5-2020

Hydrodynamic Instability Simulations Using Front-Tracking with Higher-Order Splitting Methods

Dillon Trinh

Hydrodynamic Instability Simulations Using Front-Tracking with Higher-Order Splitting Methods

An Honors Thesis submitted in partial fulfillment of the requirements of

Honors Studies in Mathematics

By

Dillon Trinh

Spring 2020

Mathematics

J. William Fulbright College of Arts and Sciences

The University of Arkansas

Acknowledgements

This research was funded in part by the University of Arkansas Honors College. This work used the Extreme Science and Engineering Discovery Environment (XSEDE), which is supported by National Science Foundation grant number ACI-1548562. Specifically, it used the Bridges system at the Pittsburgh Supercomputing Center (PSC), which is supported by NSF award number ACI-1445606. The PSC resource was used to run simulations for this project. I would like to give a special thanks to Dr. Tulin Kaman, assistant professor, Lawrence Jesser Toll Jr. Endowed Chair in the Department of Mathematical Sciences, and the faculty advisor of SIAM and AWM Student Chapters at the University of Arkansas, for mentoring me through this project. I would also like to thank the committee members for the defense of this thesis for their time.

Table of Contents

Abstract

The Rayleigh-Taylor Instability (RTI) is an instability that occurs at the interface of a lighter density fluid pushing onto a higher density fluid in constant or time-dependent accelerations. The Richtmyer-Meshkov Instability (RMI) occurs when two fluids of different densities are separated by a perturbed interface that is accelerated impulsively, usually by a shock wave. When the shock wave is applied, the less dense fluid will penetrate the denser fluid, forming a characteristic bubble feature in the displacement of the fluid. The displacement will initially obey a linear growth model, but as time progresses, a nonlinear model is required. Numerical studies have been performed in the past to accurately approximate this nonlinear model. A techniques called front tracking has provided an enhanced resolution and zero numerical diffusion that is helpful with the sharp discontinuities of the fluid properties in simulations involving RTI and RMI. Weighted essentially non-oscillatory (WENO) finite difference schemes are used for accurate and precise results in both early and late time of fluid mixing simulations. In more traditional projects, WENO schemes utilized Lax-Friedrichs flux splitting. However, an alternative type of splitting developed by Gilbert Strang splits a two-dimensional problem into two one-dimensional problems that are easier and faster to solve. His splitting method was shown to achieve up to second-order accuracy. For this research, such a splitting method was derived for higher-order accuracy in three-dimensional problems. RTI simulations utilizing this newly derived model were used to incorporate front tracking technique, WENO, and operator splitting in a way that has not been done for a three-dimensional problem.

1 Introduction

1.1 Rayleigh-Taylor Instability and Richtmyer-Meshkov Instability

The Rayleigh-Taylor Instability (RTI) occurs at the interface of a lighter density fluid pushing onto a higher density fluid in constant or time-dependent accelerations. Richtmyer-Meshkov Instability (RMI) occurs when two fluids of different densities are separated by a perturbed interface that is accelerated impulsively, usually by a shock wave. As a result, when the shock wave is applied, the lighter fluid will penetrate the heavier fluid, forming a characteristic bubble feature in the displacement of the fluid. The displacement will initially obey a linear growth model, but as time progresses, a nonlinear model is required [9].

In Zhou's (2017) paper, all relevant experiments regarding RTI and RMI are reviewed. Lewis performed an RTI experiment and saw an approximately sinusoidal initial perturbation to the interface by oscillating the blade of a small paddle near the interface. The RTI was generated by accelerating liquids down a vertical tube using air pressure. Emmons, et al., improved Lewis' experiment by using a thin tank constrained by guide rails. They also accelerated the fluids using a stretched rubber tubing. Similarly, Cole and Tankin used a double-ended air cylinder driven by compressed air to propel a tank containing air and water. Most experiments for RMI were accomplished using a thin plate and withdrawing it. Meshkov was the first to use a nitrocellulose membrane in his RMI experiments. This membrane was already shaped into a sinusoid to provide the perturbation. However, the incident shock wave shattered the membrane, and the fragments were carried with the flow. Bouzgarrou, et al., described the fragments as likely to corrupt the optical measurements and would also interact with the flow. Jones and Jacobs developed a membrane free approach in 1997. Instead of a membrane, they used a vertical shock tube with horizontal slots for the fluids to enter. This was found to form a stagnation point flow at the interface, and a reproducible perturbation was applied by disturbing the shock tube to generate a standing wave [9].

1.2 Front Tracking

Numerical simulations of RTI and RMI are helpful to investigate and analyze the dynamics of the growth exponent that describes the outer edge of the mixing zone. One of the first RMI studies was performed by Holmes, Grove, and Sharp in 1995. They measured the growth rates of RMI through the use of front tracking. When two fluids are subjected to an interface like that of RTI and RMI, there are sharp discontinuities of some fluid properties such as density and pressure. Front tracking is an adaptive computational tool which provides enhanced resolution and zero numerical diffusion in simulations of the shock acceleration perturbation by tracking the interface. Holmes [5] and Glimm [3] found that simulations involving front tracking exhibited a significant improvement in agreement compared to those performed without front tracking. Front tracking eliminates numerical diffusion that occurs in Eulerian codes without front tracking. Front-tracking along with subgrid scale modeling (SGS) for Large eddy simulations (LES) is used for modeling turbulent flows for predictions of turbulence behavior [4].

1.3 WENO Scheme

The numerical analysis by Chi-Wang Shu was important in improving accuracy in computational fluid dynamics (CFD). The methods used to analyze Shu's numerical studies in CFD were high order finite difference weighted essentially non-oscillatory (WENO), finite volume WENO, and discontinuous Galerkin method. Numerical studies with the WENO scheme yielded accurate and precise results in both early and late time of fluid mixing. This method was advantageous in that it did not contain a tuning parameter. Although this method complicates the code required to carry out experiments, it yielded a more robust and accurate result than the second order total variation diminishing (TVD) scheme on the same mesh. The purpose of the TVD schemes is to capture sharper shock predictions without any misleading oscillations common with higher-order schemes. At early times, second order TVD schemes were found to yield acceptable data, but as time progressed, they did not fit the suggested model of flow. Thus, finite difference WENO schemes were used for the numerical solution of Euler equations [8].

