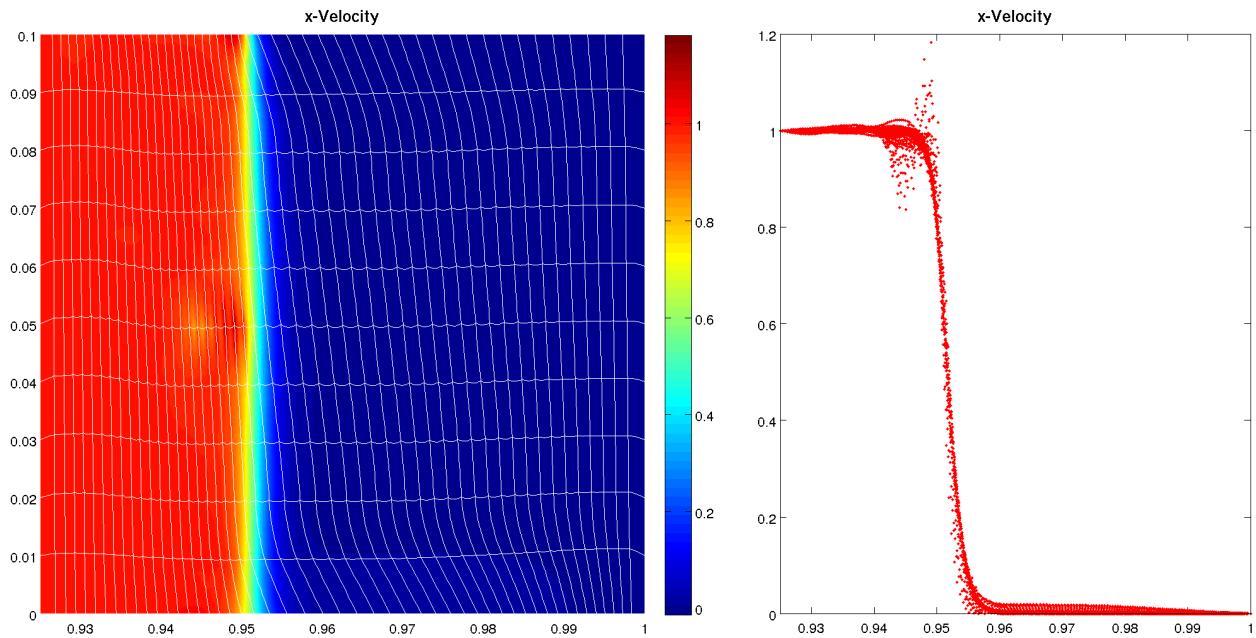


(a) Pseudocolor plot of  $x$ -velocity (*left*). Scatter plot of  $x$ -velocity (*right*).

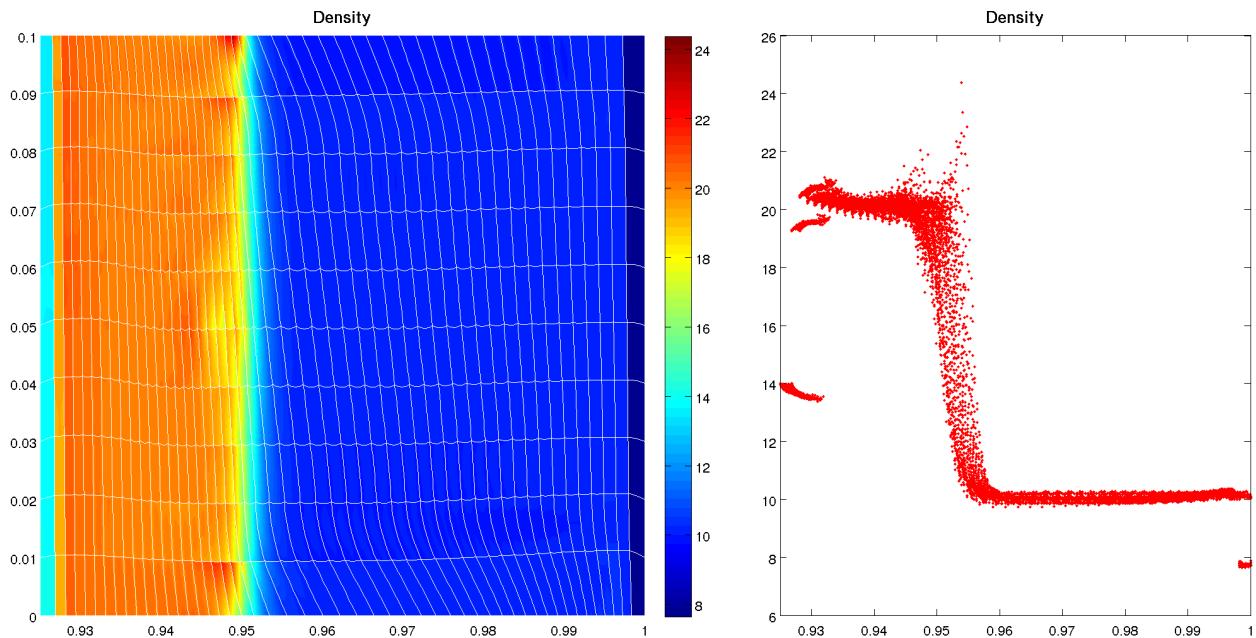


(b) Density pseudocolor plot (*left*). Density scatter plot (*right*).

**Figure 5.18: Saltzman piston problem on  $50 \times 10$  mesh using  $Q_2-\hat{Q}_2$  elements without hourglass filtering at  $t=0.925$**



(a) Pseudocolor plot of  $x$ -velocity (*left*). Scatter plot of  $x$ -velocity (*right*).

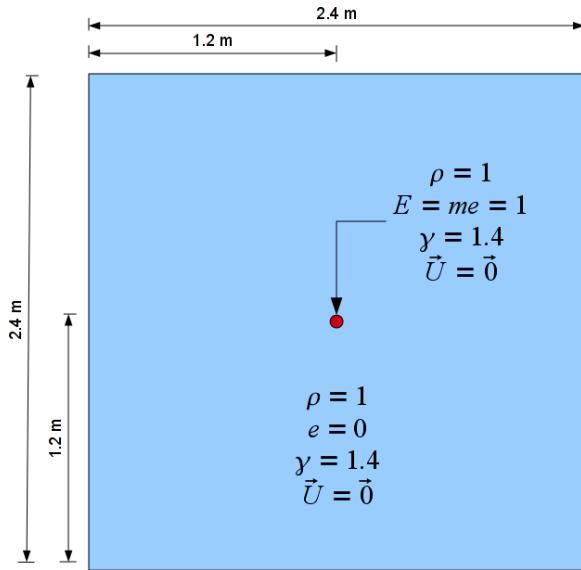


(b) Density pseudocolor plot (*left*). Density scatter plot (*right*).

**Figure 5.19:** Saltzman piston problem on  $50 \times 10$  mesh using  $Q_2-\hat{Q}_2$  elements with hour-glass filtering at  $t=0.925$

## 5.4 The Sedov Blast Wave

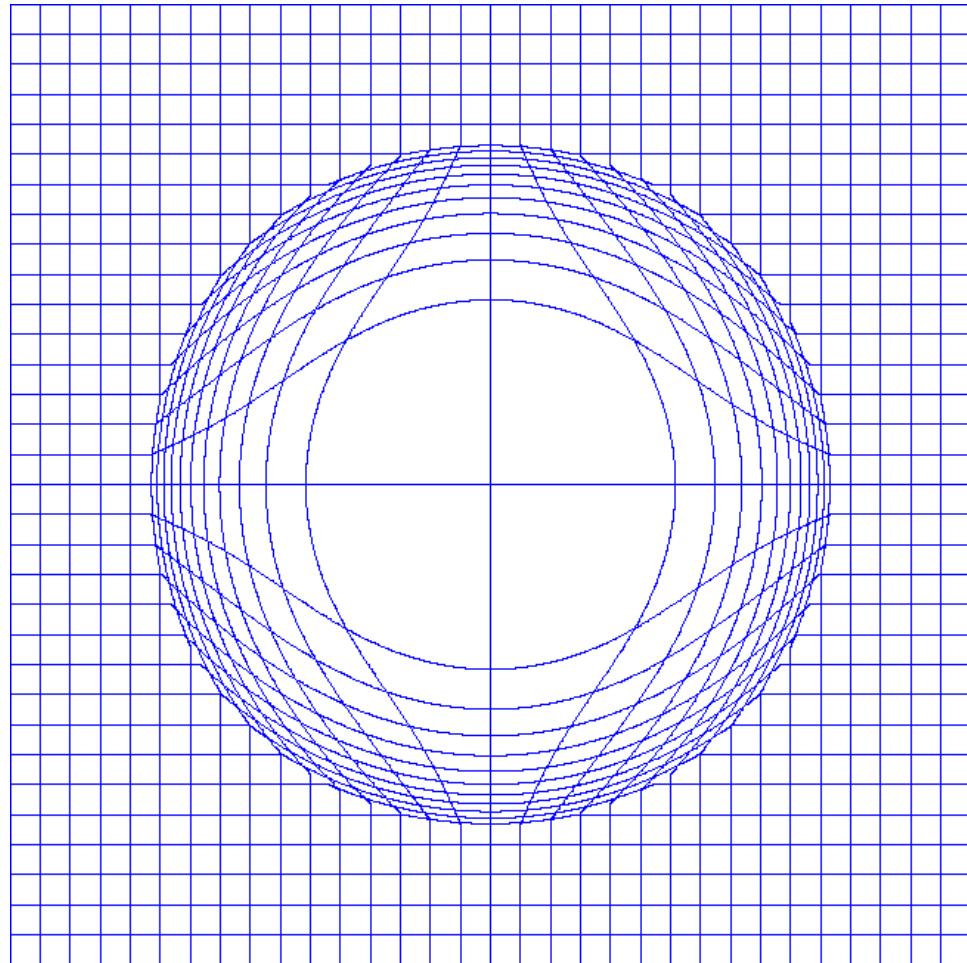
The previous problems were informative, but they did not allow us to demonstrate the true power of high order elements - curvilinear zones. The Sedov explosion problem[31] brings these curvilinear capabilities to the forefront. In two dimensions, the Sedov problem is a point-blast wave in an initially static ideal gas medium of constant density, zero pressure, with a delta function initial source of energy at the origin, as illustrated in Figure 5.20. In plain terms, this is a simplification of an explosion: a lot of energy is released in a small area and then it expands into the surrounding domain. The problem runs to  $t=1.0$  which allows the shock wave to expand to a radius of 1.0.



**Figure 5.20:** Problem setup for Sedov blast wave problem

We mentioned previously that the Sedov blast wave presents one of the most compelling arguments for using curvilinear elements. By its very nature, this problem seeks to curve geometries that may have been initially Cartesian. Indeed, when we apply the exact solution of this problem to an initially Cartesian grid, as we do in Figure 5.21, it becomes obvious that anything other than curvilinear

elements would introduce undesirable inaccuracies into the mesh geometry, even under refinement.



**Figure 5.21:** Exact solution to the Sedov problem applied to an initially Cartesian grid

#### 5.4.1 Low Order $Q_1$ - $Q_0$

Historically, this problem has presented significant challenges to  $Q_1$ - $Q_0$  because hourglass mode instabilities cause the solution to crash if they are not filtered out. We would like to compare each of our methods with and without an hourglass filter, but it is simply impossible to obtain an hourglass filter-less

solution with  $Q_1$ - $Q_0$ . If we do apply an hourglass filter we are able to get a stable solution, as seen in Figure 5.22 with a very coarse  $10 \times 10$  mesh.

