

Multimaterial Simulations using Ghost Fluid Methods

Knut Sverdrup

Abstract

The unsteady, compressible Euler equations for multimaterial flow in one dimension have been solved numerically by employing level-set based tracking of fronts and two versions of the Ghost Fluid Method. Equations of state for stiffened ideal gases are supported, to allow simulations with water. An exact Riemann solver is demonstrated, in addition to usage of both centered and upwind higher-order time-marching schemes, namely MUSCL-Hancock with HLLC and SLIC. Results from several varied test cases show that the implementation works in accordance with that presented in previous literature.

1. Introduction

Leonhard Euler first presented the momentum and continuity equations in 1757^[1], which were completed by the adiabatic condition presented by Laplace in 1816^[2]. The energy balance equation, which is the last of what is now called the Euler equations, was not properly incorporated until the late nineteenth century^[3], and although the much more general Navier-Stokes equations have been developed^[4,5], the continued interest for and usefulness of the Euler equations is undisputable. They provide a robust framework for analyzing ideal fluids when viscous effects are negligible, but cannot generally be solved analytically. Applications of the Euler equations include aerodynamics^[6–8], atmospheric modelling and weather forecasts^[9,10], astrophysics^[11] and detonations and explosives^[12–14], to name a few. Accordingly, precise and efficient methods for solving these non-linear, hyperbolic partial differential equations on arbitrary domains has been, and still is, a major field in modern computational fluid dynamics.

The Euler equations govern adiabatic and inviscid flow of a fluid. In the Froude limit (no external body forces) in one dimension, with density ρ , velocity u , total energy E and pressure p , they are given in dimensionless form as

$$\partial_t \mathbf{U} + \partial_x \mathbf{F}(\mathbf{U}) = 0, \quad (1)$$

where the vector of conserved quantities \mathbf{U} and their fluxes $\mathbf{F}(\mathbf{U})$ are given by

$$\mathbf{U} = \begin{bmatrix} \rho \\ \rho u \\ E \end{bmatrix}, \quad \mathbf{F} = \begin{bmatrix} \rho u \\ \rho u^2 + p \\ u(E + p) \end{bmatrix}.$$

It is sometimes convenient to work in terms of the primitive variables $\mathbf{W} = (\rho, u, p)^T$. The total energy is the sum of the kinetic and potential energy of the system, i.e.

$$E = \frac{1}{2} \rho u^2 + \rho e, \quad (2)$$

where e is the internal energy, related to the other variables through the equation of state. The equation of state is of fundamental importance for two reasons. Firstly, it allows us to treat the materials in question from a continuum point of view. It is thus the connection between the physics taking place on a microscopic scale and the thermodynamics which we apply macroscopically. Secondly, it provides the missing fourth equation to our four unknowns (ρ, u, p and E), so that the system becomes mathematically determined. For an ideal gas, the equation of state is

$$e = \frac{p}{(\gamma - 1)\rho}, \quad (3)$$

where γ denotes the ratio of specific heats for the gas. Several other fluids can be approximated by a so-called stiffened ideal gas equation of state,

$$e = \frac{p + \gamma p_\infty}{(\gamma - 1)\rho}. \quad (4)$$

Here, a material-dependent stiffening parameter p_∞ has been introduced. Note that Eq. (4) reduces to Eq. (3) for materials with $p_\infty = 0$.

Many important milestones have led us to the current state of the art of solving the Euler equations numerically. Riemann identified and worked on the initial value problem for Eq. (1) with discontinuous initial conditions as early as 1860^[15], but the first exact solution did not arrive until Godunov proposed an iterative scheme a century later^[16]. Today, efficient approximate solvers such as HLLC (Harten-Lax-van Leer-Contact)^[17] and Rotated-hybrid Riemann solvers^[18] are readily available. The development of numerical analysis methods for partial differential equations got its first real boost after the famous paper by Courant, Friedrichs and Lewis^[19], but local Riemann problems^[16], conservative methods^[20] and finite volume formulations^[21] were necessary before the first proper, three-dimensional simulations of the Euler equations could be performed in the 1980s^[22]. After the Monotonic Upstream-centered Scheme for Conservation Laws (MUSCL)^[23] was introduced by van Leer as the first higher-order Total Variation Diminishing (TVD) scheme in 1979, several others have been developed, notably Toro's Weighted Average Flux (WAF)^[24] and Flux/Slope Limiter Centred (SLIC/FLIC)^[25] schemes.

Considerations of multimaterial interactions and flow are among the more recent (less than twenty year old) developments within numerical solutions for the Euler equations. From a continuum fluid mechanics point of view, multimaterial interactions are used to describe problems where there is a relatively sharp interface between two materials. Examples of such cases are water/oil separation (liquid-liquid interaction), penguins swimming through water (solid-liquid interaction) and water splashing into the air as a result of a surface disruption (liquid-gas interaction). In reality, all these interfaces are continuous from one material to the other. Since the length scale of the interface is far smaller than even the discrete control volumes used for simulation, however, it is a good approximation to treat the interfaces as sharp discontinuities. It is the goal of this report to demonstrate some techniques available for multimaterial simulations of the Euler equations in one dimension. Different fluids are characterized by their material properties, such as γ , and thus special care must be taken in the treatment of boundaries separating them.

Firstly, the location of the boundary must be

tracked as time evolves. The most common way of doing this is by the use of level set methods, as proposed by Osher and Sethian^[26]. The level set function for a region is initiated such that it is negative inside the region, positive outside it and zero on the boundary. By taking into consideration the convection of the fluids under consideration, the level set function is then evolved so that its zeros move with the domain boundary. Osher and Fedkiw^[27] provide an excellent overview of level set methods and their applications.

Secondly, the interaction between different fluids across the material interfaces must be modelled accurately. In order to do so, Fedkiw et al. developed the Original Ghost Fluid Method (OGFM)^[28]. In ghost fluid methods, each material has ghost cells on the side of the boundary where the other material exists. In the OGFM, the pressures and velocities at these ghost cells are similar to that of the other material, and the densities are calculated by enforcing constant entropy across the interface. This worked well for cases except when there were strong shock waves, since the entropy condition does not apply. As a response, the Modified Ghost Fluid Method (MGFM) was developed^[29,30]. It is based on the realization that Riemann problems do not in general require the states on each side of the discontinuity to correspond to the same material. The cells in the ghost fluid are therefore updated by solving a local Riemann problem. A further development is the real Ghost Fluid Method (rGFM)^[31], in which a more accurate interfacial boundary condition is applied. The implementation of the rGFM is outside the scope of this report.

The numerical methods which have been employed are explained in Section 2, before several test cases and their results are discussed in Section 3. In Section 4, we propose possibilities for further work, while Section 5 concludes the report.

2. Numerical methods

2.1. Riemann solvers

Given a conservation equation and two sets of piecewise constant states separated by a single discontinuity, the initial value problem of evolving this system in time is called a Riemann problem. It is very useful in the study of the Euler equations for two reasons. Firstly, it allows for exact (up to an arbitrary accuracy) solutions for systems obeying Eq. (1) which have a single contact discontinuity.

The other benefit of Riemann solvers is that the discretization of space which is inevitable in computational schemes for solving differential equations, allows for precise solvers of conservation equations based on the solutions of many local Riemann problems.

For the Euler equations and initial conditions

$$\mathbf{W}(x, t = 0) = \begin{cases} \mathbf{W}_L, & x \leq 0 \\ \mathbf{W}_R, & x > 0 \end{cases},$$

typical states the system can have are shown in Figure 1, in addition to characteristics for waves propagating in space-time. There are two types of resultant waves that propagate through space, in addition to the contact wave (dashed). The first is a shock wave, depicted as a thick line on the left, while the second is a rarefaction wave, shown as several gradually decaying lines on the right. Four different combinations of shock and rarefaction waves can occur on the left and right sides of the contact discontinuity, and the result is only dependent on $\mathbf{W}(x, 0)$.

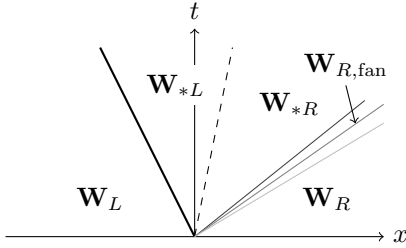


Figure 1: Possible wave configurations for the Riemann problem for Euler's equations in one dimension.

As seen from Figure 1, there are (up to) four unknown states for the Riemann problem: the left and right star states \mathbf{W}_{*K} located between the left and right waves, and additionally the states inside rarefaction waves $\mathbf{W}_{K,\text{fan}}$. Note that the density in the K star state depends on whether the corresponding wave is of shock or rarefaction type. In order to solve the Riemann problem, one must therefore be able to compute all these states, and also determine which of them is the correct one based on the speeds of the waves.

Exact solver

The exact solver, as first introduced by Godunov^[16], has been implemented following Chapter 4 of Toro's comprehensive overview of Riemann

solvers and their uses^[32], but extended to take into consideration that different (stiffened) ideal gases may be present on each side of the discontinuity.

It can be proven that the pressure in the star states p_* is given by the root of the nonlinear function

$$f(p) = f_L(p) + f_R(p) - (u_L - u_R). \quad (5)$$

Here,

$$f_K(p) = \begin{cases} \frac{2a_K}{\gamma_K - 1} \left[\left(\frac{p + p_{\infty,K}}{p_K + p_{\infty,K}} \right)^{\frac{\gamma_K - 1}{2\gamma_K}} - 1 \right], & p \leq p_K \\ (p - p_K) \left(\frac{A_K}{B_K + p} \right)^{\frac{1}{2}}, & p > p_K \end{cases},$$

where $a_K = \sqrt{\frac{\gamma(p_K + p_{\infty,K})}{\rho_K}}$ is the speed of sound and

$$A_K = \frac{2}{(\gamma_K + 1)\rho_K}, \quad B_K = \frac{(\gamma_K - 1)p_K + 2\gamma_K p_{\infty,K}}{\gamma_K + 1}.$$

In the exact solver, the pressure in the star region is found by applying an iterative scheme to Eq. (5). Due to the nature of $f(p)$, it is well-suited for iterative schemes, but it is important to ensure that the pressure stays positive. For this project, Newton-Raphson iterations were implemented, taking the initial guess as $\frac{1}{2}(p_L + p_R)$. After finding p_* , the velocity in the star region is computed as

$$u_* = \frac{u_L + u_R - [f_L(p_*) - f_R(p_*)]}{2}. \quad (6)$$

When the pressure and velocity in the star region have been computed, one can calculate the wave speeds (one per shock wave, two per rarefaction wave and one for the contact discontinuity) and use this to sample the state at a given point $S = x/t$ in space-time. Pseudocode for this procedure is given in Algorithm 1. Formulas for wave speeds and densities in different states have been left out for brevity, but they in general depend on the initial conditions (including material properties).

