A Second Order Primitive Preconditioner For Solving All Speed Multi-Phase Flows

Samet Y. Kadioglu*, Mark Sussman†, Stanley Osher‡,
Joseph P. Wright§, and Myungjoo Kang¶

February 25, 2005

Abstract

In this paper, we present a new second order primitive preconditioner technique for solving all speed multi-phase flow problems. With this technique, one can compute both compressible and incompressible flows with Mach-uniform accuracy and efficiency (i.e., accuracy and efficiency of the method are independent of Mach number). The new primitive preconditioner can handle both strong and weak shocks, providing highly resolved shock solutions together with correct shock speeds. In addition, the new technique performs very well at the zero Mach limit. In the case of multi-phase flow, the new primitive preconditioner technique enables one to accurately treat phase boundaries in which there is a large impedance mismatch. The present method is tested on a variety of problems from low (low speed) to high Mach number (high speed) flows including multi-phase flow tests, i.e, computing the growth and collapse of adiabatic bubbles for study of underwater explosions. The numerical results show that the newly proposed method supersedes existing up-to-date numerical techniques in its category.

Key Words: Preconditioner; All speed flow; Multi-phase flow; Compressible flow; Incompressible flow ; Mach-uniform; Underwater explosion

*e-mail:skadio@math.fsu.edu
†e-mail:sussman@math.fsu.edu
‡e-mail:sjo@levelset.com
§e-mail:wright@wai.com (Weidlinger Associates, Inc.)
¶e-mail:mkang@levelset.com

1 Introduction

We are interested in solving all speed flow phenomena including multi-material flow systems. We will say that a flow is weakly compressible if the Mach number satisfies $0 < M \leq 0.2$, and compressible if $M > 0.2$. There are many applications in fluid dynamics where compressible and weakly compressible flow occur simultaneously [52]. A possible short list of such applications could be given as (1)solving flow problems in the inlet of internal combustion engines; the flow is weakly compressible in the bulk of the inlet channel and compressible near the valve, (2)predicting flow fields around aircrafts during take-offs and landings; the flow shows compressible regimes around the wings and weakly compressible regimes around the rest of the body, and (3)(the topic of our current research) solving adiabatic bubble growth and collapse to simulate underwater explosions; the numerical treatment in water requires a compressible treatment around the shock front, and water behaves as a weakly compressible fluid elsewhere. There exist a variety of good numerical techniques designed specifically for compressible or specifically for incompressible flows. The difficulty arises when we use such numerical techniques in the case when both compressible and weakly compressible flows exist at the same time. For instance, if we apply a standard density-based compressible formulation to solve weakly compressible flows, we must expect loss of accuracy and efficiency due to weak coupling between pressure and density [18, 51, 60]. Such compressible methods are generally based on explicit time integration methods that impose severe CFL (Courant Friedreichs Lewy) restrictions for stability. Applying these kinds of schemes to low Mach number flows without special treatment will lead to impractical computations (due to the large sound speed), especially for three-dimensional problems. The situation is even worse when we tackle multi-phase flow problems. Unphysical pressure oscillations and other computational inaccuracies are often reported at or near material interfaces [22, 33]. This typically occurs with large density ratios across a material interface (e.g., 1:1000 for air-water), or with a "stiff" equation of state on one side of the interface. One way to overcome these difficulties is to introduce a unified numerical procedure for computing both compressible and incompressible flows with Mach-uniform accuracy and efficiency (i.e., the accuracy and efficiency of the scheme are independent of the Mach numbers from subsonic to supersonic values [46]). The procedure must include an effective way of eliminating unphysical oscillations associated with multi-material discontinuities.

There are two approaches for developing a unified numerical technique for multi-material flow. The first approach extends the functionality of explicit methods in order to effectively handle low Mach number flows. The second approach extends the functionality of semi-implicit (e.g. projection methods or all-speed methods) methods in order to effectively handle flows with strong shocks.

Of the unified methods that fall under the first approach, the most popular methods are the so-called explicit preconditioning techniques [42, 43, 50, 25]. These techniques involve multiplying the time derivative part of the system of the compressible equations by a suitable preconditioning matrix, and then solving the resulting equations. However there are drawbacks. First of all, since only the time derivatives are modified, temporal

accuracy is lost. And due to their explicit nature, they are still subject to severe CFL restrictions. Furthermore, the preconditioning matrix may become singular in the limit $M \downarrow 0$, and this impairs the robustness of the computation [34]. An alternative unified method that attempts to extend the functionality of an explicit approach to low-Mach number flows is based on asymptotic expansions of the governing equations in terms of Mach number [14, 23, 24]. Unfortunately, this methodology is applicable only for small Mach numbers (e.g., $M \leq 0.2$).

As an alternative to unified methods which extend functionality of high Mach number schemes to low-Mach number regimes, one can extend the functionality of a low-Mach number, semi-implicit, scheme to be able to handle high Mach number flow. One of the earliest examples in this class is Harlow and Amsden [16]. In their original work known as ICE (Implicit Continuous Eulerian), they pointed out that one has to separate out the incompressible part of the flow and treat it implicitly. Implicit treatment relaxes the severe CFL restrictions, especially in low Mach regions. Detailed information about ICE methods can be found in [16, 8, 18] and the references therein. ICE methods perform well for flows involving moderate shocks, but ICE is problematic when one wants to compute strong shocks. This is because their nonconservative method leads to large conservation errors. Yet another nonconservative approach in this class is Colella and Pao [11]. This method is based on Hodge decomposition of flow variables [10]. In other words, they decompose the velocity field into two orthogonal components; one being the divergence free part and the other orthogonal component being the curl-free part. Such a treatment enables a natural extension from incompressible flow, in which there is no curl-free component, to compressible flow. Although they obtained reasonably accurate results for low Mach number flows, they did not report results for high Mach numbers. Recently, a conservative approach is taken by van der Heul *et al* [45, 46] generalizing the MAC (Marker-and-Cell) method of Harlow and Welch [17]. This scheme is based on pressure correction on a staggered grid. The main theme in [45, 46] is essentially to improve the shock capturing ability of earlier schemes [6, 53, 54]. In [6, 53, 54], they discretized the continuity and momentum equations in conservation form, but they used a nonconservative discretization for the energy equation. As a result, such nonconservative representation caused inaccurate calculation of the shock speed. So in [45, 46], they extended their nonconservative schemes [6, 53, 54] to a fully conservative one. Although they achieved some improvements, their method still suffers from unphysical oscillations near the shocks.

All of the methods mentioned above are designed to solve single-fluid flow problems. When we have a multi-fluid system, we need not only a Mach-uniform scheme but also an effective way of eliminating unphysical oscillations near multi-material discontinuities. In addition, we need a good interface capturing technique for accurate representation of surfaces between different materials (i.e., gas with a liquid or solid). According to Lax theorem [27], the best flow description for the hyperbolic conservation laws is provided through the conservative formulation of the governing equations. Karni [22] noted that this may not be true in the case when we solve multi-phase flow problems. In fact, she showed that conservative discretization gives rise to unphysical pressure oscillations

near material interfaces (contact discontinuities). Alternatively, she introduced non-conservative (primitive) models to capture oscillation-free contact discontinuities [22]. Later Abgrall *et al* [1, 2] improved Karni's work by introducing a quasi-conservative approach. Their work is based on the GFM (Ghost Fluid Method) of Fedkiw *et al* [12, 13], i.e, they solve single-fluid problem for each fluid by defining ghost points on each side, and then they reconstruct the material interface by solving the level set equation. There are several other multi-material formulations [19, 47, 48], but none of them including Karni's and Abgrall's methods are all speed. Now we focus our attention on all speed multi-phase flow solvers with good shock capturing ability and good multi-material discontinuity representations. Yabe *et al* [33, 57, 58, 59, 60, 61] have put substantial effort in this direction. They utilize a CIP (Cubic Interpolated Polynomial [41, 58]) based time splitting predictor-corrector technique. In other words, they separate the governing equations into advection and non-advection parts by the time splitting method [9], then they solve the advection terms including the interface advection by the CIP method, finally they employ a pressure-based predictor-corrector method to update the flow variables. Their early works are based on a primitive formulation of both the fluid and interface equations [57, 61]. These early methods [57, 61] worked reasonably well for solving low speed flows and gave fairly good multi-material solutions, but they failed to calculate correct shock speeds due to their purely primitive formulations. More recently, they added artificial viscosity [33, 60] and also adopted partially or fully conservative procedures [59, 56] to calculate correct shock solutions, but they still have over and undershoots (Gibbs phenomenon) at shocks and contact discontinuities. In addition, the fully conservative approach [56] is untested for low speed or multi-phase flows, and more importantly these methods are only first order accurate in time.

A review of the literature cited above clearly indicates that an efficient and accurate method including good shock and multi-material discontinuity representations for all speed multi-phase flows has not been developed yet. We know that fully conservative methods create problems at contact discontinuities [22], and primitive (non-conservative) methods fail to calculate correct shock speeds [57, 61]. On the other hand, conservative techniques have the advantage of capturing correct shock speeds and primitive techniques are good at representing accurate multi-material discontinuities. Thus we want to use the advantages of both representations and introduce a new unified numerical procedure that will handle flows at any speed and provide accurate interface representations in the case of multi-phase flow phenomena. Our method consists of two steps. At the first step we solve the governing equations by one of our favorite explicit solvers, and at the second step (preconditioning step) we correct the explicitly calculated fluid variables by an implicit correction algorithm. The explicit step provides sharp shock capturing together with correct shock speeds, and the implicit correction step (the preconditioning step) enables us to use relatively large time steps, compared to typical explicit CFL restrictions, together with better material interface representations. The idea behind our method is similar to [37] in that we treat the linear part implicitly and the nonlinear part explicitly. We note that the explicit step, as a part of multi-material flows, can be performed separately for each material without regard to material boundary conditions,

i.e, the ghost fluid treatments [12, 13, 3, 4, 32] that respect material jump conditions are unnecessary for the explicit part. Instead, the pressure and velocity continuity conditions are enforced during the preconditioning step.

The outline of the present paper is as follows. In Section 2, we present the mathematical formulation of the problem. In Section 3, we give the general structure of the preconditioner including two examples of explicit methods which are to be preconditioned. In Section 4, in order to validate the proposed method, numerical simulations are performed on a variety of problems from low (low speed) to high Mach number (high speed) flows including multi-phase flow tests, i.e, computing the growth and collapse of adiabatic bubbles for study of underwater explosions, and computing oscillating water column problem in a 1-D tube. In Section 5, some concluding remarks are provided.

