# Exploration of a Superposition and Reconciliation Based Approach to Cell-Centered Lagrangian Hydrodynamic Methods

By
Lindsey Anne Gilman

B.S., Chemistry
Valparaiso University, 2010

SUBMITTED TO THE DEPARTMENT OF NUCLEAR SCIENCE AND ENGINEERING IN PARTIAL FULFILLMENT OF THE REQUIREMENTS FOR THE DEGREES OF

MASTER OF SCIENCE IN NUCLEAR SCIENCE AND ENGINEERING

AT THE

MASSACHUSETTS INSTITUTE OF TECHNOLOGY

May 11, 2012

Signature of Author:______________________________
Department of Nuclear Science and Engineering
May 11, 2012

Certified by:______________________________
Linn Hobbs
Professor of Nuclear Science and Engineering; and Materials Science and Engineering
Thesis Supervisor

Certified by:______________________________
Kord Smith
Korea Electric Power Company Professor of the Practice of Nuclear Science and Engineering
Thesis Supervisor

Certified by:______________________________
Scott Runnels
Computational Physics Division Scientist IV, Los Alamos National Lab
Thesis Supervisor

Accepted by:______________________________
Mujid Kazimi
TEPCO Professor of Nuclear Engineering
Chair, Department Committee on Graduate Students

# Exploration of a Superposition and Reconciliation Based Approach to Cell-Centered Lagrangian Hydrodynamic Methods

By
Lindsey Anne Gilman

Submitted to the Department of Nuclear Science and Engineering on May 11, 2012 in partial fulfillment for the requirements of the degrees of

MASTER OF SCIENCE
in NUCLEAR SCIENCE AND ENGINEERING

## ABSTRACT

Applications and experiments involving the hypervelocity deformation of solids are difficult to devise, implement, and occur on microsecond time scales. As a result, simulations play a large role in the study of hypervelocity deformation. This study explored a superposition and reconciliation based approach using cell-centered Lagrangian hydro methods. The reconciliation forces that are not explicitly calculated for mesh movement were analyzed on an existing hydrocode by Pierre-Henri Maire (PHM) and a truncated form of the Runnels-Gilman method (implemented without using the reconciliation forces as additional forces to form a new hydro method called the Runnels-Gilman method). Results from both the 1D Piston and Saltzman test problems illustrate that the unaccounted reconciliation forces are acting on the mesh both at the shock front and behind the shock wave in PHM's method, while in the truncated Runnels-Gilman method, reconciliation forces are acting only on the vertices at the shock front. In test problems using PHM's method, reconciliation forces may be capturing the additional forces that account for more stable density and internal energy solution during shock wave propagation as compared to the truncated Runnels-Gilman method.

**THESIS SUPERVISOR:** Linn W. Hobbs, D.Phil
**TITLE** Professor of Nuclear Science and Engineering; and Materials Science and Engineering

**THESIS SUPERVISOR:** Kord Smith, Ph.D.
**TITLE:** Korea Electric Power Company (KEPCO) Professor of the Practice of Nuclear Science and Engineering

**THESIS SUPERVISOR:** Scott Runnels, Ph.D.
**TITLE:** University Liaison for LANL's Advanced Scientific Computing Program/Scientist IV

# ACKNOWLEDGEMENTS

I would first like to thank my thesis supervisor, Professor Linn Hobbs, for all of his support in my first two years at MIT. I would also like to thank Professor Kord Smith for his willingness to help with the details of my code development. Additionally, I express thanks to both Prof. Hobbs and Prof. Smith for their patience with my project. I would also like to express my gratitude to Los Alamos National Lab, and in particular their LDRD-RD Cell-Centered Lagrange research project, that instigated the work by Dr. Scott Runnels into the development of a superposition and reconciliation based approach to cell-centered Lagrangian hydro methods.

I would like to extend my personal gratitude to Dr. Scott Runnels, who spent countless hours patiently teaching me how to code in C++, the background and development of hydrocodes, and brainstorming with me to devise a method to solve for the reconciliation forces. Further, I would like to thank Scott for all of his academic, research, and professional advice and guidance since the start of my work on this research project. Finally, I would like to thank the entire Runnels family- Joy, Brandon, Joel, Rebecca, and Catherine for being such great friends during my stay in Los Alamos over the summer.

My friends have also been vital in providing me with the support I needed to succeed. In particular I would like to thank my officemate and roommate Brittany Guyer, and also Bren Phillips who provided endless support and patience with me during this past year; along with my office neighbors Koroush Shirvan, Giancarlo Lenci, and Rosie Sugrue. Also, I would like to thank the MIT Women's Volleyball Club members who I immensely enjoy playing volleyball with. Finally, I would also like to acknowledge my very close friends from Valpo, Alyssa and David, who have been a constant source of support!

My family has always backed me up in my pursuits and dreams, and for that I want to express my deepest gratitude. None of my successes would have been possible without their help and belief in my abilities. I especially would like to thank my Mom, Dad, sister (Rachel) and brother (Chris). Also I want to thank my grandparents, Dick and Mavis, who moved me to MIT all the way from South Bend, Indiana and also helped me in my undergraduate education. Finally, I express thanks to my Aunt Linda who gave me great advice in preparing for and choosing a graduate school.

All figures and content are the author's original work, unless otherwise noted. Proper acknowledgment is provided for items not generated by the author.

## Table of Contents

**List of Figures**

## List of Tables

# 1 Introduction

## 1.1 Motivation

New and improved hypervelocity large deformation simulation techniques are needed to provide robust and numerically accurate schemes to investigate a variety of applications such as the multimaterial compressible fluid flow for Inertial Confinement Fusion (ICF) [1], [2]. Also, in the hypervelocity deformation of a solid, the current stress state of the material is dependent on the time history of its strain, strain rates, and damage. As a result, the solid's history can play a very large role in the calculation, and the history of each point in a solid can vary significantly when under extreme loading conditions. For this reason, numerical methods must be used that maintain the identity of each point in the material, and Lagrangian methods are the leading method for these simulations. Simulation techniques used to solve hypervelocity large deformation and finite strain problems are referred to as hydrocodes and were originally developed for the modeling of defense problems, but have since expanded. Hydrocodes are also of interest to the fission nuclear field because of their ability to model void growth in ductile materials and to analyze high strain-rate problems [1].

Lagrangian methods, which allow the grid to move with the material, are of particular interest because, as mentioned above, these methods allow cells to retain information during the simulation by tracking the solid material in a single cell and because of their ability to handle sharp interfaces that are colliding, separating, or sliding. One of the computational and numerical difficulties in cell-centered Lagrangian hydro methods involves the motion of the vertices of the mesh, also referred to as nodes. Cell-centered Lagrangian scheme inherently treats cell interfaces, but the movement of the nodes to maintain material continuity is not achieved without additional constraints.

Lagrangian cell-centered approaches integrate all conservation laws on each cell. The movement of cells is computed by the flux across the cell boundary, which is often computed with a one-dimensional Riemann solver that provides information in the normal direction to the boundary [3]. The obstacle encountered with this method is how to determine movements of the nodes in addition to cell boundaries to maintain connectivity of the mesh. One method of maintaining mesh integrity is to compare the Riemann solver velocity to the vertex velocity and minimizing errors between these. This approach maintains grid continuity, but is an expensive computational treatment [4].

Another approach to overcome this mesh integrity complication is to use a nodal solver where the nodal velocity is simultaneously computed with the interface fluxes in an entropy-consistent Lagrangian scheme, as proposed by Pierre Henri Maire (PHM) [5]. This method has also been improved to preserve spherical symmetry in a two-dimensional cylindrical geometry by also utilizing an area-weighted scheme [6]. Although these cell-centered methods provide a robust approach, the motivation for this thesis is to develop a more thorough understanding of the forces acting on cells during the nodal solver step and their ultimate impact on hypervelocity large deformation numerical simulations.

## 1.2 Objective

Lagrangian Godunov cell-centered hydro methods often use a one-dimensional Riemann solution at cell boundaries to provide an estimate of the cell boundaries' velocity and pressure at each time step as mentioned briefly in section 1.1 [7]. However, since they are one-dimensional in nature, the Riemann velocities and pressures are incomplete because they do not account for additional forces that hold the mesh together in two-dimensional situations, which are a focus of this investigation.

Computational methods deal with the problem of mesh continuity in different ways; this step is usually called the nodal solver. In particular, two methods will be explored here. A recently published method by PHM [5], uses a weighted least squares fit to the vertices surrounding Riemann face velocities to determine the vertex velocity. Thus, the linear momentum of cell corners are adjusted from the one-dimensional Riemann solution. PHM's method can also be explained as using a combination of entropy conditions with momentum conservation around a massless volume. In a method proposed by Runnels and Gilman [8], the vertex velocity is computed by enforcing conservation of linear momentum surrounding the vertex where each corner's linear momentum must be adjusted, but the sum remains unchanged.

In both approaches, pressures in addition to the one-dimensional Riemann solution's pressure are at work. PHM's method solves for the sum of the pressures concurrently, justifying the solution method as one that satisfies an entropy condition and by reconciling velocities on faces provided by the Riemann Solver through a weighted least squares fit at the vertices. Runnels and Gilman are proposing to solve for the Riemann and additional pressures in a separate step using a momentum conservation argument.

The objective of this research project is to understand the difference between the one-dimensional Riemann forces at cell interfaces and the additional forces computed by PHM. The forces that supplement the Riemann forces are referred to as "reconciliation forces" throughout this investigation because they are responsible for reconciling nodal motions to maintain mesh continuity. To improve the understanding of how the nodal solver affects the conservation laws, a method of analysis is established to explicitly solve for reconciliation forces based on an argument of linear momentum conservation on the staggered cell. This method also suggests a fundamentally new hydro method, which here will be referred to as the Runnels-Gilman method, that is based on the idea of superposition reconciliation forces with Riemann forces.

The new method proposed by Runnels and Gilman is further developed and tested using well-known verification problems such as the piston, Noh [9], and Saltzman [10] problems. The fully developed method can then be used to solve the fluid dynamic Equations for compressible flow. Examples for application of this scheme include simulations where the history and specific material properties are important and necessary [11].

# 2 Hydrocode Background Information

## 2.1 Hydrocode History

A code used to solve hypervelocity large deformation problems on materials is often referred to as a hydrocode. Hydrocodes were originally developed for modeling of defense problems, and were emphasized in the Stockpile Stewardship and Management Program initiated by the United States Department of Energy. The purpose of this program was to ensure that the reliability of the nuclear weapons stockpile of the United States is maintained without nuclear testing, and thus simulations were studied to fill this gap of knowledge.

Current applications for hydrocodes now extend into a variety of fields of study. In astrophysics, hydrocodes can be used to simulate the collapse of stars or the impact of a large meteorite with earth. In materials studies, particularly in the nuclear field, hydrocodes are used in modeling void growth in ductile materials, the analysis of high strain rate experiments [1], the multimaterial compressible fluid flow and for Inertial Confinement Fusion (ICF) [12], [2].

## 2.2 Lagrangian versus Eulerian Methods

The modeling of large deformation in solid mechanics generally employs two techniques using finite element and analysis: Lagrangian or Eulerian. The main difference between the Eulerian and Lagrangian methods can be illustrated by how the material flows as compared to the mesh. In Eulerian codes, the mesh is held immobile while the material flows through it. This method is preferred for fluid flows that would highly contort the mesh if grids were required to follow fluid motion through the calculation [13]. In contrast, Lagrangian methods allow the grid to flow with the fluid. This means that in each cell of the grid, also called a zone, no mass crosses the boundaries; rather the zone's movement is tracked over time which is important to accurately capture results when investigating solid materials.

In the investigation of hypervelocity problems, where a disturbance in a material moves faster than the propagation of information through the material, stress waves and pressure shocks are an important part of the solution. Traditionally, explicit time marching schemes are used for shock simulations because they limit the propagation of information to one zone per time step to capture the correct physics and provide an accurate solution. The Riemann solution on cell interfaces in cell-centered hydrocodes provides information for the shock physics [14]. An alternate method for including the shock physics is to use a flux limiter [1]. The flux-corrected transport has been further analyzed to understand the diffusive errors introduced by the flux-corrected process so that it can be applied in general fluid systems, multidimensional problems, and curvilinear geometry [15].

Since the Lagrangian method allows the mesh to flow with the fluid, it is typically preferred when the fluid exhibits only small motion. When using the Lagrangian method, the grid can prematurely break down from a high degree of distortion, causing an early simulation termination or the production of inaccurate results. In some cases, a mesh remap can be performed using the method of arbitrary Lagrangian Eulerian (ALE) [16], [17] to remove the grid distortion and continue the simulation [1].

Since the original implementation of ALE in the 1970s [16], more robust and accurate techniques for this method have been explored. One such method for executing ALE uses a mesh remapper that can handle polygonal shaped cells. It further discretizes each cell and conservatively gathers the momentum, internal energy, and kinetic energy into the discretized subcells. The subcells are then remapped onto the new subcells of a reformatted Lagrangian mesh. Density, nodal velocity, and cell-centered specific internal energy values are transferred to the new mesh. This method was shown to remove unphysical distortion that can occur during simulations [18]. Another method that was first developed for one-dimension can be used on unstructured grids. It uses a finite volume approach that also further divides the cell into smaller control volumes [19]. This allows for high-order reconstruction of the original cell volume while being able to remove or reduce spurious vorticity. This method was further developed for both two- [20] and three-dimensional systems [21].

As will be discussed in more detail in section 2.3, grid tangling caused by vorticity can form in non-uniform meshes, including the distortion that occurs in one-dimensional non-rotational flows. Instead of introducing the more commonly used Neumann-Richtmyer model which can be generalized to two and three-dimensions [22], or a method developed using a discrete viscosity tensor [23], the vorticity can be removed, or filtered out, using one of two methods. In the first method, the known flow divergence and the true vorticity computed by the transport equation are used to reconstruct the velocity. The second, more efficient method, calculates a new velocity that captures the correct vorticity by subtracting a divergence-free correction from the velocity. Both of these two methods have been successfully applied in a two-dimensional shock refraction problem [24].

Another recent study on the formulation of Lagrangian hydrodynamics investigates the definition of the zonal volume and specifically, how this definition can affect total energy conservation. The most common method to define a zonal volume is by utilizing a function based on the coordinate points that define the cell. A proposed method that exactly conserves total energy uses a time integration of the continuity equation for the volume of the zone. An investigation of these two definitions showed that there is a slight difference in these two zonal definitions that can be described as a type of entropy error. It was further demonstrated that this entropy error is insignificant unless the calculation becomes numerically unstable. Therefore, this new zonal definition could be used to indicate when a calculation encounters numerical difficulties, since this can occur even while the local conservation of total energy still holds through the calculation [25].

Lagrangian methods are of interest in the nuclear field for the information they can retain during the simulation, as described in Section 1.1. For example, a Lagrangian hydrocode can handle the sharp interfaces of the cells (the cell boundaries that hit each other and rub against each other) so that contact discontinuity can be sharply captured in multi-material flows [26]. Additionally, materials can retain their history since damage and plastic deformation information can be retained within each cell throughout the analysis. This can further effect how the material responds to stresses. This information is important in the study of nuclear materials so material failure can be predicted using the history of the material.

## 2.3 Cell-centered versus Staggered Grid

Two methods for grid movement are typically used in hydrodynamic codes: cell-centered or staggered. The staggered grid discretizes the mesh to specify the position, velocity, and kinetic energy to be associated and centered at the vertices of the mesh. The density, pressure, and specific internal energy are defined and calculated within each cell [26]. It was first introduced for one-dimensional flows [12] and then adapted to two-dimensional problems [27]. Although this method has drawbacks, it has been widely used in the last 40 years [28] with many improvements implemented in the last ten years. Examples include a unified coordinate system to prevent premature grid breakdown [29], the Lagrangian Discontinuous Galerkin-type method [30], and as will be discussed below, various artificial viscosity treatments and anti-hourglass methods.

A potential undesirable mode of unphysical grid distortion that can occur with a staggered grid is referred to as “hourglassing” when it is evident in short spatial grid scales (shown in Figure 2.1) and “spurious vorticity” when visible in longer lengths compared to the scale of the mesh [31]. Vorticity and hourglassing are not caused by the characteristics of the flow in the simulation, but rather from numerical error or an underconstrained grid respectively. This can occur with the use of quadrilateral or hexahedral grids in two or three-dimensions. Common methods to combat this problem use subzonal masses, forces [32], pressures, and densities to prevent unphysical grid movements by providing forces that resist the spurious motion. In particular, the use of a pressure gradient though the cell is effective for dampening the unphysical movement [33].

The introduction of artificial viscosity has also led to more robust staggered Lagrangian schemes [5], [34] and is vital for capturing the correct solutions. This artificial viscosity must be implemented carefully because it can cause high numerical errors in strong shock calculations [35].

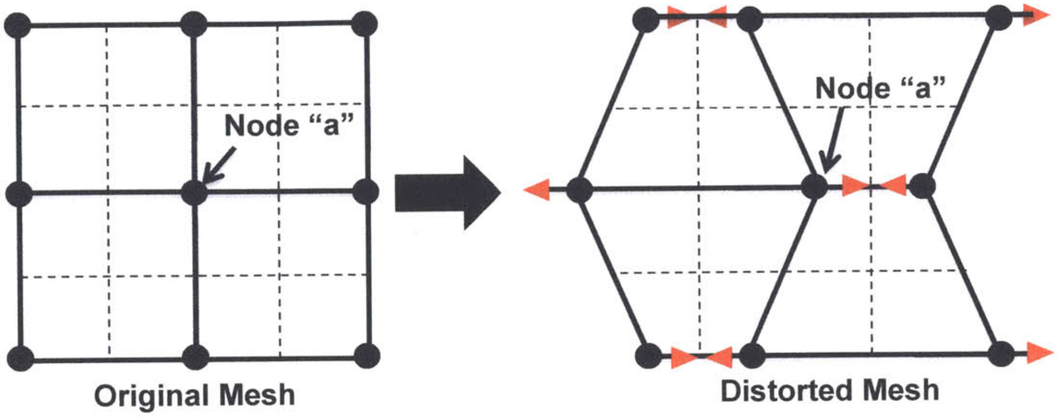


**Figure 2.1 Hourglass motion in two dimensions where the velocity field direction alternates at every node (shown for arbitrary node "*a*" in a mesh). The arrows indicate the direction of the velocity field, the solid lines the cell boundaries, the dotted lines the proposed subzonal boundaries, and the solid circles the nodes of the mesh.**

In the Godonuv cell-centered method, all conserved quantities are evolved at the cell center. The original mesh structure is also illustrated by the sketch on the left of Figure 2.1. The approach in this investigation solves the Reimman problem to determine the face velocities of the cells, and this is still the popular approach for Lagrangian scheme development [36], [37]. The remaining quantity required to move the mesh is the vertex velocity, which must be completed while upholding the geometric conservation law.

In an effort to maintain flux discretization compatibility with vertex movement, the nodal motion can be coherently solved with the fluxes at the faces using a semi-discrete entropy requirement and this has been shown to be a numerically stable scheme [38] even in two-dimensional flow on an unstructured meshes [39]. In general, this entropy requirement does not explicitly mandate the conservation laws to be upheld when the nodes are moved to maintain mesh continuity. This research studies the cell-centered method, in particular the process covered by the nodal solver. It also seeks to determine if the semi-discrete entropy requirement scheme by PHM [5] upholds conservation of linear momentum over the mass in the cell as the mesh is propagated in time.

# 3 Runnels-Gilman Code Development

## 3.1 General Approach

The Runnels-Gilman method began development under Dr. Scott Runnels as part of the LDRD-DR (Laboratory Directed Research and Development) Cell-Centered Lagrange research project at Los Alamos National Laboratory (LANL). It was coded into what is referred to as the "DiscoverHydro code" in C++. It was further developed during the 2011 XCP Computational Physics Student Summer where, as a visiting student, the author became familiar with the existing code, and began contributing to the development of the reconciliation forces solver and iteration scheme for implementing reconciliation forces that provide information for moving the mesh.

The proposed reconciliation and superposition based cell-centered method involves breaking each time step into three phenomena which are solved for consistently to determine how each cell progresses with the mesh. These three parts include (1) the forces on each cell that cause a change in internal energy and linear momentum, giving it a trajectory in space, (2) the deformation of the cell that occurs from these forces, and (3) the collective work of the aforementioned processes (cell forces, deformation and trajectory) to keep the mesh connected [8].

## 3.2 Governing Equations

Although the goal here is to model solids, gas dynamics equations are the starting point to serve as a test-bed for the development of hydrocodes. The multidimensional gas dynamics equations are shown in Equations 3.1, 3.2, and 3.3.

$$\rho \frac{d}{dt}\left(\frac{1}{\rho}\right) - \nabla \cdot \boldsymbol{u} = 0 \qquad 3.1$$

$$\rho\frac{d}{dt}(\boldsymbol{u}) - \nabla P = 0 \qquad 3.2$$

$$\rho\frac{d}{dt}(E) - \nabla\cdot(P\boldsymbol{u}) = 0 \qquad 3.3$$

In these gas dynamics equations, ρ represents the density, $\boldsymbol{u}$ the velocity, $E$ the specific total energy, and $\frac{d}{dt}$ the time derivative of the material. Equation 3.1 is the volume conservation Equation, or GCL. The momentum and total energy conservation equations are represented by Equations 3.2 and 3.3 respectively. For thermodynamic closure, the equation of state is used where

$$P = P(\rho, \varepsilon) \qquad 3.4$$

In Equation 3.4, $\varepsilon$ represents the specific internal energy,

$$\varepsilon = E - \frac{\boldsymbol{u}^2}{2} \qquad 3.5$$

## 3.3 Grid Construction

In every cell, each of the 4 corners is divided into two "iota," producing 8 iota in each cell and 8 iota surround each interior node (see Figure 3.1). The velocities on the three corners of the iota are interpolated from the cell's velocity and the facial velocities. The cell's velocity is modeled at the cell-center and this is also the velocity of one point of each iota. The point on the node (vertex) is the vertex velocity, and the point on the center of the face is the average of the two vertices connected to that face.