HLLC

By far the most computationally expensive part of the exact solver is the iterative root finding procedure for finding p_* . Since most modern schemes require the solutions of local Riemann problems between all points in the computational domain, high gains in terms of computational efficiency can be

Data: $S, \mathbf{W}_L, \mathbf{W}_R, p_*, u_*$, material properties

Result: $\mathbf{W}(S)$

```

if  $S \leq$  speed of contact discontinuity then
  if left wave is rarefaction then
    if  $S \leq$  speed of rarefaction head then
      | return  $\mathbf{W}_L$ ;
    else
      if  $S >$  speed of rarefaction tail then
        | return  $\mathbf{W}_{*L, \text{fan}}$ ;
      else
        | return  $\mathbf{W}_{L, \text{fan}}$ ;
      end
    end
  else
    if  $S \leq$  speed of shock wave then
      | return  $\mathbf{W}_L$ ;
    else
      | return  $\mathbf{W}_{*L}$ ;
    end
  end
else
  perform similar analysis for sampling point
  on right side of contact discontinuity;
end

```

Algorithm 1: Sample exact solution of Riemann problem given pressure and velocity in star states.

achieved by employing approximate solvers instead. One such solver is the HLLC solver^[17], which has been implemented for this project.

When compared to the exact solver, the main simplifications in HLLC are that the pressure in the star region is approximated instead of found by iterative schemes; that states inside rarefactions are not taken into consideration, and that wave speeds are estimations, either direct or pressure-based. In this contribution, the pressure is estimated as

$$p_* \approx \frac{1}{2}(p_L + p_R) + \frac{1}{8}(u_L - u_R)(\rho_L + \rho_R)(a_L + a_R), \quad (7)$$

and set equal to zero if the approximation is negative. Based on the approximation for p_* , the wave speeds S_K of the left and right travelling waves are computed, with the type of wave taken into consideration. The speed of the contact discontinuity is then approximated as

$$S_* \approx \frac{\rho_L u_L (S_L - u_L) - \rho_R u_R (S_R - u_R) - (p_L - p_R)}{\rho_L (S_L - u_L) - \rho_R (S_R - u_R)}. \quad (8)$$

There are four possible states, separated by the three (approximated) wave speeds. Contrary to the exact solver, where the output is the state $\mathbf{W}(S)$, the HLLC solver outputs a flux $\mathbf{F}^{\text{HLLC}}(\mathbf{W}_L, \mathbf{W}_R)$ based on the state corresponding to $S = 0$. This is because the flux at the intermediate point between cells is the quantity employed in the update formula as introduced in Subsection 2.2. For an overview of the approximate states and the flux calculation employed in the HLLC solver, see Chapter 9 of Toro's book^[32].

2.2. Schemes for the Euler equations

The Euler equations as given in Eq. (1) are said to be a system of conservation equations in differential form. In integral form, on a spatial domain $[x_L, x_R]$, the same system of equations can be written

$$\int_{x_L}^{x_R} \mathbf{U}(x, t_2) dx = \int_{x_L}^{x_R} \mathbf{U}(x, t_1) dx - \int_{t_1}^{t_2} \mathbf{F}(\mathbf{U}(x_R, t)) dt + \int_{t_1}^{t_2} \mathbf{F}(\mathbf{U}(x_L, t)) dt. \quad (9)$$

From Eq. (9), it is straightforward to discretize the Euler equations to produce a conservative time-marching scheme through the Finite Volume Method (FVM). Let the spatial domain $x \in [0, 1]$ be divided into N equal cells of width $\Delta x = 1/N$. We denote by x_i the center of the i -th cell, i.e. $x_i = (i + 1/2)\Delta x$. Temporal discretization is done with variable time step Δt^n , so that $t^n = \sum_{i=1}^n \Delta t^i$.

We write the discrete approximation of $\mathbf{U}(x_i, t^n)$ as \mathbf{U}_i^n , and let it be the weighted average of $\mathbf{U}(x, t^n)$ in the cell with boundaries at $x_{i-1/2}$ and $x_{i+1/2}$:

$$\mathbf{U}_i^n \approx \frac{1}{\Delta x} \int_{x_{i-1/2}}^{x_{i+1/2}} \mathbf{U}(x, t^n) dx \quad (10)$$

Similarly, the fluxes at each interface are approximated as

$$\mathbf{F}_{i+1/2}^n \approx \frac{1}{\Delta t^n} \int_{t_n}^{t_{n+1}} \mathbf{F}(\mathbf{U}(x_{i+1/2}, t)) dt. \quad (11)$$

By applying Eq. (9) on each cell $[x_{i-1/2}, x_{i+1/2}]$, and by inserting the discrete approximations in Eqns. (10) and (11), a conservative, time-marching scheme for evolving the Euler equations in time is

$$\mathbf{U}_i^{n+1} = \mathbf{U}_i^n - \frac{\Delta t^n}{\Delta x} (\mathbf{F}_{i+1/2} - \mathbf{F}_{i-1/2}). \quad (12)$$

The Courant-Friedrichs-Lewis condition for convergence demands that

$$\Delta t^n \leq \frac{\Delta x}{S_{\max}^n}, \quad (13)$$

so we let $\Delta t^n = C_{\text{CFL}} \Delta x / S_{\max}^n$, where $0 \leq C_{\text{CFL}} \leq 1$. In our implementation, we use $C_{\text{CFL}} = 0.9$. Transmissive boundary conditions are implemented on both boundaries, so that

$$\mathbf{W}_{-i}^{n+1} = \mathbf{W}_{-(i-1)}^n, \quad \mathbf{W}_{N+i}^{n+1} = \mathbf{W}_{N+(i-1)}^n, \quad (14)$$

for $i = 1, \dots, N_{GC}$ where the number of ghost cells N_{GC} equals the order of the scheme employed.

All the most common schemes used for solving conservation laws are of the same form as Eq. (12), and the difference between them is in the evaluation of the fluxes $\mathbf{F}_{i\pm 1/2}$. For this project, we have implemented so-called total variation diminishing (TVD) high resolution methods for this purpose. High resolution methods are accurate to at least second order; they avoid spurious oscillations in the solution, capture discontinuities and shock waves well and satisfy a discrete form of the entropy condition. There are two main classes of higher-order flux evaluation schemes; the upwind schemes are based on the solution of local Riemann problems, while the centered schemes are extensions of the First ORder CEntered (FORCE) flux:

$$\mathbf{F}^{\text{FORCE}}(\mathbf{U}_L, \mathbf{U}_R) = \frac{1}{2} \left[\mathbf{F}_0 + \frac{1}{2} (\mathbf{F}_L + \mathbf{F}_R) \right] - \frac{1}{4} \frac{\Delta x}{\Delta t^n} (\mathbf{U}_R - \mathbf{U}_L). \quad (15)$$

Here, $\mathbf{F}_K = \mathbf{F}(\mathbf{U}_K)$ and

$$\mathbf{U}_0 = \frac{1}{2} (\mathbf{U}_L + \mathbf{U}_R) - \frac{1}{2} \frac{\Delta t^n}{\Delta x} (\mathbf{F}_R - \mathbf{F}_L). \quad (16)$$

In general, upwind schemes offer higher accuracy than centered ones, but they are computationally more demanding and difficult to implement.

The MUSCL-Hancock^[23] and SLIC^[25] schemes, as mentioned in the Introduction, exemplify upwind and centred schemes, respectively. They are similar in many regards, and both consist of three main steps.

First, the data cell averages \mathbf{U}_i^n are replaced locally by linear functions so that at the left and right

cell interfaces, the boundary extrapolated values are $\mathbf{U}_i^L = \mathbf{U}_i^n - \frac{1}{2} \Delta_i$ and $\mathbf{U}_i^R = \mathbf{U}_i^n + \frac{1}{2} \Delta_i$. Here, the slope vector Δ_i has been introduced, which in our case is chosen such that the schemes are TVD. For the MUSCL-Hancock scheme, we have implemented limited slopes^[7], while SLIC employs slope limiters^[25].

After calculating the boundary extrapolated values for each cell, these are advanced to time $t^n + \frac{1}{2} \Delta t^n$ to obtain

$$\bar{\mathbf{U}}_i^{L,R} = \mathbf{U}_i^{L,R} - \frac{1}{2} \frac{\Delta t^n}{\Delta x} [\mathbf{F}(\mathbf{U}_i^R) - \mathbf{F}(\mathbf{U}_i^L)]. \quad (17)$$

The third and final step is the actual evaluation of the intercell flux $\mathbf{F}_{i+1/2}$ based on the partially evolved boundary extrapolated values. For MUSCL-Hancock, this is done by using the HLLC solver to solve the local Riemann problem across each cell interface, i.e.

$$\mathbf{F}_{i+1/2}^{\text{MUSCL}} = \mathbf{F}^{\text{HLLC}}(\bar{\mathbf{W}}_i^R, \bar{\mathbf{W}}_{i+1}^L). \quad (18)$$

For SLIC, however, the intercell flux is directly evaluated as the FORCE flux with

$$\mathbf{F}_{i+1/2}^{\text{SLIC}} = \mathbf{F}^{\text{FORCE}}(\bar{\mathbf{U}}_i^R, \bar{\mathbf{U}}_{i+1}^L). \quad (19)$$

2.3. Level-set method

In order to represent the interfaces between materials, we employ a level-set function $\phi(x)$ whose sign is a boolean indicator of the material present. The interfaces between materials are thus located at the points where $\phi = 0$. Figure 2 shows the simple case of a single material interface separating materials A and B; several interfaces have been incorporated by letting ϕ be piecewise linear throughout the domain in a zig-zag pattern.

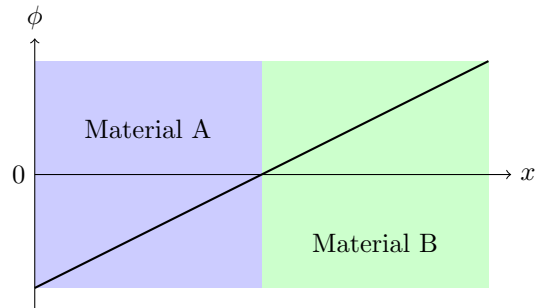


Figure 2: Level-set function for a single material interface. In the left region, where $\phi(x) < 0$, material A is present.