1.4 Euler Conservation Equations

The two-dimensional Euler Conservation equation is given as

$$\frac{\partial \boldsymbol{U}}{\partial t} + \frac{\partial \boldsymbol{F}}{\partial x} + \frac{\partial \boldsymbol{G}}{\partial y} = 0 \tag{1}$$

and the three-dimensional Euler Conservation equation is given as

$$\frac{\partial \boldsymbol{U}}{\partial t} + \frac{\partial \boldsymbol{F}}{\partial x} + \frac{\partial \boldsymbol{G}}{\partial y} + \frac{\partial \boldsymbol{H}}{\partial z} = 0 \tag{2}$$

where

$$\boldsymbol{U} = \begin{pmatrix} \rho \\ \rho u_1 \\ \rho u_2 \\ \rho u_3 \\ E \end{pmatrix}, \quad \boldsymbol{F} = \begin{pmatrix} \rho u_1 \\ \rho u_1^2 \\ \rho u_1 u_2 \\ \rho u_1 u_3 \\ (E+p)u_1 \end{pmatrix} \quad \boldsymbol{G} = \begin{pmatrix} \rho u_2 \\ \rho u_1 u_2 \\ \rho u_2^2 \\ \rho u_2 u_3 \\ (E+p)u_2 \end{pmatrix} \quad \boldsymbol{H} = \begin{pmatrix} \rho u_3 \\ \rho u_1 u_3 \\ \rho u_2 u_3 \\ \rho u_3^2 \\ (E+p)u_3 \end{pmatrix}$$

where ρ is density, $\boldsymbol{u} = (u_1\ u_2\ u_3)$ are velocities, E is the total energy per unit volume, and p is pressure. The notation is simplified by letting $X = \frac{\partial \boldsymbol{F}}{\partial x}$, $Y = \frac{\partial \boldsymbol{G}}{\partial y}$, $Z = \frac{\partial \boldsymbol{H}}{\partial z}$. The derivative of x is what Shu sought to approximate with finite difference WENO using Lax-Friedrichs flux splitting [8]. However, this research deals with a different type of splitting method.

2 Strang Splitting

Splitting methods have been developed to split the higher dimension problems into single dimension equations. MacNamara and Strang [6] introduce the ordinary differential equation (ODE)

$$\frac{du}{dt} = (\boldsymbol{A} + \boldsymbol{B})u \tag{3}$$

which has a solution of the form $u(t) = u(0)e^{t(\boldsymbol{A}+\boldsymbol{B})}$. A first order splitting scheme is shown where

$$e^{t(\boldsymbol{A}+\boldsymbol{B})} \approx e^{t\boldsymbol{A}}e^{t\boldsymbol{B}}. \tag{4}$$

This is said to be first order splitting because upon inspection of the local error, the Taylor series

$$e^{t(\boldsymbol{A}+\boldsymbol{B})} = I + t(\boldsymbol{A}+\boldsymbol{B}) + \frac{1}{2!}t^2(\boldsymbol{A}+\boldsymbol{B})^2 + \ldots \tag{5}$$

agrees with the approximation of $e^{t\boldsymbol{A}}e^{t\boldsymbol{B}}$ only up to the first order term. Strang developed a symmetric splitting where

$$e^{t(\boldsymbol{A}+\boldsymbol{B})} \approx e^{\frac{1}{2}t\boldsymbol{A}}e^{t\boldsymbol{B}}e^{\frac{1}{2}t\boldsymbol{A}} \tag{6}$$

is seen to be second order accurate with Equation (5). In context of this research, a second order symmetric Strang splitting would have the form $X^{\frac{1}{2}t}Y^{t}X^{\frac{1}{2}t}$ in two dimensions and $X^{\frac{1}{2}t}Z^{\frac{1}{2}t}Y^{t}Z^{\frac{1}{2}t}X^{\frac{1}{2}t}$ in three dimensions.

2.1 Splitting Methods in Two Dimensions

2.1.1 Second-Order Strang Splitting

For the two-dimensional Euler Conservation equation (1), the solution $\boldsymbol{U}(x, y, t)$ to the problem

$$\begin{aligned} &\frac{\partial \boldsymbol{U}}{\partial t} + \frac{\partial \boldsymbol{F}}{\partial x} + \frac{\partial \boldsymbol{G}}{\partial y} = 0 \qquad && (x, y) \in \mathbb{R}^2,\ t > 0 \\ &\boldsymbol{U}(x, y, 0) = \boldsymbol{U}_0(x, y) \qquad && (x, y) \in \mathbb{R}^2 \end{aligned} \tag{7}$$

where $\boldsymbol{U}_0(x,y)$ is given, can be approximated using splitting methods. For the case of the second-order Strang splitting, first approximate the x-sweep by fixing $y \in \mathbb{R}$ and solving

$$\frac{\partial \boldsymbol{U}}{\partial t} + \frac{\partial \boldsymbol{F}}{\partial x} = 0, \qquad \boldsymbol{U}(x,y,t^n) = \boldsymbol{U}^n(x,y), \qquad x \in \mathbb{R},\ t \in \left(t^n, t^{n+\frac{1}{2}}\right] \tag{8}$$

for $\tilde{\boldsymbol{U}}^{n+\frac{1}{2}}(x,y)$ where $t^{n+\frac{1}{2}} = t^n + \frac{\Delta t}{2}$. This is then used as the initial condition for the y-sweep where $x \in \mathbb{R}$ is then fixed to solve

$$\frac{\partial \boldsymbol{U}}{\partial t} + \frac{\partial \boldsymbol{G}}{\partial y} = 0, \qquad \boldsymbol{U}(x,y,t^n) = \tilde{\boldsymbol{U}}^{n+\frac{1}{2}}(x,y), \qquad y \in \mathbb{R},\ t \in \left(t^n, t^{n+1}\right] \tag{9}$$

for $\tilde{\tilde{\boldsymbol{U}}}^{n+1}(x,y)$ where $t^{n+1} = t + \Delta t$. The splitting is completed by solving the rest of the x-sweep where again $y \in \mathbb{R}$ is fixed to solve

$$\frac{\partial \boldsymbol{U}}{\partial t} + \frac{\partial \boldsymbol{F}}{\partial x} = 0, \qquad \boldsymbol{U}(x,y,t^n) = \tilde{\tilde{\boldsymbol{U}}}^{n}(x,y), \qquad x \in \mathbb{R},\ t \in \left(t^{n+\frac{1}{2}}, t^{n+1}\right] \tag{10}$$

for $\boldsymbol{U}_1^{n+1}(x,y)$ which is the final approximation using the second-order Strang splitting [1].