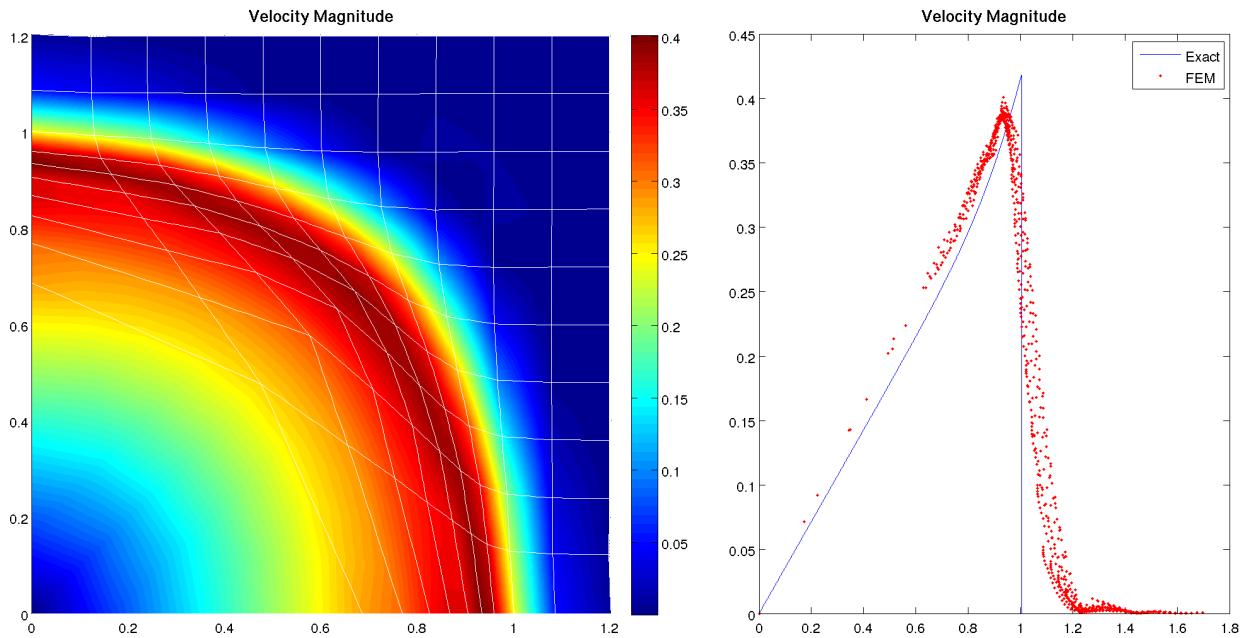
The Sedov problem is inherently circular. A point source of energy should produce a circular blast wave expanding out from the origin. Therein we see a significant limitation of low order methods. We are only able to produce straight-edged elements with  $Q_1$ - $Q_0$  which interfere with the curved nature of the problem. With this coarse mesh we can see how the zone at the origin (which doesn't have enough DOFs to represent a curved edge) induces a kinked shock wave where we should see a nice smooth radius. This problem can be mitigated by adding more elements, but it still exists as a significant source of error when modeling fundamentally curved phenomena.

You will also notice that the shock has been smeared considerably. This is primarily the fault of the piece-wise constant representation for density and pressure. We can thank artificial viscosity for smoothing the shock discontinuity out enough to allow a numerical solution, but we still expect to see sharp gradients of density, pressure, and velocity. The low order method can only take one step in density or pressure per zone, thus we see the shock being spread over several zones to smooth out the sharp change in value. We can alleviate this shock smoothing by tweaking the artificial viscosity coefficients, but the fact remains that low order methods are fundamentally limited in their ability to capture sharp gradients.

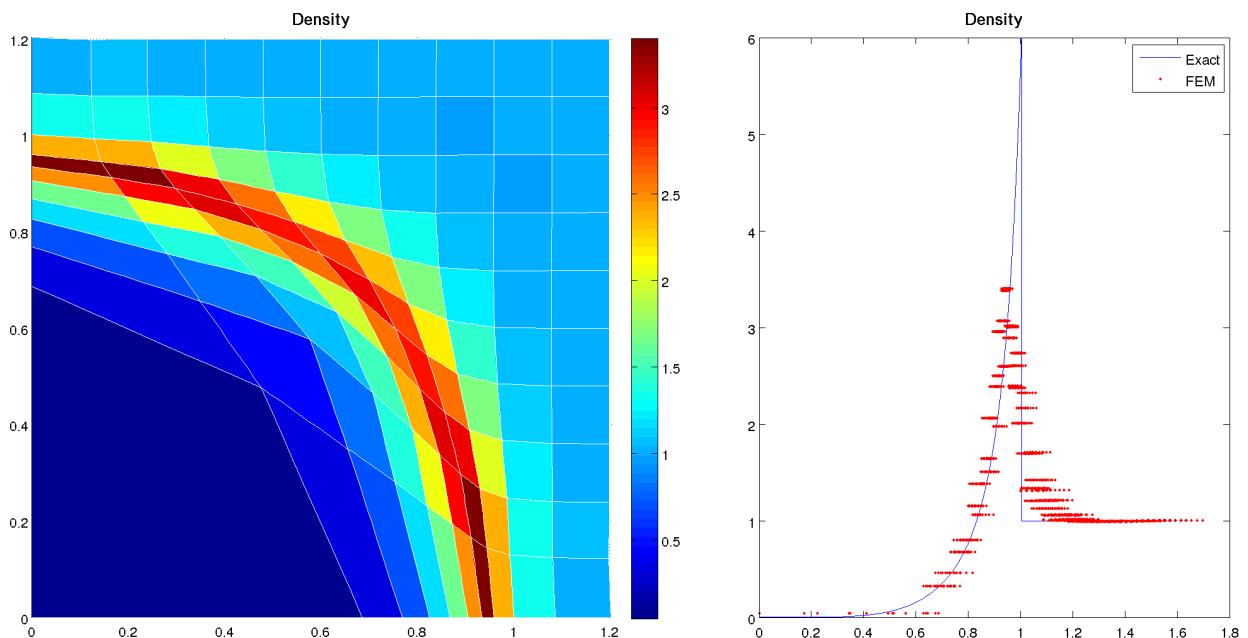
This low order representation of density and pressure also limits our ability to accurately predict the maximum density and pressure in the shock wave. The constant density interpolation is only able to represent averages over a zone without any additional fluctuations within that zone. Thus, for rapidly changing phenomena like a blast wave, we are very limited in what we can learn without

extensive mesh refinement.

We are able to see nominal improvement if we double the resolution in both the  $x$ - and  $y$ - directions, as seen in Figure 5.23. While higher resolution attains closer correspondence to the exact solution, the density spike is still grossly underpredicted, and the zone at the origin is still fails to resemble the exact solution in Figure 5.21.

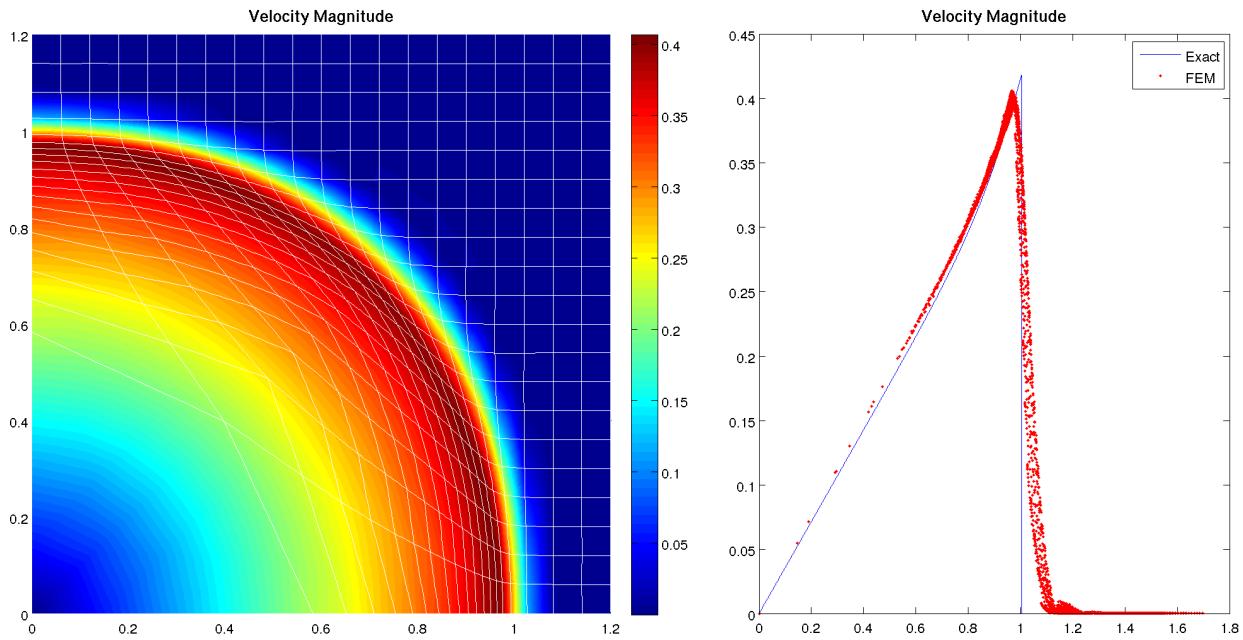


(a) Velocity magnitude pseudocolor plot (*left*). Velocity magnitude scatter plot compared to exact solution (*right*).

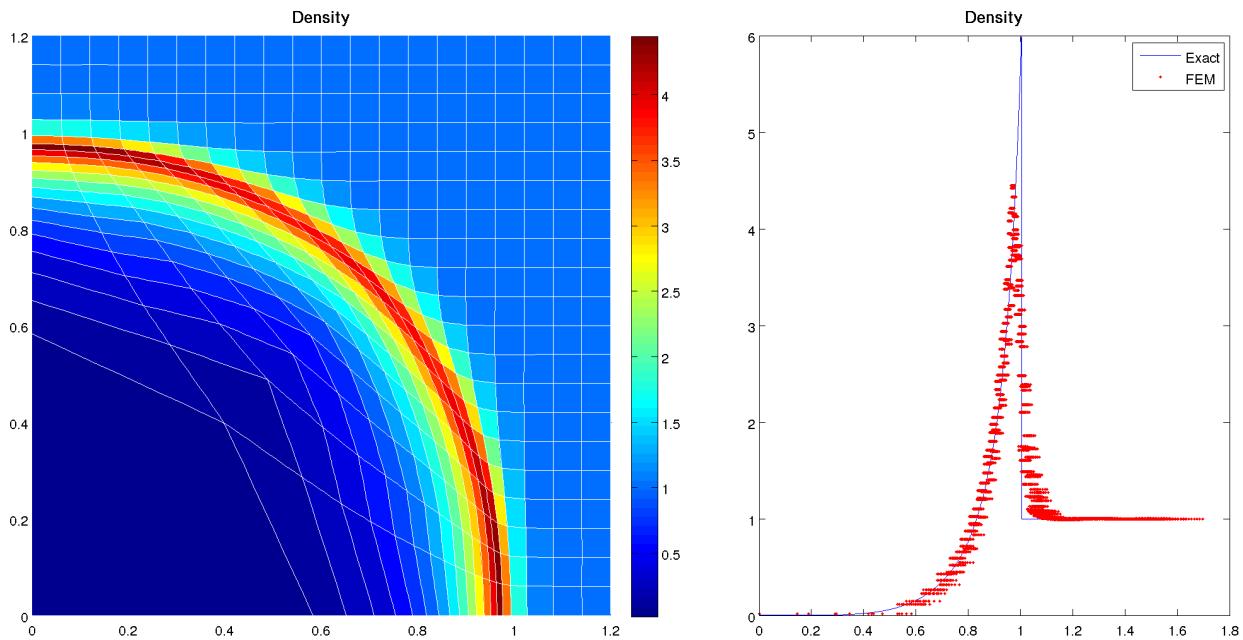


(b) Density pseudocolor plot (*left*). Density scatter plot compared to exact solution (*right*).