Initializing the level-set function is trivial, but it is a good idea to make sure that the zeros end up between cell centers, so that we avoid grid points where the level set function is exactly zero. By doing so, each grid point has well-defined material properties. In order to capture the evolution of the system, the level-set function itself must be advanced in time. Since ϕ is a consequence of the state of the fluid rather than a property, it is not coupled to the other variables in the system. Thus it is governed by the advection equation

$$\partial_t \phi + u \partial_x \phi = 0, \quad (20)$$

Due to frequent discontinuities in density at material interfaces, conservative schemes lead to smearing of the locations of the zeros. Since the only interesting information contained in ϕ are these locations, however, non-conservative methods can be safely implemented as long as ϕ remains smooth and not too sharp. In our implementation, we have used the first-order upwind scheme

$$\phi_i^{n+1} = \begin{cases} \phi_i^n - u_i^n \frac{\Delta t^n}{\Delta x} (\phi_i^n - \phi_{i-1}^n), & u_i^n > 0 \\ \phi_i^n - u_i^n \frac{\Delta t^n}{\Delta x} (\phi_{i+1}^n - \phi_i^n), & u_i^n \leq 0 \end{cases}. \quad (21)$$

Note that u_i^n is the velocity of material A at x_i if $\phi_i^n < 0$, and the velocity of material B otherwise. In order to avoid the level set function becoming too sharp, we reinitialize it after every update by demanding $|\phi_{i+1}^n - \phi_i^n| = \Delta x$. We also make sure that the zeros of ϕ lie between grid points.

2.4. Ghost fluid methods

Multimaterial simulations require one final consideration in addition to interface tracking; namely which boundary conditions to enforce across these interfaces. These dynamic boundary conditions should be thermodynamically consistent in order to capture the physical interactions between materials. Presence of thermodynamical inconsistencies leads to spurious entropy generation, which in turn results in unphysical evolution of the system. In the computational implementation, the Euler equations are solved on a separate domain for each material, where that material is present in all cells. For the regions where $\phi(x) < 0$, fluid A is present in domain A, while *ghost fluid* B is present in domain B. The opposite is true when $\phi(x) > 0$. The only interesting cells in the ghost fluids are the N_{GC} points located immediately adjacent to a material interface,

since these affect the solution in the actual fluid. It is the goal of ghost fluid methods to update these ghost cells in a thermodynamically consistent manner.

Original ghost fluid method

The first successful attempt at implementing a ghost fluid method was performed by Fedkiw et al. in 1998^[28]. In this approach, which we refer to as OGFM, the entropy S is held constant across the ghost fluid interface, in order to give conditions for thermodynamic consistency. The assumption of constant entropy is not entirely true in the case of shocks, so one does not expect excellent capturing of large shock waves. The pressure and velocity in the ghost cells are set equal to the pressure and velocity in the same cells of the other material domain. It is the density which is extrapolated by enforcing constant entropy. To derive how, we note that the entropy of an ideal gas is given by

$$S = c_V \ln \left(\frac{p}{\rho^\gamma} \right) + \text{const}, \quad (22)$$

so if two cells i and j have the same entropy and contain the same material, they must satisfy

$$\rho_i = \left(\frac{p_i}{p_j} \right)^{1/\gamma} \rho_j. \quad (23)$$

In order to avoid “overheating” across the material boundary, an isobaric fix has been implemented for OGFM by enforcing constant entropy from the neighbour of the boundary cell to the boundary as well as from the boundary and into the ghost fluid. A schematic illustration, including the isobaric fix, is given in Figure 3.

Modified ghost fluid method

In order to be able to solve a wider range of problems, including those with strong shock waves or large density gradients, the MGFM was developed^[29,30]. In this approach, the intermaterial boundary conditions are enforced in a way which is based on solving a local Riemann problem across the interface. As explained in Subsection 2.1, it is unproblematic to solve a Riemann problem with different material properties on each side of the initial discontinuity. Given the two states immediately adjacent to the material interface, \mathbf{W}_L and \mathbf{W}_R , and the corresponding material properties, we calculate the star states \mathbf{W}_{*L} and \mathbf{W}_{*R} , and set

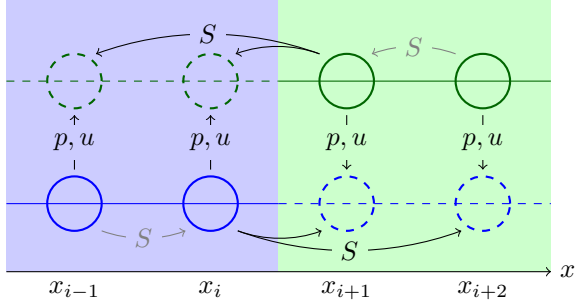


Figure 3: Updating of cells in ghost fluids as implemented in OGF. The pressure and velocities are inherited from the other material, while entropy is kept constant across the ghost fluid interface. With the isobaric fix (grey), the entropy condition is enforced from further inside the actual fluid, to reduce overheating.

the ghost fluid cells equal to them. Thus every extrapolated value in the ghost fluid corresponds to a physical state of the system, and we have thermodynamical consistency. By looking at Figure 1, it is easy to convince oneself that the chosen star states are valid choices for the boundary cells. In order to compute the star states, the exact solver is employed.

2.5. Final notes

With all separate parts of the computational procedure explained, it is helpful to explain how it all fits together. As such, Algorithm 2 exhibits pseudocode summarizing the complete procedure. Our code is written in C++, but for easy input customization and result visualization we have built a user interface using Qt, a cross-platform application framework.

3. Results

The different tests described in this section aim to show how the implemented methods cope under a range of circumstances. Since there are quite many different choices one can make for each method, we inform that unless otherwise specified, fluxes are computed using the MUSCL-Hancock scheme; slope limiters are of minbee type; OGF is implemented with the isobaric fix. The reference solutions, where available, are obtained using the exact solver with 5000 points. Although the amount of grid points varies throughout the results, only 50 points in the solutions are included in all plots, so as to avoid the high-accuracy tests overshadowing the information from the lower accuracy ones.

Data: initial configuration of system,
choice of scheme, N , t_{end}
Result: configuration of system at time t_{end}
set $t = 0$;
initiate computational domain;
while $t < t_{\text{end}}$ **do**
 find locations of material interfaces;
 for each interface do
 | update ghost fluid using GFM;
 end
 calculate Δt^n from Eq. (13);
 if $t + \Delta t^n > t_{\text{end}}$ **then**
 | $\Delta t^n \rightarrow t_{\text{end}} - t$;
 end
 update boundaries according to Eq. (14);
 advance ϕ according to Eq. (21);
 for each material do
 | compute fluxes using Eq. (18) or (19);
 | evolve system in time with Eq. (12);
 end
 $t \rightarrow t + \Delta t^n$;
end
if initially there was one interface **then**
 | compute exact solution for reference;
end

Algorithm 2: Multimaterial solver

3.1. Moving contact discontinuity

Test A

To illustrate how the combination of a level set function and GFMs allow capture of a moving contact discontinuity, test A has a single discontinuity in density, and the system moves with constant speed to the right. Full test specifications are given in Table 1. Air occupies the entire domain, so both single-material solvers and multimaterial GFM solvers are employed.

$x \in$	ρ	u	p	γ
$[0, 0.25]$	1.000	0.5	1	1.4
$(0.25, 1]$	0.138	0.5	1	1.4

Table 1: Initial conditions for test A.

Results for test A are shown in Figure 4. Please note that only half of the spatial domain is included, since the discontinuity is located here. As is evident, both ghost fluid methods capture the discontinuity far better than the single-material solvers, in fact no error at all is apparent. Of the schemes used for the single-material solver, the MUSCL-Hancock

scheme provides a higher accuracy around the discontinuity, as expected.

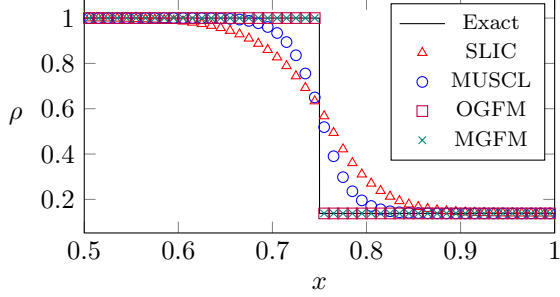


Figure 4: Results from test A with $N = 100$ and $t_{\text{end}} = 1$.

Test B

Test B is identical to test A except for the fact that the discontinuity now separates two different fluids; namely air and helium ($\gamma = 1.67$). Results are shown in Figure 5, which demonstrates that the solvers work well for the multimaterial case as well.

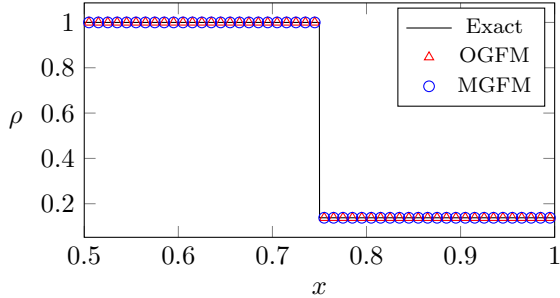


Figure 5: Results from test B with $N = 100$ and $t_{\text{end}} = 1$.

3.2. Simple ghost fluid tests

In order to check how well the GFM solvers capture general single-material contact discontinuities, a set of five standard tests for these schemes are considered. They are taken from Chapter 4 in Toro’s book^[32], and are typical Riemann problems, i.e. they have left and right states separated at time $t = 0$. For all cases considered, the initial discontinuity is located in the middle of the domain ($x = 0.5$), and the material considered is air, so $\gamma = 1.4$. The state parameters and final simulation times for all tests are given in Table 2. For readability, the full set of results from these test cases have been moved to Appendix A.

Test 1

The first test is a fairly mild test tube problem, but is nevertheless a useful indicator that all possible types of wave propagations are captured by the scheme. Two regions of a tube containing a gas are separated by a membrane, and the densities and pressures differ across this interface. At time $t = 0$ the membrane is removed, and the states start interacting. The solution contains (from left to right) a rarefaction wave, a contact wave and a shock wave. Since it was introduced by G.A. Sod in 1978^[33], it has gradually become the go-to test case for investigating the behaviour of simple Euler schemes, hence its alias “the Sod test tube”. Figure 6 exhibits the results for density for both ghost fluid methods. In Figure A.14 in Appendix A, results for all variables with several different resolutions are displayed. All features are captured very well by both models.