2.1.2 Fourth-Order Strang Splitting

Descombes [2] showed a fourth-order splitting method for a two-dimensional problem which requires five steps, similar to the ones in the last section. The first step is solving the x-sweep over the time interval $\left(t^n, t^{n+\frac{1}{4}}\right]$ with $t^{n+\frac{1}{4}} = t^n + \frac{\Delta t}{4}$ to solve for $\tilde{\boldsymbol{U}}^{n+\frac{1}{4}}(x,y)$. The next step is the y-sweep for the time interval $\left(t^n, t^{n+\frac{1}{2}}\right]$ with initial condition $\tilde{\boldsymbol{U}}^{n+\frac{1}{4}}(x,y)$. This step solves for $\tilde{\tilde{\boldsymbol{U}}}^{n+\frac{1}{2}}(x,y)$. The third step is solving the x-sweep again over the time interval $\left(t^{n+\frac{1}{4}}, t^{n+\frac{3}{4}}\right]$ from initial condition $\tilde{\tilde{\boldsymbol{U}}}^{n+\frac{1}{2}}(x,y)$, solving for $\tilde{\boldsymbol{U}}^{n+\frac{3}{4}}(x,y)$. The fourth step is completing the y-sweep in the time interval $\left(t^{n+\frac{1}{2}}, t^{n+1}\right]$ using the initial condition $\tilde{\boldsymbol{U}}^{n+\frac{3}{4}}(x,y)$ and solving for $\tilde{\tilde{\boldsymbol{U}}}^{n+1}(x,y)$. The last step of this method is completing the x-sweep. This is solved over the time interval $\left(t^{n+\frac{3}{4}}, t^{n+1}\right]$, using the initial condition $\tilde{\tilde{\boldsymbol{U}}}^{n+1}(x,y)$ and solving for $\boldsymbol{U}_2^{n+1}(x,y)$. To complete the fourth-order approximation, the approximation, $\boldsymbol{U}_1^{n+1}(x,y)$, found in Equation (10) and $\boldsymbol{U}_2^{n+1}(x,y)$ are used. These two computations are multiplied using the following weights for the final approximation

$$\boldsymbol{U}^{n+1}(x,y) = \frac{4}{3}\boldsymbol{U}_2^{n+1}(x,y) - \frac{1}{3}\boldsymbol{U}_1^{n+1}(x,y) \tag{11}$$

where $\frac{4}{3}$ and $-\frac{1}{3}$ are weights.

2.2 Splitting Methods in Three Dimensions

2.2.1 Second-Order Strang Splitting

For the three-dimensional Euler Conservation equation (2), the solution $\boldsymbol{U}(x,y,z,t)$ to the problem

$$\begin{aligned} \frac{\partial \boldsymbol{U}}{\partial t} + \frac{\partial \boldsymbol{F}}{\partial x} + \frac{\partial \boldsymbol{G}}{\partial y} + \frac{\partial \boldsymbol{H}}{\partial z} &= 0 \qquad (x,y,z) \in \mathbb{R}^3,\ t > 0 \\ \boldsymbol{U}(x,y,z,0) &= \boldsymbol{U}_0(x,y,z) \qquad (x,y,z) \in \mathbb{R}^3 \end{aligned} \tag{12}$$

where $\boldsymbol{U}_0(x,y)$ is given, can be approximated using splitting methods with an XZY scheme, where X, Y, Z are defined in Section 1.4, and the scheme refers to the order of the sweeps of the corresponding variable, as shown below. First, a second-order Strang splitting method is used. This method will take five steps to complete. Similar to the two-dimensional problem, x-sweep is first performed by fixing $y, z \in \mathbb{R}$ and solving

$$\frac{\partial \boldsymbol{U}}{\partial t} + \frac{\partial \boldsymbol{F}}{\partial x} = 0, \quad \boldsymbol{U}(x,y,z,t^n) = \boldsymbol{U}^n(x,y,z), \quad x \in \mathbb{R},\ t \in \left(t^n, t^{n+\frac{1}{2}}\right] \tag{13}$$

for $\tilde{\boldsymbol{U}}^{n+\frac{1}{2}}(x,y,z)$ where $t^{n+\frac{1}{2}} = t^n + \frac{\Delta t}{2}$. Next, the z-sweep is started in the time interval $\left(t^n, t^{n+\frac{1}{2}}\right]$ using $\tilde{\boldsymbol{U}}^{n+\frac{1}{2}}(x,y,z)$ as the initial condition and solving for $\tilde{\tilde{\boldsymbol{U}}}^{n+\frac{1}{2}}(x,y,z)$. The third step starts the y-sweep in the time interval $(t^n, t^{n+1}]$ with the initial condition $\tilde{\tilde{\boldsymbol{U}}}^{n+\frac{1}{2}}(x,y,z)$ solving for $\tilde{\tilde{\tilde{\boldsymbol{U}}}}^{n+1}(x,y,z)$. The fourth step involves completing the z-sweep in the time interval $\left(t^{n+\frac{1}{2}}, t^{n+1}\right]$ where the initial condition is $\tilde{\tilde{\tilde{\boldsymbol{U}}}}^{n+1}(x,y,z)$ and is used to solve for $\tilde{\tilde{\boldsymbol{U}}}^{n+1}(x,y,z)$. The final step is finishing the x-sweep, for which the time interval is $\left(t^{n+\frac{1}{2}}, t^{n+1}\right]$ with the initial condition $\tilde{\tilde{\boldsymbol{U}}}^{n+1}(x,y,z)$ and solution $\boldsymbol{U}_1^{n+1}(x,y,z)$.