**Figure 5.22:** Sedov explosion problem on  $10 \times 10$  mesh using  $Q_1-Q_0$  elements with hour-glass filtering



(a) Velocity magnitude pseudocolor plot (*left*). Velocity magnitude scatter plot compared to exact solution (*right*).

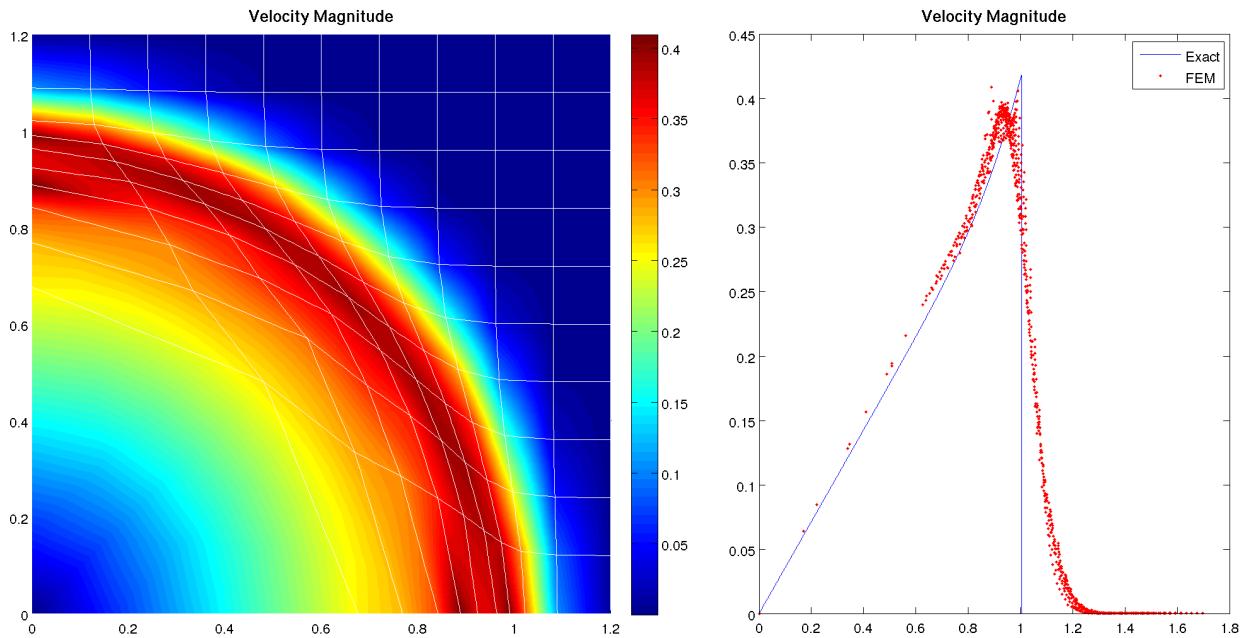


(b) Density pseudocolor plot (*left*). Density scatter plot compared to exact solution (*right*).

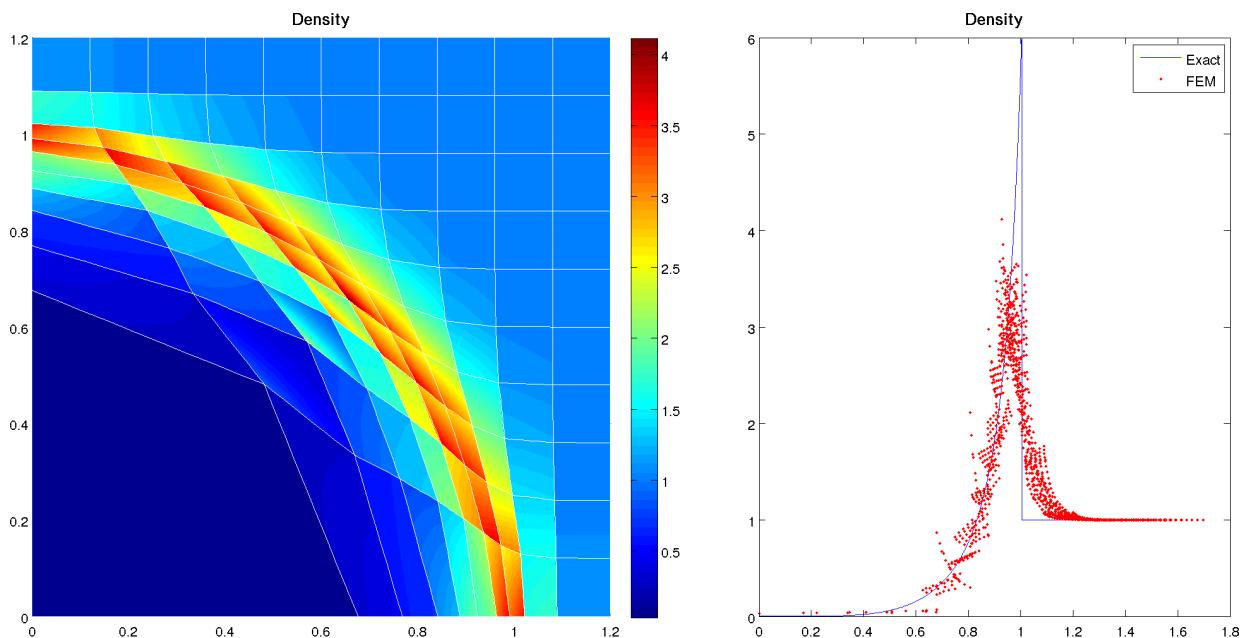
**Figure 5.23:** Sedov explosion problem on  $20 \times 20$  mesh using  $Q_1-Q_0$  elements with hour-glass filtering

### 5.4.2 Low Order $Q_1$ - $\hat{Q}_1$

As discussed previously, the main benefit of the  $Q_1$ - $\hat{Q}_1$  element is that it appears to squelch hourglass modes in certain situations. We saw in Section 4.7.4 that this element introduces new spurious modes, but in the Sedov problem, at least, these modes appear to be negligible. The immediate improvement we see when we compare the  $10 \times 10$  results to  $Q_1$ - $Q_0$  is that the higher order density field allows the shock to better maintain a circular shape. This results in a more tightly grouped, and hence more accurate, velocity scatter plot. The density results also adhere more closely to the exact solution, approaching closer to the peak value. Despite the straight-edged zones, this is a very good solution overall, and it only gets better with more elements. The shock is still more blurred than higher order methods can give us and the density still does not quite reach the accurate peak value.

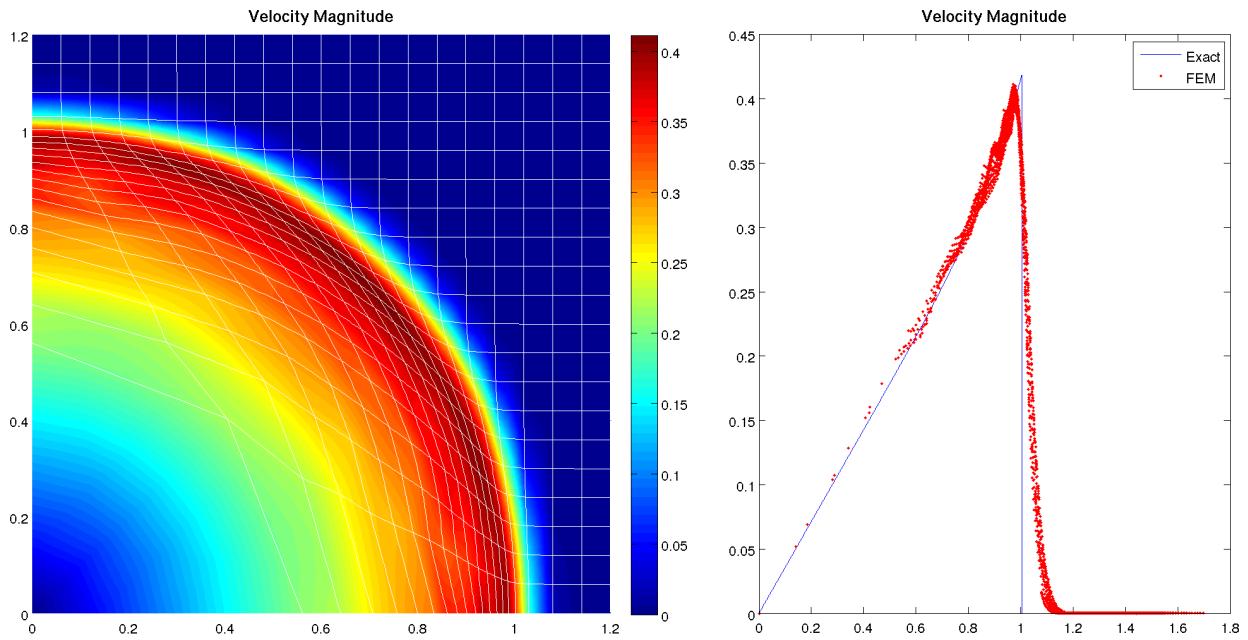


(a) Velocity magnitude pseudocolor plot (*left*). Velocity magnitude scatter plot compared to exact solution (*right*).

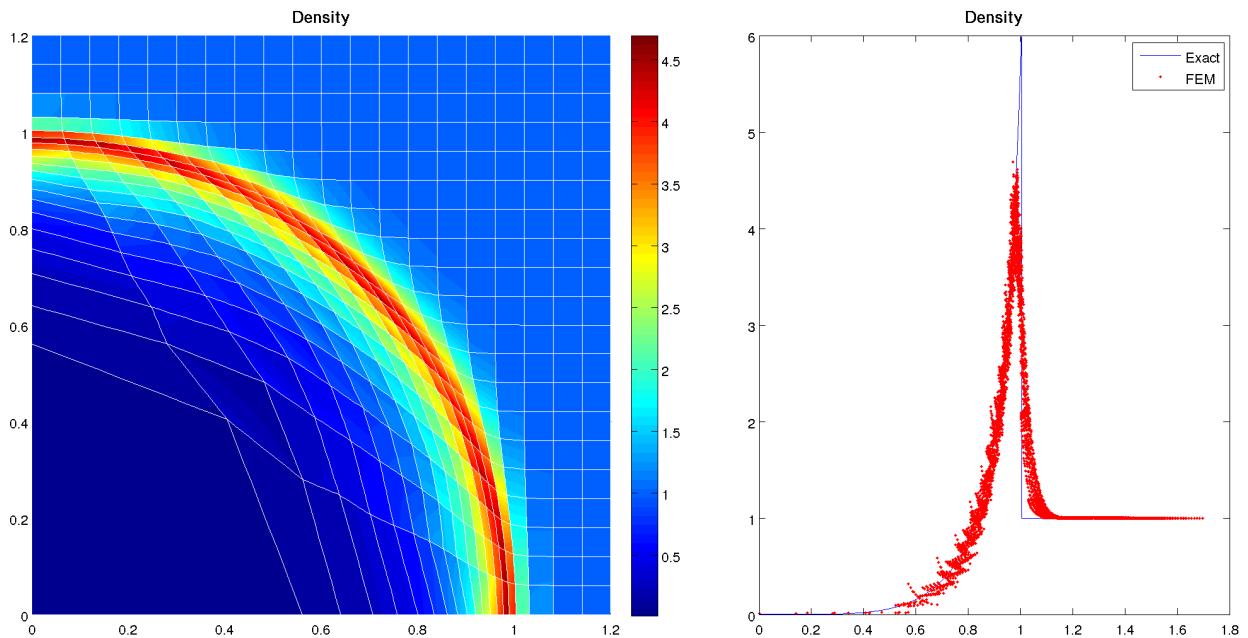


(b) Density pseudocolor plot (*left*). Density scatter plot compared to exact solution (*right*).