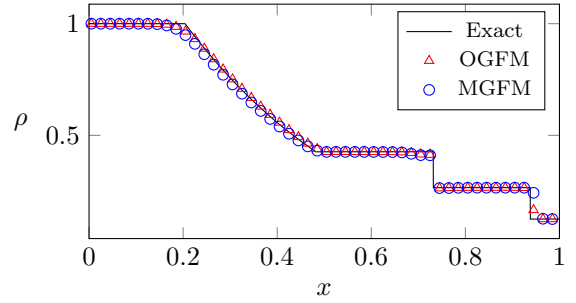


Figure 6: Density for OGFM and MGFM applied to Toro’s test 1 with $N = 100$.

This test also lends itself well to display the improvement that can be achieved in OGFM by applying the isobaric fix. From Figure 7, it is evident that the overheating in specific internal energy is reduced when the isobaric fix is implemented.

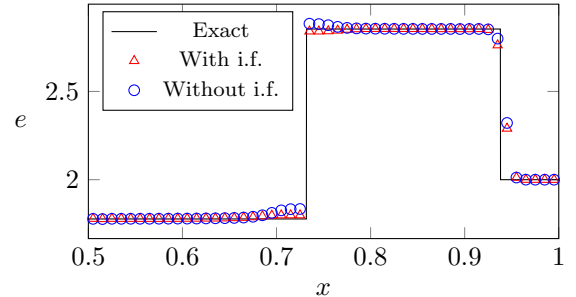


Figure 7: Internal energy for OGFM applied to Toro’s test 1 with and without isobaric fix. For both cases, $N = 100$.

Test	t_{end}	ρ_L	u_L	p_L	ρ_R	u_R	p_R
1	0.25	1.0	0.0	1.0	0.125	0.0	0.1
2	0.15	1.0	-2.0	0.4	1.0	2.0	0.4
3	0.012	1.0	0.0	1000.0	1.0	0.0	0.01
4	0.035	1.0	0.0	0.01	1.0	0.0	100.0
5	0.035	5.99924	19.5975	460.894	5.99242	-6.19633	46.0950

Table 2: Parameters for each of the five test from Toro’s book.

Test 2

Test two is also known as “the 123-problem”, and its solution consists of two strong rarefaction waves, one in each direction. Their speeds are equal, resulting in a static contact in the middle of the domain. Due to the low pressures, difficulties can also arise when estimating p . As seen from Figure A.15, both GFM’s handle this problem fairly well too. MGFM is more precise than OGFM for pressure, velocity and density, and although the OGFM yields a closer approximation to the specific internal energy in the middle of the region, it overshoots more and more when the spatial accuracy is increased. MGFM, on the other hand, converges towards the exact solution.

Test 3

Woodward and Colella investigated strong shocks in 2D in 1984^[34], and the remaining three tests are based on their work (albeit in a 1D framework). Test three is the left part of their blast wave problem, and has a huge pressure difference and a correspondingly large production of entropy. For this reason, the constant entropy condition employed in OGFM is expected to break down. The solution of the test case contains a left rarefaction, a contact, and a right shock. Our results are shown in Figure A.16, from which it is apparent that MGFM is far more suited to the task than OGFM, as expected.

The effect on the density of using a different limiter for test 3, namely superbee, is shown in Figure 8. The superbee limiter is in general less conservative than the minbee type, and can capture rapid changes more precisely. When it fails, however, oscillations can occur in the solution.

Test 4

The right part of Woodward and Colella’s blast wave problem also has a large pressure difference, but here the shock and rarefaction waves are travelling in opposite directions compared to test 3. Figure A.17 confirms the behaviour of our solvers when applied to these types of tests.

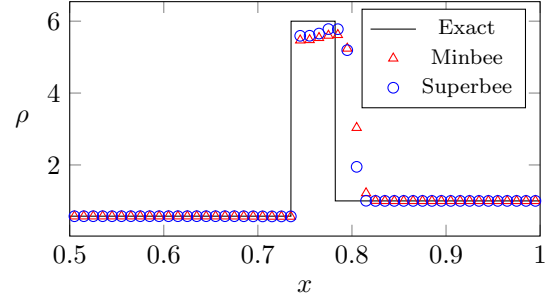


Figure 8: Density for MGFM applied to Toro’s test 3 showing the difference between minbee and superbee type limiters. For both cases, $N = 100$.

Test 5

As a last example, the fifth test case mimics what happens when the right-moving shock wave from test 3 meets the left-moving shock wave from test 4. As seen from Figure A.18, both OGFM and MGFM handle this case very well: both shock waves are captured with great accuracy. The pressure ratio between the left and right states is only a factor of ten here, so compared to the previous two test cases the entropy production across the shock is not expected to be substantial.

3.3. Multimaterial shock tubes for gases

In order to investigate how the solvers cope with the interaction of material interfaces and shock waves, the following two multimaterial shock tube test cases have been used. In both cases, the initial conditions are such that they correspond to physically realistic scales, measured in SI units.

Test C

This test case was introduced by Smadar Karni in 1996^[35], and is also one of the test cases used to verify OGFM in 1999^[28]. In it, a weak shock wave in air travels towards the right and refracts at a interface with helium gas. The result of the interfacial interaction is a narrow, reflected rarefaction wave and weak, transmitted shock wave. Since γ

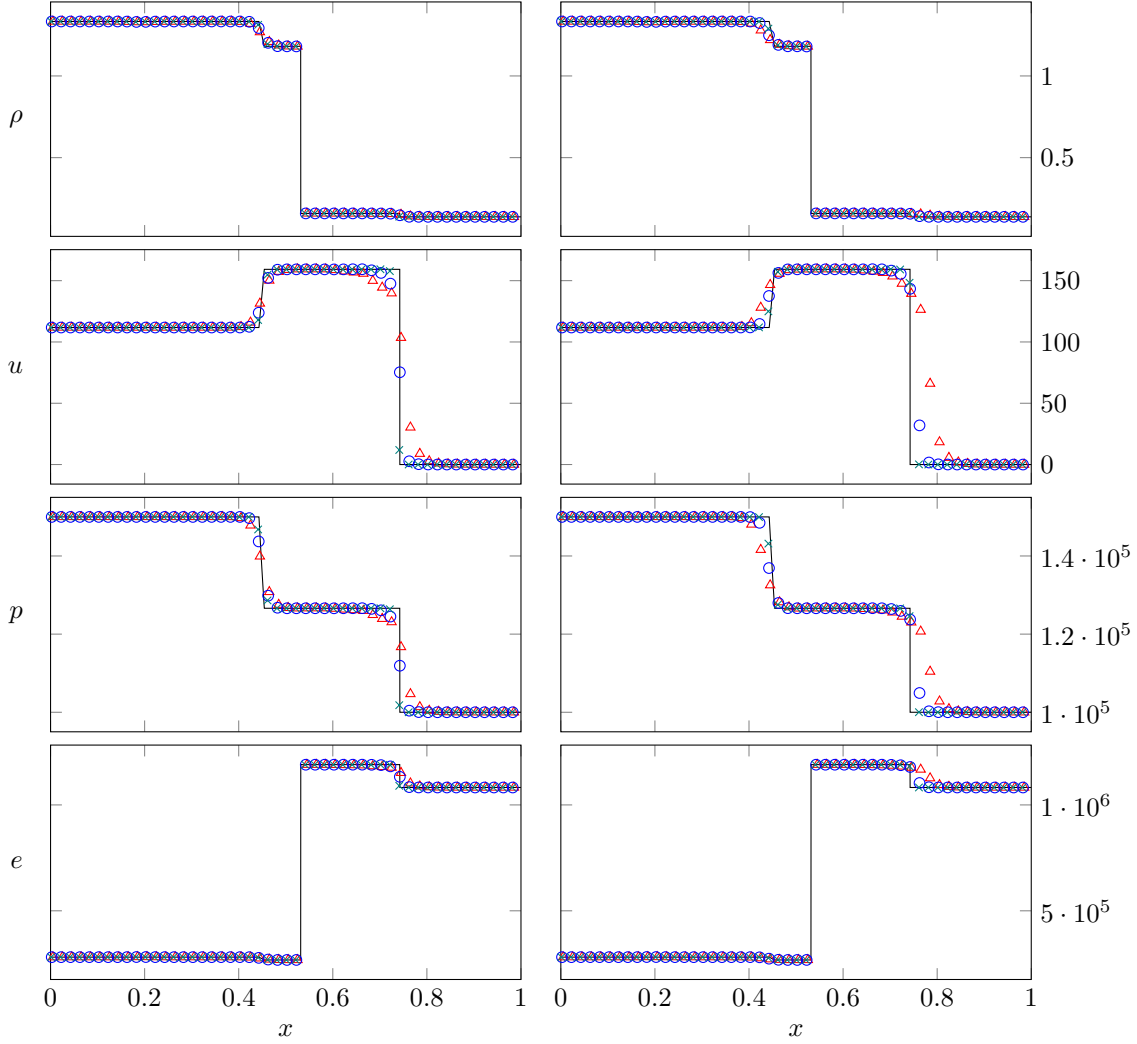


Figure 9: Results for Test C using OGFM (left) and MGFM (right). $\triangle N = 100$ $\circ N = 200$ $\times N = 400$

(and thus the speed of sound a) is higher for helium than for air, the transmitted shock wave travels faster than the incident shock wave in air. Initial parameters for the left (air postshock), middle (air preshock) and right (helium preshock) parts of the domain are tabulated in Table 3, and the simulation is run to a final time $t_{\text{end}} = 0.0012$.

$x \in$	ρ	u	p	γ
$[0, 0.05]$	1.3333	$0.3535\sqrt{10^5}$	$1.5 \cdot 10^5$	1.4
$(0.05, 0.5]$	1	0	$1.0 \cdot 10^5$	1.4
$(0.5, 1]$	0.1379	0	$1.0 \cdot 10^5$	1.67

Table 3: Initial conditions for test C.

The results for test C are depicted in Figure 9.

Since the initial data has two discontinuities, our exact solver cannot be used to find a reference solution directly in this case. As a workaround, we evolved the system until only two regions exist, namely air postshock and helium preshock, which is a single discontinuity and thus a Riemann problem. For both GFMs, the reflected rarefaction, transmitted shock and contact are captured very well.

Test D

A fascinating test case developed by Quirk and Karni^[36] simulates a rightward propagating shock wave in air hitting a slab of helium (between $x = 0.4$ and $x = 0.6$), on the right side of which is air. It is the one-dimensional equivalent of a shock wave hitting a bubble of helium in air. See Table 4 for the

initial conditions, and Figure 10 for a plot of the initial density distribution. Due to the presence of a local Riemann problem in the left air region in addition to two material interfaces, the resulting pattern becomes quite complex. The shock wave refracts at both material interfaces, so the end result (with $t_{\text{end}} = 0.0014$) is a superposition of multiple reflected and transmitted shock waves and rarefactions.