2.2.2 Fourth-Order Strang Splitting

For the fourth-order splitting method, there will have nine steps. Similar to the two-dimensional fourth-order splitting method, the method starts with the x-sweep over the time interval $\left(t^n, t^{n+\frac{1}{4}}\right]$ with initial condition $\boldsymbol{U}^n(x,y,z)$ to solve for $\tilde{\boldsymbol{U}}^{n+\frac{1}{4}}(x,y,z)$. The next step is starting the z-sweep over the time interval $\left(t^n, t^{n+\frac{1}{4}}\right]$ with initial condition $\tilde{\boldsymbol{U}}^{n+\frac{1}{4}}(x,y,z)$ and solution $\tilde{\tilde{\boldsymbol{U}}}^{n+\frac{1}{4}}(x,y,z)$. Now, the y-sweep is initiated over the time interval $\left(t^n, t^{n+\frac{1}{2}}\right]$ with initial condition $\tilde{\tilde{\boldsymbol{U}}}^{n+\frac{1}{4}}(x,y,z)$ to solve for $\tilde{\tilde{\tilde{\boldsymbol{U}}}}^{n+\frac{1}{2}}(x,y,z)$. For the fourth step, another z-sweep is performed over the time interval $\left(t^{n+\frac{1}{4}}, t^{n+\frac{1}{2}}\right]$ using $\tilde{\tilde{\tilde{\boldsymbol{U}}}}^{n+\frac{1}{2}}(x,y,z)$ as the initial condition and solving for $\tilde{\tilde{\boldsymbol{U}}}^{n+\frac{1}{2}}(x,y,z)$. Now, another x-sweep is performed over the time interval $\left(t^{n+\frac{1}{4}}, t^{n+\frac{3}{4}}\right]$ with the initial condition $\tilde{\tilde{\boldsymbol{U}}}^{n+\frac{1}{2}}(x,y,z)$ used to solve for $\tilde{\boldsymbol{U}}^{n+\frac{3}{4}}(x,y,z)$. The next step is another z-sweep over the time interval $\left(t^{n+\frac{1}{2}}, t^{n+\frac{3}{4}}\right]$ using the initial condition $\tilde{\boldsymbol{U}}^{n+\frac{3}{4}}(x,y,z)$ to solve for $\tilde{\tilde{\boldsymbol{U}}}^{n+\frac{3}{4}}(x,y,z)$. The y-sweep is then completed over the time interval $\left(t^{n+\frac{1}{2}}, t^{n+1}\right]$ with the initial condition $\tilde{\tilde{\boldsymbol{U}}}^{n+\frac{3}{4}}(x,y,z)$ to solve for $\tilde{\tilde{\tilde{\boldsymbol{U}}}}^{n+1}(x,y,z)$. In the eighth step, the z-sweep is completed over the time interval $\left(t^{n+\frac{3}{4}}, t^{n+1}\right]$ to solve for $\tilde{\tilde{\boldsymbol{U}}}^{n+1}(x,y,z)$ with the initial condition $\tilde{\tilde{\tilde{\boldsymbol{U}}}}^{n+1}(x,y,z)$. Finally, the x-sweep is completed over the time interval $\left(t^{n+\frac{3}{4}}, t^{n+1}\right]$. Using the initial condition $\tilde{\tilde{\boldsymbol{U}}}^{n+1}(x,y,z)$, the approximation $\boldsymbol{U}_2^{n+1}(x,y,z)$ can be computed. Using $\boldsymbol{U}_1^{n+1}(x,y,z)$ and $\boldsymbol{U}_2^{n+1}(x,y,z)$, there is a similar equation to Equation (11) to get the final approximation of $\boldsymbol{U}^{n+1}(x,y,z)$ with

$$\boldsymbol{U}^{n+1}(x,y,z) = \frac{4}{3}\boldsymbol{U}_2^{n+1}(x,y,z) - \frac{1}{3}\boldsymbol{U}_1^{n+1}(x,y,z), \tag{14}$$

again, where $\frac{4}{3}$ and $-\frac{1}{3}$ are weights.

3 Simulations and Results

3.1 Grid Convergence and Error Analysis

In CFD studies, establishing grid convergence is an important tool for numerical analysis. As described by Roache [7], the first step is to find three mesh sizes suitable for coarse, medium, and fine simulations. Next, an order of convergence is established for the simulations. This tells how quickly the error of the simulations converge. For these simulations, the order of convergence, P, was calculated with

$$P = \frac{\log\left(\frac{e_i}{e_{i-1}}\right)}{\log(r)} \tag{15}$$

where e_i is the error with respect to the fine mesh and r is the constant refinement ratio. For these simulations, $r = \frac{1}{2}$ for all calculations. The next step is performing a Richardson Extrapolation to predict the data of the simulation at mesh size 0 according to the equation

$$\rho_0 = \rho_i + \frac{\rho_{i-1} - \rho_{i-2}}{2^P - 1} \tag{16}$$

where ρ_i represents the max density value for the fine mesh size, ρ_{i-1} represents the max density value for the medium mesh size, and ρ_{i-2} represents the max density value for the coarse mesh size. This ρ_0 is an estimate of the density value when there is zero grid spacing. Next, the Grid Convergence Index (GCI) for the medium and fine meshes is calculated to report grid convergence quality by the equation

$$\mathrm{GCI}_{i,i-1} = \frac{F_s|e_i|}{2^P - 1} \tag{17}$$

where $F_s = 1.25$ is the safety factor. GCI measures the percentage that the computed value is away from the value of the asymptotic numerical value. Lastly, the grids chosen are checked to see if they are in the asymptotic range of convergence by verifying that

$$\frac{\mathrm{GCI}_{i,i-1}}{r^P \cdot \mathrm{GCI}_{i-1,i-2}} \cong 1 \tag{18}$$

where $\mathrm{GCI}_{i,i-1}$ is the GCI of the fine mesh size, and $\mathrm{GCI}_{i-1,i-2}$ represents the GCI of the medium mesh size.

3.2 Finding Optimum Mesh Sizes

The first simulations performed were to find the optimum mesh sizes and the number of processors necessary to perform the three-dimensional RM simulations. Table 1 shows the simulations stopped at time $T = 0.1$ ms, where m is the mesh size, np is the number of processor that performed the simulation, and the max density values. The max density (ρ) values are recorded from initial time $T = 0$ ms. The time to solution each simulation took was also recorded.

TABLE 1: Simulations on mesh 10, 20, 40, 80, 160, 320 stopped at $T = 0.1$ ms

m	np	time to solution (hh:mm:ss)	max ρ (at $T = 0$ ms)
$10 \times 10 \times 40$	$1 \times 1 \times 1$	00 : 00 : 03	2.206
$20 \times 20 \times 80$	$1 \times 1 \times 1$	00 : 00 : 26	2.221
$40 \times 40 \times 160$	$1 \times 1 \times 1$	00 : 04 : 23	2.229
$80 \times 80 \times 320$	$1 \times 1 \times 1$	01 : 01 : 27	2.233
$80 \times 80 \times 320$	$4 \times 4 \times 16$	00 : 00 : 33	2.233
$160 \times 160 \times 640$	$4 \times 4 \times 16$	00 : 09 : 33	2.235
$320 \times 320 \times 1280$	$2 \times 2 \times 8$	–	–

The first three simulations and the mesh 80 with $np = 256$ can be seen to have completed very quickly. When the simulation with the size 80 mesh and $np = 1$ was performed, the simulations took much longer to perform. Increasing np to 256 did help with speed, but it could be seen that more resources were required for the same data points. Clearly, it is seen that as mesh size (m) increased, the interface became much smoother. However, the mesh size 320 was unable to complete due to the disk allocation requirements in PSC Bridges involved with such a fine-sized mesh. Thus, the rest of the simulations that were performed using the coarse mesh with $m = 20 \times 20 \times 80$, the medium-sized mesh with $m = 40 \times 40 \times 160$, and the fine mesh with $m = 80 \times 80 \times 320$, each with $np = 1$.