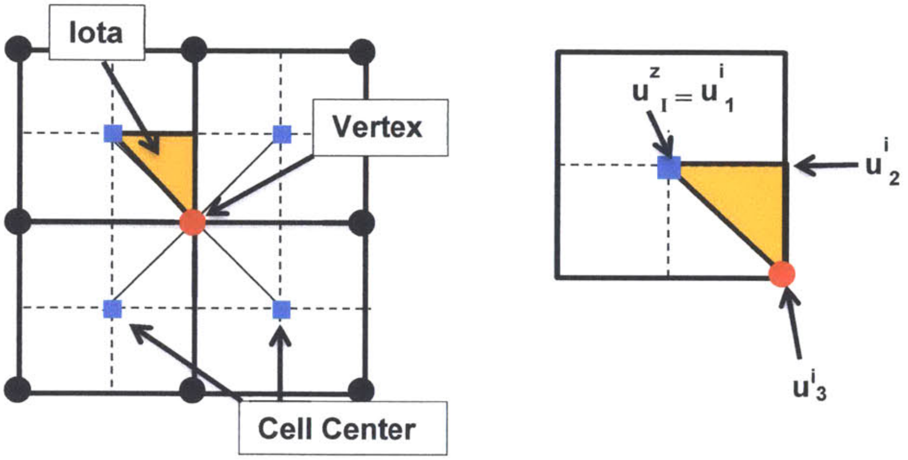


**Figure 3.1 The grid construct of the mesh showing a sample cell with the 3 point velocities of "*iota i.*"**

Using linear Lagrange polynomials, denoted by $\phi^i_j$, these velocities are interpolated to give an iota velocity, as shown in Equation 3.6. Using the three iota corner velocities, the iota linear momentum and kinetic energy can be calculated, and the cellular (zonal) linear momentum is computed by using a vector sum of the values for iota contained in the cell.

$$\boldsymbol{u}^i\,(x,y) = \sum_{j=1}^{3} \boldsymbol{u}^i_j\,\phi^i_j(x,y) \qquad 3.6$$

## 3.4 Scheme Development

### 3.4.1 Conservation Laws on a Cell

To maintain consistency with the physical conservation laws, the rate of change of the cell volume must be coherently computed with the motion of the nodes to satisfy the necessary geometric conservation law (GCL), often referred to as the volume conservation equation [26]. For the space discretization, the mass in each cell is described by discrete variables: specific volume, velocity, and specific total energy which are denoted by $\tau_i, \boldsymbol{V}_i,$ and $E_i$ respectively. This allows a set of discrete equations to be formulated using an integral over the cell as shown in Equations 3.7, 3.8, and 3.9. In these equations, $m_i$ denotes the mass in the cell defined by the domain $\Omega_i$, and $P$ is the pressure given by the equation of state [5].

$$m_i \frac{d}{dt}(\tau_i) - \int_{d\Omega_i} \boldsymbol{V} \cdot \boldsymbol{N}\, dl = 0 \tag{3.7}$$

$$m_i \frac{d}{dt}(\boldsymbol{V_i}) + \int_{d\Omega_i} P \cdot \boldsymbol{N}\, dl = 0 \tag{3.8}$$

$$m_i \frac{d}{dt}(E_i) + \int_{d\Omega_i} P\boldsymbol{V} \cdot \boldsymbol{N}\, dl = 0 \tag{3.9}$$

### 3.4.2 Riemann Solver

This superposition and reconciliation based method calculates intercellular forces in part from the Riemann solver, as shown in Figure 3.2. The calculation provides an initial facial trajectory for each cell based on the pressure and velocity differences of neighboring cells shared faces. The Riemann solver only provides a velocity in the direction parallel to its own normal, and if used directly without modification, the mesh would become disconnected at vertices, often referred to as nodes. A correction is used to reconcile this discontinuity and maintain integrity of the mesh. In order to achieve this mesh connectivity, the cell may distort or twist to reconnect nodes, resulting in additional forces that are needed to uphold conservation laws. These forces are determined in the Runnels-Gilman method through an iteration scheme developed to conserve linear momentum around each node within each cell [8].

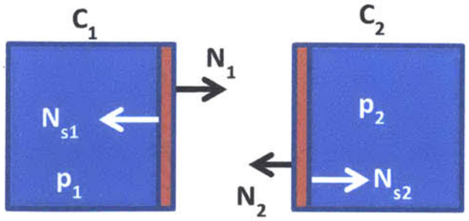


**Figure 3.2 A shared face (in red) between two cells in the mesh with the corresponding normal forces [8].**

For the simple Riemann Solver method, refer to Figure 3.2 above. The cells, $C_1$ and $C_2$, share a face and contain area-weighted normals in the outward direction (denoted by the capitalized "N"). Each cell contains a corresponding pressure, $p_1$ and $p_2$, and impedance, $\mu_1$ and $\mu_2$. By applying the conservation of momentum on $C_1$ in the area surrounding the shared face (shown in red) Equation 3.10 is obtained ($a$ represents the acceleration) [40].

$$\begin{aligned} -p_1\boldsymbol{N}_{s1} - p_f\boldsymbol{N}_1 &= m\dot{\boldsymbol{u}} \\ &= A_1\rho a\Delta t\dot{\boldsymbol{u}} \\ &= \mu_1(\boldsymbol{u}_f - \boldsymbol{u}_1)A_1 \end{aligned} \qquad 3.10$$

By dividing both sides by the area, $A_1$, the area-weighted normals become simple unit normal vectors (denoted by the lowercase "$n$") and shown in Equation 3.11.

$$-p_1\boldsymbol{n}_{s1} - p_f\boldsymbol{n}_1 = \mu_1(\boldsymbol{u}_f - \boldsymbol{u}_1) \qquad 3.11$$

Likewise, a similar Equation (3.12) is derived for cell 2 and shown below.

$$-p_2\boldsymbol{n}_{s2} - p_f\boldsymbol{n}_2 = \mu_2(\boldsymbol{u}_f - \boldsymbol{u}_2) \qquad 3.12$$

As indicated by Figure 3.2, $\mathbf{n}_2 = -\mathbf{n}_1 = -\mathbf{n}_{s2} = \mathbf{n}_{s1}$. Equations 3.11 and 3.12 can be arranged to the form shown in Equations 3.13 and 3.14.

$$(p_1 - p_f)\boldsymbol{n}_1 = \mu_1(\boldsymbol{u_f} - \boldsymbol{u_1}) \tag{3.13}$$

$$(-p_2 + p_f)\boldsymbol{n}_1 = \mu_2(\boldsymbol{u_f} - \boldsymbol{u_2}) \tag{3.14}$$

Solving for the final pressure by subtracting Equation 3.14 (referring to cell 2) multiplied by $\mu_1$ from Equation 3.13 (referring to cell 1) multiplied by $\mu_2$, Equation 3.15 is obtained.

$$p_f\boldsymbol{n}_1 = \frac{\mu_1\mu_2(\boldsymbol{u_1} - \boldsymbol{u_2}) + (\mu_2 p_1 + \mu_1 p_2)\boldsymbol{n_1}}{(\mu_1 + \mu_2)} \tag{3.15}$$

To solve for the final facial velocity determined by this Riemann solver, one simply adds the two equations and solve for $\boldsymbol{u}_f$ to give Equation 3.16 [8].

$$\boldsymbol{u}_f = \frac{(p_1 - p_2)\boldsymbol{n}_1 + \mu_1\boldsymbol{u}_1 + \mu_2\boldsymbol{u}_2}{(\mu_1 + \mu_2)} \tag{3.16}$$

A similar method is used to compute the facial pressures and velocities on the external problem boundaries of the mesh. This is computed using the prescribed $\boldsymbol{u}_f$ or $p_f$ provided as a boundary condition to the calculation. Thus, for a velocity boundary condition, the pressure is computed using Equation 3.17 and for a pressure boundary condition, the velocity is computed using Equation 3.18 [8], [40].

$$p_f = p_1 - \mu_1(\boldsymbol{u_1} - \boldsymbol{u_{1f}}) \cdot \boldsymbol{n}_1 \tag{3.17}$$

$$\boldsymbol{u_f} = \frac{(p_1 - p_f)\boldsymbol{n}_1}{\mu_1} - \boldsymbol{u_1} \tag{3.18}$$

### 3.4.3 Computing Linear Momentum and Kinetic Energy

Using the smallest unit of the grid construct (the iota), the linear momentum can be computed using the three velocity components described in Section 3.3 on the corners of each iota. Thus the linear momentum of each iota becomes a function of the linear interpolation of the velocity field as shown in Equation 3.19.

$$\boldsymbol{L}^i(\boldsymbol{u}^i) = \int_i \rho \left[ \sum_{j=1}^{3} \boldsymbol{u}_j^i \, \phi_j^i(x,y) \right] dA \tag{3.19}$$

Here, $\boldsymbol{L}^i(\boldsymbol{u}^i)$ represents the linear momentum of *"iota i,"* $\rho$ is the density of the material, and $\phi_j^i$ again represents the linear Lagrange polynomials. For simpler notation, one can define a variable, $l_j^i$, so that Equation 3.19 can be rewritten as Equation 3.21 [8].

$$l_j^i \equiv \int_i \rho \left[ \phi_j^i(x,y) \right] dA \tag{3.20}$$

$$\boldsymbol{L}^i(\boldsymbol{u}^i) = \sum_{j=1}^{3} \boldsymbol{u}_j^i \, l_j^i \tag{3.21}$$

After linear momentum is computed for each iota, the linear momentum of each cell (or zone) is computed by simply summing the linear momentum of each iota contained within it, as shown in Equation 3.22 where the summation is over all iota "*i*" in zone "*z*." In a similar fashion, the kinetic energy on each iota is computed and shown in Equation 3.23. Again, the kinetic energy of a zone is determined by summing the kinetic energy of all the iota contained within it, and is shown in Equation 3.24 [8].

$$\boldsymbol{L}^z = \sum_i^z \boldsymbol{L}^i(\boldsymbol{u}^i) \tag{3.22}$$

$$\boldsymbol{KE}^i(\boldsymbol{u}^i) = \frac{1}{2} \int_i \rho \left[ \sum_{j=1}^{3} \boldsymbol{u}_j^i \, \phi_j^i(x,y) \right]^2 dA \tag{3.23}$$

$$\boldsymbol{KE}^z = \sum_i^z \boldsymbol{KE}^i \qquad 3.24$$

### 3.4.4 Conservation of Linear Momentum and Kinetic Energy

Conservation of both cellular momentum and energy is achieved by calculating the stresses on the boundaries associated with both the Riemann solver and continuity forces. The Riemann solver provides a stress on the side of the iota connected to the cell face and is denoted by $\boldsymbol{\sigma}^{f-R}$ . This provides a stress for each iota denoted by $\boldsymbol{\sigma}^i$ that is a sum of the one-dimensional stress predicted by the Riemann solver and the continuity force $(\boldsymbol{\sigma}^{i-con})$ that accounts for two-dimensional stresses that are required to maintain continuity of the mesh such that $\boldsymbol{\sigma}^i = \boldsymbol{\sigma}^{f-R} + \boldsymbol{\sigma}^{i-con}$ . At the beginning of each time step, the continuity force (or reconciliation force) is first assumed to be zero, and the conservation equation is imposed using the cell's linear momentum and shown in Equation 3.25 [8].

$$\frac{d}{dt}\boldsymbol{L}^z = \sum_i^z \boldsymbol{\sigma}^i \boldsymbol{N}^i \qquad 3.25$$

Using the forward Euler approximation, a new linear momentum for the cell is determined for the next time step, but this linear momentum has yet to take into account the continuity stresses described previously. The facial velocities are calculated from this new linear momentum, and the mesh may become disconnected. The process of reconciling the vertex velocities is achieved by imposing that the sum of the linear momentum of all the iota around each vertex be preserved through this process, as shown in Equation 3.26.

$$\boldsymbol{L}^p\left(\boldsymbol{u}^{i-con}\right) = \boldsymbol{L}^p\left(\boldsymbol{u}^{i-R}\right) \quad where\, p = 1, 2, \dots., N_p \qquad 3.26$$

Equation 3.26 enforces conservation of linear momentum with both the continuity forces and the Riemann calculated forces around each point in the mesh [8].

Although the sum of the linear momentum of the iota is preserved in this process, each individual iota's value may have been altered to achieve connectivity and can be calculated as shown in Equation 3.27 using the basis functions. This in turn, slightly alters the linear momentum of the cell in which it is contained, which must be accounted for.

$$\Delta \boldsymbol{L}^i = \sum_{j=2}^{3} \left(\boldsymbol{u}^{i-con} - \boldsymbol{u}^{i-R}\right) l_j^i \qquad 3.27$$

This alteration in zonal linear momentum occurs from the uncalculated stresses developed during the reconnecting process that are not represented in the original Riemann solver. These stresses compose both tangential and normal stresses between the iota around a point. First, a local numbering system is established for the iota touching the vertex where each iota is given a number *1* through $n_i$, where $n_i$ is the number of iota touching the vertex. Second, a numbering system is established for the normal and tangential forces on each boundary. These numbering systems are shown in Figure 3.3 and Figure 3.4. This results in Equations 3.28 and 3.29 for the net forces in the x- and y-directions respectively, where *N* represents the area weighted normal for the intercellular direction and *H* represents the area weighted normal for the intracellular direction (hypotenuse of the iota) [8].

$$F_x^i = \left(-f_i N_x^i - f_{i+n} H_x^i\right) + \left(t_i N_y^i + t_{i+n} H_y^i\right) \qquad 3.28$$

$$F_y^i = \left(-f_i N_y^i - f_{i+n} H_y^i\right) + \left(-t_i N_x^i - t_{i+n} H_x^i\right) \qquad 3.29$$

By utilizing Newton's third law of motion where the forces on a shared side of an iota are equal in magnitude and opposite in direction, the numbering system is re-written to reduce the number of forces to compute. This reduces the number of forces from the original 32 (16 from each of the normal and tangential forces) to 16 total (8 from both the normal and tangential forces). These new numbering systems are shown in Figure 3.5 and Figure 3.6.

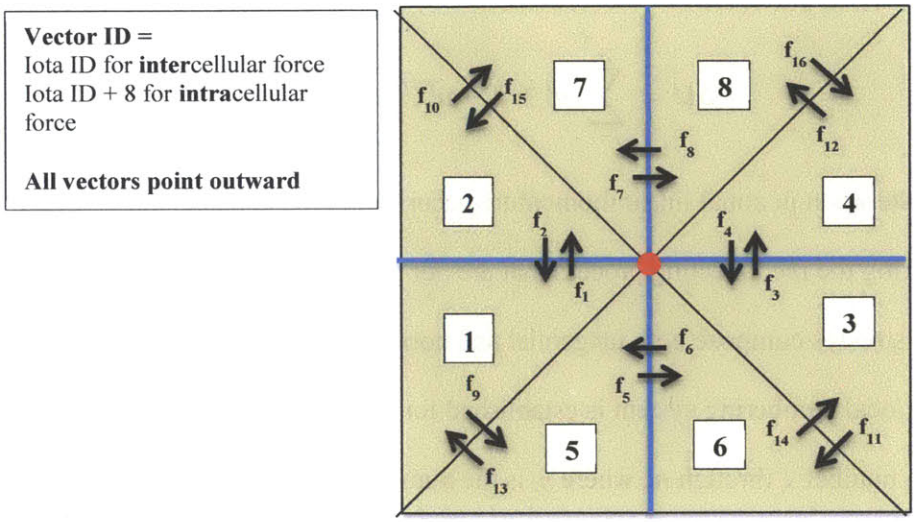


**Figure 3.3 The numbering system for touching a vertex is shown in the shadowed boxes with the numbering system for the normal on each surface with their sign convention. The blue lines and red point represent cell boundaries and the vertex respectively [8].**

**Vector ID =**
Iota ID for intercellular force
Iota ID + 8 for intracellular force

**All vectors point counterclockwise**

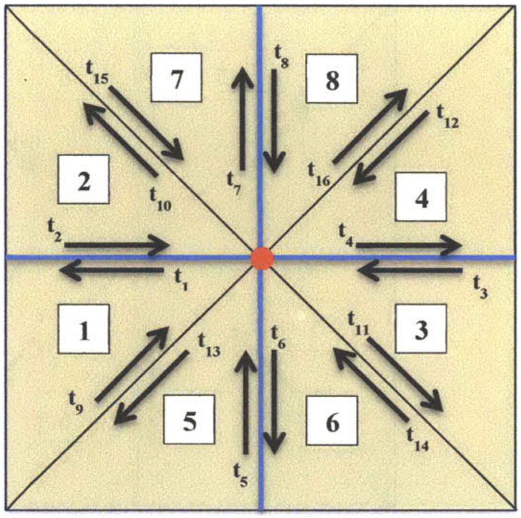


**Figure 3.4 The numbering system for the iota touching a vertex with the numbering system for the tangential forces on each surface with their sign convention [8].**

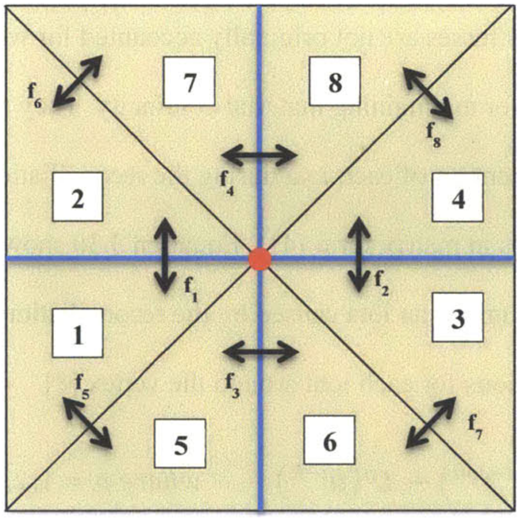


**Figure 3.5 The re-numbered normal forces, after utilizing Newton's third law of motion, on the iota. The force direction notation is kept for the lower numbered iota of a shared edge, and the force is now equivalent to a compressive or tensile stress [8].**

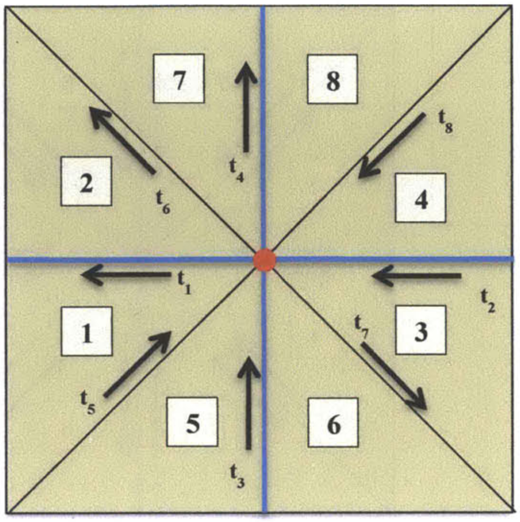


**Figure 3.6 The re-numbered tangential forces, after utilizing Newton's third law of motion, on the iota. The force direction notation is kept for the lower numbered iota of a shared edge [8].**

These additional stresses are not originally accounted for when using only the Riemann solver but are necessary for maintaining material continuity. They are computed by solving for the change in linear momentum of each iota during the reconciliation process and solving for the resulting continuity forces in matrix form [41]. Equation 3.30 shows the calculation for the change in linear momentum for an iota caused by the reconciliation process and is computed for the two coordinate directions for each iota around the vertex [8].