$x \in$	ρ	u	p	γ
$[0, 0.05]$	1.3333	$0.3535\sqrt{10^5}$	$1.5 \cdot 10^5$	1.4
$(0.05, 0.4]$	1	0	$1.0 \cdot 10^5$	1.4
$(0.4, 0.6]$	0.1379	0	$1.0 \cdot 10^5$	1.67
$(0.6, 1]$	1	0	$1.0 \cdot 10^5$	1.4

Table 4: Initial conditions for test D.

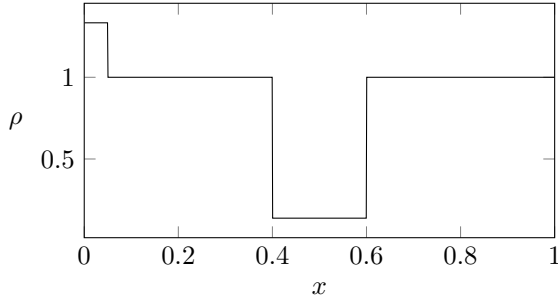


Figure 10: Initial density distribution for test D. A slab of helium gas is located in the region $0.4 < x \leq 0.6$, while the rest of the domain is filled with hydrogen.

Results from runs with both GFMs are visualized in Figure 11. Since no exact solution is available, we have included results obtained with high-resolution meshes ($N = 2000$), plotting every second point with a small dot. Both GFMs yield results that converge to the expected pattern, which can be found e.g. in a paper by Wang et al. from 2004^[37]. Note that in that paper, the internal energy is plotted instead of the *specific* internal energy, which explains the apparent difference in the bottom plots of Figure 11 when compared to their results.

3.4. Water-gas shock tube test

So far, all test cases have only included materials assumed to behave as ideal gases, i.e. which have $p_\infty = 0$ in Eq. (4). An example of a fluid which is approximated quite well by the stiffened gas equation of state, is water. As a final validation test case, we therefore verify the applicability of our model to water.

Test E

This test case is a standard shock-tube problem, but where the high-density region is filled with water, following Chinnaya et al.^[38]. Table 5 shows the initial conditions for this case. For water, we use $p_\infty = 6.0 \cdot 10^8$, which means there is a huge difference in the equations of state between the two materials. The solution of the test case consists of a left rarefaction, a rightward moving contact and a right shock, and is it the latter two which can become difficult to distinguish due to this huge difference. Test E is run to a final time $t_{\text{end}} = 237.44 \cdot 10^{-6}$.

$x \in$	ρ	u	p	γ	p_∞
$[0, 0.7]$	1000	0	10^9	4.4	$6.0 \cdot 10^8$
$(0.7, 1]$	50	0	10^5	1.4	0

Table 5: Initial conditions for test E.

The results for test E are shown in Figure 12. Although advised that OGFM was not expected to deal well with this case, we decided to run the tests with OGFM as well, out of curiosity. To our (pleasant) surprise, OGFM captures all aspects of the solution, albeit not as well as MGFM. In any event, the results show that even air/water interfaces are solved in a satisfactory manner by GFMs.

3.5. More interfaces!

The results from test D show that our level set function can track two material interfaces simultaneously without a problem. We wanted to ensure that it could track an arbitrary amount of interfaces, however, and therefore created an extra test where the amount of interfaces has been doubled compared to the previous maximum.

Test F

Table 6 shows the initial conditions for test F, which is run to a final time $t_{\text{end}} = 0.1$. The leftmost, center and rightmost regions contain hydrogen gas, and two slabs of helium are placed between these. Due to the initial discontinuities, shock waves travel toward the interior of both helium slabs, where they collide and interact. This is not necessarily a very realistic case physically, but it does test the ability of the scheme to generate a zig-zagging level set function and track it correctly.

Results generated by applying MGFM to test F are shown in Figure 13. Since no exact solution is possible, and no comparison is provided in the literature, we ran the code with a high resolution mesh

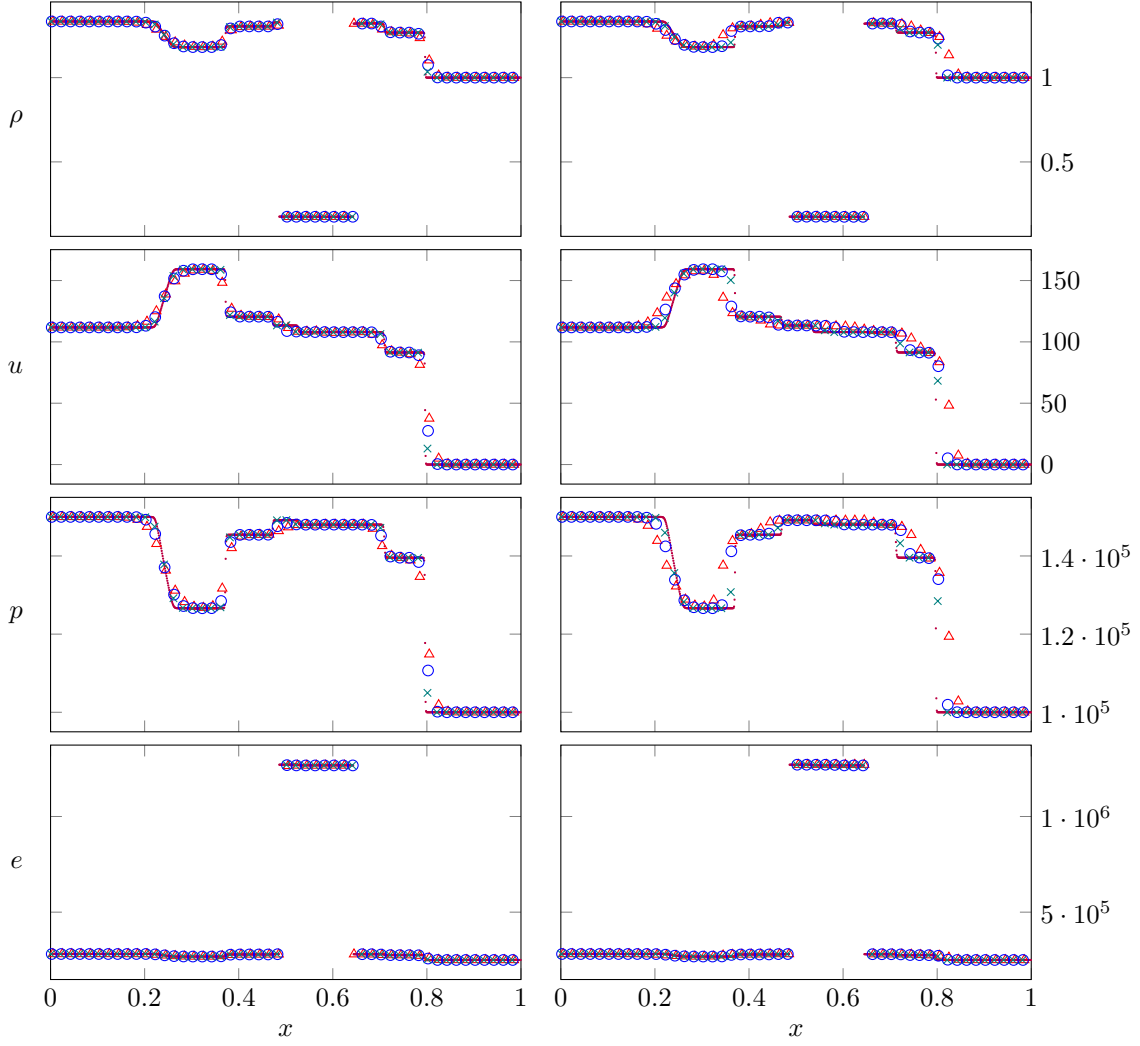


Figure 11: Results for Test D using OGFM (left) and MGFM (right). $\triangle N = 100$ $\circ N = 200$ $\times N = 400$ $\cdot N = 2000$

$x \in$	ρ	u	p	γ
$[0, 0.2]$	1	0	1	1.4
$(0.2, 0.4]$	0.125	0	0.1	1.67
$(0.4, 0.6]$	1	0	1	1.4
$(0.6, 0.8]$	0.125	0	0.1	1.67
$(0.8, 1]$	1	0	1	1.4

Table 6: Initial conditions for test F.

($N = 2000$). The resulting patterns are complex and difficult to interpret from just a snapshot, but animating the results in our graphical user interface show that the system does indeed behave in an intuitive manner.

4. Proposed further work

For future development of the code, we suggest extensions to two (and three) spatial dimensions, so as to facilitate simulations of more interesting physical problems that are relatable to real-world applications. In order to do so, we would advise implementing dimensional operator splitting. For these types of problems, it would also be interesting to investigate arbitrary domain shapes and mesh refinement techniques. Since the computational expense grows quickly with the amount of dimensions, it would at this point be feasible to put some effort into speeding up the runtime. Luckily, the most expensive job is the calculation of fluxes, which can be done independently for each point in the domain.