**Figure 5.24:** Sedov explosion problem on  $10 \times 10$  mesh using  $Q_1-\hat{Q}_1$  elements without hourglass filtering



(a) Velocity magnitude pseudocolor plot (*left*). Velocity magnitude scatter plot compared to exact solution (*right*).



(b) Density pseudocolor plot (*left*). Density scatter plot compared to exact solution (*right*).

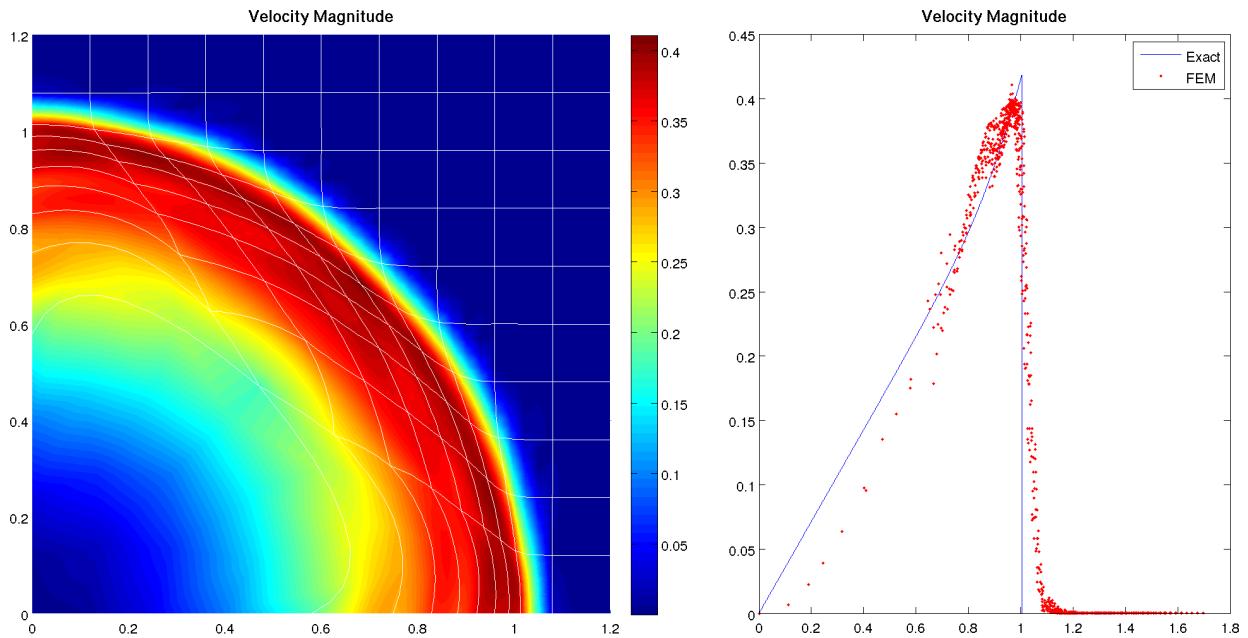
**Figure 5.25:** Sedov explosion problem on  $20 \times 20$  mesh using  $Q_1-\hat{Q}_1$  elements without hourglass filtering

### 5.4.3 High Order $Q_2$ - $\hat{Q}_1$

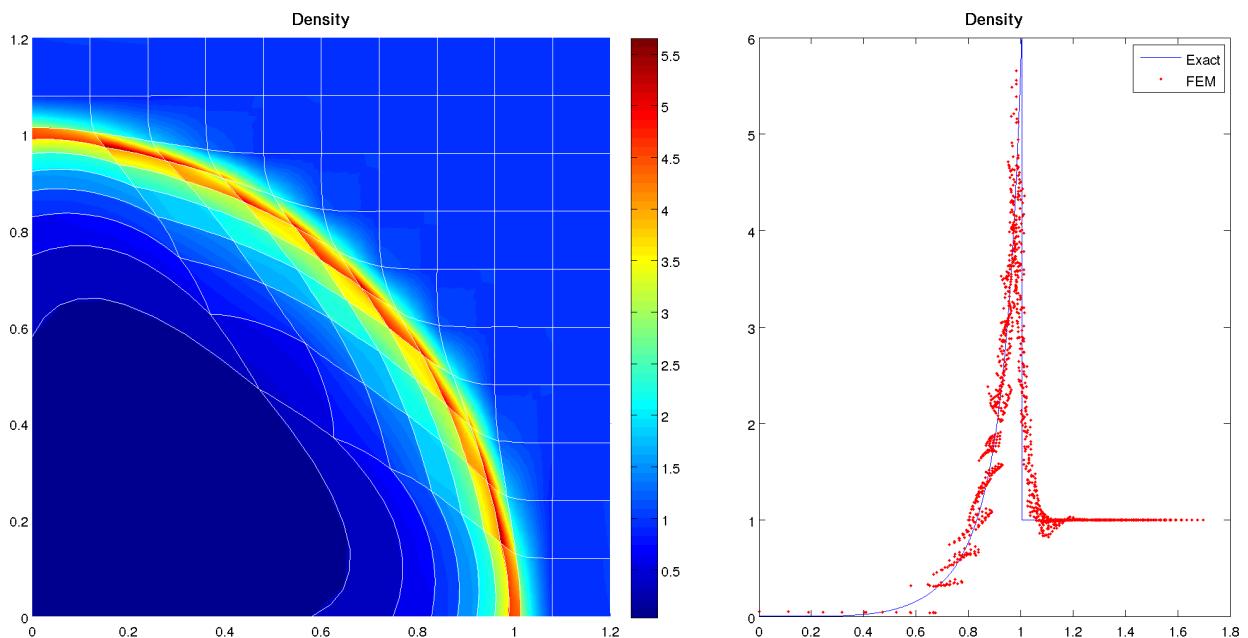
The most prominent feature of the  $Q_2$ - $\hat{Q}_1$  solution is the curved original zone. Without an hourglass filter it looks almost as if it were bowing out from the force of the explosion. This bowing is actually non-physical, but the overall curvature of the element lends very favorably to the accuracy of the scheme. We also see post-shock oscillations in the velocity field. Despite these hourglass-based penalties, the shock capturing is much sharper. At many points we can see the entire shock captured within one zone.

If we apply a simple hourglass filter by leaving 25% of linear artificial viscosity term on in expansion, the results look very fine indeed. The curved zones are reminiscent of the exact solution of the mesh geometry in Figure 5.21. This also eliminates the oscillations and gives us a solution very close the exact. The results only get better under refinement. At  $20 \times 20$  mesh refinement, we see very little variation from the exact answer.

In order to completely level the playing field concerning the limitations of  $Q_1$ - $Q_0$  for curved geometries, we should consider the same problem using the same number of DOFs with curvilinear elements. One of the most significant limitations of low order elements on the Sedov problem was that for a coarse enough grid ( $10 \times 10$ ), the zone at the origin induced a kinked shape on the shock wave downstream. If we wish to compare this result with curvilinear elements with the same number of DOFs, we would need to use the almost absurd mesh resolution of  $5 \times 5$ . We can see in Figure 5.29 that this coarse  $Q_2$ - $\hat{Q}_2$  solution avoids the angular mesh problem and produces a surprisingly symmetric and reasonable result.

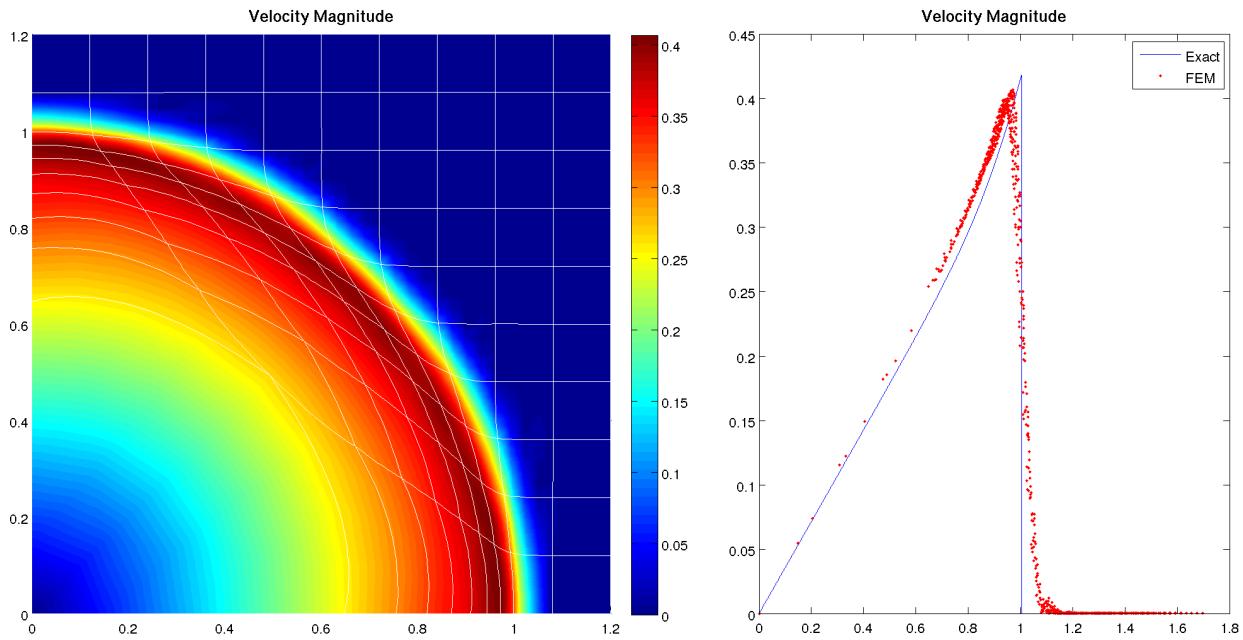


(a) Velocity magnitude pseudocolor plot (*left*). Velocity magnitude scatter plot compared to exact solution (*right*).

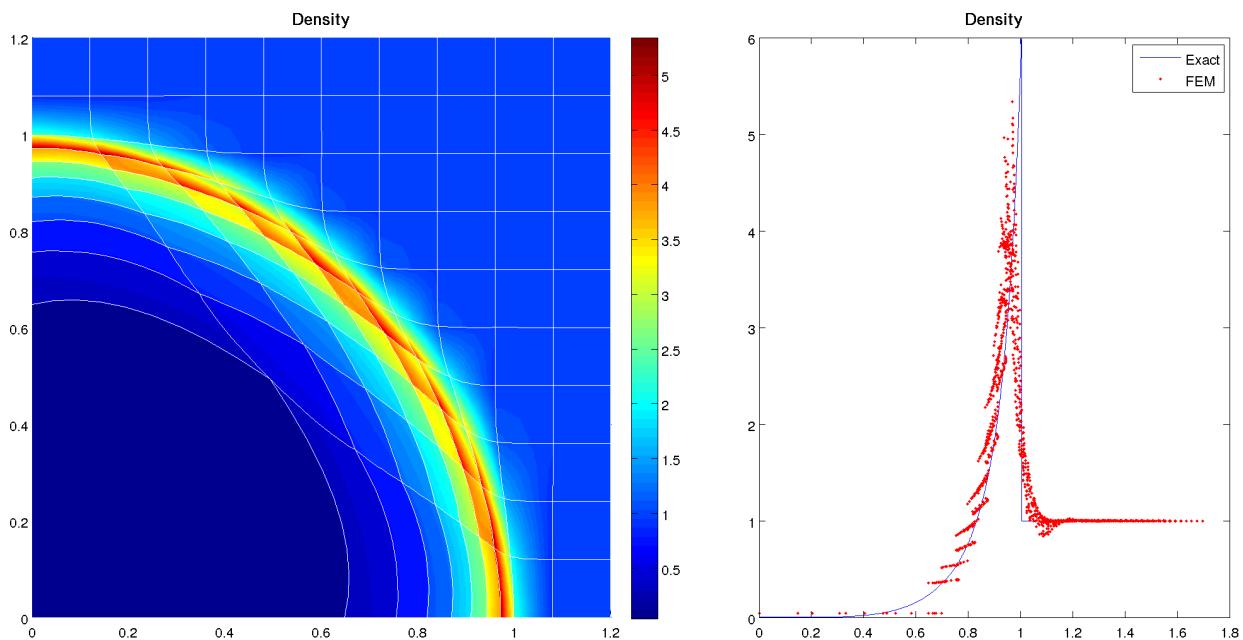


(b) Density pseudocolor plot (*left*). Density scatter plot compared to exact solution (*right*).