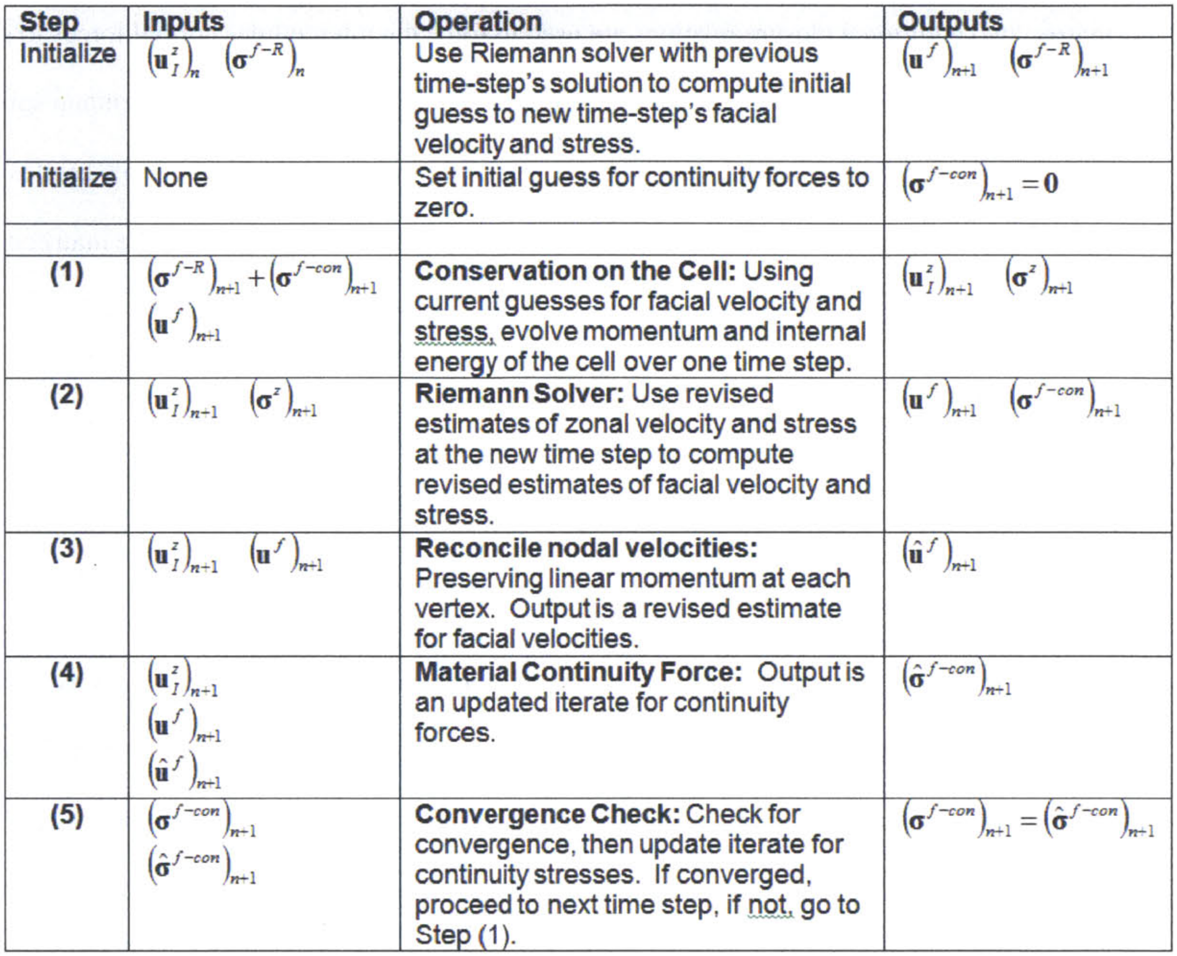
$$\Delta \boldsymbol{L} = \boldsymbol{L}^{p}\left(\boldsymbol{u}^{i-con}\right) - \boldsymbol{L}^{p}\left(\boldsymbol{u}^{i-R}\right) \qquad where\; p = 1, 2, \dots., N_p \tag{3.30}$$

A total of 16 vectors are shown in Figure 3.5 and Figure 3.6 which are computed for the two coordinate directions. The stresses are denoted as $w_j^{p-con}$ in Equation 3.31 (computed for the two coordinate directions), where "*p*" denotes the point (vertex), the term "*con*" refers to the fact that these forces are responsible for the two-dimensional continuity, and $\boldsymbol{A}$ is the matrix containing the normal direction of the stresses. Thus Equation 3.31 provides 16 equations using the change in linear momentum for each iota in the two coordinate directions. To complete the matrix, two additional closure relations are used to force the intercellular normal forces to sum to zero in both the x and y-directions. These values are already computed by the Riemann solver, so the reconciliation forces in addition to the Riemann solution must add to zero. This creates a non-singular matrix that can be used to solve for the continuity forces because the matrix now contains 16 non-linear dependent equations to solve for the 16 unknowns [8].

$$\sum_{j=1}^{16} \boldsymbol{A} w_j^{p-con} = \boldsymbol{L}^i\left(\boldsymbol{u}^{i-con}\right) - \boldsymbol{L}^i\left(\boldsymbol{u}^{i-R}\right) \qquad where\ i = 1, 2, \dots, 8 \qquad 3.31$$

### 3.4.5 Time Marching Scheme using the Reconciliation Forces

The time marching scheme proposed is first order but implicit. It strives to achieve consistency between the facial velocities, facial stresses, linear momentum of the cell, linear momentum around a vertex, and the prediction of the Riemann solver. At each time step, the steps described in Figure 3.7 are followed. This iteration scheme is the next step to study the reconciliation process and to create a new hydro method [8].



| Step | Inputs | Operation | Outputs |
|---|---|---|---|
| Initialize | $\left(\mathbf{u}_I^z\right)_n$ $\left(\boldsymbol{\sigma}^{f-R}\right)_n$ | Use Riemann solver with previous time-step's solution to compute initial guess to new time-step's facial velocity and stress. | $\left(\mathbf{u}^f\right)_{n+1}$ $\left(\boldsymbol{\sigma}^{f-R}\right)_{n+1}$ |
| Initialize | None | Set initial guess for continuity forces to zero. | $\left(\boldsymbol{\sigma}^{f-con}\right)_{n+1}=\mathbf{0}$ |
| | | | |
| (1) | $\left(\boldsymbol{\sigma}^{f-R}\right)_{n+1}+\left(\boldsymbol{\sigma}^{f-con}\right)_{n+1}$<br>$\left(\mathbf{u}^f\right)_{n+1}$ | **Conservation on the Cell:** Using current guesses for facial velocity and stress, evolve momentum and internal energy of the cell over one time step. | $\left(\mathbf{u}_I^z\right)_{n+1}$ $\left(\boldsymbol{\sigma}^z\right)_{n+1}$ |
| (2) | $\left(\mathbf{u}_I^z\right)_{n+1}$ $\left(\boldsymbol{\sigma}^z\right)_{n+1}$ | **Riemann Solver:** Use revised estimates of zonal velocity and stress at the new time step to compute revised estimates of facial velocity and stress. | $\left(\mathbf{u}^f\right)_{n+1}$ $\left(\boldsymbol{\sigma}^{f-con}\right)_{n+1}$ |
| (3) | $\left(\mathbf{u}_I^z\right)_{n+1}$ $\left(\mathbf{u}^f\right)_{n+1}$ | **Reconcile nodal velocities:** Preserving linear momentum at each vertex. Output is a revised estimate for facial velocities. | $\left(\hat{\mathbf{u}}^f\right)_{n+1}$ |
| (4) | $\left(\mathbf{u}_I^z\right)_{n+1}$<br>$\left(\mathbf{u}^f\right)_{n+1}$<br>$\left(\hat{\mathbf{u}}^f\right)_{n+1}$ | **Material Continuity Force:** Output is an updated iterate for continuity forces. | $\left(\hat{\boldsymbol{\sigma}}^{f-con}\right)_{n+1}$ |
| (5) | $\left(\boldsymbol{\sigma}^{f-con}\right)_{n+1}$<br>$\left(\hat{\boldsymbol{\sigma}}^{f-con}\right)_{n+1}$ | **Convergence Check:** Check for convergence, then update iterate for continuity stresses. If converged, proceed to next time step, if not, go to Step (1). | $\left(\boldsymbol{\sigma}^{f-con}\right)_{n+1}=\left(\hat{\boldsymbol{\sigma}}^{f-con}\right)_{n+1}$ |

**Figure 3.7 Description of the Runnels-Gilman Hydro Method time implicit marching scheme. This iteration occurs until convergence is reached for the continuity stresses [8].**

# 4 Pierre-Henri Maire Method

## 4.1 General Approach

The method devised by Pierre-Henri Maire (PHM) also uses the Lagrangian Godunov cell-centered hydro method. It was coded according to the description by PHM [5] into the DiscoverHydro code by Dr. Scott Runnels in order to investigate reconciliation forces that are acting during the calculation. A one-dimensional Riemann solution is computed at cell boundaries to provide an estimate of the cell boundaries' velocity and pressure at each time step. As discussed previously, since they are one-dimensional in nature, the Riemann velocities and pressures are incomplete and do not account for additional forces that hold the mesh together in two and three-dimensional situations [5].

PHM's method differs from the Runnels-Gilman method in the nodal solver scheme. PHM's nodal solver, that deals with the problem of maintaining mesh continuity after using the Riemann solver, uses a weighted least squares fit to the vertices surrounding Riemann face velocities as the vertex velocity [5]. In both the Runnels-Gilman and PHM approaches, pressures in addition to the one-dimensional Riemann solution's pressure are at work. PHM's method solves for the sum of the pressures concurrently, justifying the solution method as one that satisfies an entropy condition. These pressures are determined by enforcing the momentum equation on the zero-mass volume using only pressure forces. PHM's method also uses discontinuous pressures at cell boundaries while the Runnels-Gilman method has a continuous pressure across the cell boundary. Thus, the nodal velocity is simultaneously computed with the fluxes on the interfaces in an entropy-consistent Lagrangian scheme.

## 4.2 Governing Equations

As in the Runnels-Gilman method, gas dynamics equations (Equations 3.1, 3.2, and 3.3 for two-dimensional problems) are obtained in Lagrangian formalism and used in the development of PHM's method.

## 4.3 Grid Construction

PHM uses a grid constructed of polygons that do not overlap and an example polygon cell is shown in Figure 4.1. Here, the vertices are denoted by $M_r$, where $r$ ranges from one to the number of vertices of the polygon (which also corresponds to the number of faces). These indices are computed in a counterclockwise direction with periodic numbering. Looking at an edge defined by two vertices, such as $[M_r, M_{r+1}]$, the length of this edge is given by $L_{r,r+1}$ with a tangent vector in counterclockwise orientation given by $\boldsymbol{T}_{r,r+1}$ and a unit outward normal vector given by $\boldsymbol{N}_{r,r+1}$. The frame provides an orthonormal basis ($\mathbf{e}_x$, $\mathbf{e}_y$) that is completed by the vector defined by $e_z = e_x \times e_y$ [5].

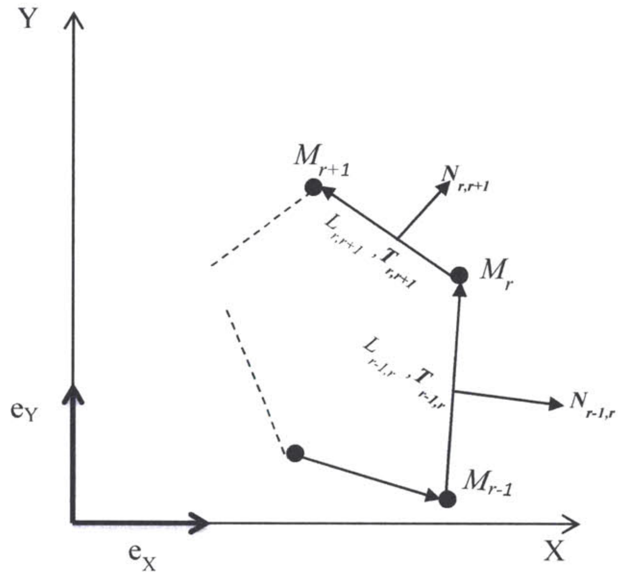


**Figure 4.1 An example polygon cell shape and the notation used for the vertices, unit outward normal vectors, unit tangent vectors, and the length of the edges [5].**

These sign and notation conventions shown in Figure 4.1 produce Equations 4.1 and 4.2.

$$L_{r,r+1}\boldsymbol{T}_{\boldsymbol{r,r+1}} = \boldsymbol{M_r M_{r+1}} \tag{4.1}$$

$$L_{r,r+1}\boldsymbol{N}_{\boldsymbol{r,r+1}} = L_{r,r+1}\boldsymbol{T}_{\boldsymbol{r,r+1}} \times e_z \tag{4.2}$$

## 4.4 Scheme Development

### 4.4.1 Conservation Laws on a Cell

For the space discretization of the mesh, the mass in each cell is described by discrete variables as it is in the Runnels-Gilman method described in section 3.4.1 shown again in Equations 4.3, 4.4, and 4.5 [5].

$$m_i \frac{d}{dt}(\tau_i) - \int_{d\Omega_i} \boldsymbol{V} \cdot \boldsymbol{N}\, dl = 0 \tag{4.3}$$

$$m_i \frac{d}{dt}(\boldsymbol{V}_i) + \int_{d\Omega_i} P \cdot \boldsymbol{N}\, dl = 0 \qquad 4.4$$

$$m_i \frac{d}{dt}(E_i) + \int_{d\Omega_i} P\boldsymbol{V} \cdot \boldsymbol{N}\, dl = 0 \qquad 4.5$$

In these Equations, $m_i$ denotes the mass in the cell area defined by the domain $\Omega_i$, and $P$ is the pressure given by the equation of state. Equations 4.3, 4.4, and 4.5 can be written to express the discrete face fluxes using the notation shown in Figure 4.1 using a summation. These face fluxes are shown in Equations 4.6, 4.7, and 4.8 where the volume energy flux is denoted by $\boldsymbol{V}^*_{r,r+1} \cdot \boldsymbol{N}_{r,r+1}$, the momentum flux by $P^*_{r,r+1} \cdot \boldsymbol{N}_{r,r+1}$, and the total energy flux by $(P\boldsymbol{V})^*_{r,r+1} \cdot \boldsymbol{N}_{r,r+1}$ for the face given by the vertices $[M_r, M_{r+1}]$ [5].

$$L_{r,r+1}\boldsymbol{V}^*_{r,r+1} \cdot \boldsymbol{N}_{r,r+1} = \int_{M_r}^{M_{r+1}} \boldsymbol{V} \cdot \boldsymbol{N}\, dl \qquad 4.6$$

$$L_{r,r+1}P^*_{r,r+1}\boldsymbol{N}_{r,r+1} = \int_{M_r}^{M_{r+1}} P\boldsymbol{N}\, dl \qquad 4.7$$

$$L_{r,r+1}(P\boldsymbol{V})^*_{r,r+1} \cdot \boldsymbol{N}_{r,r+1} = \int_{M_r}^{M_{r+1}} P\boldsymbol{V} \cdot \boldsymbol{N}\, dl \qquad 4.8$$

The mathematical definitions of the discrete face fluxes allows Equations 4.3, 4.4, and 4.5 to be re-written using a summation of these flux equations and are shown in Equations 4.9, 4.10, and 4.11 [5].

$$m_i \frac{d}{dt}(\tau_i) - \sum_{r=1}^{R(i)} L_{r,r+1}\boldsymbol{V}^*_{r,r+1} \cdot \boldsymbol{N}_{r,r+1} = 0 \qquad 4.9$$

$$m_i \frac{d}{dt}(\boldsymbol{V}_i) + \sum_{r=1}^{R(i)} L_{r,r+1}P^*_{r,r+1}\boldsymbol{N}_{r,r+1} = 0 \qquad 4.10$$

$$m_i \frac{d}{dt}(E_i) + \sum_{r=1}^{R(i)} L_{r,r+1}(\boldsymbol{PV})^*_{r,r+1} \cdot \boldsymbol{N}_{r,r+1} = 0 \qquad 4.11$$

The ensuing steps are then to compute the face fluxes and node velocities while ensuring the mesh connectivity by maintaining that the cell volume change and mesh motion agrees. To achieve this, PHM proposes to coherently calculate the node velocities with the face fluxes. This is accomplished by linking the volume flux $\boldsymbol{V}^*_{r,r+1} \cdot \boldsymbol{N}_{r,r+1}$ to the node velocity $\boldsymbol{V}^*_{r}$ [5].

Each face connecting two vertices is divided into two equal lengths, subsequently providing "half-face pressures" for these half-lengths of each face. An example is denoted by $P^{*i}_{r+\frac{1}{2},r}$ in Figure 4.2. The face fluxes can now be defined by the values shown in Figure 4.2 to provide the new set of equations shown in Equations 4.12, 4.13, and 4.14 [5].

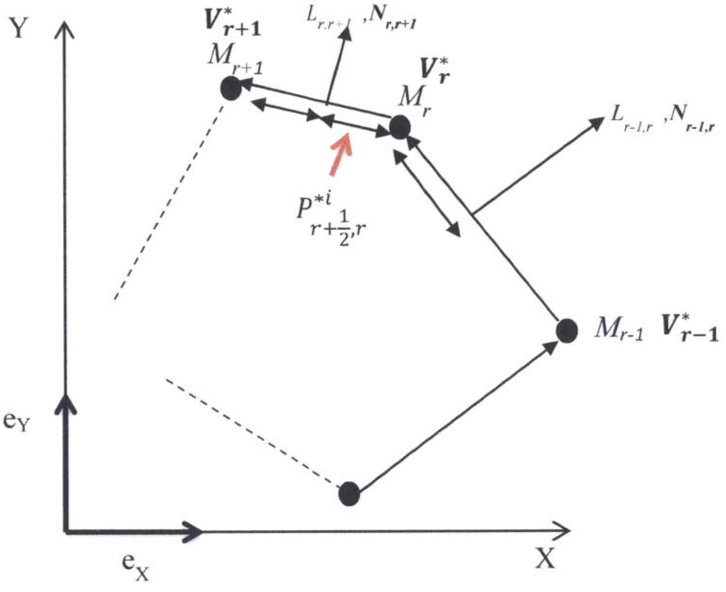

Y
$\boldsymbol{V}^*_{r+1}$
$M_{r+1}$
$L_{r,r+1}, \boldsymbol{N}_{r,r+1}$
$\boldsymbol{V}^*_{r}$
$M_r$
$L_{r-1,r}, \boldsymbol{N}_{r-1,r}$
$P^{*i}_{r+\frac{1}{2},r}$
$M_{r-1}$ $\boldsymbol{V}^*_{r-1}$
$e_Y$
$e_X$
X

**Figure 4.2 An example polygon cell shape and the notation used for the half-face pressures [5].**

$$\boldsymbol{V}^{*}_{\boldsymbol{r,r+1}} = \frac{1}{2}(\boldsymbol{V}^{*}_{\boldsymbol{r}} + \boldsymbol{V}^{*}_{\boldsymbol{r+1}}) \qquad 4.12$$

$$P^{*,i}_{r,r+1} = \frac{1}{2}(P^{*,i}_{r,r+\frac{1}{2}} + P^{*,i}_{r+\frac{1}{2},r+1}) \qquad 4.13$$

$$(P\boldsymbol{V})^{*,i}_{r,r+1} = \frac{1}{2}(P^{*,i}_{r,r+\frac{1}{2}}\boldsymbol{V}^{*}_{\boldsymbol{r}} + P^{*,i}_{r+\frac{1}{2},r+1}\boldsymbol{V}^{*}_{\boldsymbol{r+1}}) \qquad 4.14$$

The nodal solver can then be constructed to solve for the vertex velocity and two half-pressures around each node. This is done by the global conservation of momentum (by enforcing the conservation on each vertex individually) and the newly proposed local entropy inequality described in Section 4.4.3. Consequently, the two constraints of this method used to solve for the mesh motion are the conservation laws and the dissipation of entropy [5].