2 The Mathematical Formulation

We will present our numerical procedure starting with single-fluid flow models, then in the later section we will extend the numerical method to multi-phase flow problems. As a single-fluid flow model, we consider the one dimensional time dependent inviscid Euler equations for representing the dynamics of one dimensional inviscid compressible flows. The one dimensional Euler equations in conservation form can be written as the non-linear hyperbolic system of partial differential equations, i.e,

$$\frac{\partial U}{\partial t} + \frac{\partial F(U)}{\partial x} = 0, \tag{1}$$

where

$$U = \begin{pmatrix} \rho \\ m \\ E \end{pmatrix},$$

and

$$F(U) = \begin{pmatrix} m \\ um + p \\ uE + up \end{pmatrix},$$

where ρ, u, p, and E denotes density, velocity, pressure, and the total energy per unit volume, m denotes momentum $m = \rho u$, and $E = \rho e + \frac{1}{2}\rho u^2$ with e denoting the internal energy per unit mass. For perfect gases, the thermodynamical relation (the equation of state) is

$$e = \frac{p}{\rho(\gamma - 1)}, \tag{2}$$

where γ is the specific heat ratio.

3 The Second Order Primitive Preconditioner

We give the general structure of our second order primitive preconditioner algorithm which is based on the implicit correction of an explicit building block of the Euler equations in general form.

Basic structure of the algorithm:

Define $p^{n+1,0} = p^n, \rho^{n+1,0} = \rho^n$, and $\mathbf{u}^{n+1,0} = \mathbf{u}^n$,
Given $p^{n+1,k-1}, \rho^{n+1,k-1}$, and $\mathbf{u}^{n+1,k-1}$,
For k=1,...,M
Step 1) Explicit building block
Calculate $p^{exp,k}, \mathbf{u}^{exp,k}, \rho^{exp,k}$, and $e^{exp,k}$ by an explicit solver, (solving (1) either in conservation form or primitive form)
Step 2) Implicit correction step
Update $p^{n+1,k}, \mathbf{u}^{n+1,k}, \rho^{n+1,k}, e^{n+1,k}, m^{n+1,k}$, and $E^{n+1,k}$ by the second order primitive preconditioner.
end

In order to execute Step 2) in the flow chart, we have (in multi-dimensional notation)

$$\frac{p^{n+1,k} - p^{exp,k}}{\Delta t} = -\eta (c^{n+1,k-1})^2 \rho^{n+1,k-1} [\nabla \cdot \mathbf{u}^{n+1,k} - \nabla \cdot \mathbf{u}^{n+1,k-1}], \tag{3}$$

$$\frac{\mathbf{u}^{n+1,k} - \mathbf{u}^{exp,k}}{\Delta t} = -\eta \frac{1}{\rho^{n+1,k-1}} [\nabla p^{n+1,k} - \nabla p^{n+1,k-1}], \tag{4}$$

where c is the speed of sound

$$c^2 = \frac{dp}{d\rho}, \tag{5}$$

superscript k represents the iteration number, $(n+1,k)$ and $(n+1,k-1)$ denote the newly computed and current values of the related field variables, (exp,k) represents the computed explicit values for which we will give two different examples, one conservative and one non-conservative (in Sections 3.1 and 3.2), and η is a constant, either $\eta = 0$ or $\eta = 1$ to automatically switch the scheme to an explicit or implicit one. If we take the divergence of the velocity correction equation (4) and replace $\nabla \cdot \mathbf{u}^{n+1,k}$ in the pressure correction equation (3), we obtain the following elliptic system involving the new pressure field,

$$\eta^2\Delta t^2\nabla\cdot[\frac{\nabla p^{n+1,k}}{\rho^{n+1,k-1}}]-\frac{1}{(c^{n+1,k-1})^2\rho^{n+1,k-1}}p^{n+1,k} = \eta\Delta t[\nabla\cdot\mathbf{u}^{exp,k}-\nabla\cdot\mathbf{u}^{n+1,k-1}] - \frac{1}{(c^{n+1,k-1})^2\rho^{n+1,k-1}}p^{exp,k} + \eta^2\Delta t^2\nabla\cdot[\frac{\nabla p^{n+1,k-1}}{\rho^{n+1,k-1}}]. \quad (6)$$

After solving this elliptic system, we immediately update the velocity field. Using the newly computed pressure and velocity fields, we then compute the other primitive variables by using the following approximations,

$$\frac{\rho^{n+1,k}-\rho^{exp,k}}{\Delta t} = -\eta\rho^{n+1,k-1}[\nabla\cdot\mathbf{u}^{n+1,k}-\nabla\cdot\mathbf{u}^{n+1,k-1}], \quad (7)$$

$$\frac{e^{n+1,k}-e^{exp,k}}{\Delta t} = -\eta\frac{p^{n+1,k-1}}{\rho^{n+1,k-1}}[\nabla\cdot\mathbf{u}^{n+1,k}-\nabla\cdot\mathbf{u}^{n+1,k-1}], \quad (8)$$

and finally we update the momentum and total energy by using

$$m^{n+1,k} = \rho^{n+1,k}\mathbf{u}^{n+1,k}, \quad (9)$$

$$E^{n+1,k} = \rho^{n+1,k}e^{n+1,k}+\frac{1}{2}\rho^{n+1,k}|\mathbf{u}^{n+1,k}|^2. \quad (10)$$

3.1 Example of a conservative explicit building block

In this section, we describe how to obtain the explicit terms (the $(exp,k)^{th}$ values) conservatively for the second order primitive preconditioner algorithm when solving all speed flows with shock capturing. The conservative explicit solver uses local Lax Friedrich numerical fluxes together with a second order ENO extrapolation method in order to obtain shock solutions with high resolution. Our method is in the same spirit as the second order ENO central schemes developed by [31] in that we do not use field-by-field decomposition in order to extrapolate the characteristic variables. We remark that our proposed preconditioner is flexible in that we can use as an alternative explicit building block, the less diffusive ENO/WENO schemes [35, 36, 44, 20, 30] or third order central schemes [29]. We consider the one dimensional Euler equations (1) and use the following conservative discretization,

$$\frac{U_i^{(exp,k)}-U_i^{(n+1,0)}}{\Delta t} = -\frac{F_{i+\frac{1}{2}}^{LLF,\frac{(n+1,k-1)+(n+1,0)}{2}}-F_{i-\frac{1}{2}}^{LLF,\frac{(n+1,k-1)+(n+1,0)}{2}}}{\Delta x}, \quad (11)$$

where superscript $\frac{(n+1,k-1)+(n+1,0)}{2}$ represents the simple average of the $(k-1)^{th}$ iterated and the initial values of related flow variables, and $F_{i+\frac{1}{2}}^{LLF}$ denotes the local Lax Friedrich numerical flux function, i.e,

$$
\begin{aligned}
F^{LLF,\frac{(n+1,k-1)+(n+1,0)}{2}}_{i+\frac{1}{2}} &= F^{LLF}(U^{R,\frac{(n+1,k-1)+(n+1,0)}{2}}_{i+\frac{1}{2}}, U^{L,\frac{(n+1,k-1)+(n+1,0)}{2}}_{i+\frac{1}{2}}) \qquad (12)\\
&= \frac{1}{2}[F(U^{R,\frac{(n+1,k-1)+(n+1,0)}{2}}_{i+\frac{1}{2}}) + F(U^{L,\frac{(n+1,k-1)+(n+1,0)}{2}}_{i+\frac{1}{2}})]\\
&\quad -\frac{1}{2}\alpha_{i+\frac{1}{2}}[U^{R,\frac{(n+1,k-1)+(n+1,0)}{2}}_{i+\frac{1}{2}} - U^{L,\frac{(n+1,k-1)+(n+1,0)}{2}}_{i+\frac{1}{2}}].
\end{aligned}
$$

Here

$$\alpha_{i+\frac{1}{2}} = \max_p |\lambda^{p(L,R)}_{i+\frac{1}{2}}|, \qquad p = 1,2,3 \qquad (13)$$

with $\lambda^{p(L,R)}_{i+\frac{1}{2}}$'s denoting the left or right values of the eigenvalues of the system, and $U^L_{i+\frac{1}{2}}$, $U^R_{i+\frac{1}{2}}$ represent the left and right values of the state variables respectively. $U^L_{i+\frac{1}{2}}$ and $U^R_{i+\frac{1}{2}}$ are defined as

$$U^L_{i+\frac{1}{2}} = U_i + \frac{\Delta x}{2} U_{x,i}, \qquad (14)$$

$$U^R_{i+\frac{1}{2}} = U_{i+1} - \frac{\Delta x}{2} U_{x,i+1}, \qquad (15)$$

where

$$U_{x,i} = \begin{cases} \frac{U_{i+1}-U_i}{\Delta x} & \text{if } |U_{i+1}-U_i| < |U_i - U_{i-1}| \\ \frac{U_i - U_{i-1}}{\Delta x} & \text{otherwise.} \end{cases} \qquad (16)$$

3.2 Example of a non-conservative explicit building block

In this section, we give an alternative way of obtaining the explicit terms for the second order primitive preconditioner technique when we solve zero Mach limit flows which are dominated by acoustic waves or flows that contains no shock discontinuity. The main idea is to give up the conservative form of the Euler equations in the explicit step whenever there is no shock discontinuity in the flow. Instead, we solve the primitive Euler equations in the explicit step, and then we apply the implicit correction algorithm. The primitive Euler equations are

$$\rho_t + u\rho_x + \rho u_x = 0 \qquad (17)$$

$$u_t + uu_x + \frac{1}{\rho} p_x = 0 \qquad (18)$$

$$p_t + up_x + c^2 \rho u_x = 0. \qquad (19)$$

We discretize the primitive Euler equations as

$$
\begin{aligned}
\frac{\rho_i^{exp,k} - \rho_i^{n+1,0}}{\Delta t} &= -u_i^{\frac{(n+1,k-1)+(n+1,0)}{2}} \frac{<\rho>^{\frac{(n+1,k-1)+(n+1,0)}{2}}_{i+\frac{1}{2}} - <\rho>^{\frac{(n+1,k-1)+(n+1,0)}{2}}_{i-\frac{1}{2}}}{\Delta x} \qquad (20)\\
&\quad -\rho_i^{\frac{(n+1,k-1)+(n+1,0)}{2}} \frac{u^{\frac{(n+1,k-1)+(n+1,0)}{2}}_{i+\frac{1}{2}} - u^{\frac{(n+1,k-1)+(n+1,0)}{2}}_{i-\frac{1}{2}}}{\Delta x},
\end{aligned}
$$

$$\frac{u_i^{exp,k} - u_i^{n+1,0}}{\Delta t} = -u_i^{\frac{(n+1,k-1)+(n+1,0)}{2}} \frac{< u >_{i+\frac{1}{2}}^{\frac{(n+1,k-1)+(n+1,0)}{2}} - < u >_{i-\frac{1}{2}}^{\frac{(n+1,k-1)+(n+1,0)}{2}}}{\Delta x} \quad (21)$$
$$- \frac{1}{\rho_i^{\frac{(n+1,k-1)+(n+1,0)}{2}}} \frac{p_{i+1}^{\frac{(n+1,k-1)+(n+1,0)}{2}} - p_{i-1}^{\frac{(n+1,k-1)+(n+1,0)}{2}}}{2\Delta x},$$

and

$$\frac{p_i^{exp,k} - p_i^{n+1,0}}{\Delta t} = -u_i^{\frac{(n+1,k-1)+(n+1,0)}{2}} \frac{< p >_{i+\frac{1}{2}}^{\frac{(n+1,k-1)+(n+1,0)}{2}} - < p >_{i-\frac{1}{2}}^{\frac{(n+1,k-1)+(n+1,0)}{2}}}{\Delta x} \quad (22)$$
$$- c_i^2 \rho_i^{\frac{(n+1,k-1)+(n+1,0)}{2}} \frac{u_{i+\frac{1}{2}}^{\frac{(n+1,k-1)+(n+1,0)}{2}} - u_{i-\frac{1}{2}}^{\frac{(n+1,k-1)+(n+1,0)}{2}}}{\Delta x}.$$