**Figure 5.26:** Sedov explosion problem on  $10 \times 10$  mesh using  $Q_2-\hat{Q}_1$  elements without hourglass filtering

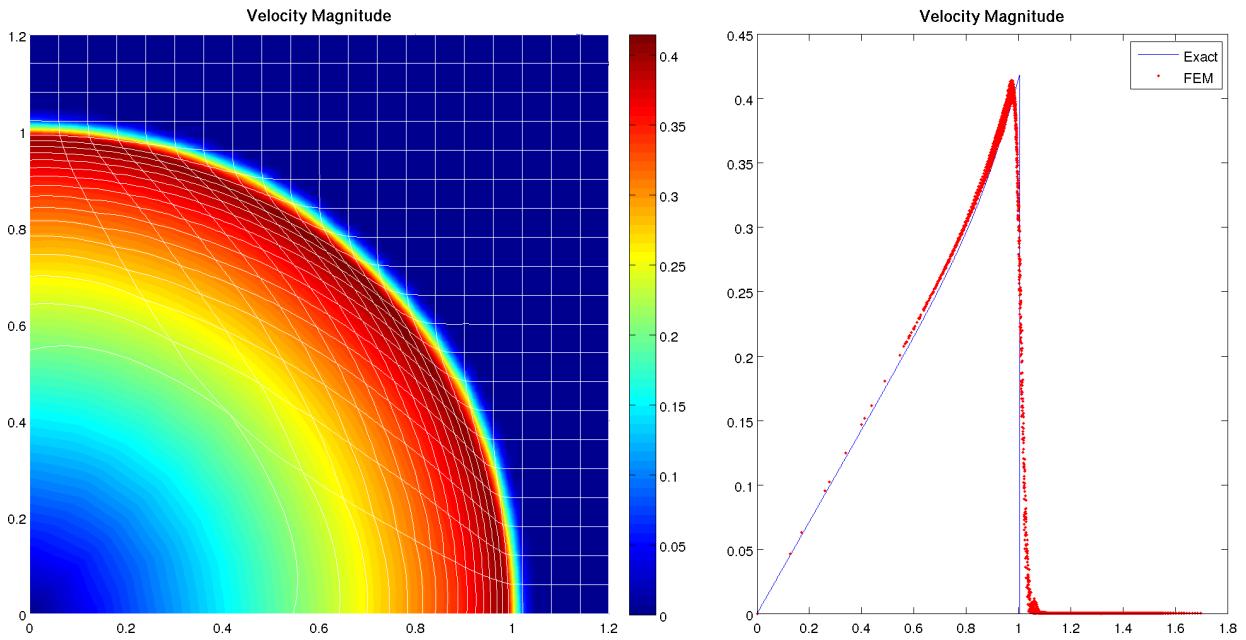


(a) Velocity magnitude pseudocolor plot (*left*). Velocity magnitude scatter plot compared to exact solution (*right*).

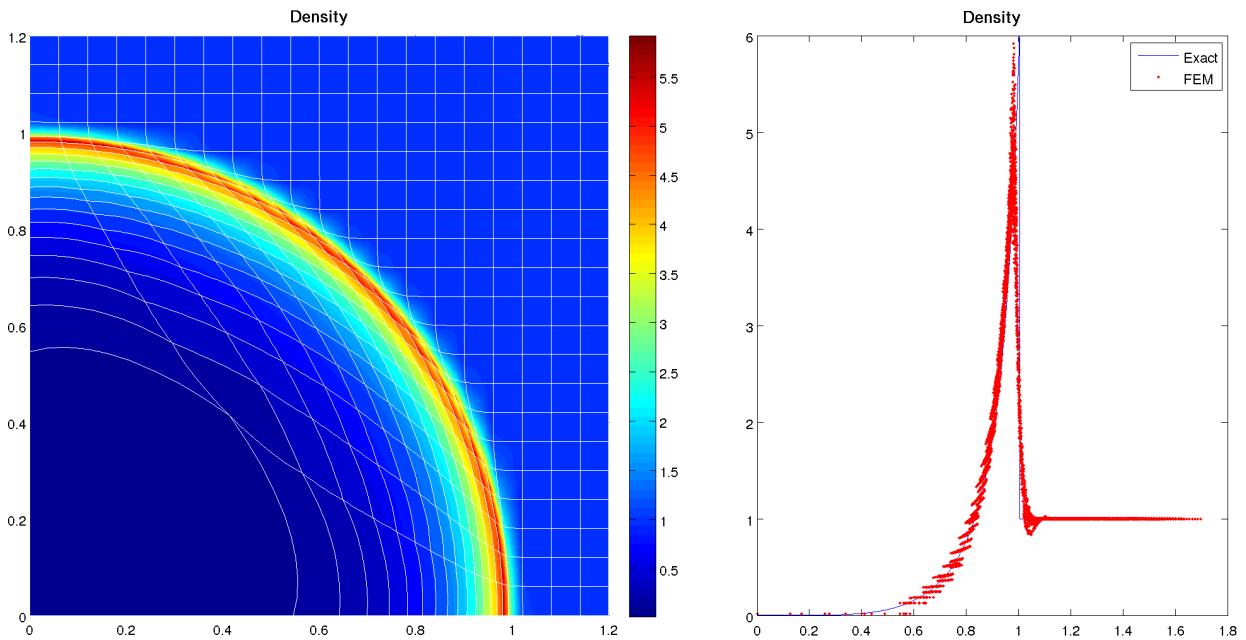


(b) Density pseudocolor plot (*left*). Density scatter plot compared to exact solution (*right*).

**Figure 5.27:** Sedov explosion problem on  $10 \times 10$  mesh using  $Q_2-\hat{Q}_1$  elements with hour-glass filtering

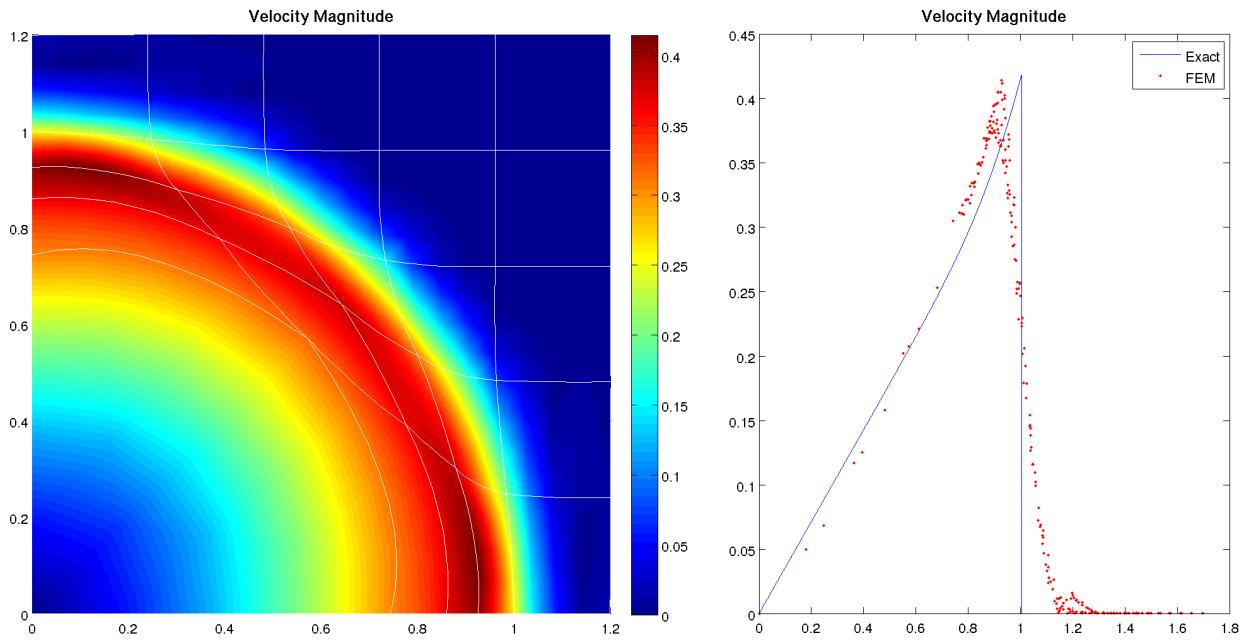


(a) Velocity magnitude pseudocolor plot (*left*). Velocity magnitude scatter plot compared to exact solution (*right*).

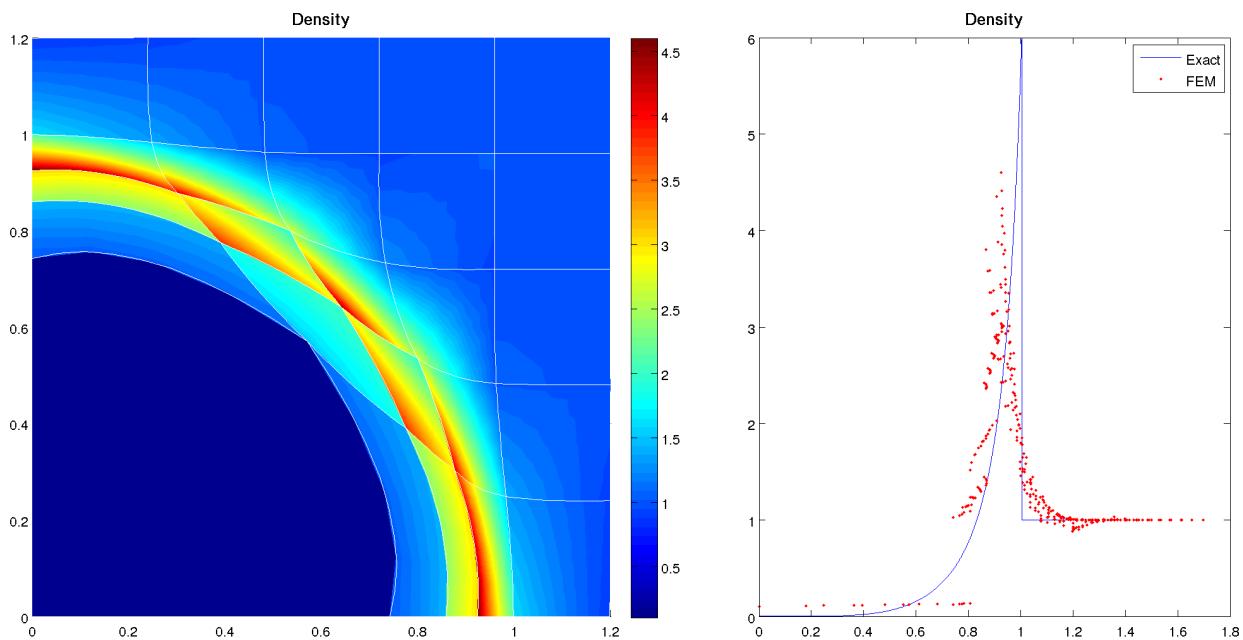


(b) Density pseudocolor plot (*left*). Density scatter plot compared to exact solution (*right*).