### 4.4.2 Enforcing the Conservation Relations on the Grid

The global conservation of momentum is enforced by summing over all the cells in the domain and is shown in Equation 4.15 where *I* denotes the total number of cells in the grid.

$$\frac{d}{dt}\left(\sum_{i=1}^{I} m_i\boldsymbol{V}_{\boldsymbol{i}}\right) = -\sum_{i=1}^{I}\sum_{r=1}^{r(i)}\frac{1}{2}\left(L^{i}_{r-1,r}\,P^{*,i}_{r-\frac{1}{2},r}\boldsymbol{N}^{\boldsymbol{i}}_{\boldsymbol{r-1,r}} + L^{i}_{r,r+1}\,P^{*,i}_{r,r+\frac{1}{2}}\boldsymbol{N}^{\boldsymbol{i}}_{\boldsymbol{r,r+1}}\right) \qquad 4.15$$

To change from cell to node notation (to use the nodal solver proposed), the summation is altered to be over all of the nodes in the mesh rather than the cells. This is shown in Equation 4.16 where *Q* represents the total number of nodes in the grid and *k* represents the cells surrounding a particular vertex. Therefore, to satisfy the global conservation of momentum, Equation 4.16 must go to zero, providing Equation 4.17.

$$\frac{d}{dt}\left(\sum_{i=1}^{I} m_i\boldsymbol{V}_{\boldsymbol{i}}\right) = -\sum_{q=1}^{Q}\sum_{k=1}^{K(q)}\frac{1}{2}(L_k\,P^{*,k}_{q,k}\boldsymbol{N}^{\boldsymbol{k}}_{\boldsymbol{k}} + L_{k+1}\,P^{*,k}_{q,k+1}\boldsymbol{N}^{\boldsymbol{k}}_{\boldsymbol{k+1}}) \qquad 4.16$$

$$\sum_{k=1}^{K(q)} \frac{1}{2}\left(L_k\, P_{q,k}^{*,k} \boldsymbol{N_k^k} + L_{k+1}\, P_{q,k+1}^{*,k} \boldsymbol{N_{k+1}^k}\right) = 0 \qquad 4.17$$

Since $\boldsymbol{V_q^*}$ is a single value for the node $M_q$, Equation 4.17 also implies the energy conservation over the mesh which is shown in Equation 4.18. Similarly, the conservation of volume is also achieved and shown in Equation 4.19. These equations are further formulated for the massless volume conservation surrounding the vertices.

$$\frac{d}{dt}\left(\sum_{i=1}^{I} m_i E_i\right) = -\sum_{q=1}^{Q} \sum_{k=1}^{K(q)} \frac{1}{2}\left(L_k\, P_{q,k}^{*,k} \boldsymbol{N_k^k} + L_{k+1}\, P_{q,k+1}^{*,k} \boldsymbol{N_{k+1}^k}\right) \cdot \boldsymbol{V_q^*} = 0 \qquad 4.18$$

$$\frac{d}{dt}\left(\sum_{i=1}^{I} m_i \tau_i\right) = \sum_{q=1}^{Q} \sum_{k=1}^{K(q)} \frac{1}{2}\left(L_k \boldsymbol{N_k^k} + L_{k+1} \boldsymbol{N_{k+1}^k}\right) \cdot \boldsymbol{V_q^*} \qquad 4.19$$

### 4.4.3 Entropy Inequality

The unique aspect of PHM's method is the introduction of the entropy inequality as a closure relation used in the nodal solver and enforced coherently with the conservation Equations. By defining $\sigma_i$ as the time variation of entropy for the cell defined by $\Omega_i$ and $T_i$ as the average temperature of that cell, Equation 4.20 represents this relation. This equation is computed in two steps. First, the time variation of the internal energy ($\frac{d}{dt}\varepsilon_i$) is computed by calculating the kinetic energy variation of the cell and subtracting that from the total energy variation. Then, the pressure work ($P_i \frac{d}{dt}\tau_i$) is calculated using Equation 4.3. This allows Equation 4.21 to be used to enforce the entropy inequality [5].

$$m_i T_i \frac{d}{dt}(\sigma_i) = m_i\left(\frac{d}{dt}\varepsilon_i + P_i \frac{d}{dt}\tau_i\right) \qquad 4.20$$

$$m_i T_i \frac{d}{dt}(\sigma_i) = \sum_{r=1}^{R(i)} \frac{1}{2}\left[L_{r-1,r}\left(P_i - P^{*,i}_{r-\frac{1}{2},r}\right)\boldsymbol{N_{r-1,r}} + L_{r,r+1}\left(P_i - P^{*,i}_{r,r+\frac{1}{2}}\right)\boldsymbol{N_{r,r+1}}\right] \cdot (\boldsymbol{V_r^*} - \boldsymbol{V_i}) \tag{4.21}$$

To enforce the entropy inequality, the right-hand side of Equation 4.21 must be positive and this provides a required condition that the closure relationship must satisfy in the nodal solver developed by PHM. PHM chooses to use Equations 4.22 and 4.23 as a condition to enforce the relation to be positive [5].

$$P_i - P^{*,i}_{r-\frac{1}{2},r} = \alpha_i(\boldsymbol{V_r^*} - \boldsymbol{V_i}) \cdot \boldsymbol{N_{r-1,r}} \tag{4.22}$$

$$P_i - P^{*,i}_{r,r+\frac{1}{2}} = \alpha_i(\boldsymbol{V_r^*} - \boldsymbol{V_i}) \cdot \boldsymbol{N_{r,r+1}} \tag{4.23}$$

To ensure the positive value of Equations 4.22 and 4.23, $\alpha_i$ must be a positive coefficient and is defined in Equation 4.24. Here, $c_i$ represents the isentropic speed of sound and is shown in Equation 4.25. This allows one to solve for the entropy variation.

$$\alpha_i = \rho_i c_i \tag{4.24}$$

$$c_i = \sqrt{\left(\frac{\partial P}{\partial \rho}\right)_\sigma} \tag{4.25}$$

Important to note is that, as is typical of Godunov-type hydro schemes, solving Equation 4.21 as described always yields a positive production of entropy. Thus, in the case of an isentropic flow, this scheme does not conserve entropy [5].

### 4.4.4 Nodal Solver Method

To solve for the velocity and pressures around a particular vertex in the mesh, the equations are reformulated to be around a generic vertex denoted by $M_q$ and shown in Equations

4.26 and 4.27, where $\alpha_k$, $P_k$, and $\boldsymbol{V}_k$represent the acoustic impedance, pressure, and velocity respectively of cell $\Omega_k$ [5].

$$P_k - P_{q,k}^{*,k} = \alpha_k(\boldsymbol{V}_q^* - \boldsymbol{V}_k) \cdot \boldsymbol{N}_k^k, \quad k = 1, \dots\dots, K(q) \qquad 4.26$$

$$P_k - P_{q,k+1}^{*,k} = \alpha_k(\boldsymbol{V}_q^* - \boldsymbol{V}_k) \cdot \boldsymbol{N}_{k+1}^k, \quad k = 1, \dots\dots, K(q) \qquad 4.27$$

The unknowns now to be calculated for a given vertex in the mesh include a total of 2**K(q)* pressure values and 2 scalar values for $\boldsymbol{V}_q^*$. A linear system is written to satisfy the velocity components $\boldsymbol{V}_q^*$ for the vertex $M_q$ by using Equation 4.17 that provides two scalar relations and Equations 4.26 and 4.27 that provide an additional 2**K(q)* relations to evaluate all the unknowns [5].

The description of the nodal solver functioning as a weighted least squares procedure in Section 1.2 comes from the use of the gradient of a quadratic functional created from the closure relations. By defining $v_k^*$ as the normal velocity that is provided by the one-dimensional Riemann solver for a specific face [$M_q$, $M_k$], Equation 4.28 is obtained.

$$v_k^* = \frac{P_{k-1} - P_k + \alpha_{k-1}\boldsymbol{V}_{k-1} \cdot \boldsymbol{N}_k^{k-1} + \alpha_k \boldsymbol{V}_k \cdot \boldsymbol{N}_k^{k-1}}{\alpha_{k-1} + \alpha_k} \qquad 4.28$$

Using this definition of the normal velocity, the closure relations produce Equation 4.29. The left-hand side of Equation 4.29 can then be written as a gradient of the quadratic functional shown in Equation 4.30 where the components of the velocity $\boldsymbol{V}_q^*$ are denoted by $(u_q^*, v_q^*)$. As a result, the solution to Equation 4.29 is the minimum of the functional $F(u_q^*, v_q^*)$, and therefore $\boldsymbol{V}_q^*$ is obtained from a weighted least squares method [5].

$$\sum_{k=1}^{K(q)} L_k(\alpha_{k-1}+\alpha_k)\left[\boldsymbol{V_q^*}\cdot\boldsymbol{N_k^{k-1}} - v_k^*\right]\boldsymbol{N_k^{k-1}} = 0 \tag{4.29}$$

$$F\left(u_q^*, v_q^*\right) = \sum_{k=1}^{K(q)} L_k(\alpha_{k-1}+\alpha_k)\left[\boldsymbol{V_q^*}\cdot\boldsymbol{N_k^{k-1}} - v_k^*\right]^2 \tag{4.30}$$

The nodal velocity can then be computed using a 2x2 linear system of equations as shown in Equations 4.31 and 4.32. The coefficients *A*, *B*, and *C* are defined by Equations 4.33, 4.34, and 4.35 respectively, where *k* again denotes the cells or edges around a specific node $M_q$. Finally, the right hand side of Equations 4.31 and 4.32 is defined by Equation 4.36 where $v_k^*$ is again the normal velocity given by the Riemann solver in Equation 4.28.

$$Au_q^* + Cv_q^* = SM_X \tag{4.31}$$

$$Cu_q^* + Bv_q^* = SM_Y \tag{4.32}$$

$$A = \sum_{k=1}^{K(q)} L_k(\rho_{k-1}c_{k-1}+\rho_k c_k)\left(N_{k,X}^{k-1}\right)^2 \tag{4.33}$$

$$B = \sum_{k=1}^{K(q)} L_k(\rho_{k-1}c_{k-1}+\rho_k c_k)\left(N_{k,Y}^{k-1}\right)^2 \tag{4.34}$$

$$C = \sum_{k=1}^{K(q)} L_k(\rho_{k-1}c_{k-1}+\rho_k c_k)N_{k,X}^{k-1}\,N_{k,Y}^{k-1} \tag{4.35}$$

$$SM = \sum_{k=1}^{K(q)} L_k(\rho_{k-1}c_{k-1}+\rho_k c_k)\,v_k^*\boldsymbol{N_k^{k-1}} \tag{4.36}$$

## 4.5 Reconciliation Forces Calculation

As described in Section 1.2, PHM uses a nodal solver where the nodal velocity is simultaneously computed with the fluxes on the interfaces in an entropy-consistent Lagrangian scheme that conserves momentum around each massless volume. Doing so means that the linear momentum of the associated iota are adjusted from what the one-dimensional Riemann solution would have them be. PHM reasons that the entropy inequality constraint enforces the conservation laws for the grid, but by using the reconciliation forces solver described in the Runnels-Gilman method, the forces used to retain mesh continuity will be computed explicitly to investigate if this entropy-consistent scheme accounts for all the forces used when moving the mesh.

# 5 Piston Test Problem

## 5.1 Description of 1D Piston Test Problem

The 1D Piston Test Problem is a hydrocode test using a constant-state, constant-velocity shock of infinite strength. This is commonly referred to as the piston-driven shock where the reference frame is chosen so that the piston (the side positioned at *x=0*) is initially at rest. The shock of infinite strength means that the initial pressure of the system (pre-shock pressure) is set equal to zero ($P^- = P^0 = 0$). Since hydrocodes are developed initially using gas dynamics, the shock is generated in a perfect gas, and thus $\gamma = \frac{5}{3}$. The gas is also initially cold so that the specific internal energy is zero ($\varepsilon^0 = 0$) and is approximated as such by setting $\varepsilon^0 = 1.0e^{-04}$. Additionally, the initial density is set as $\rho^0 = 1$ and the initial velocity for the piston is $u^0 = 1$ [35]. Equation 5.1 provides the value for the pre-shock pressure as well as the initial conditions.

$$P^0 = (\gamma - 1)\rho^0\varepsilon^0 = 0 \tag{5.1}$$

The analytical solution to this classic piston test problem is shown in Figure 5.1 after a time of t=0.6μs and where *S* represents the constant shock speed. The values of the post-shock solution for the velocity, density, specific internal energy, and pressure are $u^+ = 0,\ \rho^+ = 4, \varepsilon^+ = \frac{1}{2}$, and $P^+ = \frac{4}{3}$ respectively [35].

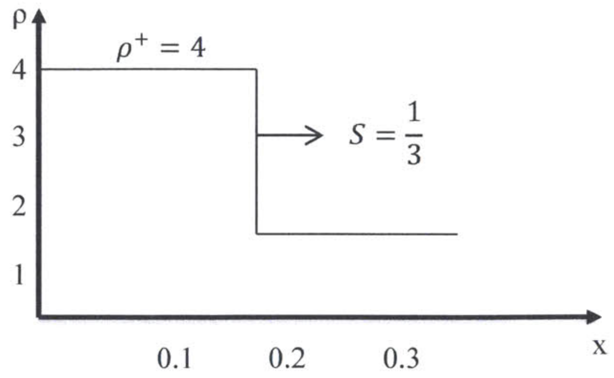


**Figure 5.1 The exact solution at t=0.6µs to Noh's generic constant-velocity shock problem (piston test problem).**

## 5.2 Grid Parameters

The piston test problem was completed on a [0,1]x[0,1] mesh in the x and y-coordinate directions. The parameters were used as defined in section 5.1 with equally partitioned cells in a mesh comprised of 50x2 cells in the x and y-directions respectively. The results were also compared with a grid of 20x4 cells and shown to provide similar results for density, pressure, velocity, and internal energy at t=0.6µs .

## 5.3 Runnels-Gilman Method Test Results

### 5.3.1 Results of the Original Piston Test Problem

The piston test problem parameters were implemented using the Runnels-Gilman method and the reconciliation forces were calculated through the simulation using time step increments of t=0.001µs. The plots for density and pressure at t=0.6µs for a uniform 50x2 cell grid are shown in Figure 5.2 and Figure 5.3 respectively. ***Important to note here is that the Runnels-Gilman method is being tested without implementing the reconciliation forces as additional forces acting during the movement of the mesh. Instead, the reconciliation forces are merely being calculated to determine their presence or absence during the simulation.***

As discussed in section 5.1, the values of the post-shock solution for the pressure and density are analytically calculated to be $\frac{4}{3}$ $Mbar$ and 4 ${}^{gm}/_{cm^3}$ respectively. The Runnels-Gilman method (implemented without the reconciliation forces iteration scheme) provides a pressure of ~4/3 *Mbar*, a density of ~4.11 ${}^{gm}/_{cm^3}$, and an internal energy ~1/2 ${}^{erg}/_{gm}$. These values are comparable to the values of the analytical solution to the constant-state, constant-velocity shock of infinite strength test problem and summarized in Table 1.

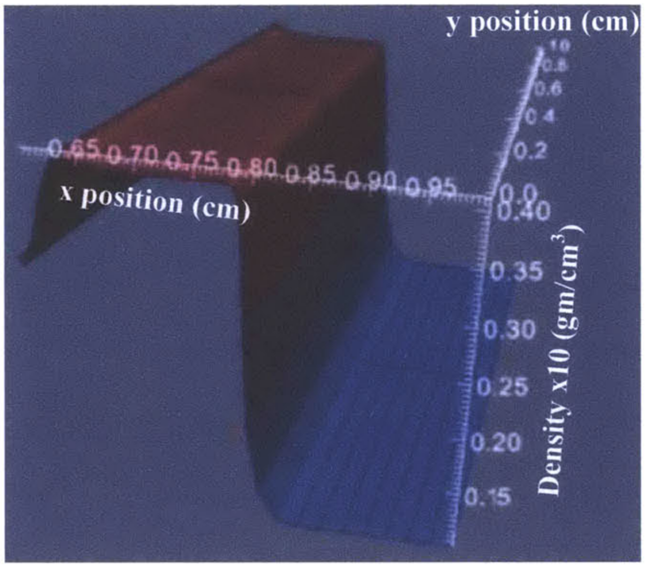


**Figure 5.2 A 50x2 grid with dimensions of [0,1]x[0,1] showing the density after t=0.6μs using the Runnels-Gilman method. Note the density is scaled by a factor of 0.1. The density behind the shock is shown to be near the analytical solution of 4.**

As is common in shock capturing simulations, the density plot in Figure 5.2 illustrates a numerical error that occurs at the front of the mesh that is referred to as "wall heating." This is a phenomenon not present in experimental results or the analytical solution [42]. Wall heating is visible in the sharp density decrease in the first few cells of the mesh. In this piston problem, it does not appear to affect the pressure calculation, which is illustrated in the constant pressure behind in the shock in Figure 5.3. Shock physics simulations are still being heavily investigated to determine the cause of this wall heating and to develop schemes that minimize wall heating errors [43]. In the Runnels-Gilman method, this phenomenon is illustrated by the density at the wall of ~3.1 ${}^{gm}/_{cm^3}$ compared to the density of ~4.11 ${}^{gm}/_{cm^3}$ at the shock front.

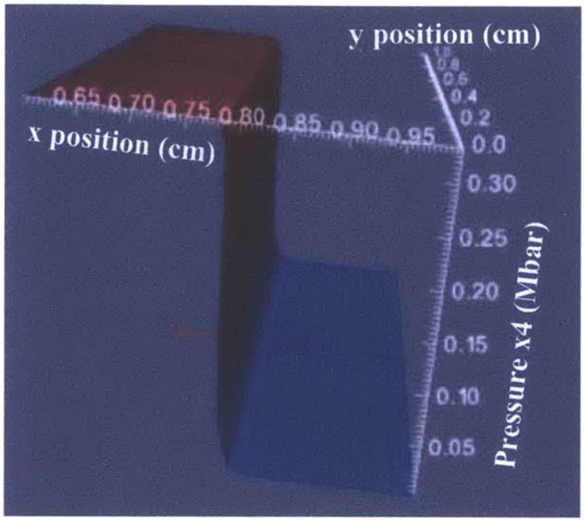


**Figure 5.3 A 50x2 grid with dimensions of [0,1]x[0,1] showing the pressure after t=0.6μs using the Runnels-Gilman method. Note the pressure is scaled by a factor of 0.25. The pressure behind the shock is shown to be near the analytical solution of 4/3.**

### 5.3.2 Results of the Rotated Grid Piston Problem

To investigate the accuracy of the reconciliation forces solver and the robustness of the code, the Piston test problem was also run on a rotated grid. This was accomplished by rotating the mesh by 45° with respect to the origin. This gave a mesh with side lengths of one, but with corners positioned at (0,1), (1,0), (2,1), and (1,2) as is shown in Figure 5.4. The results of the simulation are shown in Figure 5.5 and Figure 5.6 for the density and pressure calculations respectively. The Runnels-Gilman method implemented without the reconciliation forces iteration on the rotated grid provides a pressure of ~4/3 *Mbar*, a density of ~4.12 ${}^{gm}/_{cm^3}$, and an internal energy ~1/2 ${}^{erg}/_{gm}$ and is summarized in Table 1. These values are comparable to the values of the analytical solution and the results from the original piston test problem using the Runnels-Gilman method. This grid rotation did show a small change in the amount of wall heating compared to the original piston test problem, as the density at the wall was $\sim 2.9 \, {}^{gm}/_{cm^3}$.

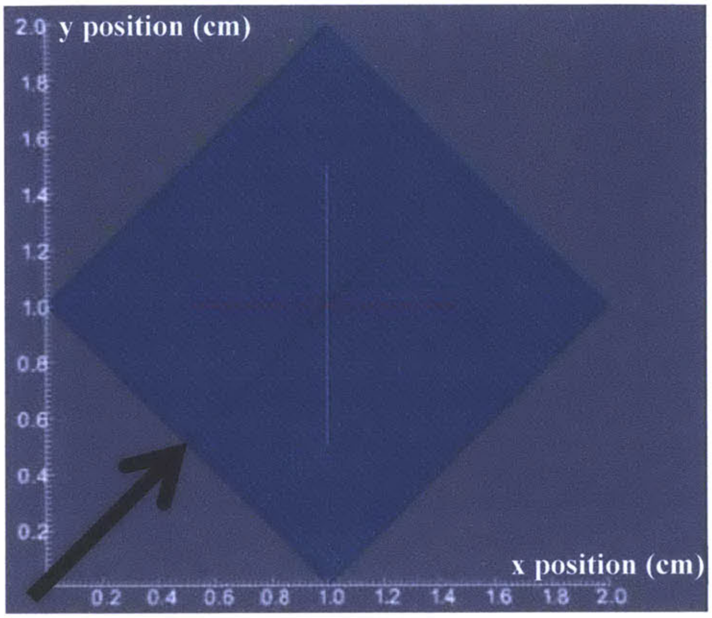


**Figure 5.4 The initial grid set-up for the Piston Test Problem on a crooked grid. The arrow indicates the face of the mesh where the normal boundary condition velocity is placed.**

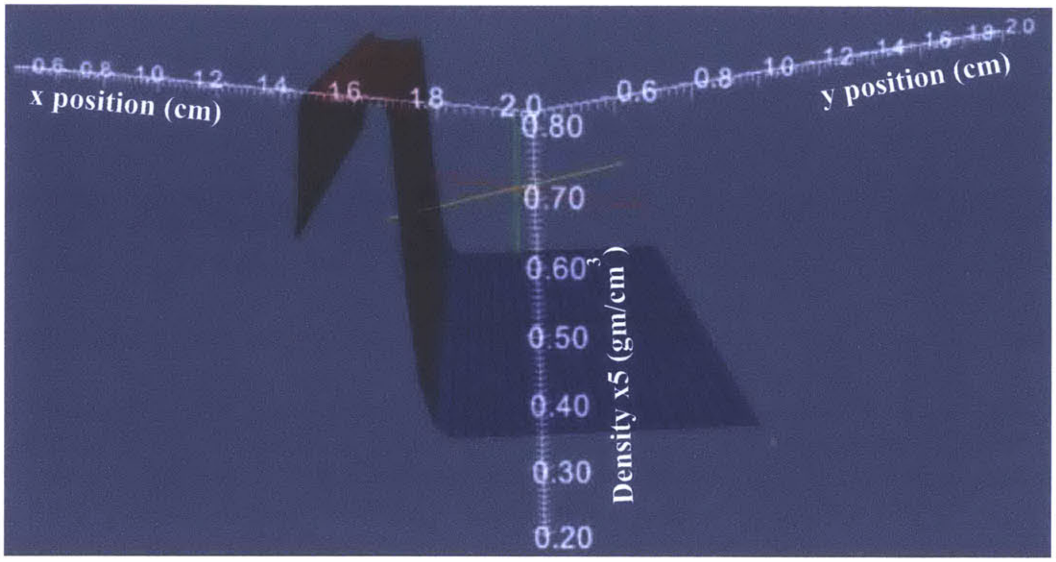


**Figure 5.5 A 50x2 grid on the rotated mesh showing the density after t=0.6μs using the Runnels-Gilman method. Note the density is scaled by a factor of 0.2. The density behind the shock is shown to be near the analytical solution of 4.**

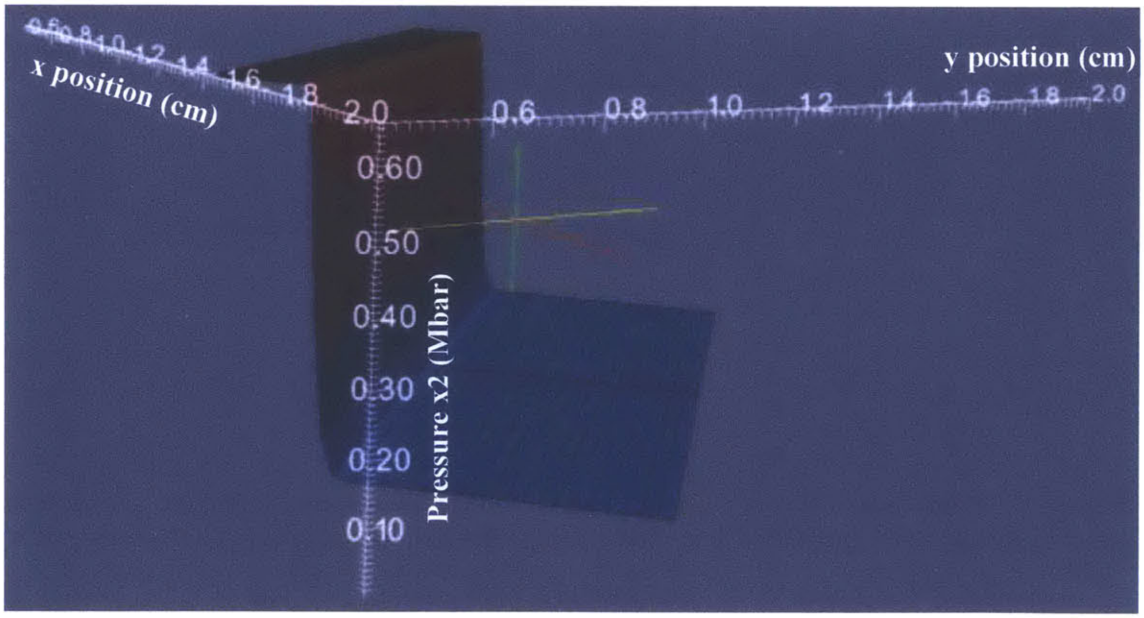


**Figure 5.6 A 50x2 grid on the rotated mesh showing the pressure after t=0.6μs using the Runnels-Gilman method. Note the pressure is scaled by a factor of 0.5. The pressure behind the shock is shown to be near the analytical solution of 4/3.**

### 5.3.3 Reconciliation Forces on the Original Piston Test Problem

The reconciliation forces in this test problem were calculated for each iota in the mesh around each internal vertex for both the x and y-directions. The net reconciliation forces for the two coordinate directions were calculated using Equations 3.28 and 3.29. Reconciliation forces acting on each iota were analyzed by calculating the net reconciliation force acting on each corner of the four cells that surround a particular vertex as shown in Figure 5.7. The net reconciliation force on each corner was calculated using the sum of the reconciliation forces of the iota contained in the corner. Finally, the net reconciliation force acting on the staggered cell around the vertex was determined using the sum of the reconciliation forces of all the iota surrounding the vertex.