Here

$$u_{i+\frac{1}{2}}^{\frac{(n+1,k-1)+(n+1,0)}{2}} = \frac{u_{i+1}^{\frac{(n+1,k-1)+(n+1,0)}{2}} + u_i^{\frac{(n+1,k-1)+(n+1,0)}{2}}}{2}, \quad (23)$$

and

$$< V >_{i+\frac{1}{2}} = \begin{cases} V_i + \frac{\Delta x}{2} V_{x,i} & \text{if } u_{i+\frac{1}{2}} > 0 \\ V_{i+1} - \frac{\Delta x}{2} V_{x,i+1} & \text{if } u_{i+\frac{1}{2}} < 0, \end{cases} \quad (24)$$

where $V_{i+\frac{1}{2}}$ denotes one of the primitive variables, i.e, ρ, u, or p and $V_{x,i}$ is defined in the same way as (16).

This procedure relaxes the CFL condition significantly, i.e, it allows us to use time steps that are $fifteen$ times larger than the typical explicit CFL condition. We remark that it is critical that in order for one to take such large time steps, determined by the condition, $|u|_{max}\Delta t < \Delta x$, the u_x terms in the primitive Euler equations must be discretized by central differencing. We also remark that Yabe's CIP scheme [58, 61] is equivalent to our preconditioning scheme except that the temporal accuracy of our scheme is second order.

4 Numerical results

The numerical examples are presented in two different subsections. The first part is dedicated to single-fluid flow problems ranging from low to high Mach numbers. In the second part, we provide numerical results of all speed multi-phase flow computations. For calculations in Section 4.1.1 through Section 4.1.7 we used M=3 (the number of iterations) , and for all other calculations we used M=2. We used $\eta = 1$ for all computations in this paper.

4.1 Application of the second order primitive preconditioner technique to single-fluid flow problems

For the single-fluid flow case, we solve similar test problems as in [56, 46] in order to explore the resolution gained by our proposed preconditioner.

4.1.1 Sod's shock tube problem

The shock tube problem considers a long, thin, cylindrical tube containing a gas separated by a thin membrane. The gas is assumed to be at rest on both sides of the membrane, but it has different constant pressures and densities on each side. At time $t = 0$, the membrane is ruptured, and the problem is to determine the ensuing motion of the gas. This problem is first studied by Riemann, and known as the Riemann problem. The solution to this problem consists of a shock wave moving into the low pressure region, a rarefaction wave that expands into the high pressure region, and a contact discontinuity which represents the interface.

The shock tube problem of Sod [38] considers the Riemann problem with the following initial data on the interval $0 \leq x \leq 1$,

$$(\rho(x,0), u(x,0), p(x,0)) = \begin{cases} (1,0,1) & \text{if } x \leq 0.5 \\ (0.125,0,0.1) & \text{if } x > 0.5. \end{cases} \tag{25}$$

When producing computational results for this test problem, we used 400 grid cells and the CFL ($|u|_{max}\Delta t$ <CFLΔx) number 0.3. And we used the conservative explicit building block given in Section 3.1. Our numerical results indicate higher resolved solutions for a given time step and given mesh size than the numerical results reported in [56, 46]. Our results, shown in Figure 1, have no spurious oscillations at any shock or contact discontinuities. A conservation study is done for the mass and total energy with different mesh resolutions. Table 1 shows that the conservation errors are within acceptable ranges. Table 1 also shows that the conservation property of our method improves when more iterations are performed.

4.1.2 Lax's shock tube problem

We present the calculations for the Riemann problem,

$$(\rho(x,0), u(x,0), p(x,0)) = \begin{cases} (0.445, 0.698, 3.528) & \text{if } x \leq 0.5 \\ (0.5,0,0.571) & \text{if } x > 0.5, \end{cases} \tag{26}$$

used by Lax in [28]. The computational domain is taken to be $[0,1]$. The grid resolution is 400 cells and the time stepping condition is CFL = 0.3 ($|u|_{max}\Delta t$ <CFLΔx). The conservative explicit building block (Section 3.1) is used in here. For this problem, again as in the previous test case, our results indicate higher resolution than [46] and comparable resolution as [56] for a given mesh size and time step. Figure 2 illustrates well resolved shock and contact solutions.

4.1.3 Strong shock tube problem

Here we present a strong shock tube problem with the following initial condition,

$$(\rho(x,0), u(x,0), p(x,0)) = \begin{cases} (1, 0, 10^{10}) & \text{if } x \leq 0.5 \\ (0.125, 0, 0.1) & \text{if } x > 0.5. \end{cases} \tag{27}$$

The computational domain is taken as $[0, 1]$ interval. For this problem, we use the conservative explicit building block (Section 3.1). As noted in [56], this initial condition creates a supersonic shock associated with extreme jumps in velocity and pressure. It is well known that purely non-conservative schemes fail to compute strong shocks due to their intrinsic inability to calculate correct shock speeds. This is also discussed in [56]. Xiao [56] computed correct shocks for this problem, yet there are still density overshoots in his results. Figure 3 shows the numerical results at $t = 2.5 \times 10^{-6}$. We used 400 grid cells during the computation while keeping the CFL ($|u|_{max}\Delta t$ <CFLΔx) at 0.3 as in [56]. As can be seen from Figure 3, we have no overshoots at any place while obtaining correct shock calculations.

4.1.4 Mach 3 shock test

The Mach 3 shock tube experiment uses the following initial conditions,

$$(\rho(x,0), u(x,0), p(x,0)) = \begin{cases} (3.857, 0.92, 10.333) & \text{if } x \leq 0.5 \\ (1, 3.55, 1) & \text{if } x > 0.5. \end{cases} \tag{28}$$

Here we use the conservative explicit building block (Section 3.1) with 400 grid cells and the CFL ($|u|_{max}\Delta t$ <CFLΔx) number 0.5 on the domain of $[0, 1]$ to produce the numerical results. Figure 4 indicates that our numerical results fairly compare to [56].

4.1.5 High Mach flow test

The High Mach number shock tube problem uses the following initial data on the interval $0 \leq x \leq 1$,

$$(\rho(x,0), u(x,0), p(x,0)) = \begin{cases} (10, 2000, 500) & \text{if } x \leq 0.5 \\ (20, 0, 500) & \text{if } x > 0.5. \end{cases} \tag{29}$$

The number of grid cells are 400 and the CFL ($|u|_{max}\Delta t$ <CFLΔx) number is 0.4 for this problem. The conservative explicit building block (Section 3.1) is used here. Figure 5 demonstrates well resolved shocks with correct shock locations and less smeared contact discontinuities than [56]. Here the Mach number can reach 240. This problem illustrates the robustness and stability of our method, even in the case of extremely high Mach number (high speed) flows.

4.1.6 Interaction of blast waves

In this subsection we present the numerical results for the problem of two interacting blast waves. This problem, introduced by Woodward and Colella [55], involves multiple interactions of strong shock waves and other discontinuities. Initial conditions are

$$(\rho(x,0), u(x,0), p(x,0)) = \begin{cases} (1, 0, 10^3) & \text{if } 0 \le x < 0.1 \\ (1, 0, 10^{-2}) & \text{if } 0.1 \le x < 0.9 \\ (1, 0, 10^2) & \text{if } 0.9 \le x < 1. \end{cases} \tag{30}$$

The boundary conditions at $x = 0$ and $x = 1$ are reflective solid wall conditions , i.e, we define auxiliary states $v^n_0, ..., v^n_{-r+1}$ for the left wall and $v^n_{M+1}, ..., v^n_{M+r}$ for the right wall by

$$\rho^n_{-i+1} = \rho^n_i, \qquad u^n_{-i+1} = -u^n_i, \qquad p^n_{-i+1} = p^n_i, \qquad for \;\; i = 1, ..., r \tag{31}$$

$$\rho^n_{M+i} = \rho^n_{M-i+1}, \qquad u^n_{M+i} = -u^n_{M-i+1}, \qquad p^n_{M+i} = p^n_{M-i+1}, \qquad for \;\; i = 1, ..., r \tag{32}$$

To make a consistent comparison with Xiao's [56] paper, we used the same number of grid cells 400 and 1600 cells to produce a numerical and a reference solution. We use the conservative explicit building block with CFL ($|u|_{max}\Delta t$ <CFLΔx) number 0.3 for this test problem. Figure 6 shows that we achieved more accurate results compared to [56] and observed no spurious oscillations which are noted in [56]. Here we note that there have been many reported higher resolution results for this problem, e.g, using ENO [35, 36, 44], but these previous results were based on using purely explicit non-preconditioned methods. We have observed with our new preconditioner that shock resolution is not adversely effected by the preconditioner itself. For instance if one uses high order ENO/WENO schemes [35, 36, 44, 20, 30] as the explicit building block, then we expect that our primitive preconditioning technique would produce similar shock resolution as explicit ENO/WENO methods [35, 36, 44, 20, 30]. This can be seen by considering the implicit correction equations in Section 3, i.e, consider the pressure correction equation,

$$\frac{p^{n+1,k} - p^{exp,k}}{\Delta t} = -c^2 \rho \nabla \cdot \delta U^{n+1,k} \tag{33}$$

where p^{exp} is produced by an explicit building block. We observed that the difference term $\nabla \cdot \delta U$ gets very small with the increasing iteration. This means that p^{n+1} gets closer to p^{exp} (i.e, $\nabla \cdot \delta U \to 0$ as $k \uparrow$, then $p^{n+1} \to p^{exp}$). This tells us that the implicit correction step spatially preserves the accuracy of the explicit building block. In other words, if one provides a high order explicit building block, then our method will spatially maintain the high order of accuracy of the given explicit building block.