3.3 Simulations without Strang Splitting

Table 2 displays the results of the three simulations that were run with a WENO scheme that was supposedly third order. The results are visualized in Figure 2.

TABLE 2: Simulations on mesh 20, 40, 80 with max time $T = 0.1$ ms

m	np	max ρ
$20 \times 20 \times 80$	$1 \times 1 \times 1$	2.156
$40 \times 40 \times 160$	$1 \times 1 \times 1$	2.163
$80 \times 80 \times 320$	$1 \times 1 \times 1$	2.164

To analyze the grid convergence, the order of convergence is first calculated, for which Equation (15) is used to solve for

$$P = \frac{\log\left(\dfrac{2.164 - 2.164}{2.163 - 2.156}\right)}{\log\left(\dfrac{1}{2}\right)} = 3$$

which gives a third order convergence. Using this, the Richardson Extrapolation data point via Equation (16) is calculated as

$$\rho_0 = 2.164 + \frac{2.163 - 2.156}{2^3 - 1} = 2.16500.$$

Equation (17) is used to get

$$
\begin{aligned}
\mathrm{GCI}_{80,40} &= \frac{F_s\left(\frac{2.164-2.163}{2.164}\right)}{2^3-1} \times 100\% = 0.0082519\% \\
\mathrm{GCI}_{40,20} &= \frac{\left(\frac{2.163-2.156}{2.163}\right)}{2^3-1} \times 100\% = 0.057790\%.
\end{aligned}
$$

Last, Equation (18) is used to get

$$
\frac{0.0082519}{\left(\frac{1}{2}\right)^3 \cdot 0.057790} = 1.14233.
$$

This shows that solutions are within the asymptotic range of convergence. These calculations also shows that the max density for mesh 0 is predicted to be 2.16500 with an error bound of 0.057790%. The WENO scheme was expected to be third order convergent which was consistent with the results. The extrapolation result gives an anticipated value since it was predicted to be a value greater than 2.164 for the max density for mesh size 0. It can be seen that the values of the densities increased with a finer mesh size, so when there is zero grid spacing, we would expect a larger value such as 2.165.

3.4 Simulations with Second Order Strang Splitting

For this set of simulations, a known second order Strang splitting method was used. The results are visualized in Figure 3.

TABLE 3: Simulations on mesh 20, 40, 80 with max time $T = 0.1$ ms using second order Strang splitting

m	np	max ρ
$20 \times 20 \times 80$	$1 \times 1 \times 1$	2.152
$40 \times 40 \times 160$	$1 \times 1 \times 1$	2.162
$80 \times 80 \times 320$	$1 \times 1 \times 1$	2.165

To analyze these results, the order of convergence is calculated, for which Equation (15) yields

$$P = \frac{\log\left(\dfrac{2.165 - 2.162}{2.165 - 2.152}\right)}{\log\left(\dfrac{1}{2}\right)} = 2.11548$$

which implies a second order convergence as expected. Using this, the Richardson Extrapolation data point via Equation (16) is

$$\rho_0 = 2.165 + \frac{2.162 - 2.152}{2^{2.11548} - 1} = 2.16800.$$

Equation (17) is then used to get

$$\begin{aligned} \mathrm{GCI}_{80,40} &= \frac{F_s\left(\frac{2.165-2.162}{2.165}\right)}{2^{2.11548} - 1} \times 100\% = 0.051963\% \\ \mathrm{GCI}_{40,20} &= \frac{\left(\frac{2.162-2.152}{2.162}\right)}{2^{2.11548} - 1} \times 100\% = 0.17426\%. \end{aligned}$$

Last, Equation (18) yields

$$\frac{0.051963}{\left(\frac{1}{2}\right)^{2.11548} \cdot 0.17426} = 1.29217.$$

This shows that the solutions are within the asymptotic range of convergence. These calculations also show that the max density for mesh 0 is predicted to be 2.16200 with an error bound of 0.17426%. Again, the extrapolation result yields an expected value since it would have been predicted to be a value greater than 2.165 for the max density for mesh size 0 given the trend in the data with finer mesh spaces. The expected second order convergence is seen here.

3.5 Simulations with Experimental Fourth Order Strang Splitting

For this set of simulations, the experimental implementation of the fourth order Strang splitting method was used. The results are visualized in Figure 4.

TABLE 4: Simulations on mesh 20, 40, 80 with max time $T = 0.1$ ms using an experimental fourth order Strang splitting method

m	np	max ρ
$20 \times 20 \times 80$	$1 \times 1 \times 1$	2.152
$40 \times 40 \times 160$	$1 \times 1 \times 1$	2.162
$80 \times 80 \times 320$	$1 \times 1 \times 1$	2.164

First, the order of convergence is calculated, for which Equation (15) gives

$$P = \frac{\log\left(\dfrac{2.164 - 2.162}{2.164 - 2.152}\right)}{\log\left(\dfrac{1}{2}\right)} = 2.58496$$

which gives a third order convergence. Using this, the Richardson Extrapolation data point via Equation (16) is

$$\rho_0 = 2.164 + \frac{2.162 - 2.152}{2^{2.58496} - 1} = 2.16700.$$

Using Equation (17) yields

$$\begin{aligned} \mathrm{GCI}_{80,40} &= \frac{F_s\left(\frac{2.164-2.162}{2.164}\right)}{2^{2.58496} - 1} \times 100\% = 0.023105\% \\ \mathrm{GCI}_{40,20} &= \frac{\left(\frac{2.162-2.152}{2.162}\right)}{2^{2.58496} - 1} \times 100\% = 0.11617\%. \end{aligned}$$

Last, Equation (18) is used to get

$$\frac{0.023105}{\left(\frac{1}{2}\right)^{2.58496} \cdot 0.11617} = 1.19334.$$

This shows that the solutions are within the asymptotic range of convergence. These calculations also give the max density for mesh 0 is predicted to be 2.16200 with an error bound of 0.11617%. Rounding the order of convergence up, the results of the simulations give a third order convergence when fourth order was expected with this scheme. This leads to the conclusion that the implementation of the C++ code for this method is not correct. However, an increase in order of convergence as a result of this implementation is seen. The max density prediction, however, does yield something since it would have been predicted to be a value greater than 2.164 for the max density for mesh size 0.