**Figure 5.28:** Sedov explosion problem on  $20 \times 20$  mesh using  $Q_2-\hat{Q}_1$  elements with hour-glass filtering



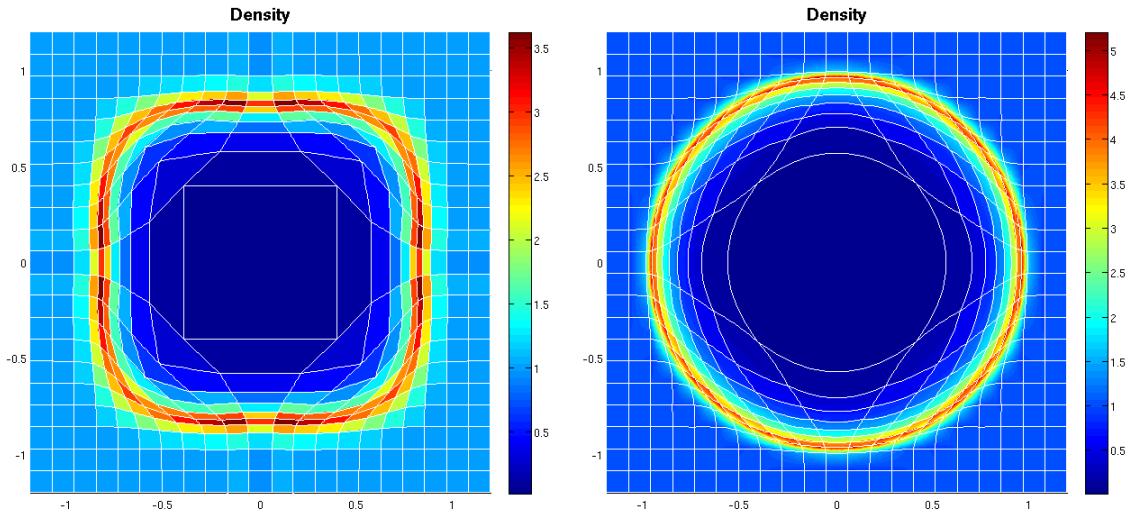
(a) Velocity magnitude pseudocolor plot (*left*). Velocity magnitude scatter plot compared to exact solution (*right*).



(b) Density pseudocolor plot (*left*). Density scatter plot compared to exact solution (*right*).

**Figure 5.29:** Sedov explosion problem on  $5 \times 5$  mesh using  $Q_2-\hat{Q}_1$  elements with hourglass filtering

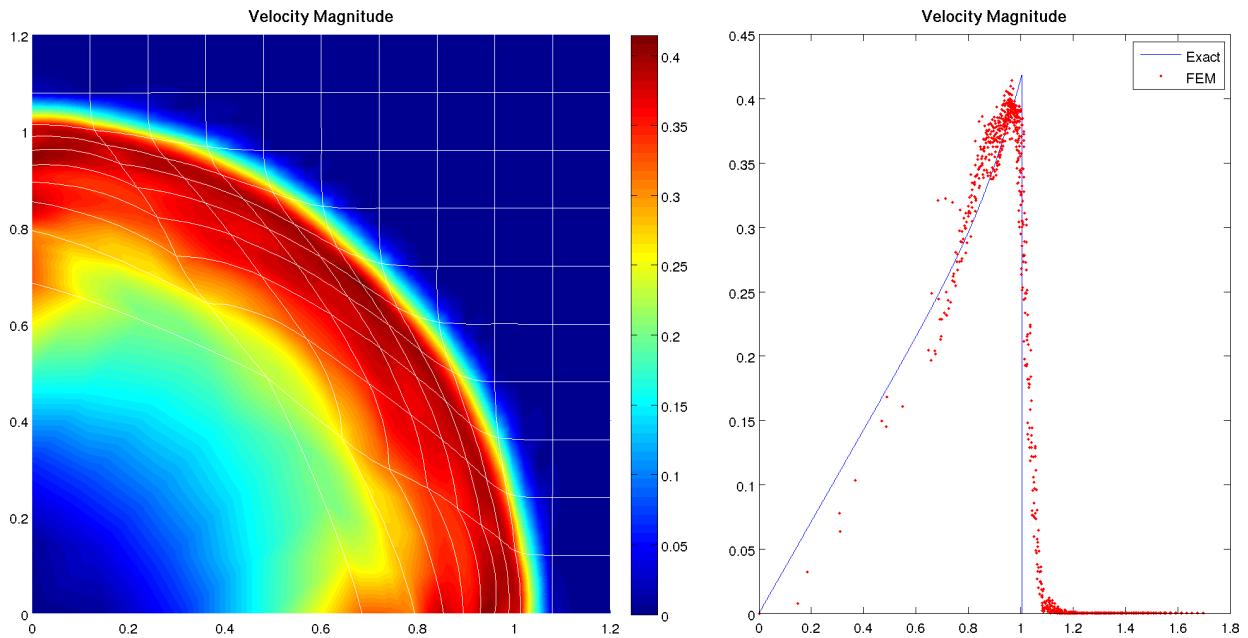
We gain some further insight when we run this simulation on a full mesh, and instead of splitting up the energy source into four cells, we energize a single cell which encompasses the origin. We can see the results of two  $21 \times 21$  runs in Figure 5.30. The  $Q_1-Q_0$  elements lack the flexibility to expand naturally from the force of the explosion, hence the original zone is forced to retain its square shape as it expands. This imposes a square shockfront and causes the corner elements to bend unnaturally until one of the corners exceeds a 180 degrees at  $t=0.868$ . This causes the Jacobian to go negative and the solution to go unstable and the simulation to stop. The bi-quadratic method, on the other hand, possesses the required flexibility to expand naturally with the force of the explosion, and the simulation progresses smoothly to the end at  $t=1.0$ . This is a rather extreme test, but curvilinear elements displays a distinct advantage of straight-edged methods.



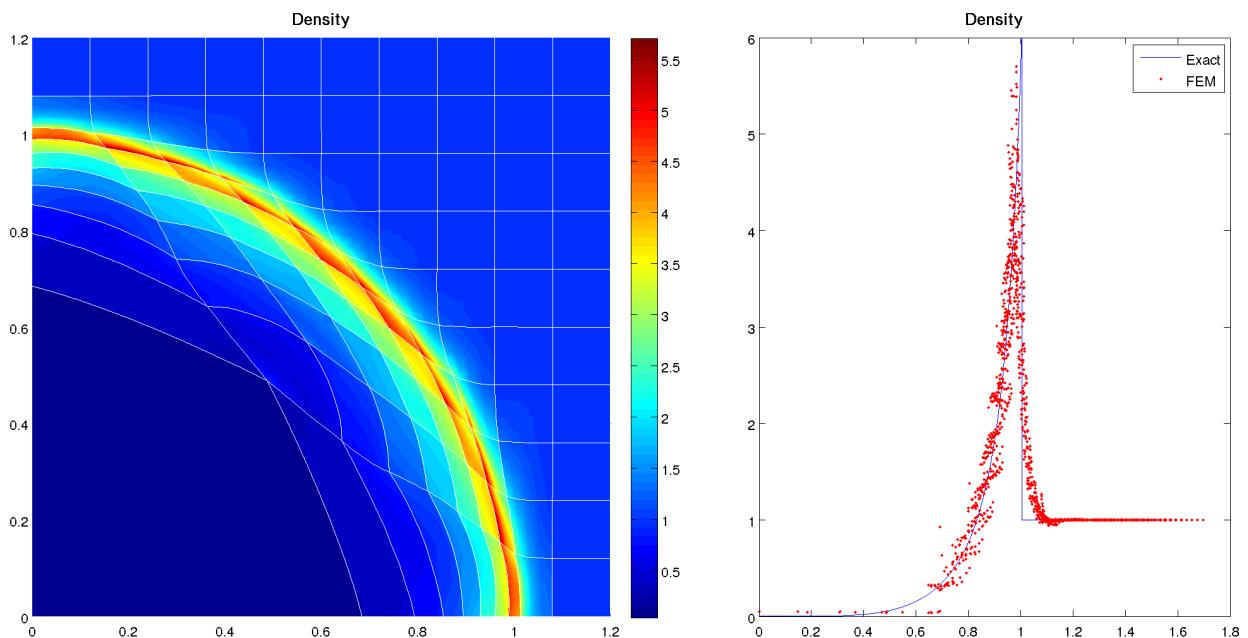
**Figure 5.30:** Comparison of  $Q_1-Q_0$  (*left*) and  $Q_2-\hat{Q}_1$  (*right*) on a full  $21 \times 21$  mesh.

#### 5.4.4 High Order $Q_2$ - $\hat{Q}_2$

$Q_2$ - $\hat{Q}_2$  does not perform quite as admirably for this problem. Without an hourglass filter we get some velocity “hot spots” at the symmetry boundaries and despite its higher order potential, the original zone is not nearly curved enough. It almost looks like one of the lower order solutions. The situation actually gets worse under refinement. The original cell and many layers beyond actually bow inward, a phenomenon that is completely non-physical. We also start to see some very bad noise in the velocity plot. This just goes to show the unpredictable nature of the  $Q_2$ - $\hat{Q}_2$  element. In some cases it appears to squelch hourglass modes, but in others spurious modes run rampant. The solution improves dramatically with an hourglass filter.

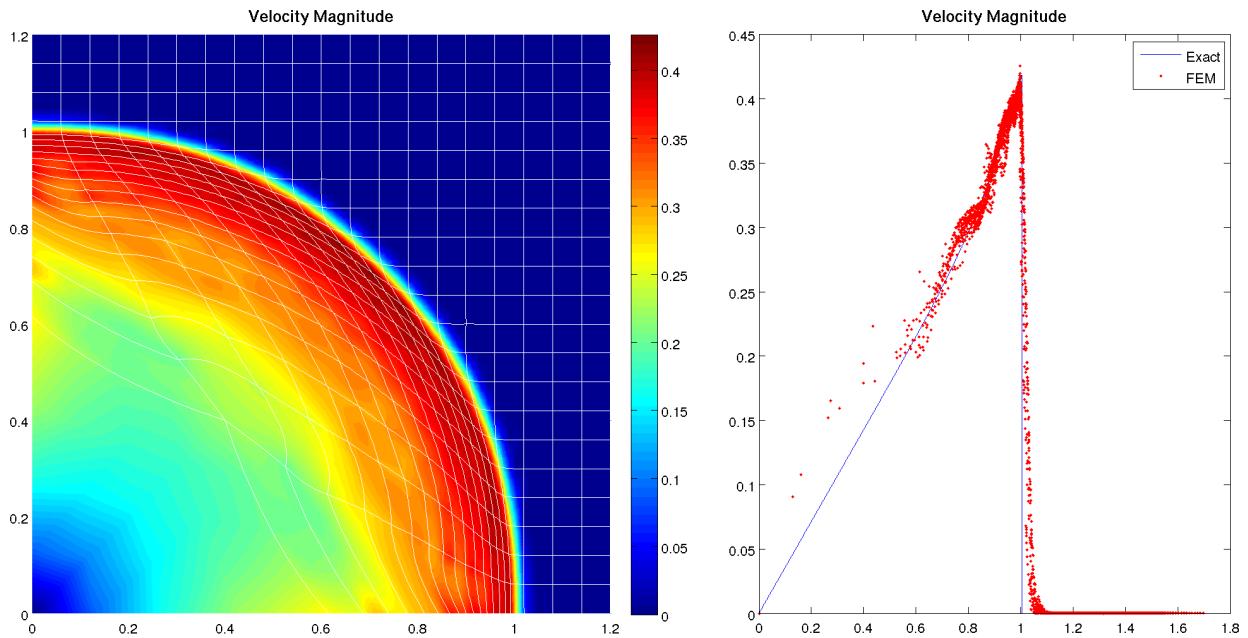


(a) Velocity magnitude pseudocolor plot (*left*). Velocity magnitude scatter plot compared to exact solution (*right*).