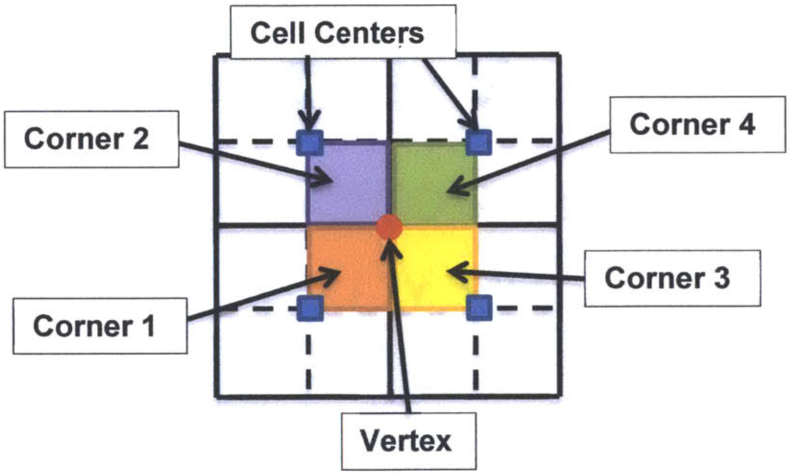


**Figure 5.7 Construct of the corners surrounding a vertex that are analyzed. The bottom left, top left, bottom right, and top right correspond to corners 1, 2, 3, and 4 respectively.**

As expected, the sum of all the reconciliation forces around each vertex was zero since the method used to reconcile the vertex velocities enforces that the sum of the linear momentum of all the iota around each vertex be preserved during this process (described in Section 3.4.4). In contrast, each individual corner surrounding a vertex need not sum to zero. The vertices that were surrounded by corners enveloping non-zero reconciliation forces were shown to propagate with the shock front through time. Note that only interior nodes were investigated for the reconciliation forces. Also, since this test problem involves only a velocity in the x-direction, the reconciliation forces in the y-direction were shown to be negligible. The shock front at t=0.0μs in Figure 5.8 can be detected by inspecting the pressure (plotted in black in the figure). The reconciliation forces on each corner, which will now be referred to as "corner reconciliation forces," are also shown in Figure 5.8. Here, it is verified visually that the sum of the reconciliation forces around each vertex is equal to zero.

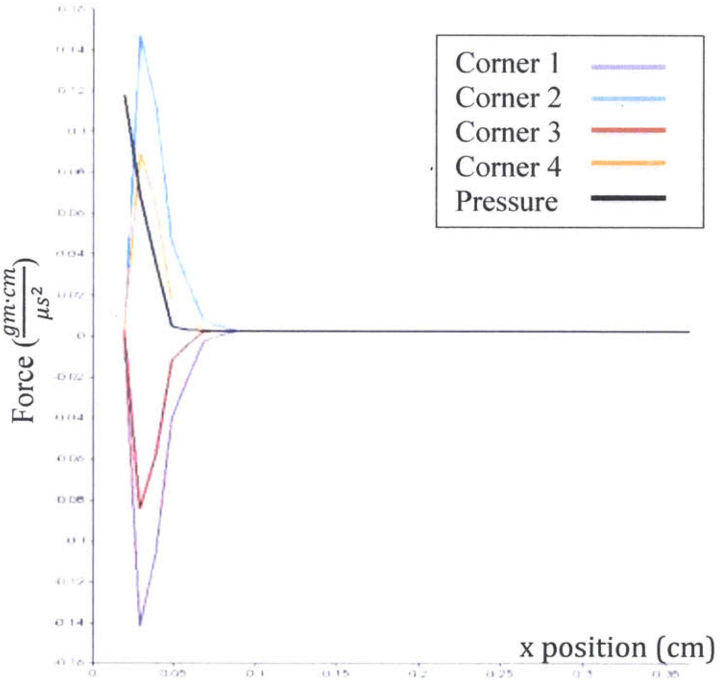


**Figure 5.8 A 50x2 grid with dimensions of [0,1]x[0,1] showing the net forces at t=0.01μs for corners 1, 2, 3, and 4 and the pressure (units of Mbar) using the Runnels-Gilman method.**

The reconciliation forces at t=0.6μs were also plotted to illustrate that the non-zero reconciliation forces propagate with the shock front through the simulation. Figure 5.9 shows the shock wave at t=0.6μs by plotting the pressure with the corner reconciliation forces (note the position of the shock front). Figure 5.9 also illustrates that the sum of the reconciliation forces in all four corners is also equal to zero, as is expected due to linear momentum preservation during the reconciling of vertex velocities.

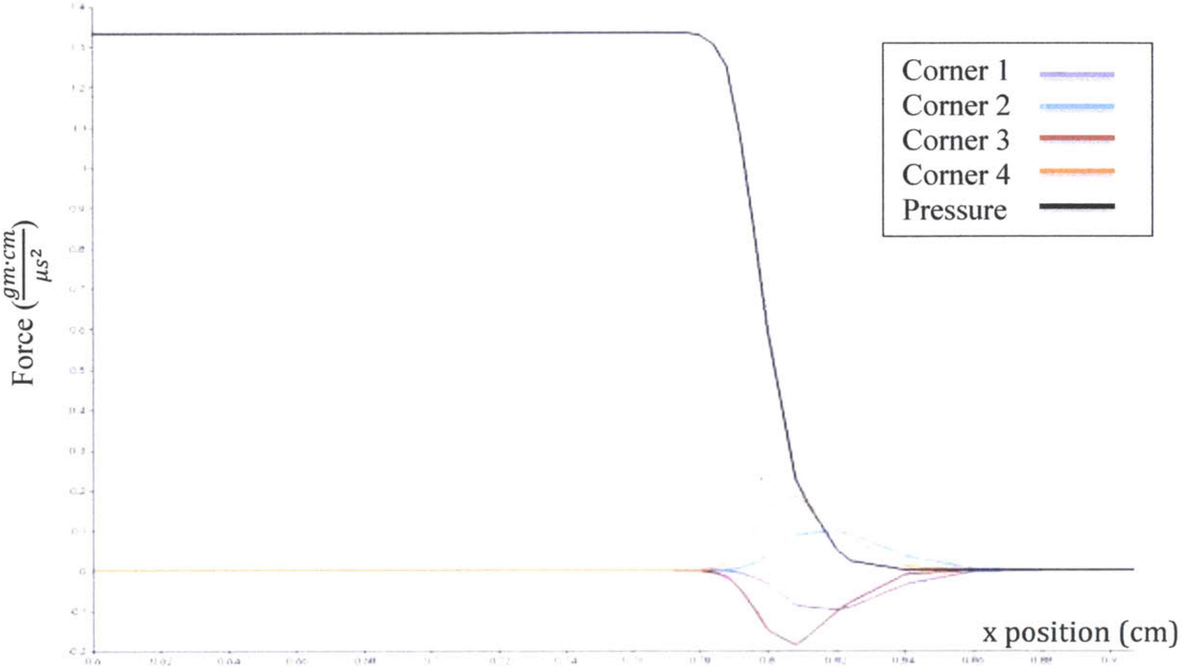


**Figure 5.9 A 50x2 grid with dimensions of [0,1]x[0,1] showing the net forces at t=0.6μs for corners 1, 2, 3, and 4 and the pressure (units of Mbar) using the Runnels-Gilman method.**

### 5.3.4 Reconciliation Forces on the Rotated Grid Piston Problem

As expected, the sum of all the reconciliation forces around each vertex was again zero while each individual corner surrounding a vertex had non-zero reconciliation forces surrounding the vertices at the shock front. The difference in the reconciliation forces values determined in this rotated grid versus the original piston test problem is due to the existence of reconciliation forces in the y-direction. This is due to the different velocity boundary condition. In the rotated grid, the velocity is initiated in both the x and y-coordinate directions in equal magnitude, whereas in the original piston problem it is only in the x-direction. This leads to reconciliation forces in the x and y-directions on the rotated grid that are equal in magnitude. Figure 5.10 illustrates the shock wave at t=0.01μs by plotting the pressure with the corner reconciliation forces (note the position of the shock front). The net forces on corners 1, 2, 3, and 4 for the y-coordinate direction are of the same magnitude and position as those illustrated in these figures.

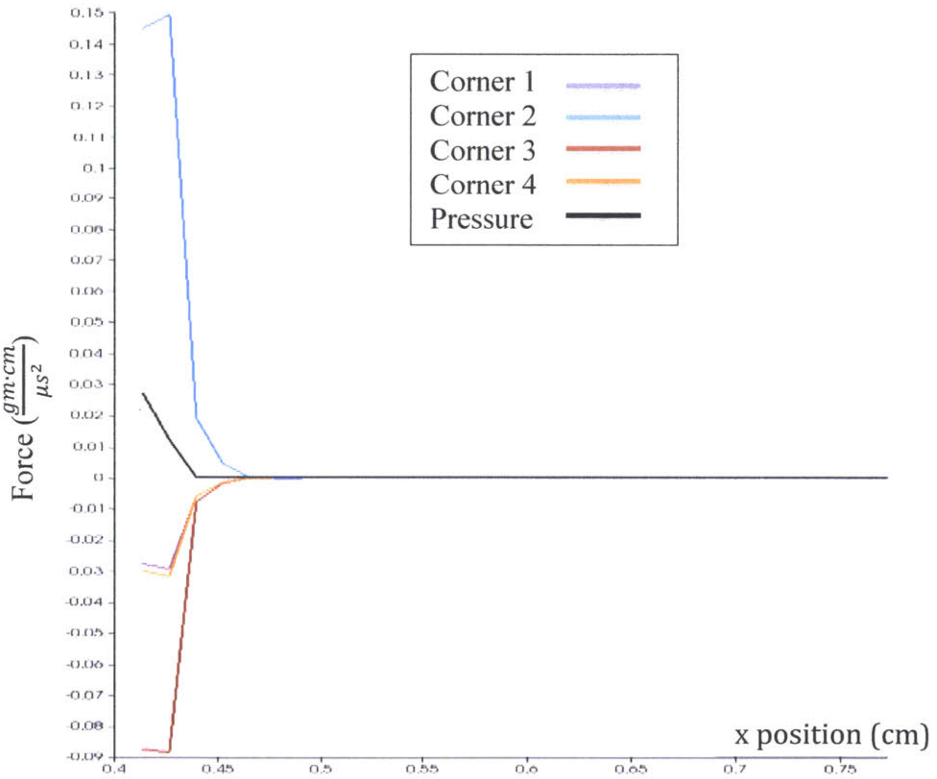


**Figure 5.10 A 50x2 grid on the rotated mesh illustrating the net reconciliation forces at corners 1, 2, 3, and 4 in the x-direction (y-direction is the same) surrounding each vertex and the pressure (units of Mbar) at t=0.01μs using the Runnels-Gilman method.**

To illustrate the propagation of reconciliation forces with the shock front through the simulation, the reconciliation forces at t=0.6μs were also plotted. Figure 5.11 illustrates the shock wave at t=0.6μs by plotting the pressure with the corner reconciliation forces (note the position of the shock front). Figure 5.11 verifies visually that the sum of the reconciliation forces in all four corners is equal to zero.

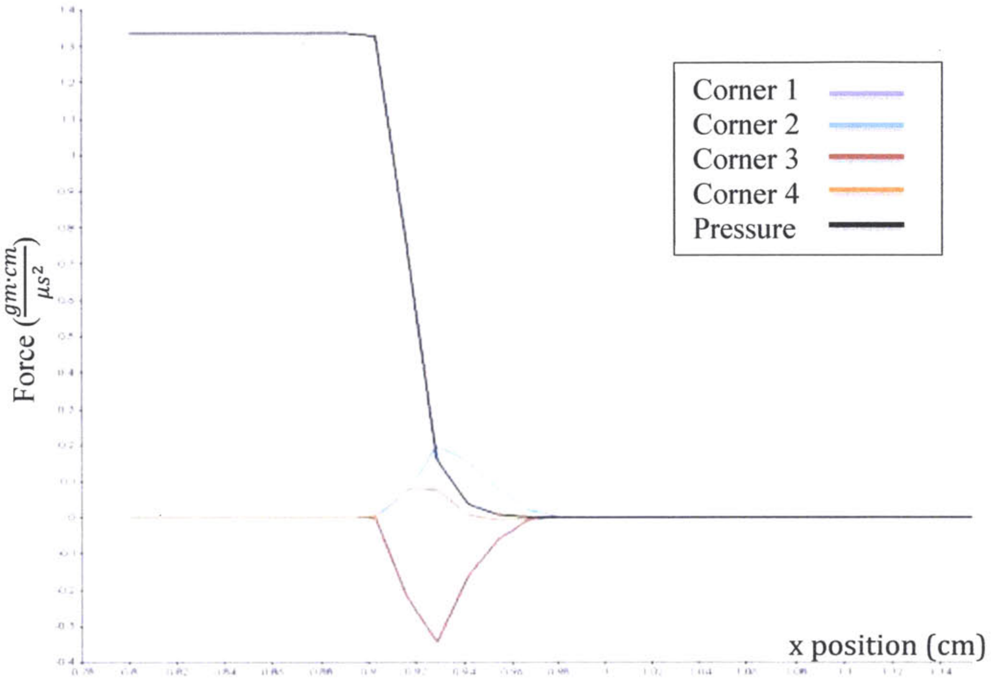


**Figure 5.11 A 50x2 grid on the rotated mesh illustrating the net reconciliation forces at corners 1, 2, 3, and 4 in the x-direction (y-direction is the same) surrounding each vertex and the pressure (units of Mbar) at t=0.6μs using the Runnels-Gilman method.**

## 5.4 PHM Method Test Results

### 5.4.1 Results of the Piston Test Problem

The piston test problem was simulated with PHM's method through the DiscoverHydro code, the plots for density and pressure after t=0.6μs, using time steps of t=0.001μs, on a 50x2 uniform cell grid are shown in Figure 5.12 and Figure 5.13 respectively.

As discussed in section 5.1, the values of the post-shock solution for the pressure and density are analytically calculated to be $\frac{4}{3}$ $Mbar$ and 4 $^{gm}/_{cm^3}$ respectively. PHM's method implemented in the DiscoverHydro code provides a pressure of ~4/3 *Mbar*, a density of ~4.02 $^{gm}/_{cm^3}$, and an internal energy ~1/2 $^{erg}/_{gm}$. These values are also comparable to the values of the analytical solution and summarized in Table 1. These figures also correspond to the figures published by PHM [5].

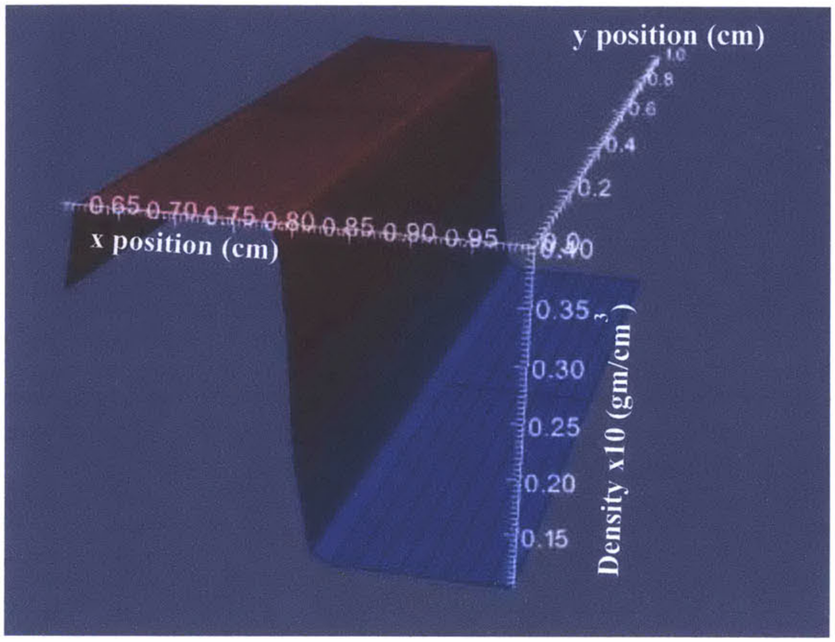


**Figure 5.12 A 50x2 grid with dimensions of [0,1]x[0,1] showing the density after t=0.6μs using PHM's method. Note the density is scaled by a factor of 0.1. The density behind the shock is shown to be near the analytical solution of 4 with a density of 3.3 at the wall.**

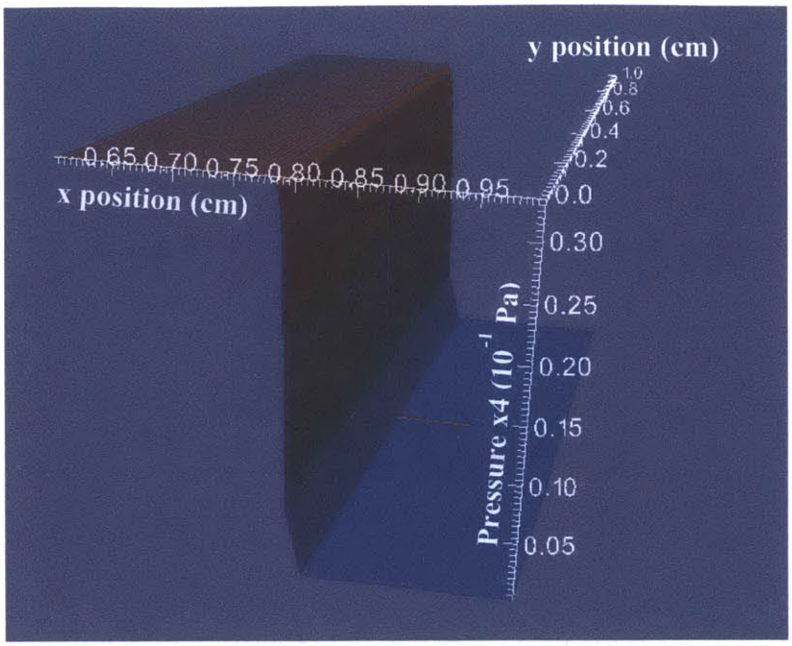


**Figure 5.13 A 50x2 grid with dimensions of [0,1]x[0,1] showing the pressure after t=0.6μs using PHM's method. Note the pressure is scaled by a factor of 0.25. The pressure behind the shock is shown to be near the analytical solution of 4/3.**

| Parameter | Analytical Value | Runnels-Gilman (original piston problem) | Runnels-Gilman (rotated grid) | PHM's method (original piston problem) |
|---|---|---|---|---|
| $\rho^{+}\left[{}^{gm}/_{cm^3}\right]$ | 4 | 4.11 | 4.12 | 4.02 |
| $\varepsilon^{+}\left[{}^{erg}/_{gm}\right]$ | $^1/_2$ | $^1/_2$ | $^1/_2$ | $^1/_2$ |
| $P^{+}\,[Mbar]$ | $^4/_3$ | $^4/_3$ | $^4/_3$ | $^4/_3$ |
| Wall heating<br>( $\rho$ $\left[{}^{gm}/_{cm^3}\right]$ at the wall) | 4 | 3.1 | 2.9 | 3.3 |

**Table 1 A comparison of the solutions of the Runnels-Gilman (both original grid setup and the rotated grid) and PHM methods to the analytical solution for the piston test problem.**

### 5.4.2 Reconciliation Forces Calculation

As expected, the sum of the reconciliation forces of all the iota around each vertex was zero and similar to the results seen in the Runnels-Gilman method, some individual corners surrounding interior vertices had non-zero reconciliation forces. Note again that only interior nodes were investigated for the reconciliation forces. Also, since this test problem involves only a velocity in the x-direction, the reconciliation forces in the y-direction were shown to be negligible. Figure 5.14 illustrates the shock wave at t=0.01µs by plotting the pressure with the corner reconciliation forces (note the position of the shock front). The vertices with corners enveloping non-zero reconciliation forces propagated with the shock front through time, but a distinct difference as compared to the Runnels-Gilman method was that the reconciliation forces were also acting behind the shock front. This is shown in Figure 5.14 because non-zero corner reconciliation forces are calculated back to the vertices at the front of the mesh. Figure 5.14 also illustrates that the sum of the reconciliation forces in all four corners is equal to zero.