4.1.7 Two symmetric rarefaction waves

Here we present the two symmetric rarefaction waves test problem. The initial conditions, on the $0 \leq x \leq 1$ interval, are

$$(\rho(x,0), u(x,0), p(x,0)) = \begin{cases} (1, -2, 0.4) & \text{if } x \leq 0.5 \\ (1, 2, 0.4) & \text{if } x > 0.5. \end{cases} \tag{34}$$

Our results are comparable to Xiao's [56] results. Figure 7 shows that there are some computational inaccuracies observed in the flow fields as in [56], when using the conservative explicit building block. For instance, we have overestimation in the pressure profile near the the expansion center, an unphysical pulse in the internal energy field near the center of the low pressure region, and inaccurate velocity representation. These inaccuracies are improved significantly (Figure 8) simply by using the second version of our algorithm, i.e, we precondition the non-conservative Euler equations (using non-conservative explicit building block, Section 3.2) at the implicit correction step. We note that the improvements with using the non-conservative explicit building block are due to the fact that there doesn't exist any shock discontinuity in the flow.

4.1.8 Smooth flow test (Mach zero limit)

In this section, we show the results of the zero Mach limit flows which are dominated by acoustic waves. We use the following smooth functions as the initial conditions,

$$\begin{aligned} u(x,0) &= 0 \\ p(x,0) &= p_0 + \epsilon p_1(x), \end{aligned} \tag{35}$$

where $p_0 = 10^6, \epsilon = 1.0$, and

$$p_1(x) = 60cos(2\pi x) + 100sin(4\pi x). \tag{36}$$

We initialize the density field by the following isentropic relation,

$$\rho(x,0) = (\frac{p(x,0)}{A})^{\frac{1}{\gamma}}, \tag{37}$$

where A is a constant which is determined by letting $p_0 = A{\rho_0}^\gamma$ with $\rho_0 = 10^{-3}$.

Here we use periodic boundary conditions on $0 \leq x \leq 1$. The numerical results are produced by applying the second version of our algorithm where we precondition the non-conservative explicit building block (Section 3.2). Preconditioning the conservative explicit building block works fine as shown in Section 4.2.4, but the computation would be less efficient due to the explicit CFL restriction. Preconditioning the non-conservative explicit building block enables us to use larger time steps, i.e, CFL= 3 ($(|u| + c)_{max}\Delta t <CFL\Delta x$). Figure 13 and 14 together with Table 2 clearly demonstrate the second order convergence of our newly proposed scheme.

4.2 Application of the second order primitive preconditioner technique to multi-phase flow problems

In this section, we present the application of the second order primitive preconditioner technique to multi-phase flow problems. The major part of this section is dedicated to the underwater explosion problem. In the first three subsections, the problem description of an underwater explosion, its numerical procedure, and some computational results will be given. In the last subsection, the oscillating water column problem in a 1-D tube and some numerical results will be presented.

4.2.1 Problem statement and mathematical description (an underwater explosion)

We are interested in studying non-linear bubble dynamics, which plays an important role in many areas of contemporary science and technology such as: oil industry, in which bubbles are essential for lifting the heavy oil to the surface; chemical reactors, in which bubbles are used to increase the contact surface between the gas-liquid phases; ship hydrodynamics, in which collapsing bubbles are the main cause of the propeller damage and limiting factors for the speed of the vessels; some medical and many other technological applications (including our current subject underwater explosions) can also be added [5, 7, 15, 62].

Here we pay special attention to underwater explosions. An underwater explosion generates a very strong shock wave and a high pressure gas bubble. Therefore an underwater explosion can be modeled as the growth and collapse of a gas bubble. For this problem, we make the following assumptions: bubble expansions are assumed to be spherically symmetric, so variation only occurs in the radial direction. The system is assumed to be adiabatic, i.e, we ignore the heat convection. We assume that surface tension and viscous effects are negligible. We also assume that the gas contents of the bubble are spatially uniform (but vary with time), i.e, we don't solve the gas-dynamics equations inside the bubble. Finally we assume that water is compressible and obeys the Tait equation of state (see equation (49) below), which is independent of internal energy [49].

With the above assumptions, the governing equations become the inviscid compressible Euler equations in spherical coordinates,

$$\frac{\partial \rho}{\partial t} + \frac{1}{r^2}\frac{\partial (r^2 m)}{\partial r} = 0, \tag{38}$$

$$\frac{\partial m}{\partial t} + \frac{1}{r^2}\frac{\partial (r^2 u m)}{\partial r} = -\frac{\partial p}{\partial r}, \tag{39}$$

where ρ, u, m, and p represent density, velocity, momentum, and pressure of water respectively.

We enforce a Dirichlet type moving boundary condition for water pressure, i.e, we specify water pressure as the bubble pressure at the bubble wall.

The time dependent bubble pressure is determined by using the adiabatic form of the JWL (Jones-Wilkins-Lee) equation of state for explosive materials,

$$p_{bubble}(t) = Ae^{-R_1(\frac{V(t)}{V_I})} + Be^{-R_2(\frac{V(t)}{V_I})} + C(\frac{V(t)}{V_I})^{-(\omega+1)}, \tag{40}$$

where V_I is the volume of the bubble at the radius R_I, and A, B, R_1, R_2, R_I, and ω are the standard constants for TNT, and finally C is an arbitrary constant which is determined by letting $V(t) = V_0$, $p_{bubble}(t) = p_{bubble0}$ at $t = 0$.

The position of the bubble interface (the material surface) is captured by updating the following level set equation,

$$\frac{\partial \phi}{\partial t} + u\frac{\partial \phi}{\partial r} = 0, \tag{41}$$

where ϕ is the level set function and u is the external velocity field (water velocity). The level set equation (41) states that ϕ is constant along particle paths. This means that if the zero level set of ϕ is initialized as a material surface (bubble interface) between water and gas, then it will always be a material surface at later times [39, 40]. We take $\phi < 0$ in the gas region and $\phi > 0$ in the water region (Figure 15). Hence, we have

$$\phi(r,t) \begin{cases} > 0 & \text{if } r \in \text{water} \\ = 0 & \text{if } r \in \Gamma \\ < 0 & \text{if } r \in \text{the gas}, \end{cases} \tag{42}$$

where Γ represents the bubble interface, and it is defined as the zero level set of ϕ,

$$\Gamma = \{r \mid \phi(r,t) = 0\}. \tag{43}$$

4.2.2 Numerical procedure

For an adiabatic underwater explosion, the governing fluid dynamics equations of water are the continuity and momentum equations, (38) and (39). Thus the same second-order primitive preconditioner algorithm given in Section 3 applies here (except for the energy part, due to the Tait equation of state). Here we shall only concentrate on the details of the moving boundary conditions; i.e., the extension procedure (extrapolation) of the water field variables into the gas region at each time step, along with a short description of the level set algorithm, (Figure 15).

We impose two different Dirichlet type of boundary conditions when solving the elliptic equation for the new pressure field (6). At the left end of the computational domain (gas region), we specify the water pressure as the bubble pressure which is calculated via the bubble volume. At the right end, we specify the hydrostatic water pressure p_{inf} as a pressure boundary condition.

We call one particular cell, in front of the bubble interface, the critical cell (Figure 16). This critical cell plays an important role during the computation, because the moving

boundary condition at the left will be determined by this cell. At each time step, we will have to find a new critical cell due to motion of the bubble interface.

For i equals the critical cell (i.e, $i = critical$) we impose the following boundary conditions when discretizing (6),

$$p^{n+1}_{critical-1} = \frac{\Delta r}{\theta} p^{n+1}_{bubble} + \frac{\theta - \Delta r}{\theta} p^{n+1}_{critical}, \tag{44}$$

where θ is the distance between the critical cell and the bubble interface.

For i equals the last cell (i.e, $i = M$) we have,

$$p^{n+1}_{M+1} = \frac{\Delta r}{L} p_{\inf} + \frac{L - \Delta r}{L} p^{n+1}_{M}, \tag{45}$$

where $p_{\inf}$ is the water pressure at some distance L.

We also apply the following boundary condition for the gradient of the pressure; i.e., for the $\nabla p^{n+1,k}$ term in (6),

$$p^{n+1}_{r,critical-\frac{1}{2}} = \frac{p^{n+1}_{critical} - p^{n+1}_{bubble}}{\theta}, \tag{46}$$

and

$$p^{n+1}_{r,M+\frac{1}{2}} = \frac{p_{\inf} - p^{n+1}_{M}}{L}. \tag{47}$$

Now we shall discuss the extension procedure for the flow variables from water region into the gas region. We must use this extension since we don't solve the gas dynamics equation inside the bubble for this particular problem, and we need the values of the flow variables in the gas region because we have to solve the level set equation in whole domain including the gas and water. Also one has to solve (1) using values in the ghost regions. Here we remark that the discretization of the explicit step does not use the ghost fluid treatment to define the ghost values. The extension procedure to define the ghost values is done through sets of simple linear extrapolation of the flow variables from water into the gas, irrespective of the jump condition that might exist at the moving boundary.

After finishing the extension procedure at each time step, we are ready to solve the level set equation in the whole domain. The level set equation (41) is solved via the following second order differencing,

$$\frac{\phi^{n+1,k}_i - \phi^{n+1,0}_i}{\Delta t} = -u_i^{\frac{(n+1,k-1)+(n+1,0)}{2}} \frac{\phi_{i+1}^{\frac{(n+1,k-1)+(n+1,0)}{2}} - \phi_{i-1}^{\frac{(n+1,k-1)+(n+1,0)}{2}}}{2\Delta r}. \tag{48}$$

4.2.3 Numerical results of the underwater explosion

Here we present the numerical results of the underwater explosion test problem, to show the performance of our second order non-oscillatory all speed multi-phase flow algorithm developed in Sections 3 and Section 4. We recall that underwater explosions can be modeled as the growth and collapse of spherically symmetric gas bubbles, and considering the other assumptions we made in Section 4.2.1 we note that the numerical

results come from 1-D calculations of the adiabatic inviscid Euler equations together with the level set equation in the radial direction.