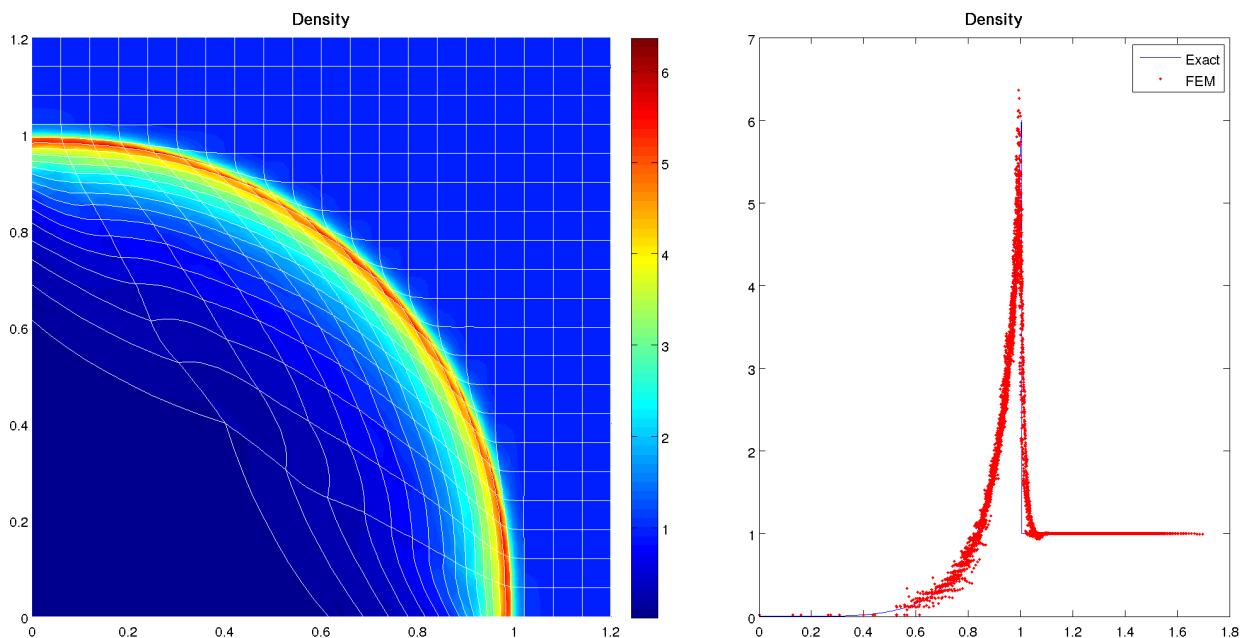


(b) Density pseudocolor plot (*left*). Density scatter plot compared to exact solution (*right*).

**Figure 5.31:** Sedov explosion problem on  $10 \times 10$  mesh using  $Q_2-\hat{Q}_2$  elements without hourglass filtering

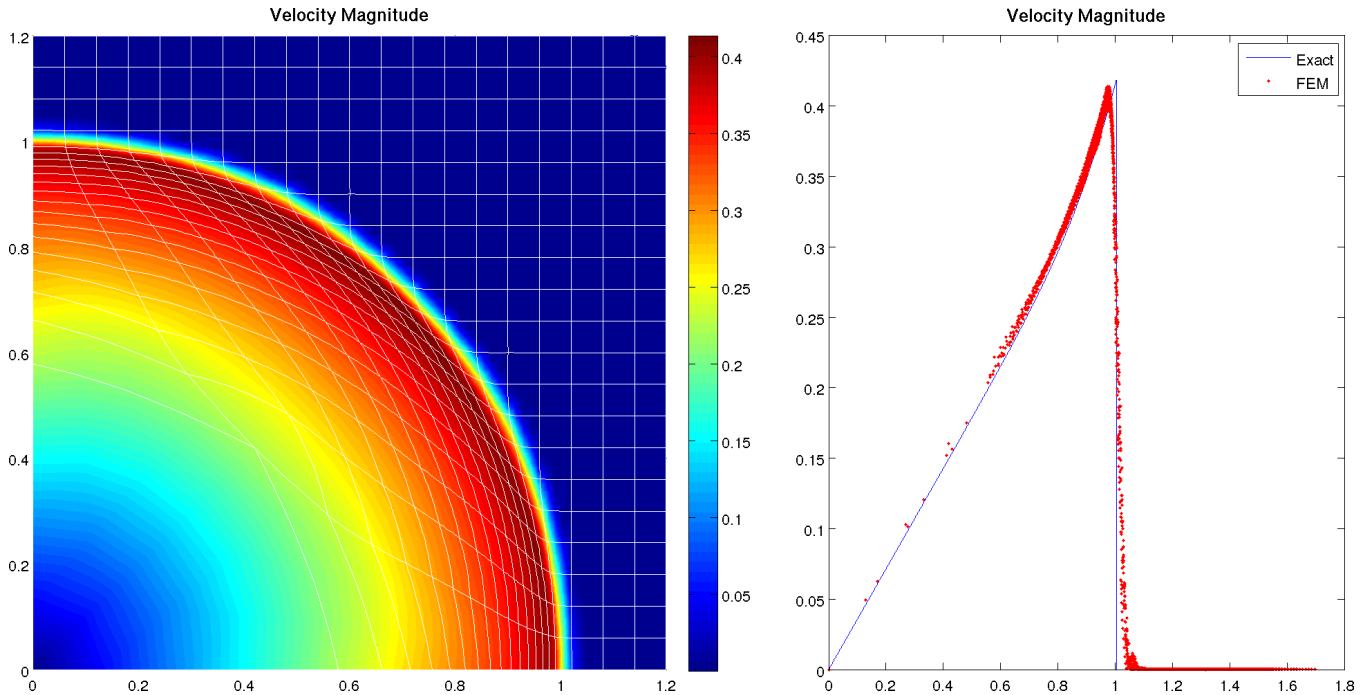


(a) Velocity magnitude pseudocolor plot (*left*). Velocity magnitude scatter plot compared to exact solution (*right*).

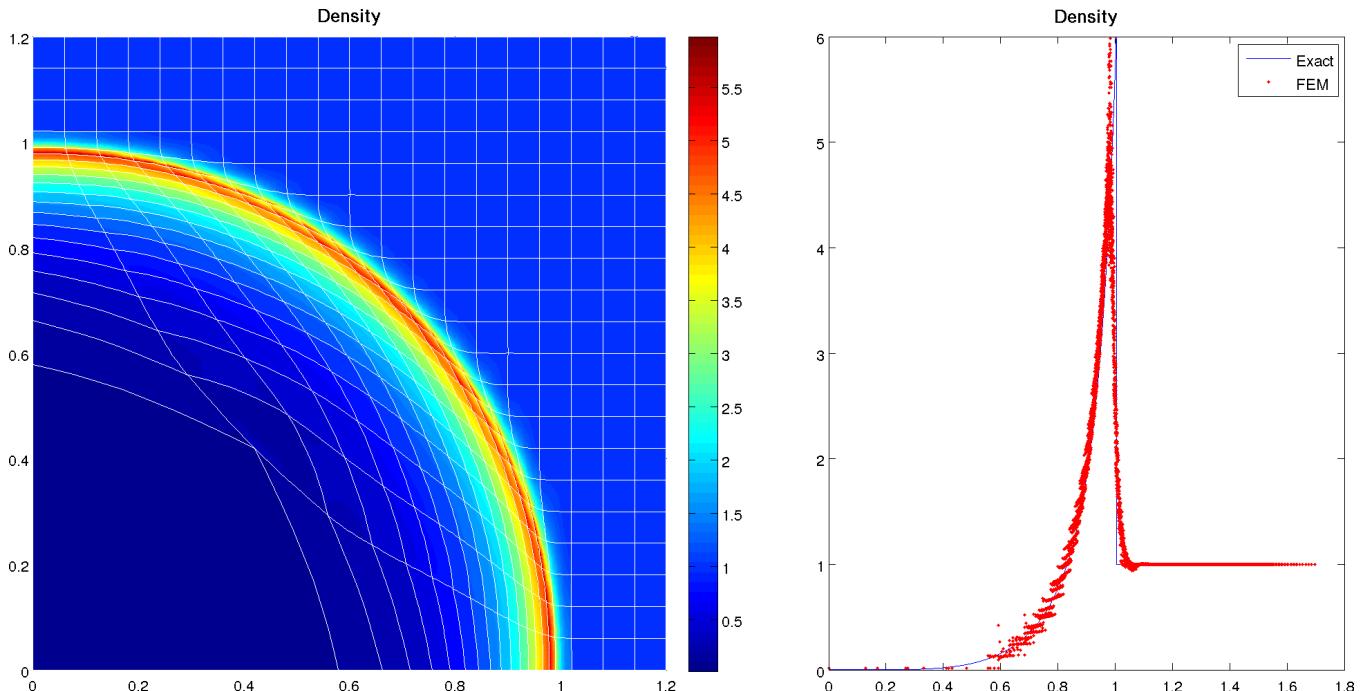


(b) Density pseudocolor plot (*left*). Density scatter plot compared to exact solution (*right*).

**Figure 5.32:** Sedov explosion problem on  $20 \times 20$  mesh using  $Q_2-\hat{Q}_2$  elements without hourglass filtering



(a) Velocity magnitude pseudocolor plot (*left*). Velocity magnitude scatter plot compared to exact solution (*right*).



(b) Density pseudocolor plot (*left*). Density scatter plot compared to exact solution (*right*).

**Figure 5.33:** Sedov explosion problem on  $20 \times 20$  mesh using  $Q_2-\hat{Q}_2$  elements with hour-glass filtering

## 5.5 Computational Efficiency

The fairest comparison of computational efficiency is to compare the runtimes of several solutions using the same number of kinematic degrees of freedom. Thus a  $5 \times 5 Q_2-\hat{Q}_1$  mesh would have the same number of kinematic degrees of freedom as a  $10 \times 10 Q_1-Q_0$  mesh. By the related nature of these two elements, the number of thermodynamic DOFs is also the same between these two meshes. High order methods become computationally attractive when we take this viewpoint. Most of the steps in the solution process proceed one-by-one through each element. If we can limit the number of zones that we loop through and distribute more work to each element, we can drastically improve the performance of the code. We ran the Sedov and Noh problems at a variety of mesh resolutions for all four methods that we have been considering and compiled the results in Table 5.1 and Table 5.2. It appears that higher order  $Q_2-\hat{Q}_1$  levels off at approximately twice as fast per degree of freedom as  $Q_1-\hat{Q}_0$ .

**Table 5.1: Sedov run times compared across the number of kinematic degrees of freedom. Time in seconds.**

Method	36 Degrees of Freedom		121 Degrees of Freedom		441 Degrees of Freedom	
	Run Time	Speedup	Run Time	Speedup	Run Time	Speedup
$Q_1-Q_0$	28.154	1.0	156.021	1.0	1265.232	1.0
$Q_1-\hat{Q}_1$	29.565	0.95	165.185	0.94	1291.628	0.98
$Q_2-\hat{Q}_1$	16.328	1.72	73.375	2.13	623.658	2.03
$Q_2-\hat{Q}_2$	17.017	1.65	77.632	2.01	663.122	1.91

When we run a profile of our code, it appears that the largest chunk of computer time is spent on the high order Jacobian calculation. This makes sense because this function is evaluated once per quadrature point per zone per timestep, and this is a somewhat complicated matrix to calculate. But from our results above, it appears that even a full bi-quadratic Jacobian matrix is not

**Table 5.2:** Noh run times compared across the number of kinematic degrees of freedom. Time in seconds.

Method	36 Degrees of Freedom		121 Degrees of Freedom		441 Degrees of Freedom	
	Run Time	Speedup	Run Time	Speedup	Run Time	Speedup
$Q_1-Q_0$	101.192	1.0	383.356	1.0	1543.319	1.0
$Q_1-\hat{Q}_1$	103.048	0.98	396.967	0.97	1590.766	0.97
$Q_2-\hat{Q}_1$	53.042	1.91	191.063	2.01	775.230	1.99
$Q_2-\hat{Q}_2$	53.605	1.89	194.381	1.97	792.363	1.95

enough to slow this method down appreciably.