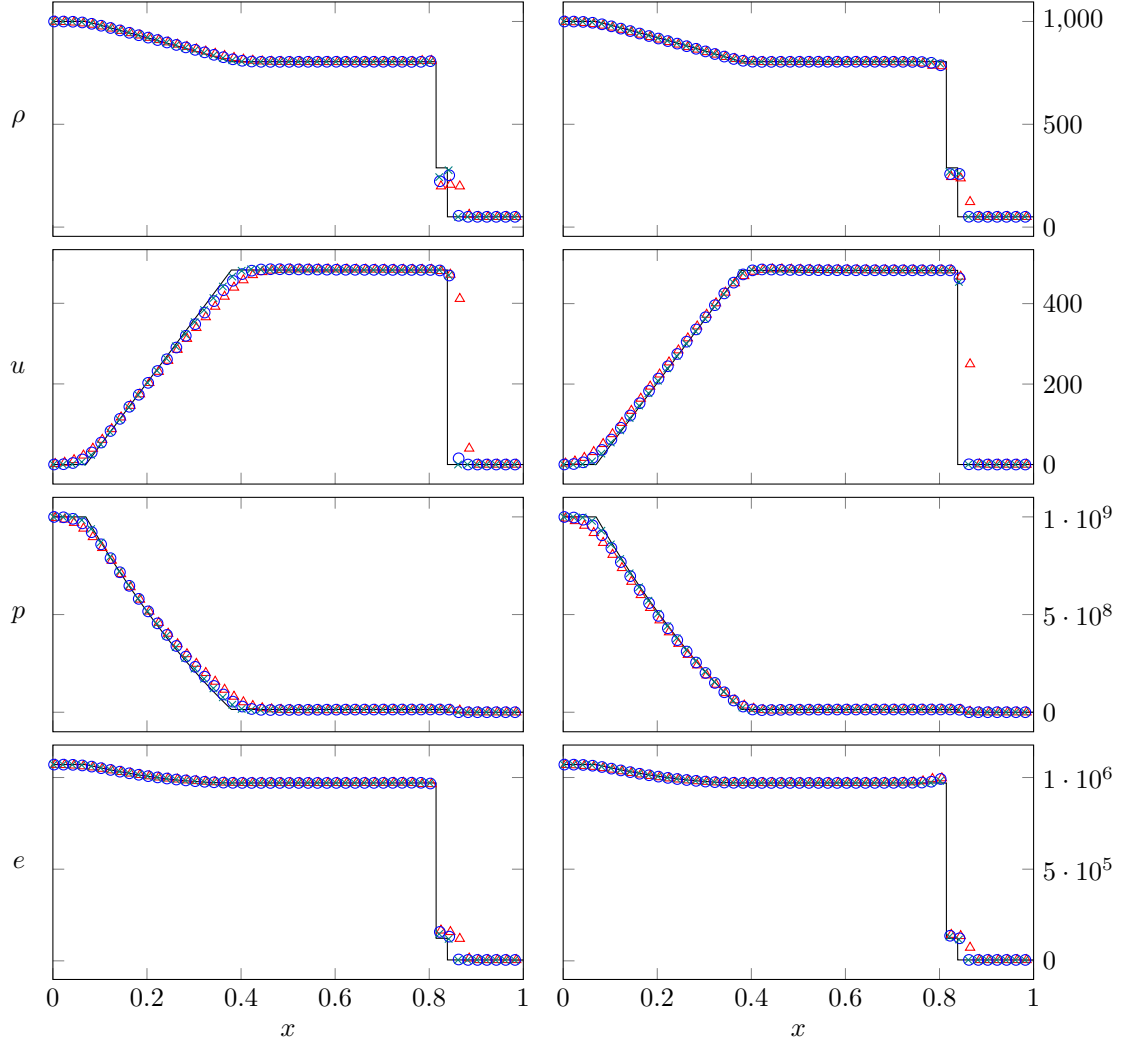


Figure 12: Results for test E using OGFM (left) and MGFM (right). $\triangle N = 100$ $\circ N = 200$ $\times N = 400$

Thus, parallelization techniques should be implemented for high gains in terms of computing time.

5. Conclusions

We have successfully implemented a numerical PDE solver for the compressible Euler equations, which allows for multimaterial continuum simulations in one spatial dimension. Although there is nothing novel about the methods and resulting code, the project report serves as a literature review which covers an introduction to the Euler equations and their numerical solution, in addition to multimaterial problems for which level set and ghost fluids methods are necessary. In addition to describing the theory, we present results for numerous test

cases which illustrate that our code is correct, robust and in agreement with what is described in the literature.

Acknowledgements

Gratitude is directed to project advisor Stephen Millmore for sharing his insight in the field and offering good advice.

References

- [1] L. Euler, Principes généraux de l'état d'équilibre d'un fluide, Académie Royale des Sciences et des Belles Lettres de Berlin 11 (1757) 217–273.

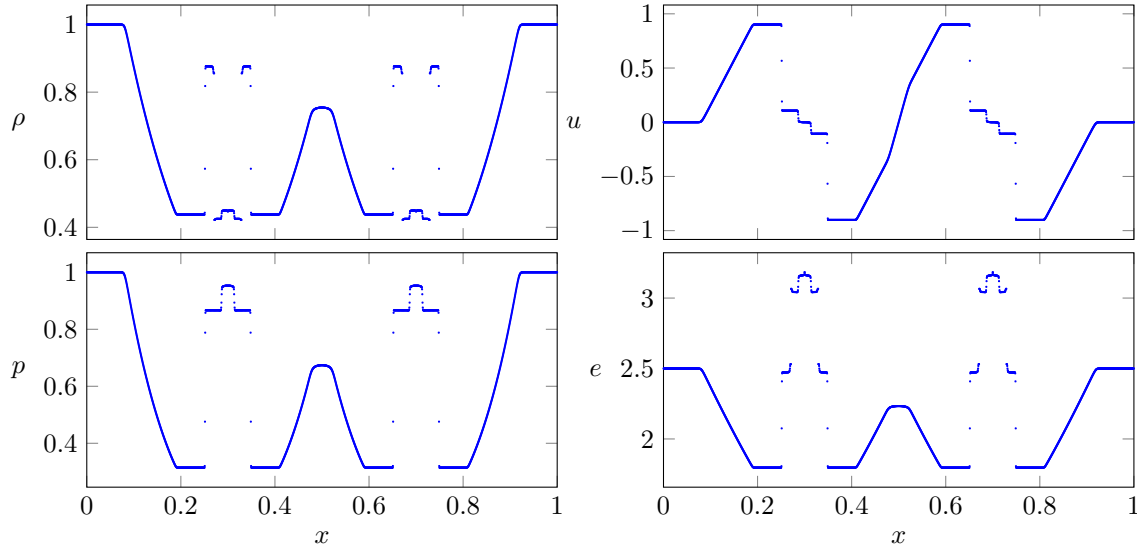


Figure 13: Results for test F using MGFM.

- [2] P.-S. de Laplace, Sur la vitesse du son dans l'air et dans l'eau, *Annales de Chimie et de Physique* 3 (1816) 283–241.
- [3] D. Christodoulou, The Euler equations of compressible fluid flow, *Bulletin of the American Mathematical Society* 44 (4) (2007) 581–602.
- [4] C. L. M. H. Navier, Memoire sur les lois du mouvement des fluides, *Mémoires de l'Académie Royale des Sciences de l'Institut de France* 6 (1822) 389–440.
- [5] G. G. Stokes, On the theories of the internal friction of fluids in motion and of the equilibrium and motion of elastic solids, *Transactions of the Cambridge Philosophical Society* 8 (1845) 287–319.
- [6] M. Drela, Two-dimensional transonic aerodynamic design and analysis using the Euler equations, Ph.D. thesis, Massachusetts Institute of Technology (1985).
- [7] W. K. Anderson, J. L. Thomas, B. Van Leer, Comparison of finite volume flux vector splittings for the Euler equations, *AIAA journal* 24 (9) (1986) 1453–1460.
- [8] G. P. Guruswamy, Unsteady aerodynamic and aeroelastic calculations for wings using Euler equations, *AIAA journal* 28 (3) (1990) 461–469.
- [9] R. Laprise, The Euler equations of motion with hydrostatic pressure as an independent variable, *Monthly weather review* 120 (1) (1992) 197–207.
- [10] W. C. Skamarock, J. B. Klemp, A time-split nonhydrostatic atmospheric model for weather research and forecasting applications, *Journal of Computational Physics* 227 (7) (2008) 3465–3485.
- [11] H. Trac, U.-L. Pen, A primer on Eulerian computational fluid dynamics for astrophysics, *Publications of the Astronomical Society of the Pacific* 115 (805) (2003) 303.
- [12] N. Nikiforakis, J. Clarke, Numerical studies of the evolution of detonations, *Mathematical and Computer Modelling* 24 (8) (1996) 149–164.
- [13] D. N. Williams, L. Bauwens, E. S. Oran, Detailed structure and propagation of three-dimensional detonations, in: *Symposium (International) on Combustion*, Vol. 26, Elsevier, 1996, pp. 2991–2998.
- [14] R. Saurel, O. Lemetayer, A multiphase model for compressible flows with interfaces, shocks, detonation waves and cavitation, *Journal of Fluid Mechanics* 431 (2001) 239–271.
- [15] B. Riemann, Über die Fortpflanzung ebener Luftwellen von endlicher Schwingungsweite, *Verlag der Dieterichschen Buchhandlung*, 1860.
- [16] S. K. Godunov, A difference method for numerical calculation of discontinuous solutions of the equations of hydrodynamics, *Matematicheskii Sbornik* 89 (3) (1959) 271–306.
- [17] E. F. Toro, M. Spruce, W. Speares, Restoration of the contact surface in the HLL-Riemann solver, *Shock Waves* 4 (1) (1994) 25–34.
- [18] H. Nishikawa, K. Kitamura, Very simple, carbuncle-free, boundary-layer-resolving, rotated-hybrid Riemann solvers, *Journal of Computational Physics* 227 (4) (2008) 2560–2581.
- [19] R. Courant, K. Friedrichs, H. Lewy, Über die partiellen Differenzengleichungen der mathematischen Physik, *Mathematische Annalen* 100 (1) (1928) 32–74.
- [20] P. Lax, B. Wendroff, Systems of conservation laws, *Communications on Pure and Applied mathematics* 13 (2) (1960) 217–237.
- [21] R. J. LeVeque, *Finite volume methods for hyperbolic problems*, Vol. 31, Cambridge university press, 2002.
- [22] A. Jameson, W. Schmidt, E. Turkel, et al., Numerical solutions of the Euler equations by finite volume methods using Runge-Kutta time-stepping schemes, *AIAA paper* 1259.
- [23] B. van Leer, Towards the ultimate conservative difference scheme. V. A second-order sequel to Godunov's method, *Journal of Computational Physics* 32 (1) (1979) 101–136.
- [24] E. Toro, A weighted average flux method for hyperbolic conservation laws, in: *Proceedings of the Royal Society of London A: Mathematical, Physical and Engineering Sciences*, Vol. 423, The Royal Society, 1989, pp. 401–

418.