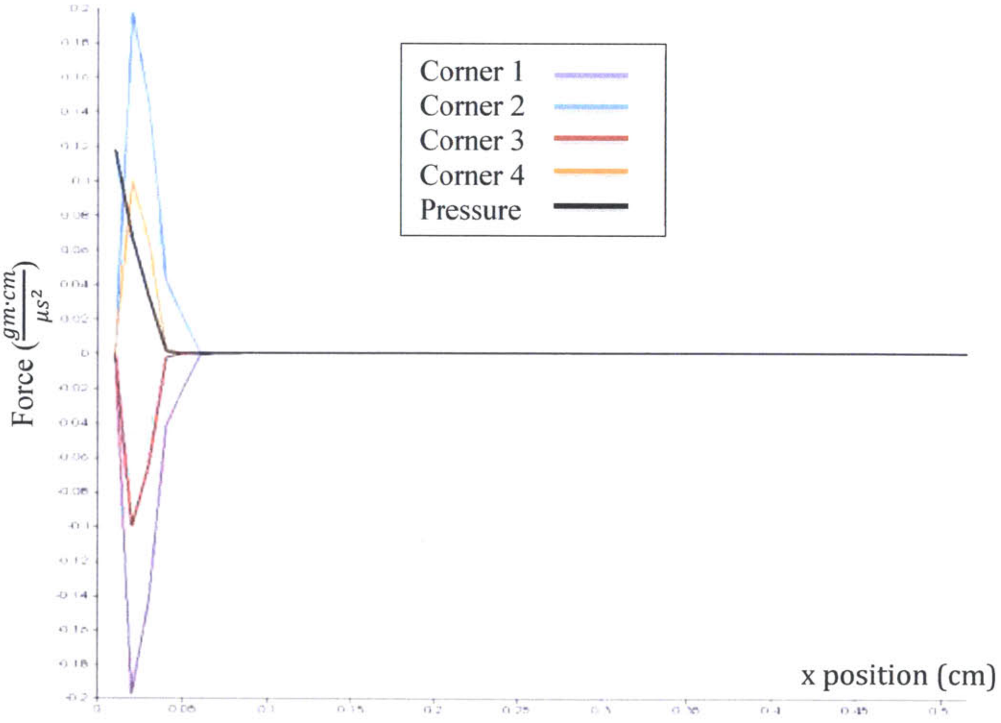


**Figure 5.14 A 50x2 grid with dimensions of [0,1]x[0,1] showing the net forces at t=0.01µs for corners 1, 2, 3, and 4 and the pressure (units of Mbar) using PHM's method.**

To investigate the reconciliation forces at the end of the simulation, the plot in Figure 5.15 illustrates the shock wave at t=0.6μs by showing the pressure (note the position of the shock front) with the corner reconciliation forces. The results shown here more clearly illustrate that the vertices affected by the reconciliation forces include all internal vertices behind the shock wave.

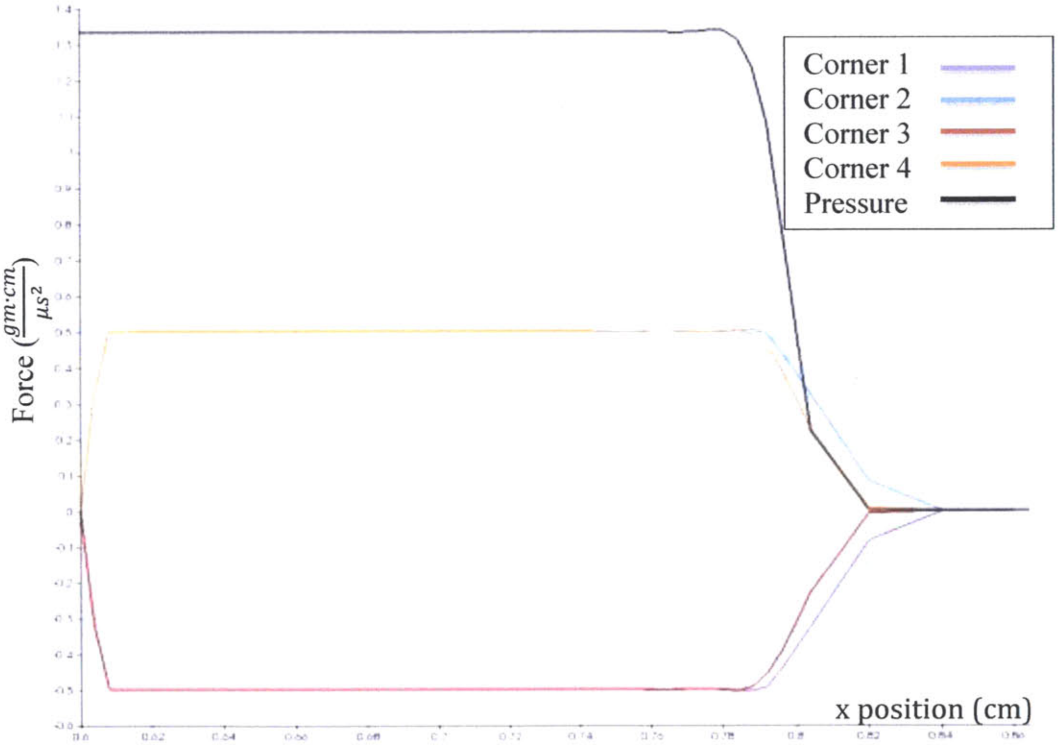


**Figure 5.15 A 50x2 grid with dimensions of [0,1]x[0,1] showing the net forces at t=0.6μs for corners 1, 2, 3, and 4 and the pressure (units of Mbar) using PHM's method. Note that not only the vertices at the front of the shock wave have cell corners that have reconciliation forces.**

# 6 Saltzman Test Problem

## 6.1 Description of Saltzman Problem

Saltzman's shock tube problem is similar to the piston test problem [5] in that it simulates the movement of a planar shock using a Cartesian grid, but instead of using cells of uniform size, they are distorted prior to the start of the calculation. The purpose of this is to test the robustness of the hydro method when the grid is not perfectly aligned with the fluid flow [5].

Again, as similar to the piston test problem, a shock of infinite strength is generated in a perfect gas ($\gamma = \frac{5}{3}$). The gas is also initially cold so that the specific internal energy is zero ($\varepsilon^0 = 0$) and is approximated as such by setting $\varepsilon^0 = 1.0e^{-04}$. The initial density is set as $\rho^0 = 1$ and the initial velocity for the piston is $u^0 = 1$.

## 6.2 Grid Parameters

The Saltzman problem was completed on a [0,1]x[0,0.1] mesh in the x and y-coordinate directions. The parameters were used as defined in section 6.1 in a mesh comprised of 100x10 cells in the x and y-directions respectively. The cells were then distorted and stretched by using the mapping shown in Equation 6.1 [5]. Figure 6.1 shows the initial rectangular mesh with distorted cells at t=0. Cells the shock wave has not disturbed from their original position are colored blue. The color red indicates cells the shock wave has reached or passed through during the simulation. The red and yellow lines illustrate the x and y-axis directions respectively.

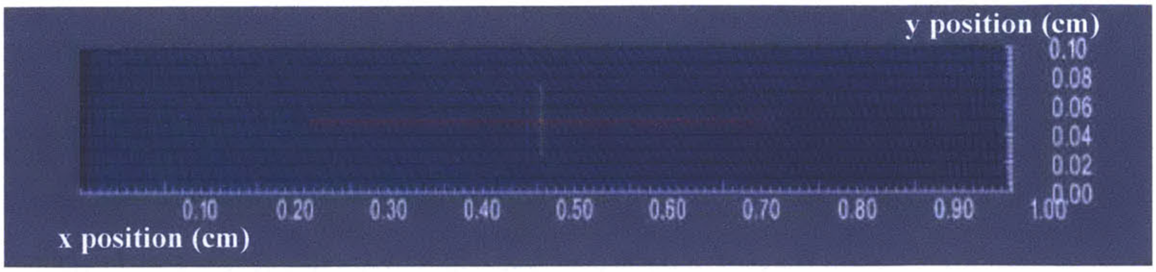


**Figure 6.1 The initial (t=0) rectangular [0,1]x[0,0.1] mesh with distorted cells used in the Saltzman test problem.**

$$x_{str} = x + (0.1 - y)\sin(x\pi)$$
$$y_{str} = y \qquad 6.1$$

## 6.3 Runnels-Gilman Method Test Results

### 6.3.1 Results of the Saltzman Test Problem

The Runnels-Gilman method implemented in the DiscoverHydro code without the reconciliation forces iteration scheme for the Saltzman problem crashes shortly after t = 0.4μs due to unphysical tangling of the mesh. The mesh becomes increasingly distorted over time as the shock wave propagates and is illustrated in Figure 6.2 after t=0.1μs, Figure 6.3 after t=0.2μs, Figure 6.4 after t=0.3μs, and Figure 6.5 after t=0.4μs.

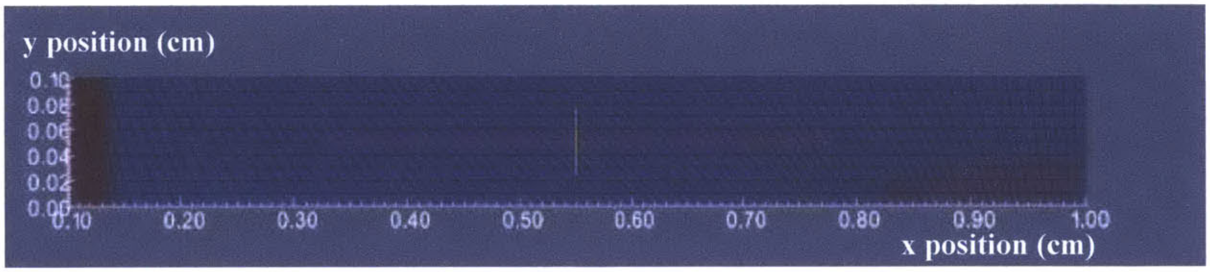


**Figure 6.2 The cell shape and configuration in the mesh after t=0.1μs for the Saltzman planar shock test problem using the Runnels-Gilman hydro method.**

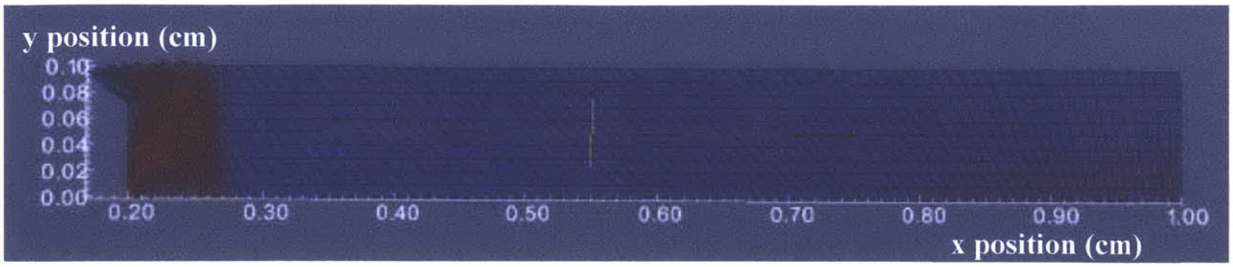


**Figure 6.3 The cell shape and configuration in the mesh after t=0.2μs for the Saltzman planar shock test problem using the Runnels-Gilman hydro method.**

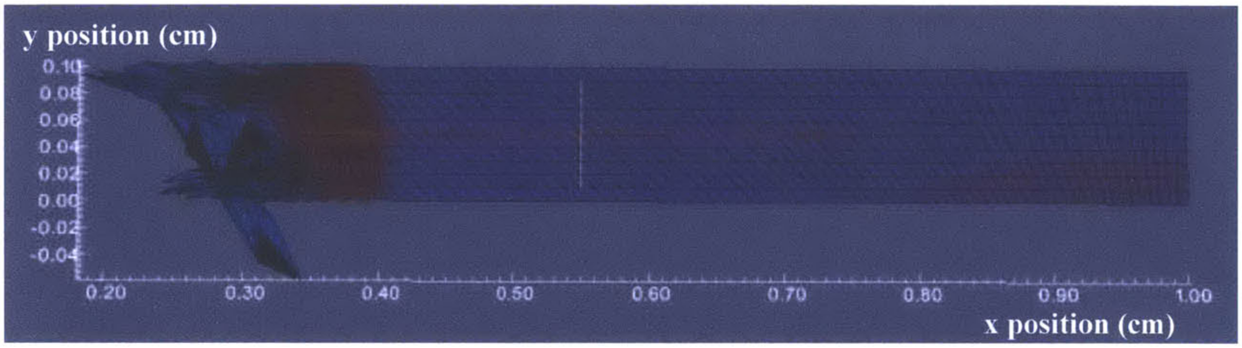


**Figure 6.4 The cell shape and configuration in the mesh after t=0.3μs for the Saltzman planar shock test problem using the Runnels-Gilman hydro method.**

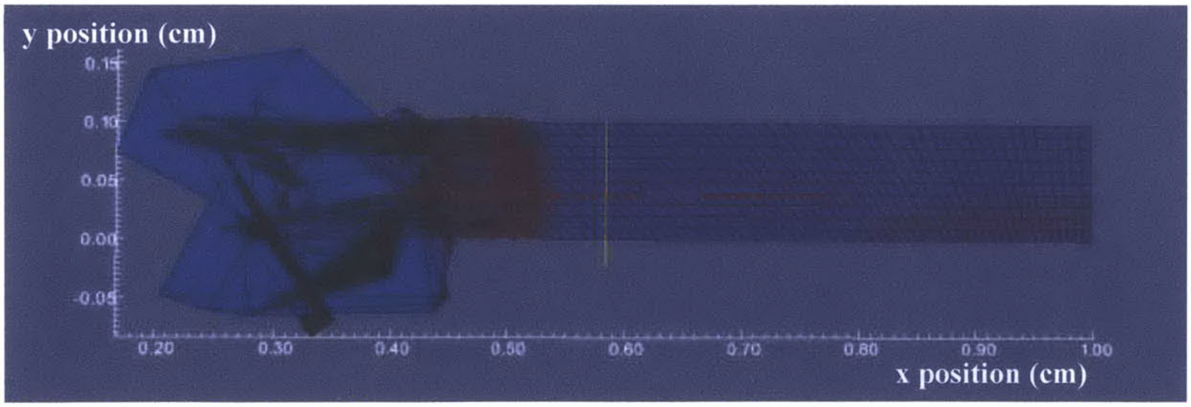


**Figure 6.5 The cell shape and configuration in the mesh after t=0.4μs for the Saltzman planar shock test problem using the Runnels-Gilman hydro method.**

Instead of plotting the results at t=0.6μs as in the piston test problem, the Saltzman test problem results are shown at t=0.4μs since this is just before the Runnels-Gilman hydrocode crashes from grid tangling. At t=0.4μs, the density shows a peak behind the shockwave of ~6.0 $^{gm}/_{cm^3}$, but the density at the shock front is near the analytical value of 4 $^{gm}/_{cm^3}$ and this is seen in Figure 6.6. Figure 6.7 shows the pressure behind the shock has a peak of ~4/3 *Mbar*, but with an average value at the shock front of ~0.9 *Mbar* as compared to the analytical solution of 4/3 *Mbar*. The internal energy values vary widely, from 0.3-34 $^{erg}/_{gm}$, likely due to the high grid distortion. These values are compared to the results from PHM's method and the analytical values in Table 2. The large mesh distortion and tangling with this method may indicate an underconstrained grid.

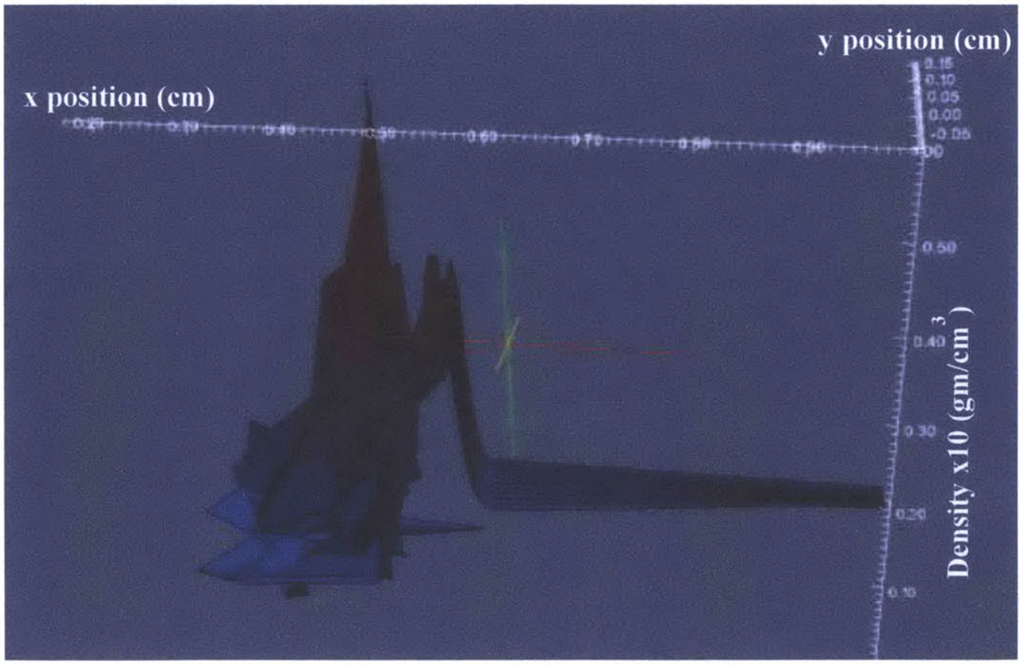


**Figure 6.6 A 100x10 grid with dimensions of [0,1]x[0,0.1] showing the density after t=0.4μs using the Runnels-Gilman method. Note the density is scaled by a factor of 0.1**

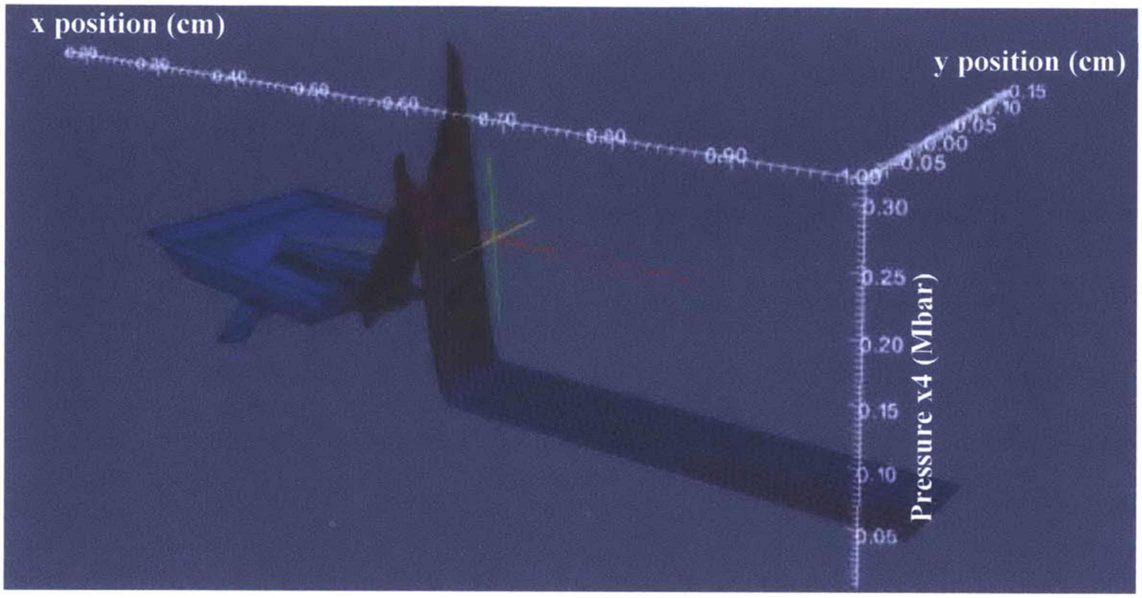


**Figure 6.7 A 100x10 grid with dimensions of [0,1]x[0,0.1] showing the pressure after t=0.4μs using the Runnels-Gilman method. Note the pressure is scaled by a factor of 0.25.**

### 6.3.2 Reconciliation Forces Calculation

The reconciliation forces for the Saltzman test problem using the truncated Runnels-Gilman method (again implemented without using the reconciliation force iteration scheme) were calculated for each iota in the mesh touching an internal vertex for both coordinate directions. The net reconciliation forces for an iota surrounding the vertex were calculated using Equations 3.28 and 3.29. These reconciliation forces acting on the iota were then used to calculate the net reconciliation force on each corner of the four cells that surround a particular vertex as shown in Figure 5.7 and described in Section 5.3.3.