The method is tested by using the following initial data:

The initial water density is taken to be 1.00037984 g/cm^3.

The water pressure is calculated by the Tait equation of state [49],

$$p = B[(\rho/\overline{\rho})^\gamma - 1] + A \tag{49}$$

where $B = 3.31E + 09d/cm^2$, $A = 1.0E + 06d/cm^2$, $\overline{\rho} = 1.0g/cm^3$, and $\gamma = 7.15$.

The initial water velocity is taken to be 0.0, and initial water pressure is $1.0E + 07d/cm^2$.

Finally the initial bubble radius is taken to be $16cm$, and the initial bubble pressure is $P_0 = 7.8039E + 10d/cm^2$.

The conservative explicit building block (Section 3.1) is used here with CFL= 0.25 ($(|u| + c)\Delta t <$CFLΔx). In Figure 17 and Figure 18, we demonstrate the second order accuracy of the proposed scheme away from the discontinuities. The code is run for four different grid settings, $250, 500, 1000$, and 2000. The enlarged part of the pressure profile, Figure 18, clearly indicates convergence of our method at discontinuities. Figure 19 shows the bubble radius as a function of time during the growth and collapse of the explosion bubble. It can be seen that our solution is in very good agreement with the benchmark data [49].

We note that this problem is a difficult problem to compute due to the high impedance mismatch at the explosive-water boundary. Nevertheless, we capture this discontinuity smoothly and with second order accuracy. This is because we avoid complicated ghost fluid procedures [12, 13, 3, 4, 32, 44] which are necessary for explicit multi-material methods. First order explicit ghost fluid treatments assume a piecewise constant pressure profile which is an incorrect assumption for "stiff fluids". Second order explicit ghost fluid treatments are unstable unless complicated regularization procedures are used to avoid "small cell" CFL constraints. In our method, the matching of pressure on either side of the material boundary is handled simultaneously during the implicit preconditioning step thus avoiding "small cell" CFL constraints. The explicit building block step in our method is written independent of the material boundaries involved.

4.2.4 Oscillating Water Column (a low Mach multi-phase flow test)

We consider a closed one dimensional tube with impermeable boundaries at the left and right ends (Figure 9). With the given initial velocity the water column starts moving from left to right compressing the air at the right and expanding the air at the left. Subsequently, a pressure difference is built up across the water column resulting that the column of water decelerates to the right, makes a stop, and then accelerates to the left. This time a reverse pressure difference is built up across the water column redirecting the flow from left to right again. As a result of this continuous process, the water column starts to oscillate. The governing equations and the numerical procedure are the same

as in Section 4.2.1 and 4.2.2 except all formulations have to be written in cartesian coordinates. The following initial data are used as in [26],

$$\frac{p + Bp_{ref}}{(1+B)p_{ref}} = (\frac{\rho}{\rho_{ref}})^{\gamma} \tag{50}$$

where the same equation of state (50) can be used for both water and air with $p_{ref} = \rho_{ref} = 1$, $\gamma_w = 7$, $\gamma_a = 1.4$, $B_w = 3000$, $B_a = 0$, $p_{w0} = p_{a0} = 1$, $u = 1$, and $x_{fs} = 0.1$. To be consistent with Koren *et al* [26], we plot the time evolution of the pressure coefficients,

$$P(x=-1,t) = \frac{p(x=-1,t) - p_a(0)}{p_a(0)}, P(x=1,t) = \frac{p(x=1,t) - p_a(0)}{p_a(0)}, \tag{51}$$

and the time evolution of the relative mass error,

$$M(t) \equiv \frac{m_a(t) - m_a(0)}{m_a(0)}, \tag{52}$$

where $m_a(t)$ is the total mass of air in the tube at time t. Figure 10 and Figure 11 are produced by using both conservative and non-conservative explicit building blocks. We used CFL= 0.8 ($(|u|+c)_{max}\Delta t$ <CFLΔx) for conservative explicit building block, and CFL= 3 ($(|u|+c)_{max}\Delta t$ <CFLΔx) for non-conservative explicit building block. As we can see, results from both building blocks are reasonably comparable, yet we notice that non-conservative explicit building block is more efficient. With this problem, we demonstrate the ability of our method for computing low Mach (weakly compressible Figure 12) multi-phase flows. We note that our method preserves mass three orders of magnitude better than Koren *et al* [26]. We also note that the amplitude of pressure profile in Figure 10 is different from Koren *et al* [26], because we are assuming constant pressure in gas whereas Koren *et al* [26] use two-fluid model.

5 Conclusions

A second order primitive preconditioner for solving all speed multiphase flows has been presented. This new technique enables us to compute both compressible and incompressible flows with Mach uniform accuracy and efficiency, i.e, we solved problems from the zero Mach number to very high Mach numbers with the same accuracy and efficiency. We showed through various test problems that our technique can efficiently handle strong or weak shocks. Throughout all the test cases we obtained highly resolved shock discontinuities together with calculating correct shock speeds. In the smooth flow test and the low Mach flow test(oscillating water column) cases, we introduced an alternative application of our preconditioner in which we preconditioned an explicit building block based on the non-conservative (primitive) form of the Euler equations. This alternative application allows us to use large time steps with high order of accuracy. The present method performs very well when solving multi-material flow problems. The underwater explosion test case shows that we have eliminated the spurious pressure oscillations at

the material surfaces. The multi-phase flow test also shows that our method can handle high impedance interface flows very efficiently and accurately.

The numerical results clearly indicate that our method is more efficient and accurate than current state of the art techniques for all speed multi/single phase flows . Some improvements could be made by adding high order ENO/WENO schemes[35, 36, 44, 20, 30, 29] as the explicit building block. An important benefit of our preconditioner is that one can use any of his/her favorite explicit schemes as an explicit building block, then precondition it by our primitive preconditioner. Our method will be combined with the automating dynamic Adaptive Mesh Refinement (AMR) technique including subcycling procedure, wherein very large time steps will be taken on coarser levels without degrading the accuracy on the finest level. The new all speed multi-phase multi dimensional AMR technique [21] will use a strategy that a non-conservative explicit building block will be applied on coarser levels where shock resolution is not needed, and a conservative explicit building block will be applied on the finest level to be able to calculate highly resolved shock discontinuities.

References

[1] R. Abgrall. How to prevent pressure oscillations in multicomponent flow calculations: A quasi conservative approach. *J. Comput. Phys.*, 125:150–160, 1996.

[2] R. Abgrall and S. Karni. Computations of compressible multi-fluids. *J. Comput. Phys.*, 169:594–623, 2001.

[3] T. Aslam. A Level Set Algorithm for Tracking Discontinuities in Hyperbolic Conservation Laws I (Scalar Equations) . *J. Comput. Phys.*, 167:413–438, 2001.

[4] T. Aslam. A Level Set Algorithm for Tracking Discontinuities in Hyperbolic Conservation Laws II (System of Equations). *J. Scien. Comput.*, 19:37–62, 2003.

[5] G.K. Batchelor. *An Introduction to Fluid Dynamics.* Cambridge University Press, 1967.

[6] H. Bijl and P. Wesseling. A unified method for computing incompressible and compressible flows in boundary-fitted coordinates. *J. Comput. Phys.*, 141:153–173, 1998.

[7] C.E. Brennen. *Cavitation and Bubble Dynamics.* Oxford University Press, 1995.

[8] V. Casulli and D. Greenspan. Pressure method for the numerical solution of transient, compressible fluid flows. *Int. J. Numer. Methods Fluids*, 4:1001, 1984.

[9] A. Chorin. A numerical method for solving incompressible viscous flow problems. *J. Comput. Phys.*, 2:12–26, 1967.

[10] A. Chorin and J.E. Marsden. *A Mathematical Introduction to Fluid Mechanics.* Springer-Verlag, 1979.

[11] P. Colella and K. Pao. A projection method for low speed flows. *J. Comput. Phys.*, 149:245–269, 1999.

[12] R. Fedkiw, T. Aslam, B. Merriman, and S. Osher. A non-oscillatory Eulerian approach to interfaces in multi-material flows (the Ghost Fluid Method). *J. Comput. Phys.*, 152:457, 1999.

[13] R. Fedkiw, T. Aslam, and S. Xu. The Ghost Fluid Method for deflagration and detonation discontinuities. *J. Comput. Phys.*, 154:393, 1999.

[14] J. Guerra and B. Gustafsson. A numerical method for incompressible and compressible flow problems with smooth solutions. *J. Comput. Phys.*, 63:377, 1986.

[15] F.G. Hammitt. *Cavitation and Multiphase Flow Phenomena.* McGraw-Hill, 1980.

[16] F.H. Harlow and A.A. Amsden. Numerical calculation of almost incompressible flows. *J. Comput. Phys.*, 3:80–93, 1968.

[17] F.H. Harlow and J.E. Welch. Numerical calculation of time dependent viscous incompressible flow of fluid with a free surface. *Phys. Fluids*, 8:2182, 1965.

[18] C.W. Hirt, A.A. Amsden, and J.L. Cook. An Arbitrary Lagrangian-Eulerian Computing Method For All Flow Speeds. *J. Comput. Phys.*, 135:203–216, 1997.

[19] X.Y. Hu and B.C. Khoo. An interface interaction method for compressible multifluids. *J. Comput. Phys.*, 198:35–64, 2004.

[20] G. Jiang and C.W. Shu. Efficient implementation of weighted ENO schemes. *J. Comput. Phys.*, 126:202, 1996.

[21] S. Y. Kadioglu, M. Sussman, and J. P. Wright. An AMR (Adaptive Mesh Refinement) Technique For Solving All Speed Multi-phase Flows. *in preparation.*

[22] S. Karni. Multicomponent Flow Calculation by a Consistent Primitive Algorithm. *J. Comput. Phys.*, 112:31–43, 1994.

[23] S. Klainermann and A. Majda. Compressible and incompressible fluids. *Comm. Pure Appl. Math.*, 35:629, 1982.

[24] R. Klein. Semi-implicit extension of a Godunov-type scheme based on low Mach number asymptotics I: One-dimensional flow. *J. Comput. Phys.*, 121:213, 1995.

[25] B. Koren. Improving Euler computations at low Mach numbers. *Int. J. Comp. Fluid Dyn.*, 6:51–70, 1996.

[26] B. Koren, M. R. Lewis, E.H. van Brummelen, and B. van Leer. Riemann problem and level set approaches for homentropic two-fluid flow computations. *J. Comput. Phys.*, 181:654–674, 2002.

[27] P. Lax and B. Wendroff. Systems of conservation laws. *Commun. Pure Appl. Math.*, 13:217–237, 1960.