- [25] E. Toro, S. Billett, Centred TVD schemes for hyperbolic conservation laws, *IMA Journal of Numerical Analysis* 20 (1) (2000) 47–79.
- [26] S. Osher, J. A. Sethian, Fronts propagating with curvature-dependent speed: algorithms based on Hamilton-Jacobi formulations, *Journal of Computational Physics* 79 (1) (1988) 12–49.
- [27] S. Osher, R. P. Fedkiw, Level set methods: an overview and some recent results, *Journal of Computational physics* 169 (2) (2001) 463–502.
- [28] R. P. Fedkiw, T. Aslam, B. Merriman, S. Osher, A non-oscillatory Eulerian approach to interfaces in multimaterial flows (the ghost fluid method), *Journal of Computational Physics* 152 (2) (1999) 457–492.
- [29] T. Liu, B. Khoo, K. Yeo, Ghost fluid method for strong shock impacting on material interface, *Journal of Computational Physics* 190 (2) (2003) 651–681.
- [30] S. K. Sambasivan, H. Udaykumar, Ghost fluid method for strong shock interactions part 1: Fluid-fluid interfaces, *AIAA Journal* 47 (12) (2009) 2907–2922.
- [31] C. Wang, T. Liu, B. Khoo, A real ghost fluid method for the simulation of multimediuim compressible flow, *SIAM Journal on Scientific Computing* 28 (1) (2006) 278–302.
- [32] E. F. Toro, *Riemann solvers and numerical methods for fluid dynamics: a practical introduction*, Springer Science & Business Media, 2013.
- [33] G. A. Sod, A survey of several finite difference methods for systems of nonlinear hyperbolic conservation laws, *Journal of Computational Physics* 27 (1) (1978) 1–31.
- [34] P. Woodward, P. Colella, The numerical simulation of two-dimensional fluid flow with strong shocks, *Journal of Computational Physics* 54 (1) (1984) 115–173.
- [35] S. Karni, Hybrid multifluid algorithms, *SIAM Journal on Scientific Computing* 17 (5) (1996) 1019–1039.
- [36] J. J. Quirk, S. Karni, On the dynamics of a shock-bubble interaction, *Journal of Fluid Mechanics* 318 (1996) 129–163.
- [37] S.-P. Wang, M. H. Anderson, J. G. Oakley, M. L. Corradini, R. Bonazza, A thermodynamically consistent and fully conservative treatment of contact discontinuities for compressible multicomponent flows, *Journal of Computational Physics* 195 (2) (2004) 528–559.
- [38] A. Chinnayya, E. Daniel, R. Saurel, Modelling detonation waves in heterogeneous energetic materials, *Journal of Computational Physics* 196 (2) (2004) 490–538.

Appendix A. Results for Toro’s tests

The results for all five of Toro’s tests are shown in Figures A.14-A.18 on the following pages.

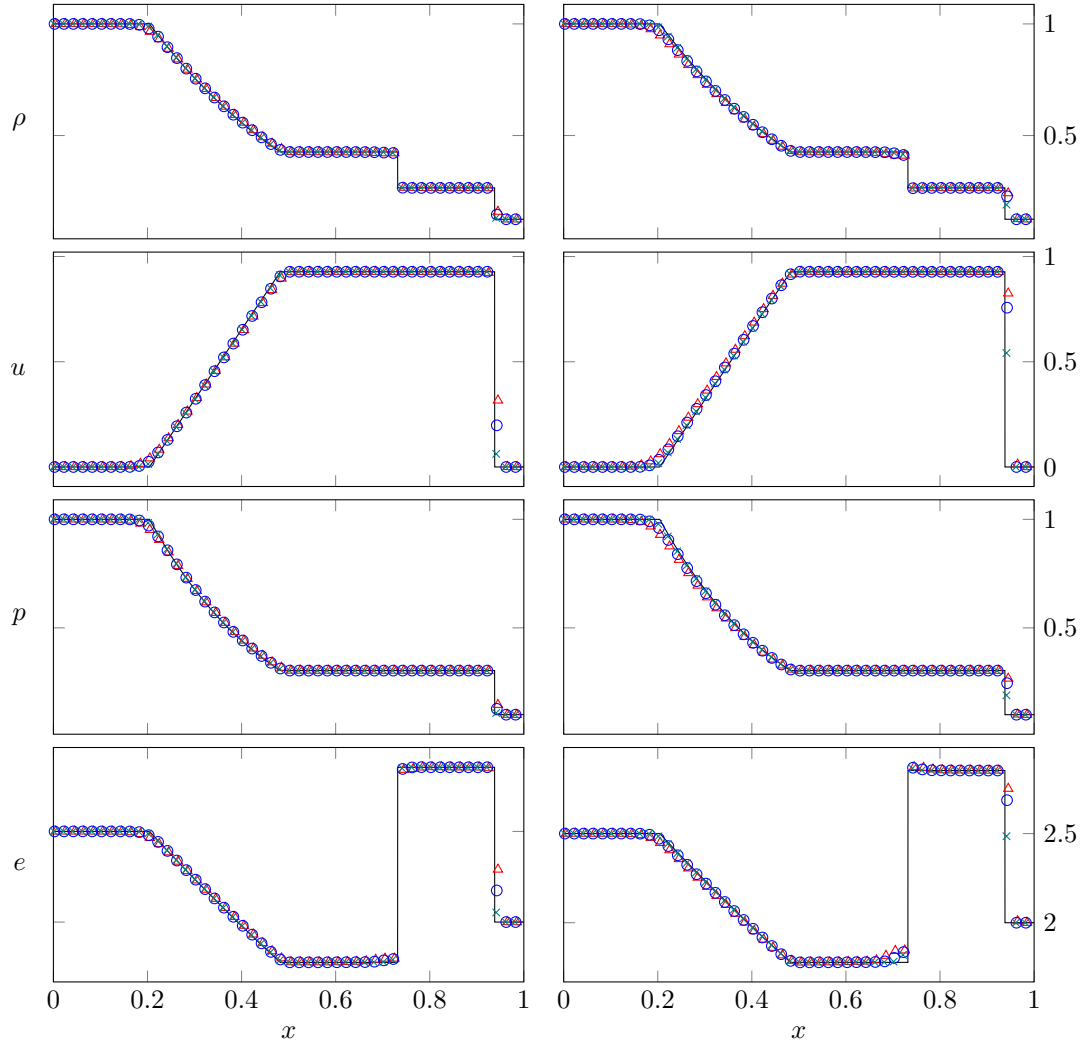


Figure A.14: Results for Toro's test 1 using OGFM (left) and MGFM (right). $\triangle N = 100$ $\circ N = 200$ $\times N = 400$

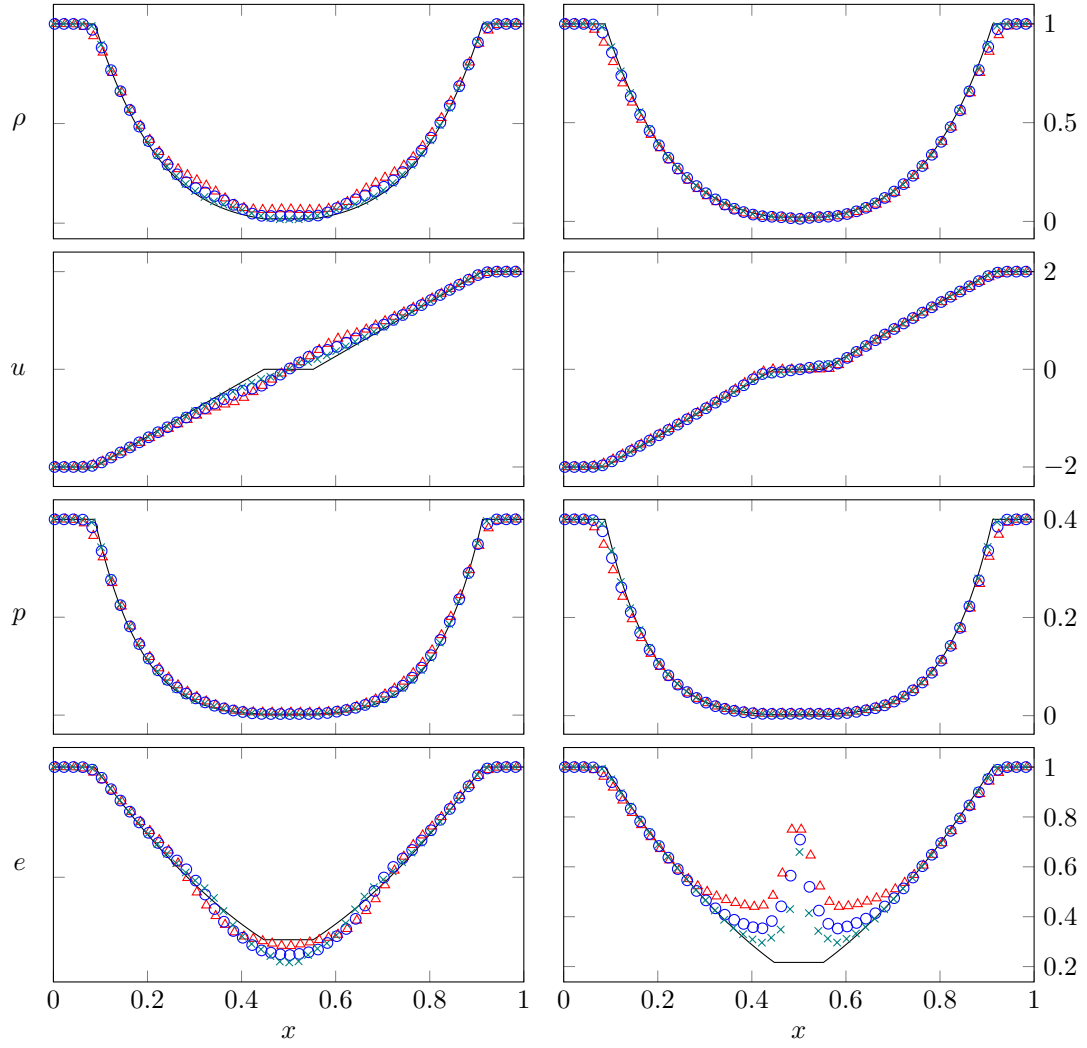


Figure A.15: Results for Toro's test 2 using OGFM (left) and MGFM (right). $\triangle N = 100$ $\circ N = 200$ $\times N = 400$