Reconciliation forces are acting in both coordinate directions for this test problem, as is expected since the cell shapes are distorted and the grid is not perfectly aligned with the fluid flow. The reconciliation forces for each of the four corners in both the x and y-directions are shown in Figure 6.8 and Figure 6.9 respectively at t=0.1μs. These figures also illustrate how the sum of the four corner reconciliation forces around a vertex is equal to zero for both coordinate directions. Additionally, small reconciliation forces are calculated behind the shockwave, which is likely due to the unphysical mesh distortion and cell tangling that occurs behind the shock front through the calculation, which can be seen in Figure 6.6.

Figure 6.8 and Figure 6.9 also illustrate the slightly varying values for the reconciliation forces for vertices positioned at approximately the same position on the x-axis. These variations are due to the cell distortion that is performed on the mesh as described in section 6.2 prior to initiation of the shock. Since reconciliation forces are dependent on the length of the sides of the iota, the slight differences in cell sizes and shapes cause the reconciliation forces to vary as well. Also note that the reconciliation forces calculated in this Saltzman test problem are smaller than the Piston test problem due to the smaller cell size.

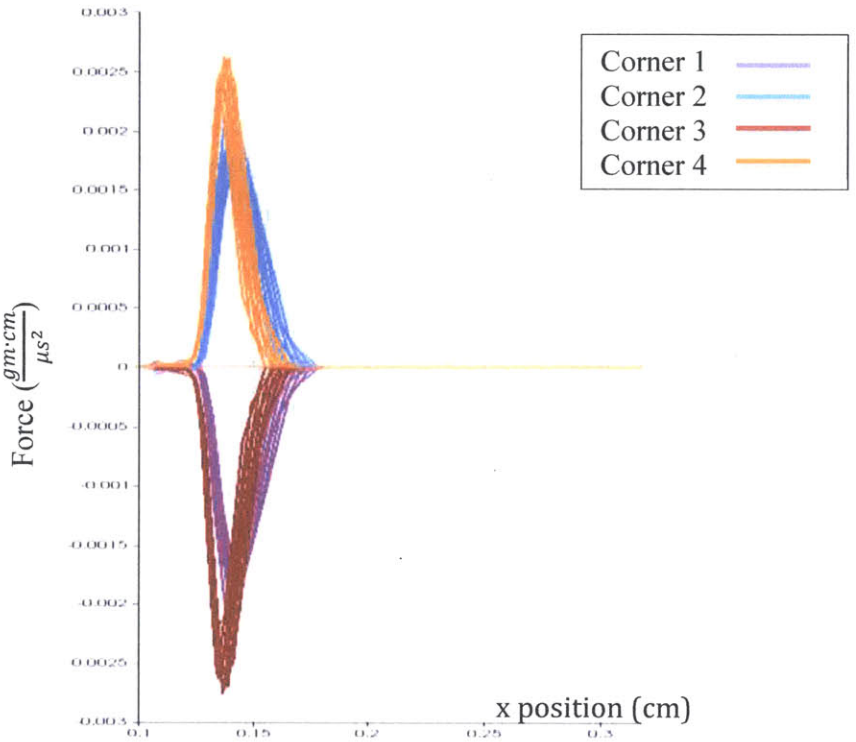


**Figure 6.8 A 100x10 grid with dimensions of [0,1]x[0,0.1] showing the net forces in the x-direction at t=0.1μs for corners 1, 2, 3, and 4 using the Runnels-Gilman method on the Saltzman problem. The vertices affected are those located at the shock front as shown.**

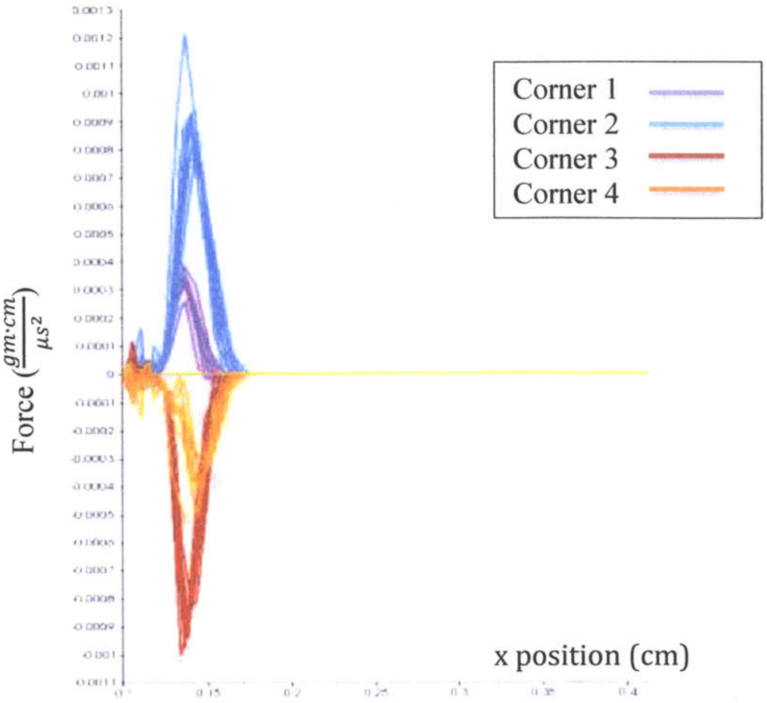


**Figure 6.9 A 100x10 grid with dimensions of [0,1]x[0,0.1] showing the net forces in the y-direction at t=0.1μs for corners 1, 2, 3, and 4 using the Runnels-Gilman method on the Saltzman problem. The vertices affected are those at the shock front.**

## 6.4 PHM Method Test Results

### 6.4.1 Results of the Saltzman Test Problem

The results from the Saltzman test problem using PHM's method implemented in the DiscoverHydro code illustrates the robustness of the method since the simulation can be run for t>0.6μs. At t=0.6μs, the Saltzman test problem provides a density peak of ~4.7 $^{gm}/_{cm^3}$, but the density behind the shockwave oscillates around the analytical solution of 4 $^{gm}/_{cm^3}$ and is shown in Figure 6.13. Also, the pressure behind the shock is ~4/3 *Mbar*, but with a peak at the shock front of ~1.5 *Mbar* as compared to the analytical solution of 4/3*Mbar* and shown in Figure 6.14. The internal energy was also found to vary behind the shock front, with values from 0.5 to 0.67 $^{erg}/_{gm}$ and the higher energies being at the front of the mesh. These results are presented in Table 2 to compare to the Runnels-Gilman method as well. The Saltzman test problem illustrates how cell distortion affects the results of the calculation since these results are not as close to the analytical solution as the piston test problem. Figure 6.13 and Figure 6.14 also agree with the figures published by PHM [5]. The mesh is shown in Figure 6.10 after t=0.2μs, Figure 6.11 after t=0.4μs, and Figure 6.12 after t=0.6μs to illustrate how the cell shapes change through the simulation.

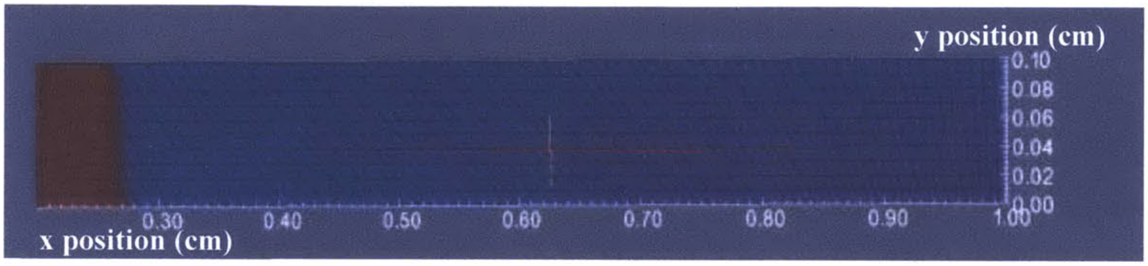


**Figure 6.10 The cell shape and configuration in the mesh after t=0.2μs using PHM's method for the Saltzman planar shock test problem.**

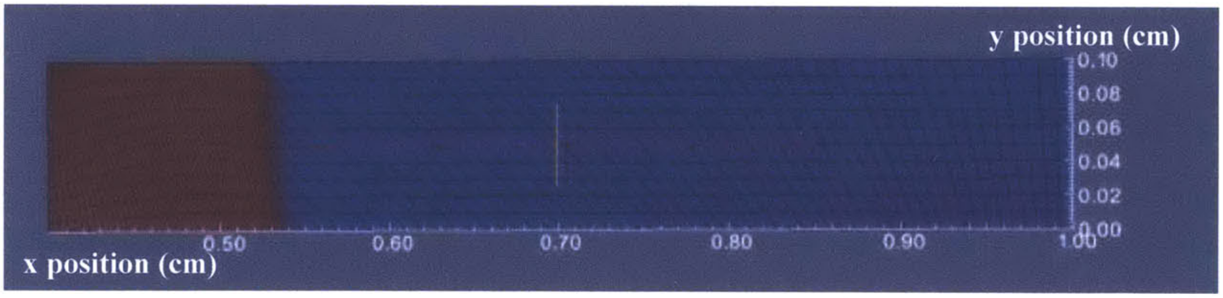


**Figure 6.11 The cell shape and configuration in the mesh after t=0.4μs using PHM's method for the Saltzman planar shock test problem.**

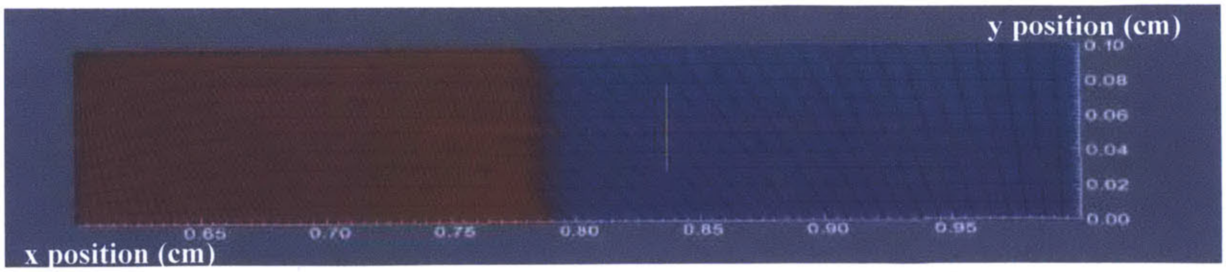


**Figure 6.12 The cell shape and configuration in the mesh after t=0.6μs using PHM's method for the Saltzman planar shock test problem.**

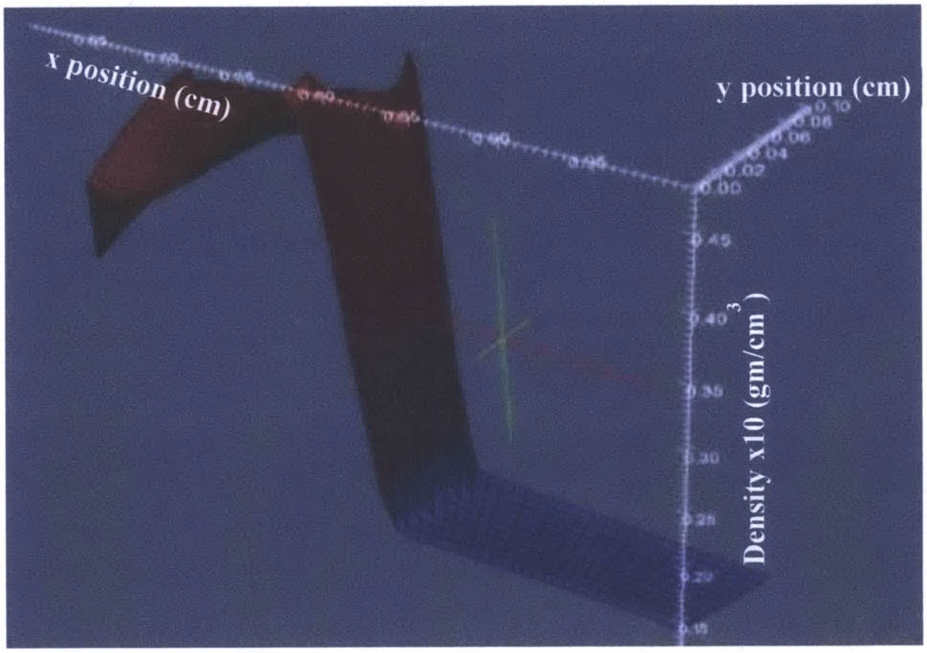


**Figure 6.13 A 100x10 grid with dimensions of [0,1]x[0,0.1] showing the density after t=0.6μs using PHM's method. Note the density is scaled by a factor of 0.1. The density behind the shock oscillates around the analytical solution of 4.**

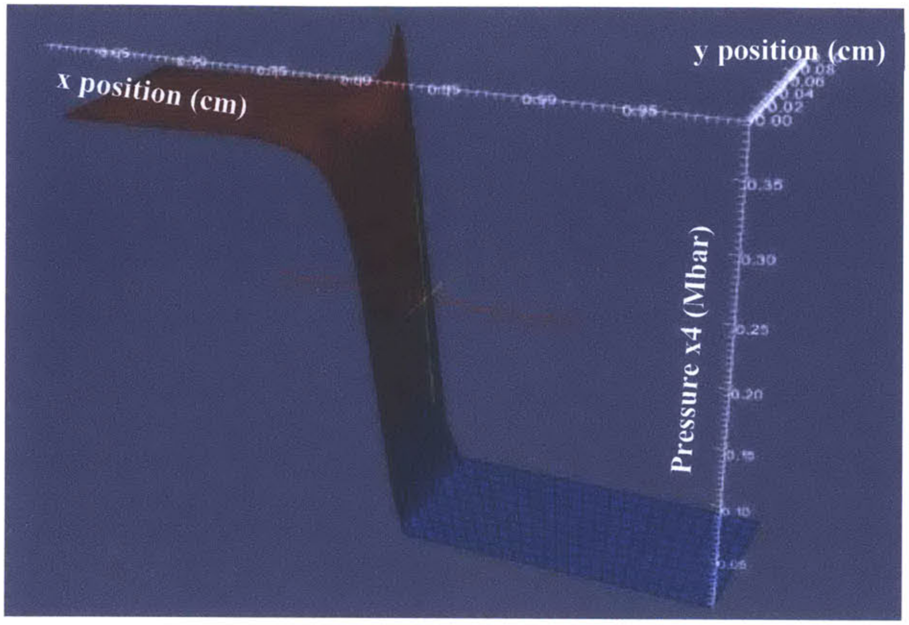


**Figure 6.14 A 100x10 grid with dimensions of [0,1]x[0,0.1] showing the pressure after t=0.6μs using PHM's method. Note the density is scaled by a factor of 0.25. The pressure behind the shock is near the analytical solution of 4/3, but also has a sharp peak at the shock front.**

| **Parameter** | **Analytical Value** | **Runnels-Gilman (Saltzman problem)** | **PHM's Method (Saltzman Problem)** |
|---|---|---|---|
| $\rho^{+}\left[{}^{gm}/_{cm^3}\right]$ | 4 | Peak: 6.0<br>Shock Front: ~4 | Peak: 4.7<br>Shock Front: ~4 |
| $\varepsilon^{+}\left[{}^{erg}/_{gm}\right]$ | ${}^{1}/_{2}$ | 0.3 to 30 | 0.5 to 0.67 |
| $P^{+}\ [Mbar]$ | ${}^{4}/_{3}$ | Peak: ${}^{4}/_{3}$<br>Shock Front: ~0.9 | Peak: 1.5<br>Shock Front: ~ ${}^{4}/_{3}$ |

**Table 2 A comparison of the solutions of the Runnels-Gilman and PHM methods to the analytical solution for the Saltzman test problem.**

### 6.4.2 Reconciliation Forces Calculation

The reconciliation forces for the Saltzman test problem using PHM's method were also calculated for each iota surrounding an internal vertex of the mesh for both coordinate directions using Equations 3.28 and 3.29. The net reconciliation force on each corner of the four cells that surround an internal vertex were also calculated. As expected, the sum of the reconciliation forces of all the iota around each vertex was zero, and like the results seen in the Runnels-Gilman method, each individual corner surrounding a vertex did not always sum to zero.

The reconciliation forces for each of the four corners in both the x and y-directions are shown in Figure 6.15 and Figure 6.16 respectively for t=0.1μs in the simulation. Reconciliation forces are also acting in both coordinate directions for this test problem, as is expected due to cell shape distortion. Also, the smaller cell size, due to the finer mesh, results in smaller reconciliation forces than seen in the Piston test problem. These figures also illustrate how the sum of the four corner reconciliation forces around a vertex equals zero. A significant difference from the Runnels-Gilman hydrocode results is that the reconciliation forces calculated for mesh continuity using PHM's method affect vertices on the cells behind the shock wave and at the shock front.

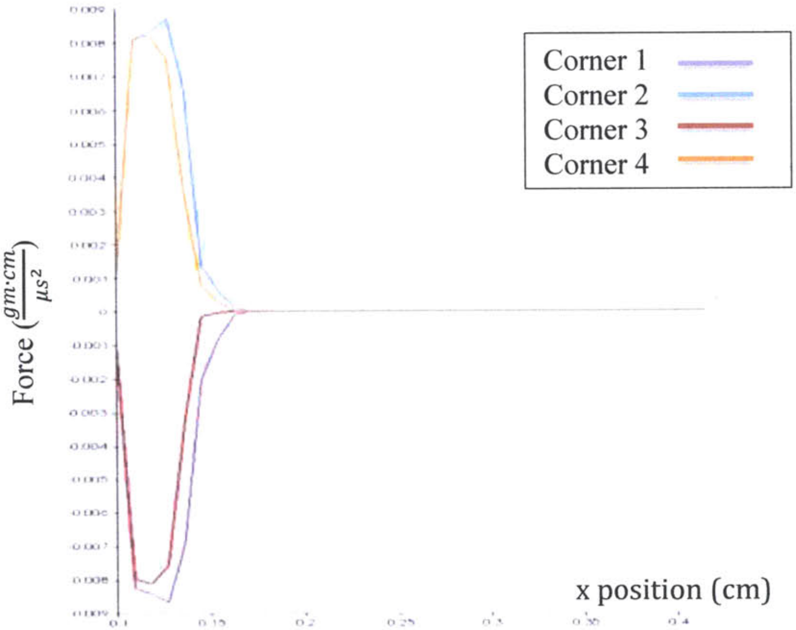


**Figure 6.15 A 100x10 grid with dimensions of [0,1]x[0,0.1] showing the net forces in the x-direction at t=0.1μs for corners 1, 2, 3, and 4 using PHM's method on the Saltzman problem. The vertices affected are those at the shock front and behind the wave.**

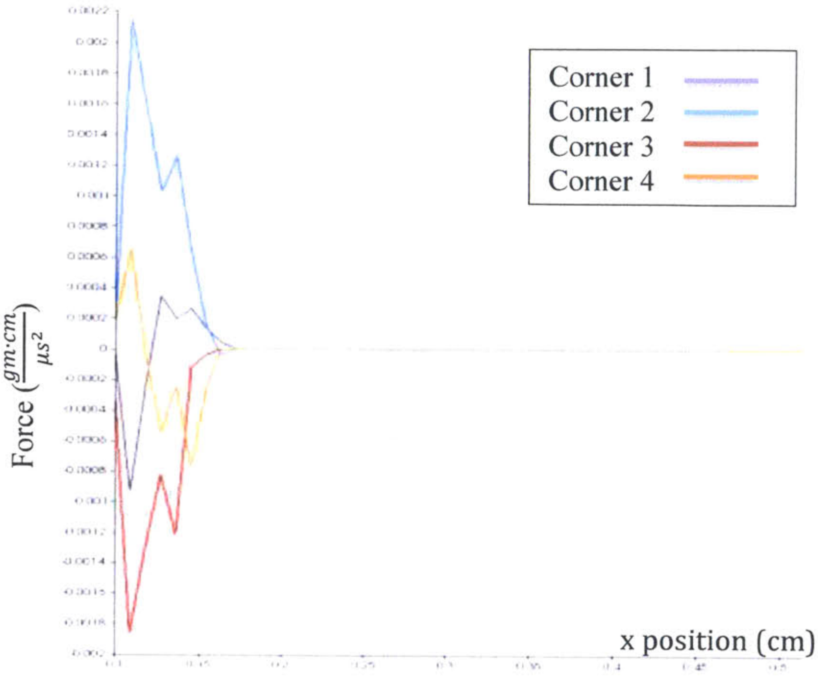


**Figure 6.16 A 100x10 grid with dimensions of [0,1]x[0,0.1] showing the net forces in the y-direction at t=0.1µs for corners 1, 2, 3, and 4 using PHM's method on the Saltzman problem. The vertices affected are those at the shock front and behind the wave.**

To investigate the results at the end of the simulation, the reconciliation forces at t=0.6µs were also plotted. Figure 6.17 and Figure 6.18 show the net forces on corners 1, 2, 3, and 4 at t=0.6µs for the x and y-coordinate directions respectively to demonstrate that the sum of the forces in all four corners is equal to zero for each vertex. Again, note the difference in the reconciliation forces calculated for the mesh as compared to the Runnels-Gilman method. In PHM's method, the vertices affected by reconciliation forces are those both located at the shock front and the behind the shock wave as well.

Figure 6.17 and Figure 6.18 also illustrate the different values for the reconciliation forces around vertices, even at same position along the x-axis in the mesh. This is due to the cell distortion that is performed on the mesh as described in section 6.2. The values for the reconciliation forces are dependent on the lengths of the iota sides, so the slight differences in cell sizes and shapes cause the forces to be slightly different on each iota.