[28] P.D. Lax. Weak solutions of nonlinear hyperbolic equations and their numerical approximations. *Comm. Pure Appl. Math.*, 7:159–93, 1954.

[29] X. D. Liu and S. Osher. Convex ENO High Order Multi-dimensional Schemes without Field by Field Decomposition or Staggered Grids. *J. Comput. Phys.*, 142:304–330, 1998.

[30] X.D. Liu, S. Osher, and T. Chan. Weighted essentially non-oscillatory schemes. *J. Comput. Phys.*, 115:200, 1994.

[31] H. Nessyahu and E. Tadmor. Non-oscillatory central differencing for hyperbolic conservation laws. *J. Comput. Phys.*, 87:408–463, 1990.

[32] R. Nourgaliev, N. Dinh, and T. Theofanous. Direct Numerical Simulation of Compressible Multiphase Flows: Interaction of Shocks Waves with Dispersed Multiphase Media. In 5^{th} *International Conference on Multiphase Flow, ICMF04*, 2004. Yokohama, Japan.

[33] Y. Ogata and T. Yabe. Shock capturing with improved numerical viscosity in primitive Euler representation. *Comput. Phys. Commun.*, 119:179–193, 1999.

[34] R. Rudnik, S. Melber, A. Ronzheimer, and O. Brodersen. Three-Dimensional Navier-Stokes Simulations for transport aircraft high lift configurations. *AIAA J.*, 38:895–903, 2001.

[35] C.W. Shu and S. Osher. Efficient implementation of essentially non-oscillatory shock capturing schemes. *J. Comput. Phys.*, 77:439, 1988.

[36] C.W. Shu and S. Osher. Efficient implementation of essentially non-oscillatory shock capturing schemes II. *J. Comput. Phys.*, 83:32, 1989.

[37] P. Smereka. Semi-Implicit Level Set Methods for Curvature and Surface Diffusion Motion. *J. Scien. Comput.*, 19:439–456, 2003.

[38] G.A. Sod. A survey of several finite difference methods for systems of nonlinear conservation laws. *J. Comput. Phys.*, 27:1–31, 1978.

[39] M. Sussman. A second order coupled level set and volume-of-fluid method for computing growth and collapse of vapor bubbles. *J. Comput. Phys.*, 187:110–136, 2003.

[40] M. Sussman and E.G. Puckett. A coupled level set and volume-of-fluid method for computing 3D and Axisymmetric incompressible two-phase flows. *J. Comput. Phys.*, 162:301–337, 2000.

[41] H. Takewaki, A. Nishiguchi, and T. Yabe. Cubic interpolated pseudoparticle method (CIP) for solving hyperbolic type equations. *J. Comput. Phys.*, 61:261–268, 1985.

[42] E. Turkel. Preconditioning techniques in computational fluid dynamics. *Annu. Rev. Fluid Mech.*, 31:385–416, 1999.

[43] E. Turkel, R. Radespiel, and H. Kroll. Assessment of preconditioning methods for multidimensional aerodynamics. *Comput. Fluids*, 26:613–634, 1997.

[44] H.S. Udaykumar, L. Tran, D.M. Belk, and K.J. Vanden. An Eulerian method for computation of multi-material impact with ENO shock capturing and sharp interfaces. *J. Comput. Phys.*, 186:136–177, 2003.

[45] D.R. van der Heul, C. Vuik, and P. Wesseling. A conservative pressure-correction method for the Euler and ideal MHD equations at all speeds. *Int. J. Numer. Meth. Fluids.*, 40:521–529, 2002.

[46] D.R. van der Heul, C. Vuik, and P. Wesseling. A conservative pressure-correction method for flow at all speeds. *Comput. Fluids.*, 32:1113–1132, 2003.

[47] J. Wackers and B. Koren. A fully conservative model for compressible two-fluid flow. *Int. J. Numer. Meth. Fluids*, 00:1–6, 2004.

[48] S.P. Wang, M.H. Anderson, J.G. Oakley, M.L. Corradini, and R. Bonazza. A thermodynamically consistent and fully conservative treatment of contact discontinuities for compressible multicomponent flows. *J. Comput. Phys.*, 195:528–559, 2004.

[49] A. Wardlaw. Underwater explosion test cases. In *Technical Report IHTR 2069, ADB238684, Office of Naval Research*, 1998.

[50] J.M. Weiss and W.A. Smith. Preconditioning applied to variable and constant density flows. *AIAA J.*, 33:2050–2057, 1995.

[51] I. Wenneker, A. Segal, and P. Wesseling. A Mach-uniform unstructured staggered grid method. *Int. J. Numer. Meth. Fluids.*, 40:1209–1235, 2002.

[52] P. Wesseling. *Principles of Computational Fluid Dynamics.* Springer Series in Computational Mathematics, 2000.

[53] P. Wesseling, van der Heul, and C. Vuik. Unified methods for computing compressible and incompressible flows. In *Proceedings European Congress on Computational Methods in Applied Sciences and Engineering, ECCOMAS*, 2000. Barcelona.

[54] P. Wesseling and D.R. van der Heul. Uniformly effective numerical methods for hyperbolic systems. *Computing.*, 66:249–267, 2001.

[55] P. Woodward and P. Colella. The numerical simulation of two-dimensional fluid flow with strong shocks. *J. Comput. Phys.*, 54:115, 1984.

[56] F. Xiao. Unified formulation for compressible and incompressible flows by using multi-integral moments I: One-dimensional inviscid compressible flow. *J. Comput. Phys.*, 195:629–654, 2004.

[57] T. Yabe. Interface capturing and universal solution of solid, liquid, and gas by CIP method. In *Proceedings Conf. High-Performance Computing on Multi-phase Flow*, 1997. Tokyo.

[58] T. Yabe and T. Aoki. A universal solver for hyperbolic equations by cubic-polynomial interpolation. *Comput. Phys. Commun.*, 66:219–242, 1991.

[59] T. Yabe, Y. Ogata, K. Takizawa, T. Kawai, A. Segawa, and K. Sakurai. The next generation CIP as a conservative semi-Lagrangian solver for solid, liquid, and gas. *J. Comput. Appl. Math.*, 149:267–277, 2002.

[60] T. Yabe, F. Xiao, and T. Utsumi. The constrained interpolation profile method for multiphase analysis. *J. Comput. Phys.*, 169:556–593, 2001.

[61] S.Y. Yoon and T. Yabe. The unified simulation for incompressible and compressible flow by the predictor-corrector scheme based on the CIP method. *Comput. Phys. Commun.*, 119:149–158, 1999.

[62] F.R. Young. *Cavitation.* McGraw-Hill, 1989.

List of Figures

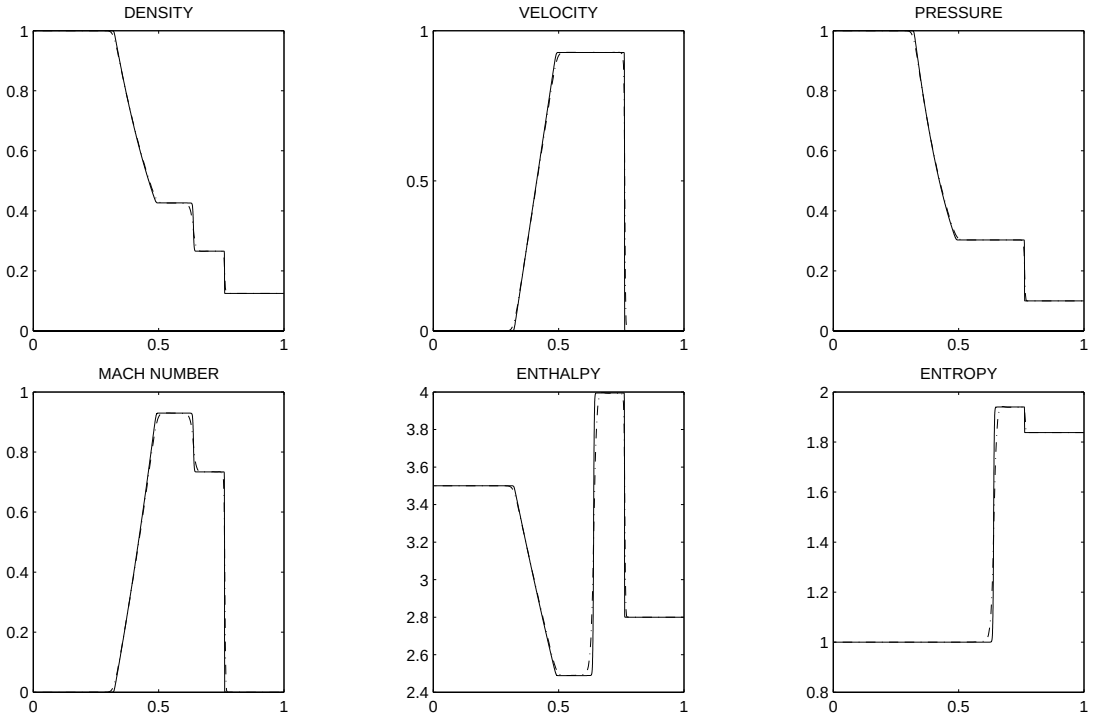

Figure 1: Numerical results of the Sod's shock tube problem at $t = 0.15$. Solid lines represent the reference solutions and dashed lines represent the Mach uniform results.