3.6 Pseudocolor Plots of Simulations

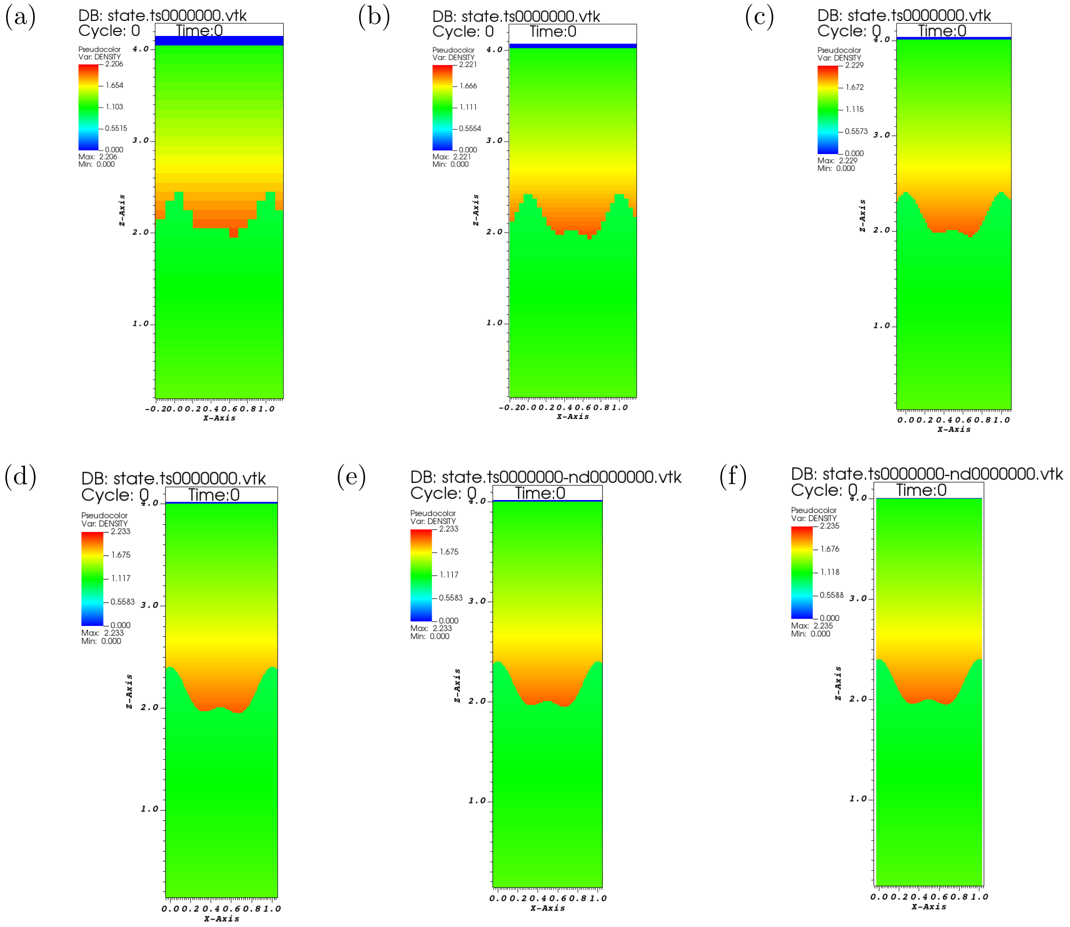

FIGURE 1: Pseudocolor plots of the density of the simulations for mesh size (a) 10 with $np = 1$, (b) 20 with $np = 1$, (c) 40 with $np = 1$, (d) 80 with $np = 1$, (e) 80 with $np = 256$ and (f) 160 with $np = 256$ with max time $T = 0.1$ ms and densities recorded at initial time $T = 0$ ms