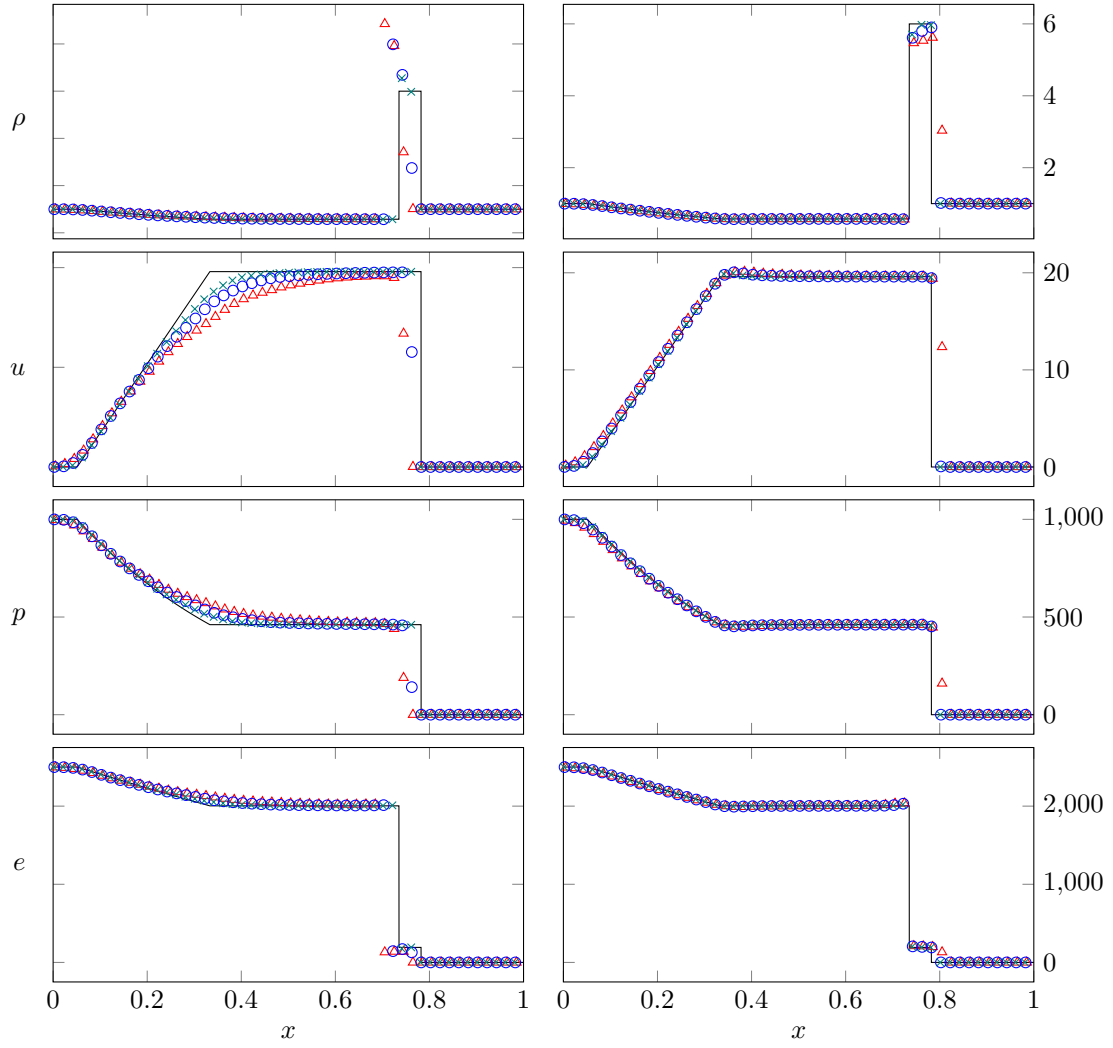


Figure A.16: Results for Toro's test 3 using OGFM (left) and MGFM (right). $\triangle N = 100$ $\circ N = 200$ $\times N = 400$

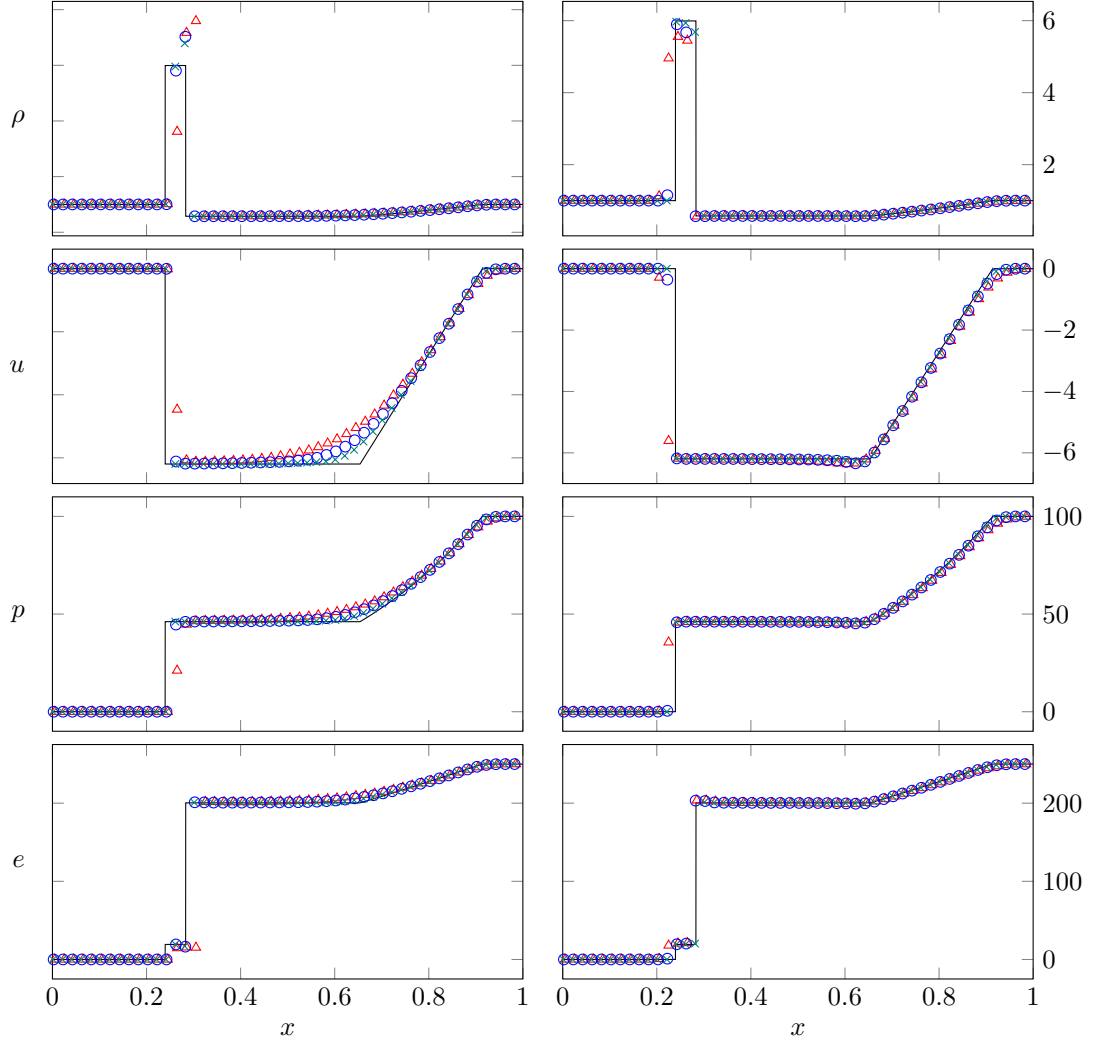


Figure A.17: Results for Toro's test 4 using OGFM (left) and MGFM (right). $\triangle N = 100$ $\circ N = 200$ $\times N = 400$

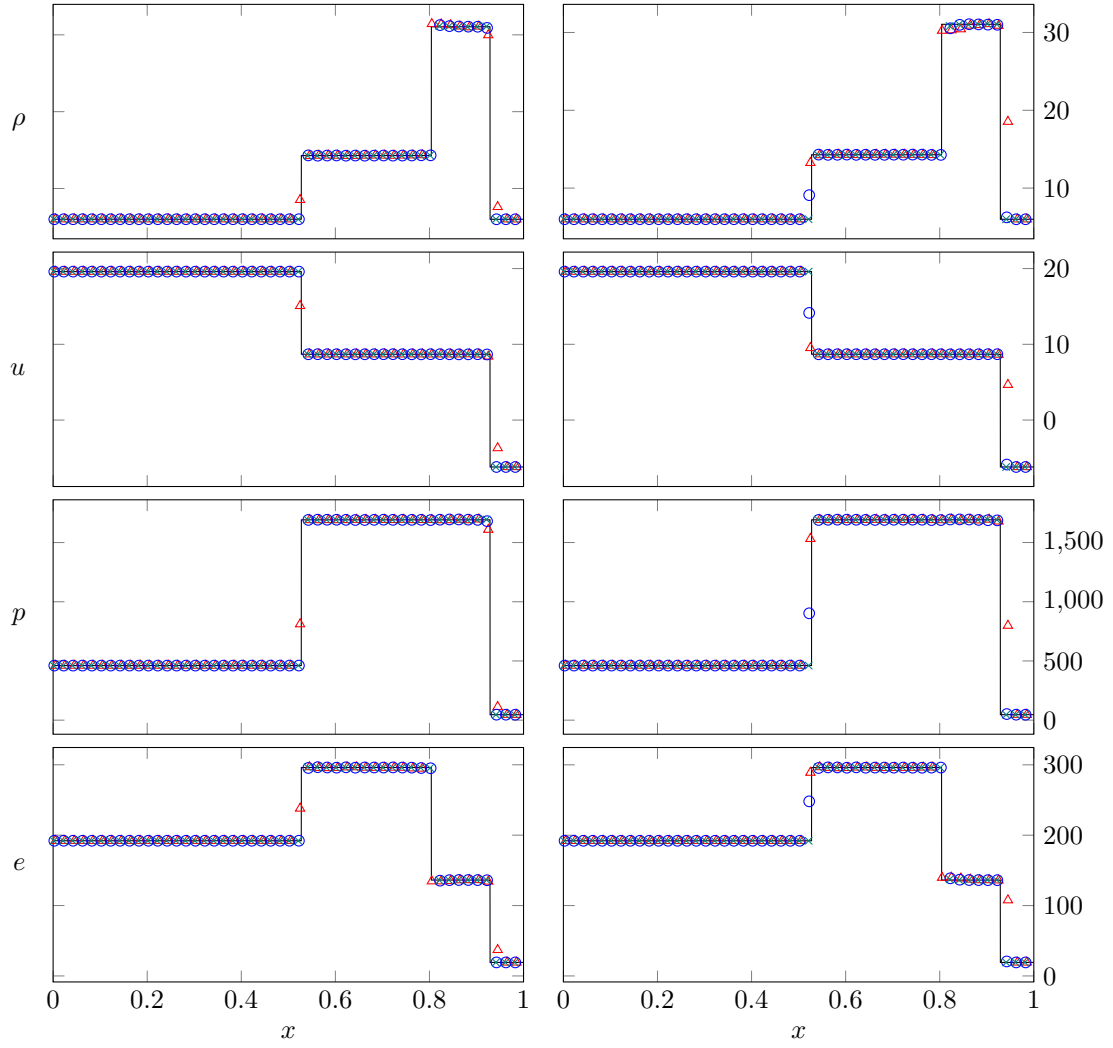


Figure A.18: Results for Toro's test 5 using OGFM (left) and MGFM (right). $\triangle N = 100$ $\circ N = 200$ $\times N = 400$