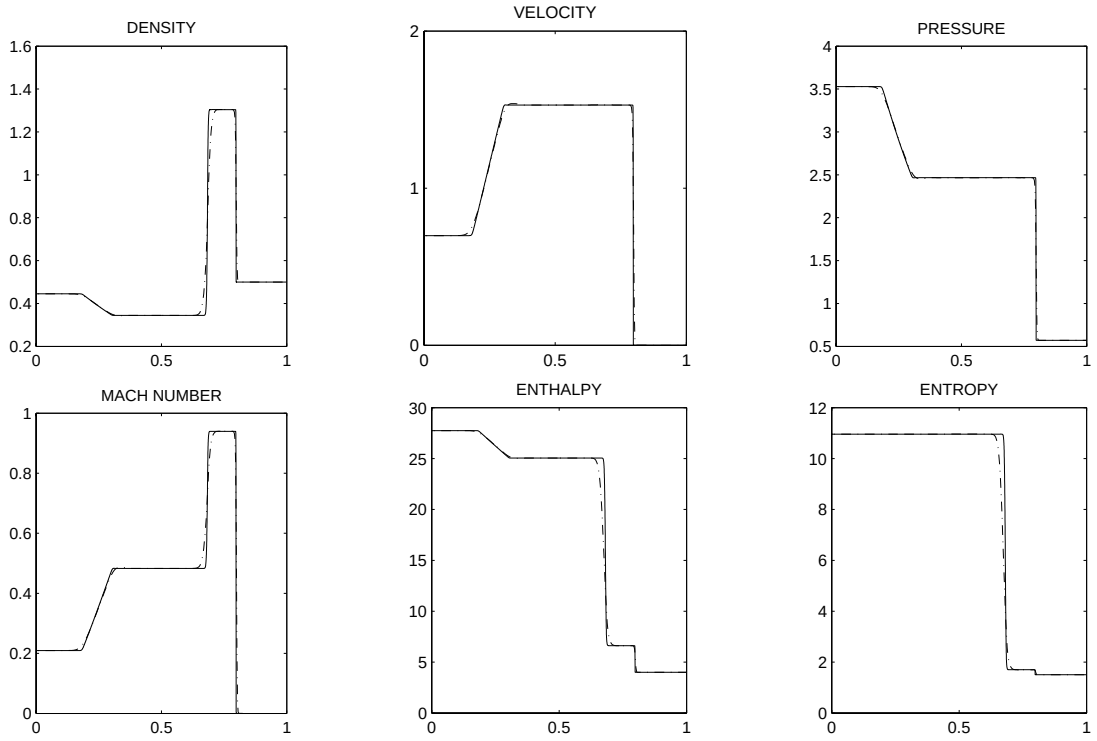

Figure 2: Numerical results of the Lax's shock tube problem at $t = 0.12$. Solid lines represent the reference solutions and dashed lines represent the Mach uniform results.

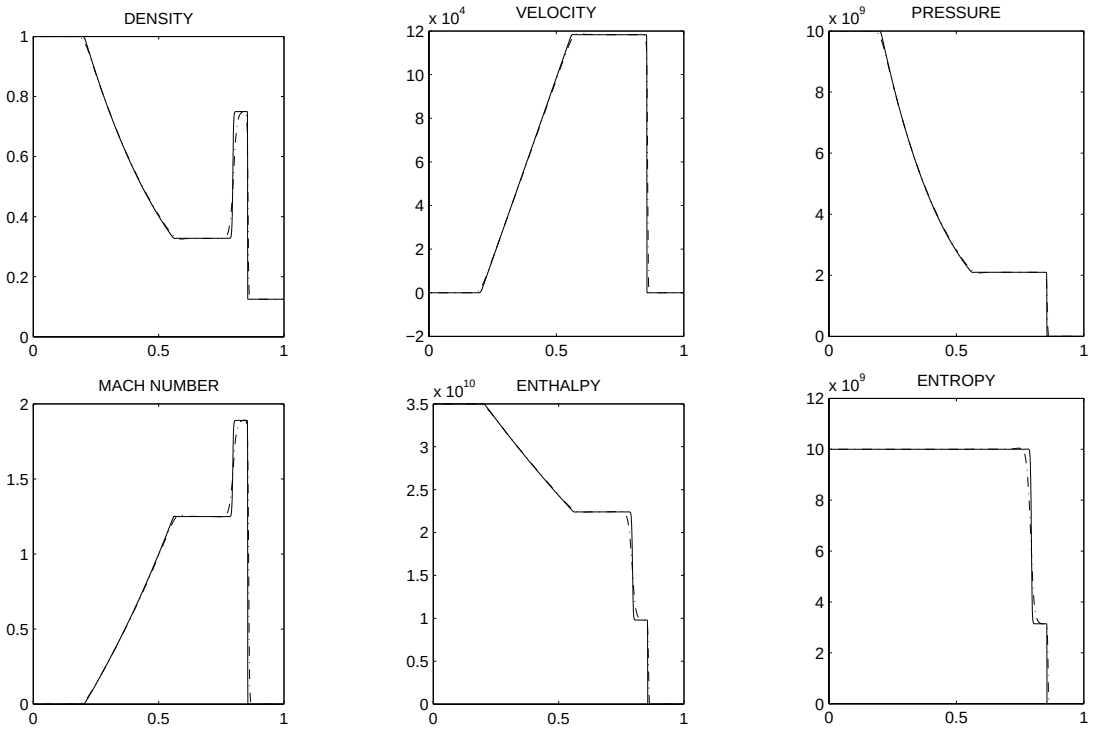

Figure 3: Numerical results of the strong shock tube problem at $t = 2.5 \times 10^{-6}$. Solid lines represent the reference solutions and dashed lines represent the results of the all speed primitive preconditioner.

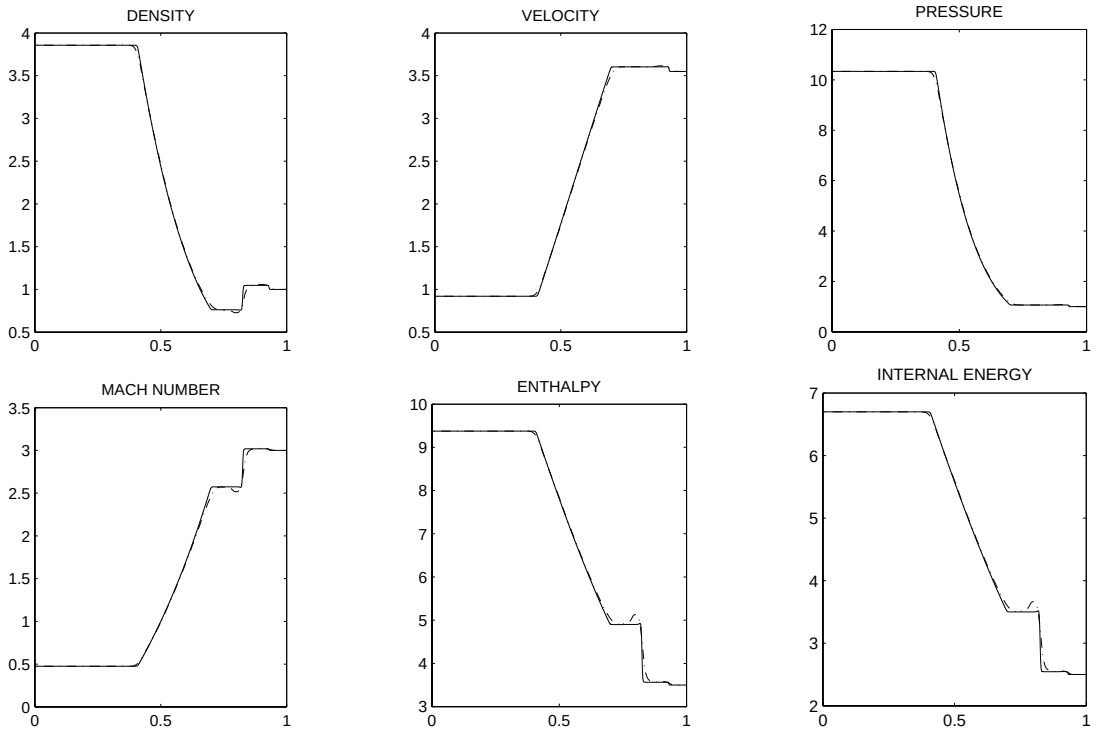

Figure 4: Numerical results of the Mach 3 shock tube problem at $t = 0.09$. Solid lines represent the reference solutions and dashed lines represent the results of the all speed primitive preconditioner.

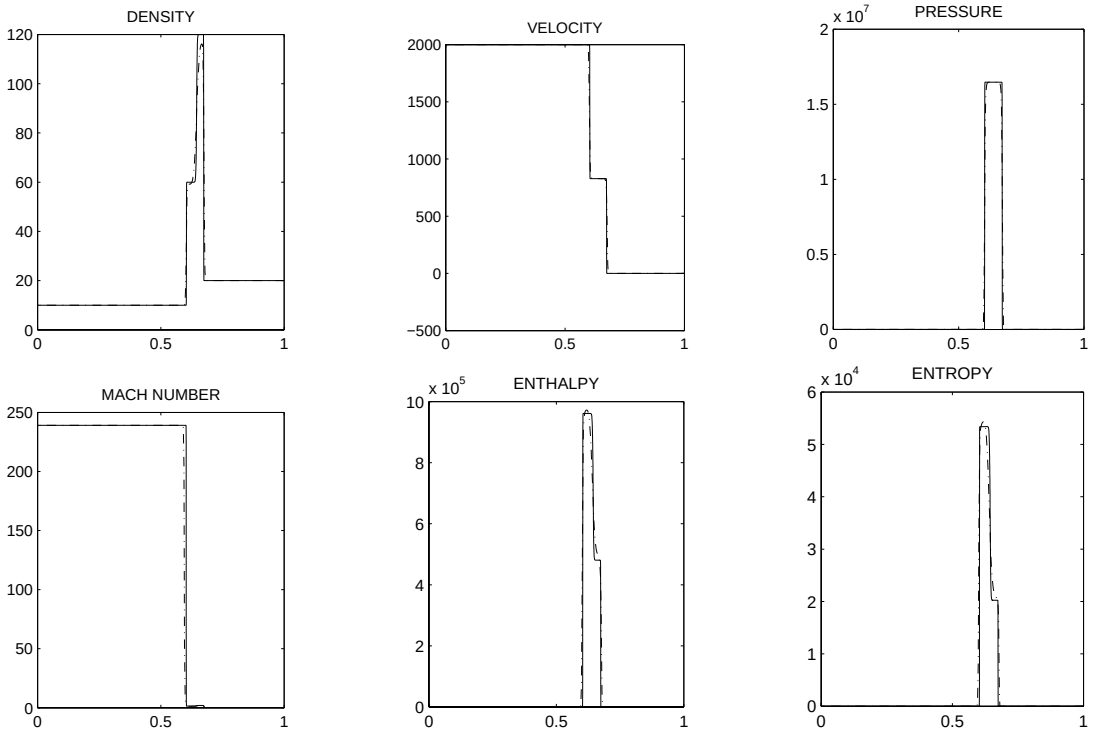

Figure 5: Numerical results of the high Mach number shock tube problem at $t = 1.75 \times 10^{-4}$. Solid lines represent the reference solutions and dashed lines represent the results of the all speed primitive preconditioner.