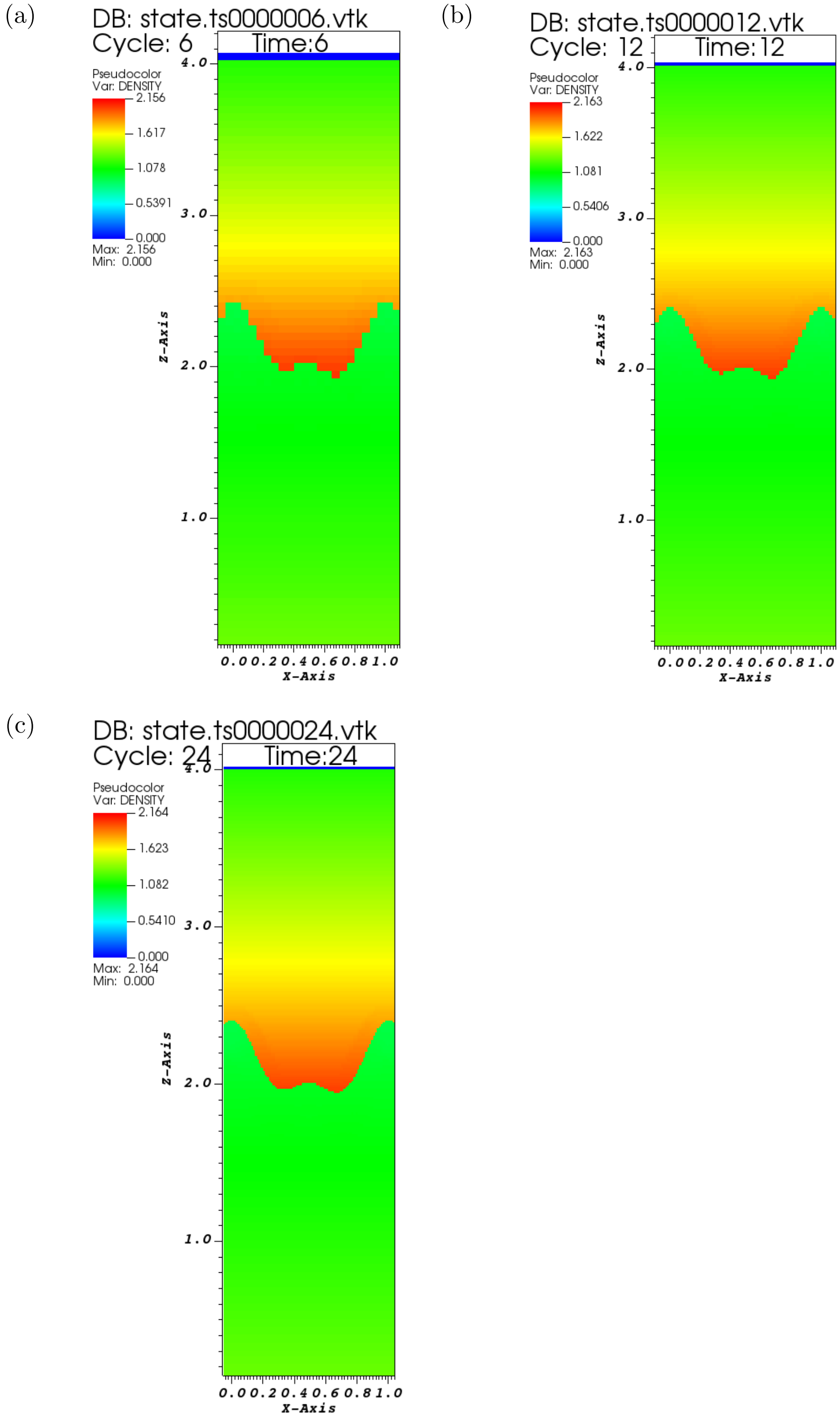

FIGURE 2: Pseudocolor plots of the density of the simulations for mesh size (a) 20, (b) 40, (c) 80 with no Strang Splitting

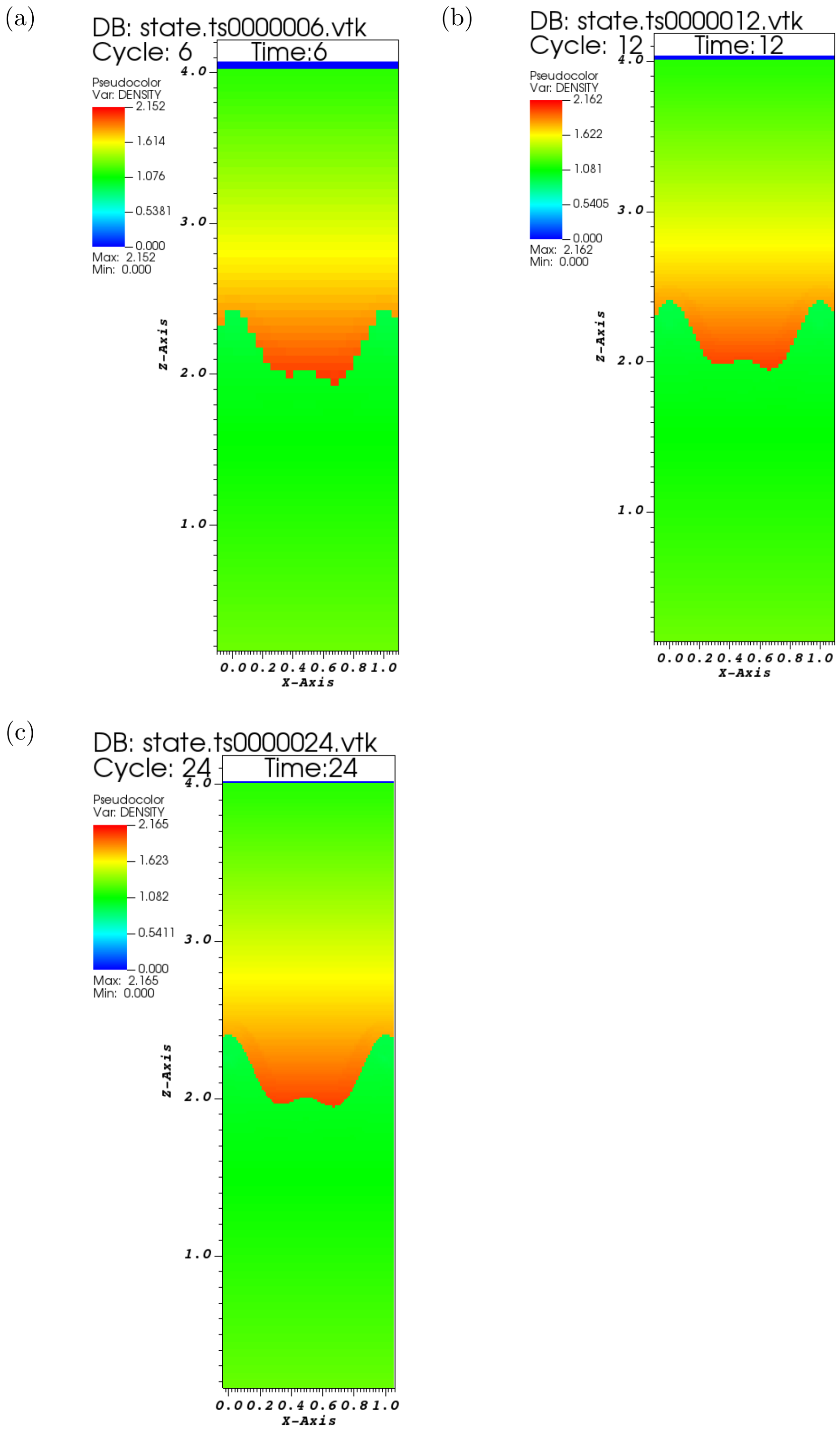

FIGURE 3: Pseudocolor plots of the density of the simulations for mesh size (a) 20, (b) 40, (c) 80 with second order Strang splitting