When we couple the faster run times with the fact that higher order methods converge to lower errors with fewer elements, high order methods become very computationally attractive.

# Chapter 6

## Conclusion

We have developed a general finite element framework for Lagrangian computational fluid dynamics. This method is energy conserving by formulation, provides a natural extension to arbitrarily high order finite element choices, and supports a tensor artificial viscosity. We have also developed **Fermium** as a corresponding hydrocode test-bed written in Matlab to implement these ideas. **Fermium** allows for the independent choice of kinematic and thermodynamic basis functions as well as several modifiers to the standard methods. We have explored a wide variety of different mixed finite element pairs and presented four for further study:  $Q_1$ - $Q_0$ ,  $Q_1$ - $\hat{Q}_1$ ,  $Q_2$ - $\hat{Q}_1$ , and  $Q_2$ - $\hat{Q}_2$ .

The bi-quadratic kinematic methods allow for curvilinear elements that show promise for more accurately modeling curved geometries. They also allow for the straightforward calculation of second derivatives in space and gradients of the thermodynamic variables sub-zonally, which could potentially aid in sub-zonal physics in a multi-material ALE code. Higher order methods appear to show less mesh-based interference to shock passage, regardless of zonal orientation. While shock capturing is much sharper for higher order methods, the problem of oscil-

lations about the shock front remains unresolved. The method of hyperviscosity has been proposed as a promising means of filtering out these high frequency oscillations for higher order methods.

The Sod shocktube showed us that higher order methods do indeed show promise for shock hydrodynamics problems and that they will converge to the correct solution faster than traditional low order methods. We also learned that in order for a low order method to achieve even comparable accuracy to a bi-quadratic solution with  $NZ$  elements you would need  $2^{nD}NZ$  bi-linear elements, where  $nD$  is the dimensionality of the problem.

The Noh problem required a huge number of low order elements to come anywhere close to the exact solution or correct post-shock density. High order methods approached these values much faster, but suffered from significant shock ringing.  $Q_2-\hat{Q}_1$  in particular, suffered from unacceptable undershoots and overshoots while  $Q_2-\hat{Q}_2$  appeared to eliminate undershoots and magnify the less dangerous overshoots.

The Saltzman piston revealed the unpredictable behavior of hourglass modes and filters. Low order methods produced better results without a filter while the higher order methods were aided by an hourglass filter. Higher order methods showed less grid distortion and sharper, straighter shocks. Despite its reliance on an hourglass filter,  $Q_2-\hat{Q}_1$  produced the best results overall.

The Sedov explosion revealed the true strength of higher order methods - the ability to produce curvilinear edges. This becomes an indispensable property when considering curved phenomena and geometries, especially in a Lagrangian sense where an initially refined mesh may expand. Low order methods produced kinked shock waves at low resolution - a property induced by the original straight-

edged zone. High order methods produced curved zones very reminiscent of the exact Sedov mesh deformation. The full mesh test further exhibits the regidity of low order methods and the contrasting flexibility of curvilinear elements to produce almost circular zones. Additionally, a large number of low order methods were required to predict the correct maximum shock density while relatively few high order zones could do a much better job. Spurious modes appear to plague both low and high order methods, but high order to a lesser degree. Throughout each of these tests we have seen sharper shock capturing abilities bi-quadratic elements.

We have compiled information a table of run times for the Sedov and Noh problems. A good rule of thumb is that in order to obtain comparable accuracy with low order methods, twice as many must be used in one dimension, four times as many in two dimensions, and eight times for full 3-D simulations. Since many calculations are done element by element, this really starts to add up. It makes more sense to place more of the computational burden within each cell rather than use many more lower order cells. For the test problems considered, bi-quadratice elements were, on average, twice as fast as the equivalent bi-linear elements.

When all of these factors are considered together it becomes clear that higher order methods, especially  $Q_2-\hat{Q}_1$  show promise for inclusion in a full ALE hydrocode. There are still significant hurdles to overcome, but they are not impossible. The ability to have curved zones, increased accuracy, sharper shock capturing, reduced shock-caused mesh tangling, the ability to calculate thermodynamic gradients and second derivatives of velocity, and reduced computational time all call for further study of high order curvilinear finite elements for Lagrangian CFD.

# Bibliography

- [1] R. Agarwal. Computational fluid dynamics of whole-body aircraft. *Annual Review of Fluid Mechanics*, 31:125–169, 1999.
- [2] A. Barlow. A compatible finite element multi-material ALE hydrodynamics algorithm. *Internat. J. Numer. Methods Fluids*, 56:953–964, 2007.
- [3] M. Berger and M. Aftosmis. Analysis of slope limiters on irregular grids. Technical Report 2005-0490, American Institute of Aeronautics and Astronautics.
- [4] D. E. Burton. Consistent finite-volume discretization of hydrodynamic conservation laws for unstructured grids. In *Eighth Nuclear Explosives Code Developers' Conference*, 1994.
- [5] J. C. Campbell and M. J. Shashkov. A tensor artificial viscosity using a mimetic finite difference algorithm. *J. Comput. Phys.*, 172:739–765, 2001.
- [6] E. J. Caramana, D. E. Burton, M. J. Shashkov, and P. P. Whalen. The construction of compatible hydrodynamics algorithms utilizing conservation of total energy. *J. Comput. Phys.*, 146:227–262, 1998.
- [7] E. J. Caramana and M. J. Shashkov. Elimination of artificial grid distortion

- and hourglass-type motions by means of Lagrangian subzonal masses and pressures. *J. Comput. Phys.*, 142(2):521–561, 1998.
- [8] A. W. Cook and W. H. Cabot. A high-wavenumber viscosity for high-resolution numerical methods. *Journal of Computational Physics*, 195:594 – 601, 2004.
- [9] A. W. Cook and W. H. Cabot. Hyperviscosity for shock-turbulence interactions. *Journal of Computational Physics*, 203:379 – 385, 2005.
- [10] R. Courant, K. O. Friedrichs, and H. Lewy. Uber die partiellen differenzen-gleichungen der mathematisches physik. *Math. Ann.*, 100:32–74, 1928.
- [11] M. Dalenbring and J. Smith. Simulation of s-duct dynamics using fluid-structure coupled cfd. In *24th Applied Aerodynamics Conference*, 2006.
- [12] T. Dixon, A. Mann, P. Hobson, S. Plaza, F. andPennisiI, and R. Steindl. Application of cfd in the sugar industry. In *Third International Conference on CFD in the Minerals and Process Industries*, 2003.
- [13] J. Donea, S. Giuliani, and J. Halleux. An arbitrary lagrangian-eulerian finite element method for transient dynamic fluid-structure interactions. *Computer Methods in Applied Mechanics and Engineering*, 33(1-3):689 – 723, 1982.
- [14] J. Donea, A. Huerta, J.-P. Ponthot, and A. Rodriguez-Ferran. Arbitrary lagrangianeulerian methods. In *Encyclopedia of Computational Mechanics*. John Wiley & Sons, Ltd., 2004.
- [15] J. Douglas and F. Milner. Interior and superconvergence estimates for mixed methods for second order elliptic equations. *RAIRO Modl. Math. Anal. Numr.*, 19:297328, 1985.

- [16] J. W. Gibbs. Fourier series. *Nature*, 59(200), 1898.
- [17] N. Gnedin. Softened lagrangian hydrodynamics for cosmology. *Astrophysical Journal Supplement Series*, 97(2):231–257, 1995.
- [18] V. Gyrya and K. Lipnikov. Vorticity errors in multidimensional lagrangian codes. *J. Comp. Phys.*, 99:115134, 1992.
- [19] C. W. Hirt, J. L. Cook, and T. D. Butler. A lagrangian method for calculating the dynamics of an incompressible fluid with free surface. *Von Karman Inst. for Fluid Dyn. Numerical Methods in Fluid Dyn.*, 15:18–34, 1972.
- [20] T. Hughes. *The Finite Element Method: Linear Static and Dynamic Finite Element Analysis*. Dover Publications, 1987.
- [21] R. Kandasamy and S. Subramanyam. Application of computational fluid dynamics simulation tools for thermal characterization of electronic packages. *International Journal for Numerical Methods in Heat & Fluid Flow*, 15(1):61–72, 2005.
- [22] K. Lipnikov and M. Shashkov. A mimetic tensor artificial viscosity for lagrangian hydrocodes on arbitrary polygonal meshes. Technical Report LA-UR-10-00006, Los Alamos National Laboratory, 2010.
- [23] T. V. Kolev and R. N. Rieben. A tensor artificial viscosity using a finite element approach. *J. Comput. Phys.*, 228(22):8336–8366, 2009.
- [24] H. U. Mair and H. Huang. Lagrangian hydrocode modeling of underwater explosive/target interaction. In *61s Shock and Vibration Symposium*, 1990.
- [25] M. A. Martnez, E. Cueto, I. Alfaro, M. Doblar, and F. Chinesta. Updated lagrangian free surface flow simulations with natural neighbour galerkin meth-

- ods. *International Journal for Numerical Methods in Engineering*, 60:2105–2129, 2004.
- [26] W. F. Noh. Errors for calculations of strong shocks using an artificial viscosity and an artificial heat flux. *J. Comput. Phys.*, 72(1):78–120, 1987.
- [27] R. Perry. Investigation of the proposed 2009 fiaformula 1 world championship aerodynamic regulation changes using computational fluid dynamics. Master’s thesis, California Polytechnic State University, June 2008.
- [28] W. J. Rider. Revisiting wall heating. *Journal of Computational Physics*, 162:395 – 410, 2000.
- [29] Z. rong Zhang, H. Liu, S. ping Zhu, and F. Zhao. Application of cfd in ship engineering design practice and ship hydrodynamics. *Journal of Hydrodynamics, Ser. B*, 18(3, Supplement 1):315 – 322, 2006. Proceedings of the Conference of Global Chinese Scholars on Hydrodynamics.
- [30] J. Saltzman and P. Colella. Second order corner coupled upwind transport methods for lagrangian hydrodynamics. Technical Report LA-UR-85-678, Los Alamos National Laboratory, 1985.
- [31] L. I. Sedov. *Similarity and Dimensional Methods in Mechanics*. CRC Press, tenth edition, 1993.
- [32] G. Sod. Survey of several finite difference methods for systems of nonlinear hyperbolic conservation laws. *Journal of Computational Physics*, 26, 1978.
- [33] R. Stenberg and M. Suri. Mixed hp finite element methods for problems in elasticity and stokes flow. *Numerische Mathematik*, 72:367–389, 1996.

- [34] J. C. Tannehill, D. A. Anderson, and R. H. Pletcher. *Computational Fluid Mechanics and Heat Transfer*. series in computational and physical processes in mechanics and thermal sciences. Taylor & Francis, 2nd edition, 1997.
- [35] R. Tipton. CALE Lagrange step. Technical report, Lawrence Livermore National Laboratory, October 1990. Unpublished.
- [36] E. F. Toro. *Riemann Solvers and Numerical Methods for Fluid Dynamics - A Practical Introduction*. Springer - Verlag, 2006.
- [37] J. Zukas. *Introduction to Hydrocodes*. Number 49 in Studies in Applied Mechanics. Elsevier, 2004.