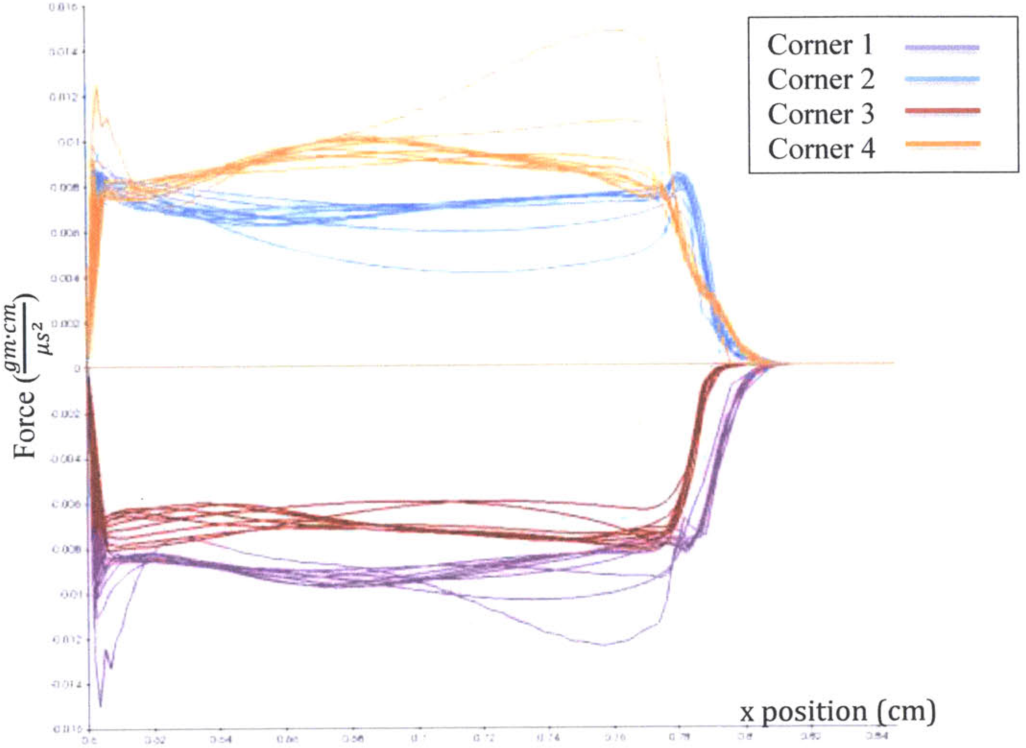


**Figure 6.17 A 100x10 grid with dimensions of [0,1]x[0,0.1] showing the net forces in the x-direction at t=0.6μs for corners 1, 2, 3, and 4 using PHM's method on the Saltzman problem. The vertices affected are those at the shock front and behind the wave.**

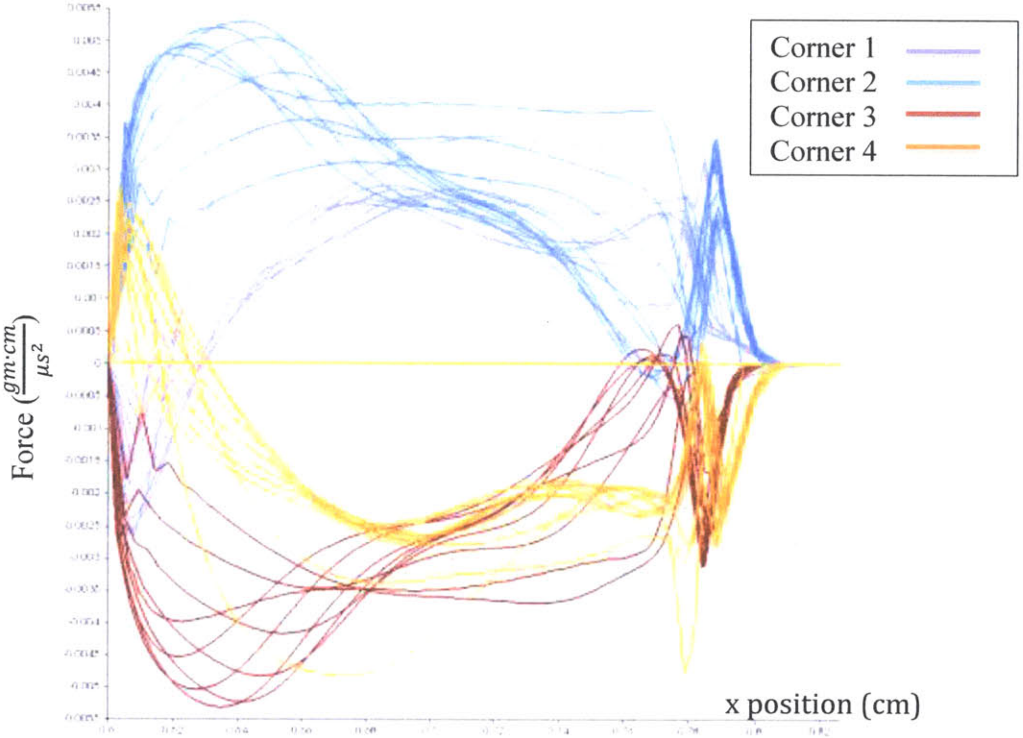


**Figure 6.18 A 100x10 grid with dimensions of [0,1]x[0,0.1] showing the net forces in the y-direction at t=0.6µs for corners 1, 2, 3, and 4 using PHM's method on the Saltzman problem. The vertices affected are those at the shock front and behind the wave.**

# 7 Conclusions

To understand the difference between the one-dimensional Riemann forces at cell interfaces and the additional forces that are included in the final forces computed by PHM, a reconciliation forces solver was developed to determine the additional forces not explicitly calculated in both PHM's method and the Runnels-Gilman method for maintaining mesh continuity. This solver was used to explore the reconciliation forces that are at work for two test problems: the original 1D Piston test problem and the Saltzman test problem.

It was determined that the 16 equations produced from using the change in linear momentum for each iota in the two coordinate directions produced a singular matrix, where two rows were linear combinations of the remaining 14 equations. Therefore, to complete the matrix, two additional closure relations are used that enforces that the intercellular normal forces add up to zero in both the x and y-directions. These values are already computed by the Riemann solver, so the reconciliation forces in addition to the Riemann solution should add to zero. This creates a non-singular matrix that was used to solve for the continuity forces. Therefore, the reconciliation forces calculated are forces acting in addition to the Riemann solution.

The findings from this investigation are summarized below:

1D Piston Test Problem:

- Both the Runnels-Gilman and PHM hydro methods calculated the density, pressure, and internal energy values to be near the analytical values for the test problem.
- In the truncated Runnels-Gilman method (without the reconciliation forces iteration scheme), the reconciliation forces were present in the x-direction for each of the four

individual corners of cells surrounding an internal vertex of the mesh. These reconciliation forces were found to be only acting at the shock front.

- PHM's method showed reconciliation forces in the x-direction for each of the four individual corners of cells surrounding an internal vertex. These reconciliation forces were acting both at the shock front and behind the shockwave. PHM's method produced a density value closer to the true analytical value and with less wall heating than the truncated Runnels-Gilman method. The reconciliation forces may be capturing the additional forces producing a more stable density solution to occur as the shock wave propagates through the mesh.

Saltzman Test Problem:

- The truncated Runnels-Gilman method crashes before the simulation reaches t=0.6μs, and the results were analyzed at t=0.4μs. At this time, there is significant grid tangling behind the shockwave and the solution has values near the analytical solution for the pressure and density while the internal energy value varies significantly (the high values for internal energy were located behind the shock wave where the grid was tangled).
- The truncated Runnels-Gilman method calculated reconciliation forces present in both the x and y-directions for each of the four individual corners of cells surrounding an internal vertex of the mesh. These reconciliation forces were again found to be only acting at the shock front.
- PHM's method ran the simulation through t=0.6μs and results were near the analytical values, but not as accurate as the 1D Piston test problem.
- PHM's method showed reconciliation forces in both the x and y-directions for each of the four individual corners of cells surrounding an internal vertex. These reconciliation forces

were acting both at the shock front and behind the shock wave. The PHM method was robust and produced internal energy and density values closer to the true analytical value. Compared to the high-degree of grid tangling that the Runnels-Gilman produced behind the shockwave, PHM's method had a low amount of cell tangling. The reconciliation forces may be capturing the additional forces that result in a more stable density and internal energy solution behind the shock wave and producing decreased grid tangling.

# 8 Future Work

## 8.1 Additional Study of Reconciliation Forces

Reconciliation forces should be studied further in additional verification test cases using different grid configurations, coordinate systems, and initial conditions. These would be able to examine use of current hydro codes to determine if they uphold the conservation laws.

An example test problem to further analyze reconciliation forces is the Multimaterial Sod's Shock Tube Problem. In this problem, a shock tube contains two separate regions, each having different densities and pressures, typically one of high pressure and density and one with low pressure and density. Initially, these regions are in a constant state with both fluids at rest. At a time t>0, the partition separating the two fluids is broken and the subsequent wave is simulated. In Lagrangian schemes, the density is usually below the analytical calculation while the internal energy is greater [44]. This makes the calculation of reconciliation forces for this problem of interest because it has the potential to indicate why the internal energy is greater than it should be.

The Noh test problem introduces a new verification technique by using a cylindrical geometry rather than the more common rectangular grid. Radial symmetry is preserved by meshing the cylinder using triangles near the center and quadrangles throughout the rest of the geometry. The test case is the implosion of this cylinder with a unit radius [5]. This would enable another test for continuity forces in a grid using a different coordinate system.

In Kidder's test case, a cylindrical shell grid is simulated for isentropic compression with a perfect gas [5]. This test problem was developed for hollow shell compression to further study inertial confinement fusion, and accordingly the analytical solution has been calculated for comparison. It has been developed so that it can be tested for problems having either planar, cylindrical or spherical symmetry [45]. This type of test case is also believed to contain numerical diffusion observed by the change in the answer to the calculation due to higher mesh refinement, which makes it a case of interest in the study of additional forces that may not be accounted for [5].

A more recently proposed test case involves a perturbation of the isentropic Kidder compression. This tests the hydrocode ability to reproduce the growth of hydrodynamic instabilities. There is an approximate analytical solution for perturbations amplified in the linear regime, that was first developed in the study of imploding plasma shells [46]. Testing a hydrocode method's ability to uphold the conservation laws while testing mesh perturbations is of interest since this simulation has a very small amplification over time [47], [5].

## 8.2 Runnels-Gilman Hydro Scheme Development

The completion of the implementation of the Runnels-Gilman method requires further development as evidenced by the results of the Saltzman test problem. The reconciliation forces appear to be a substantial and necessary component in providing an accurate solution, as evidenced by the inaccurate solutions when they are not included. The inability of the current code, to calculate a shockwave to t=0.6μs using a mesh containing distorted cells illustrates the fact that the scheme may need to incorporate additional grid constraints. By implementing the reconciliation forces that are calculated as additional forces into the Runnels-Gilman method, the development of a new hydro scheme should be pursued.

As discussed earlier, the mesh quality can be improved by implementing the rezoning and remapping procedure known as arbitrary Lagrangian-Eulerian (ALE). Specifically, employing this method in the Saltzman problem when cells begin to entangle would likely improve results and allow the simulation to extend to t=0.6μs and beyond. Future work to implement standard ALE schemes into the methods presented here should be pursued.

# 9 References

[1] J. D. Lindl, Inertial Confinement Fusion, New York, 1998.

[2] S. Weber, et al., "Modelling of Laser-Plasma Interaction on Hydrodynamic Scales: Physics Development and Code Validation," 2003.

[3] B. Einfeldt, "On Godunov-Type Methods for Gas Dynamics," *Siam Journal on Numerical Analysis,* vol. 25, pp. 294-318, 1988.

[4] D. J. Benson, "Computational methods in Lagrangian and Eulerian hydrocodes," *Computational Methods in Applied Mechanics and Engineering,* no. 99, p. 235–394, 1992.

[5] P.-H. Maire, "A cell-centered Lagrangian Scheme for Two-Dimensional Compressible Flow Problems," *SIAM Journal on Scientific Computing,* vol. 29, no. 4, 2007.

[6] P.-H. Maire, "A High-Order Cell-Centered Lagrangian Scheme for Compressible Fluid Flows in Two-Dimensional Cylindrical Geometry," *Journal of Computational Physics,* vol. 228, pp. 6882-6915, 2009.

[7] R. D. Richtmyer and K. W. Morton, Difference Methods for Initial-Value Problems, New York: Interscience Publishers, 1967.

[8] S. Runnels and L. Gilman, "Superposition- and Reconciliation-Based Cell Centered Hydro Method," LA-UR-06002, 2011.

[9] W. F. Noh, "Noh," *Journal of Computational Physics,* vol. 78, p. 72, 1987.

[10] J. Saltzman and C. P., "LA-UR-85-678," Los Alamos National Laboratory, Los Alamos, NM, 1985.

[11] S. P. Schofeld, R. V. Garimella, M. M. Francois and R. Loubere, "A second-order accurate material-order-independent interface reconstruction technique for multi-material flow simulations," *Journal of Computational Physics,* vol. 228, pp. 731-745, 2009.

[12] J. VonNeumann and R. D. Richtmyer, "A method for the numerical calculations of hydrodynamical shocks," *Journal of Applied Physics,* no. 21, pp. 232-238, 1950.

[13] T. J. R. Hughes, W. K. Liu and T. K. Zimmermann, "Lagrangian-Eulerian Finite Element Formulation for Incompressible Viscous Flows," *Computational Methods in Applied Mechanics and Engineering,* vol. 29, pp. 329-349, 1981.

[14] I. G. Cameron, "An Analysis of the Errors Caused by Using Artificial Viscosity Terms to Represent Steady-State Shock Waves," *Journal of Computational Physics,* vol. 1, pp. 1-20, 1966.

[15] D. L. Book, J. Boris and K. Hain, "Flux-Corrected Transport. II. Generalizations of the Method," *Journal of Computational Physics,* vol. 18, no. 2, pp. 248-83, 1975.

[16] C. Hirt, A. A. Amsden and J. L. Cook, "An Arbitrary Lagrangian-Eulerian Computing Method for All Flow Speeds," *Journal of Computational Physics,* vol. 14, pp. 227-253, 1974.

[17] P.-H. Maire and B. Nkonga, "Multi-scale Godunov-type method for cell-centered discrete Lagrangian hydrodynamics," *Journal of Computational Physics,* vol. 228, pp. 799-821,

2009.

[18] R. Loubere and M. J. Shashkov, "A subcell remapping method on staggered polygonal grids for arbitrary-Lagrangian-Eulerian methods," *Journal of Computational Physics,* vol. 209, no. 1, pp. 105-138, 2005.

[19] Z. J. Wang, "Spectral (Finite) Volume Method for Conservation Laws on Unstructured Grids," *Journal of Computational Physics,* vol. 178, pp. 210-251, 2002.

[20] Z. J. Wang and Y. Liu, "Spectral (Finite) Volume Method for Conservation Laws on Unstructured Grids II. Extension to Two-Dimensional Scalar Equation," *Journal of Computational Physics,* vol. 179, pp. 665-697, 2002.

[21] Y. Liu, M. Vinokur and Z. J. Want, "Spectral (finite) Volume Method for Conservation Laws on Unstructured Grids V: Extension to Three-Dimensional Systems," *Journal of Computational Physics,* vol. 212, pp. 454-472, 2006.

[22] M. L. Wilkins, "Use of Artificial Viscosity in Multidimensional Fluid Dynamics Calculations," *Journal of Computational Physics,* vol. 36, pp. 281-303, 1980.

[23] J. C. Campbell and M. J. Shashkov, "A Tensor Artificial Viscosity Using a Mimetic Finite Difference Algorithm," *Journal of Computational Physics,* vol. 172, pp. 739-765, 2001.

[24] J. K. Dukowicz and B. J. A. Meltz, "Vorticity Errors in Multidimensional Lagrangian Codes," *Journal of Computational Physics,* vol. 99, pp. 115-134, 1992.

[25] A. L. Bauer, D. E. Burton, E. J. Caramana, R. Loubere, M. J. Shashkov and P. P. Whalen,

"The Internal Consistency, Stability, and Accuracy of the Discrete, Compatible Formulation of Lagrangian Hydrodynamics," *Journal of Computational Physics,* vol. 218, pp. 572-593, 2006.

[26] P. Maire, R. Loubere and P. Vachal, "Staggered Lagrangian Discretization Based on Cell-Centered Riemann Solver Associated Hydrodynamics Scheme," *Communications in Computational Physics,* vol. 10, no. 4, pp. 940-978, 2011.

[27] P. Whalen, "Algebraic limitations on two dimensional hydrodynamics simulations," *J. Comput. Phys.,* no. 124, pp. 46-54, 1996.

[28] M. L. Wilkins, "Calculation of Elastic-Plastic Flow," *Methods in Computational Physics,* vol. 3, pp. 211-263, 1964.

[29] W. Hui, P. Y. Li and Z. W. Ki, "A Unified Coordinate System for Solving the Two-Dimensional Euler Equations," *Journal of Computational Physics,* vol. 153, no. 2, p. 596–637, 1999.

[30] R. Loubere, J. Ovadia and R. Abgrall, "A Lagrangian Discontinuous Galerkin-type method on unstructured meshes to solve hydrodynamics problems," *International Journal for Numberical Methods in Fluids,* vol. 44, pp. 645-663, 2004.

[31] E. J. Caramana and R. Loubere, "'Curl-q': A vorticity damping artificial viscosity for essentially irrotational Lagrangian hydrodynamics calculations," *Journal of Computational Physics,* 2005.

[32] E. J. Caramana, D. E. Burton, M. J. Shashov and P. P. Whalen, "The construction of compatible hydrodynamics algorithms utilizing conservation of total energy," *Journal of Computational Physics,* no. 146, pp. 227-276, 1998.

[33] E. Caramana and M. J. Shashkov, "Elimination of Artificial Grid Distortion and Hourglass-Type Motions by Means of Lagrangian Subzonal Masses and Pressures," *Journal of Computational Physics,* vol. 142, p. 521–561, 1998.

[34] K. Lipnikov and M. Shashkov, "A mimetic tensor artificial viscosity method for arbitrary polyhedral meshes," *International Conference on Computational Science,* pp. 1915-1923, 2010.

[35] W. F. Noh, "Errors for Calculations of Strong Shocks Using an Artificial Viscosity and an Artificial Heat Flux," *Journal of Computational Physics,* vol. 72, pp. 78-120, 1978.

[36] R. Loubere, P.-H. Maire and P. Vachal, "A second-order compatible staggered Lagrangian hydrodynamics scheme using a cell-centered multidimensional approximate Riemann solver," *International Conference on Computational Science,* pp. 1925-1933, 2010.

[37] J. K. Dukowicz, M. C. Cline and F. L. Addessio, "A General Topology Godunov Method," *Journal of Computational Physics,* vol. 82, pp. 29-63, 1989.

[38] B. Despres and C. Mazeran, "Lagrangian gas dynamics in two dimensions and Lagrangian Systems," *Arch. Rational Mech. Anal.,* vol. 178, pp. 327-372, 2005.

[39] P.-H. Maire, "A High-Ordered Cell-Centered Lagrangian Scheme for Two-Dimensional Compressible Fluid Flows on Unstructured Meshes," *Journal of Computational Physics,*

vol. 228, pp. 2391-2425, 2009.

[40] P. L. Roe, "Approximate Riemann Solvers, Parameter Vectors, and Difference Schemes," *Journal of Computational Physics,* vol. 43, pp. 357-372, 1981.

[41] L. E. Malvern, Introduction to the Mechanics of a Continuous Medium, Englewood Cliffs, NJ: Prentice-Hall, Inc., 1969.

[42] H. M. Glaz, P. Colella, I. I. Glass and R. L. Deschambault, "A Numerical Study of Oblique Shock-Wave Reflections with Experimental Comparisons," *Proceedings of the Royal Society London A,* vol. 398, pp. 117-140, 1985.

[43] W. J. Rider, "Revisiting Wall Heating," *Journal of Computational Physics,* vol. 162, pp. 395-410, 2000.

[44] G. A. Sod, "A Survey of Several Finite Difference Methods for Systems of Nonlinear Hyperbolic Conservation Laws," *Journal of Computational Physics,* vol. 27, pp. 1-31, 1978.

[45] R. Kidder, "Laser-Driven Compression of Hollow Shells: Power Requirements and Stability Limitations," *Nuclear Fusion,* vol. 16, pp. 3-14, 1976.

[46] S. J. Han and B. R. Suydam, "Hydrodynamic Instabilities in an Imploding Cylindrical Plasma Shell," *Physical Review A,* vol. 26, pp. 926-939, 1982.

[47] P.-H. Maire, J. Briel, L. Hallo, Olazabal-Loume and M., "Hydrodynamic Instabilities in Cylindrical Geometry. Self-Similar Models and Numberical Simulations," in *Proceedings of the 31st EPS Conference on Plasma Physics*, London, 2004.

[48] M. Shashkov and B. Wendroff, "A Composite Scheme for Gas Dynamics in Lagrangian Coordinates," *Journal of Computational Physics,* vol. 150, pp. 502-517, 1999.

[49] R. Liska and B. Wendroff, "Composite Schemes for Conservation Laws," *SIAM Journal on Numerical Analysis,* vol. 35, pp. 2250-2271, 1998.