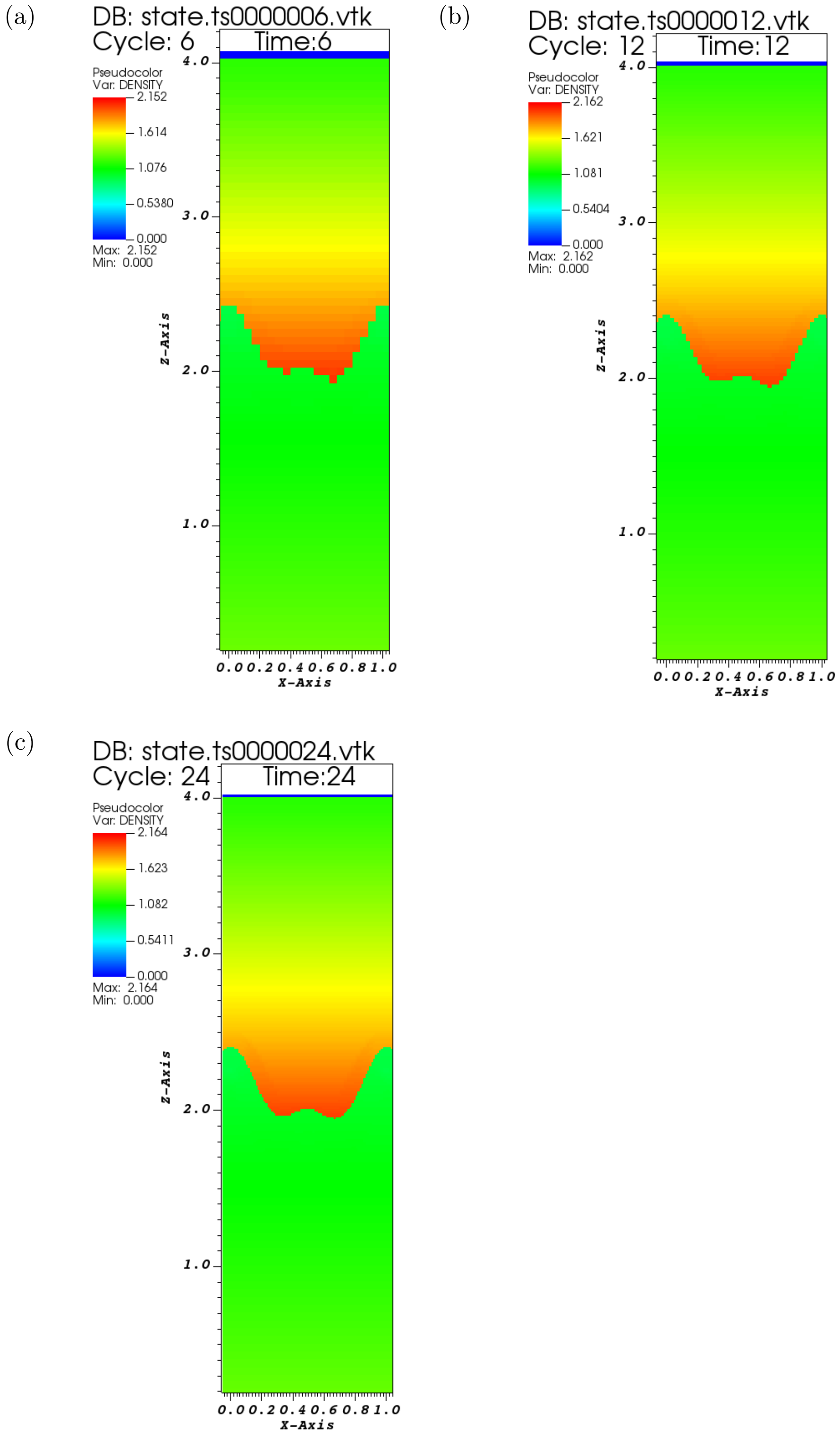

FIGURE 4: Pseudocolor plots of the density of the simulations for mesh size (a) 20, (b) 40, (c) 80 with fourth order Strang splitting

4 Conclusion

Some of the scientific applications of RTI are interstellar medium and galaxy clusters and interactions between shocked solar wind plasma and magnetospheric plasma on Earth. A practical application of RTI is found in industry with premixed combustion or transport. There are also several practical applications of the complex flow of RMI such as inertial confinement fusions and fuel injection in supersonic flows. Another application of RMI is in astrophysics where there is significant fluid mixing involved in supernovae. In all of these applications, it is easy to see that experimental studies on the actual events themselves is not feasible; issues arise from the short nature of the events and the extreme environmental conditions in which they occur. Thus, using RTI and RMI simulations for analysis is a very helpful alternative for understanding the growth of the mixing layer of the interface [9].

The results of the simulations for this research were consistent for the second order Strang splitting method, but the expected fourth order Strang splitting method was not observed as desired. Although the implementation of the fourth order Strang splitting method did not show results of fourth order convergence, it did increase the rate of convergence compared to the known-to-work second order Strang splitting method. Hence, if there were more time for implementation, it is possible that an order four convergence could be observed. A source of the error in the implementation is found in the multiplication of the flux with the weights. One other possible source of error could have been that the fine mesh of size 80 is not fine enough for accurate results. Possibly with more resources, running simulations with mesh size 320 could have potentially shown better results. Despite these possible errors, it is believed that the data observed in this study will give rise to better results in the future with more work and the observation for even higher order convergence.

References

[1] Todd Arbogast Chieh-Sen Huang and Jianxian Qiu. An Eulerian-Lagrangian WENO Finite Volume Scheme for Advection Problems. *Journal of Computational Physics*, 231(11):4028 – 4052, 2012.

[2] Stephane Descombes. Convergence of a Splitting Method of High Order for Reaction-Diffusion Systems. *Mathematics of Computation*, 70(236):1481–1501, 2001.

[3] James Glimm, John W. Grove, X. L. Li, and D. C. Tan. Robust computational algorithms for dynamic interface tracking in three dimensions. *SIAM J. Sci. Comput.*, 21(6):2240–2256, December 1999.

[4] James Glimm, David H. Sharp, Tulin Kaman, and Hyunkyung Lim. New directions for Rayleigh-Taylor mixing. *Philosophical Transactions of the Royal Society A: Mathematical, Physical and Engineering Sciences*, 371(2003):20120183, 2013.

[5] R. L. Holmes, J. W. Grove, and D. H. Sharp. Numerical investigation of Richtmyer-Meshkov instability using front tracking. *Journal of Fluid Mechanics*, 301:51–64, 1995.

[6] Shev MacNamara and Gilbert Strang. Operator splitting. In *Splitting Methods in Communication, Imaging, Science, and Engineering*, pages 95–114. Springer International Publishing, 2011.

[7] P. J. Roache. Perspective: A Method for Uniform Reporting of Grid Refinement Studies. *Journal of Fluids Engineering*, 116(3):405–413, 09 1994.

[8] C.-W. Shu. High-order Finite Difference and Finite Volume WENO Schemes and Discontinuous Galerkin Methods for CFD. *International Journal of Computational Fluid Dynamics*, 17(2):107–118, 2003.

[9] Ye Zhou. Rayleigh Taylor and Richtmyer Meshkov instability induced flow, turbulence, and mixing. I. *Physics Reports*, 720-722:1 – 136, 2017. Rayleigh-Taylor and Richtmyer-Meshkov instability induced flow, turbulence, and mixing. I.