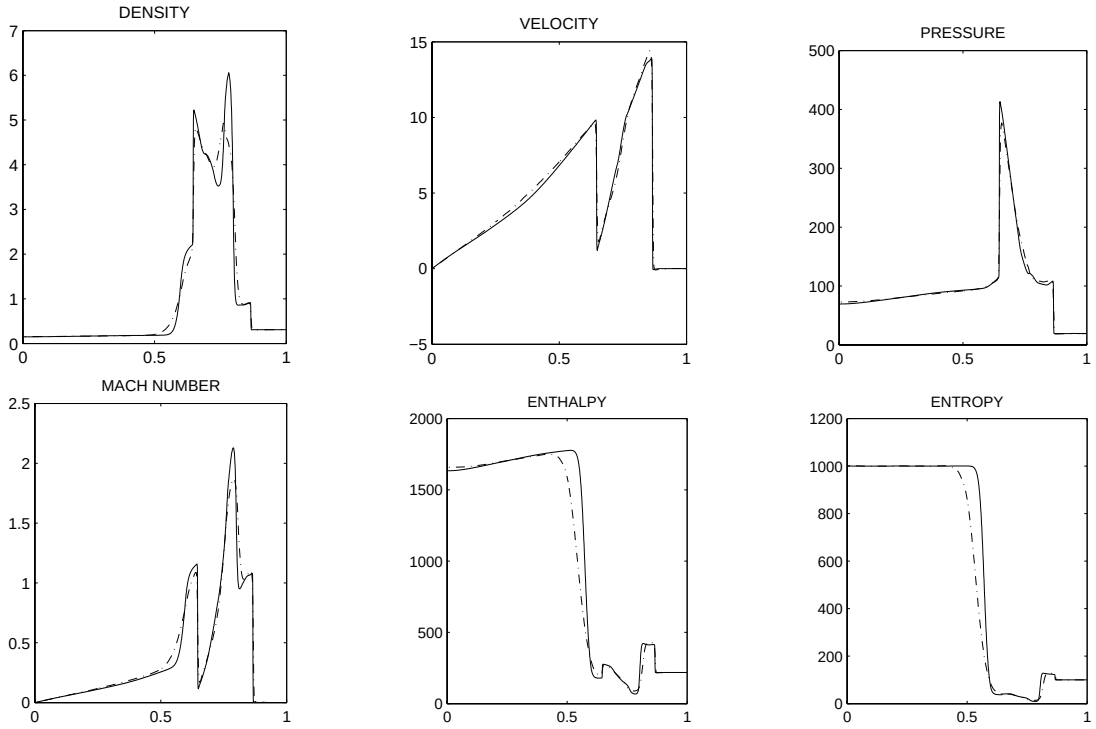

Figure 6: Numerical results of the two interacting blast waves problem at $t = 0.038$. Solid lines represent the reference solutions and dashed lines represent the results of the all speed primitive preconditioner.

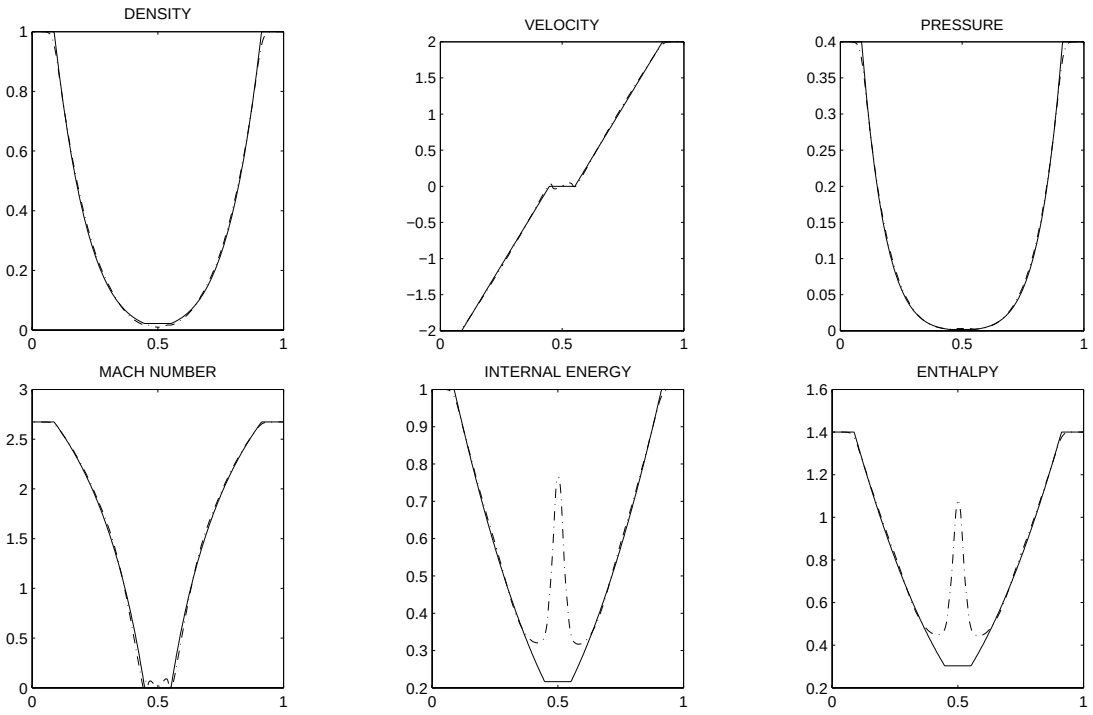

Figure 7: Numerical results of the two symmetric rarefaction waves problem at $t = 0.15$. Solid lines represent the reference solutions and dashed lines represent the results of the all speed primitive preconditioner.

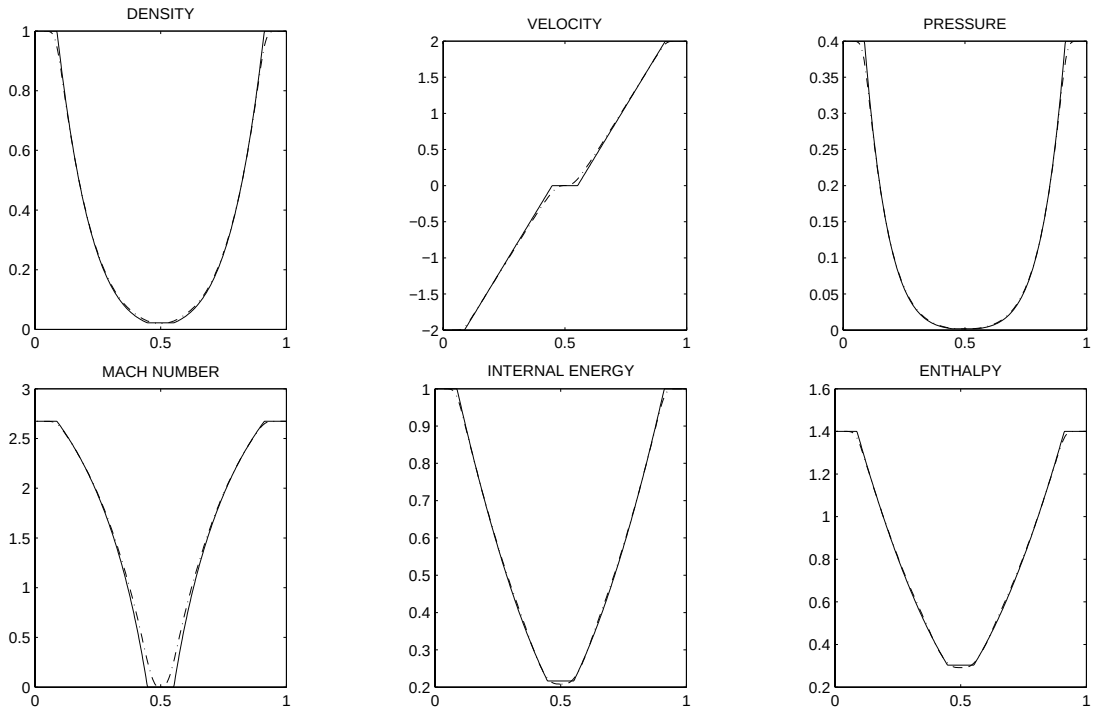

Figure 8: Numerical results of the two symmetric rarefaction waves problem computed by the second version (preconditioning an explicit building block based on the primitive Euler equations) of the algorithm at $t = 0.15$. Solid lines represent the reference solutions and dashed lines represent the results of the all speed primitive preconditioner.

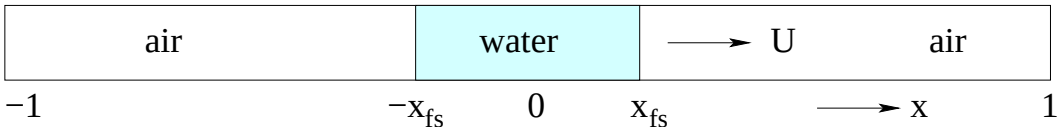

Figure 9: A closed tube in which a column of water moves from left to right with constant speed U.

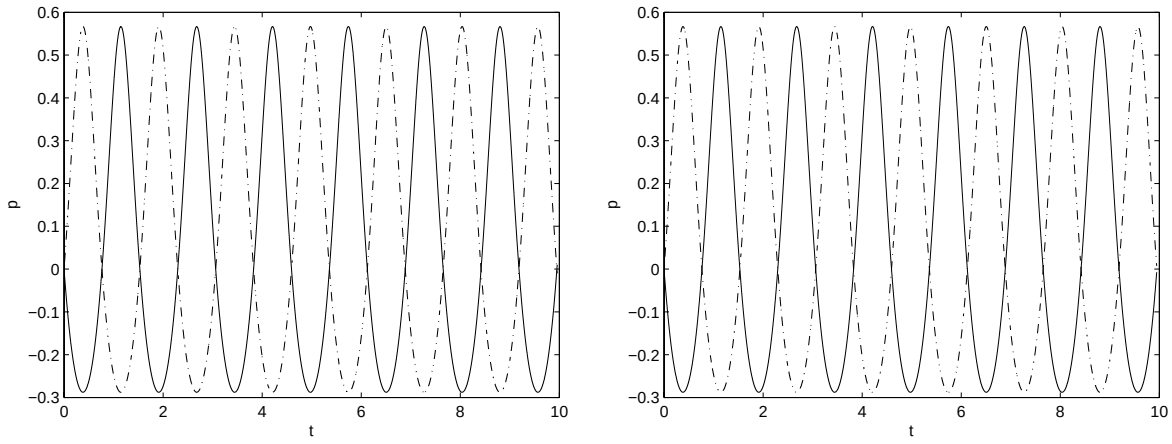

Figure 10: Time evolution of pressure coefficients at left and right boundaries. Left figure is generated by using the conservative explicit building block, and right figure is generated by using the non-conservative (primitive) explicit building block. Solid line denotes left boundary, and dashed line denotes right boundary. $M = 80$ cells points are used. CFL=0.8 on left and CFL=3.0 on right.

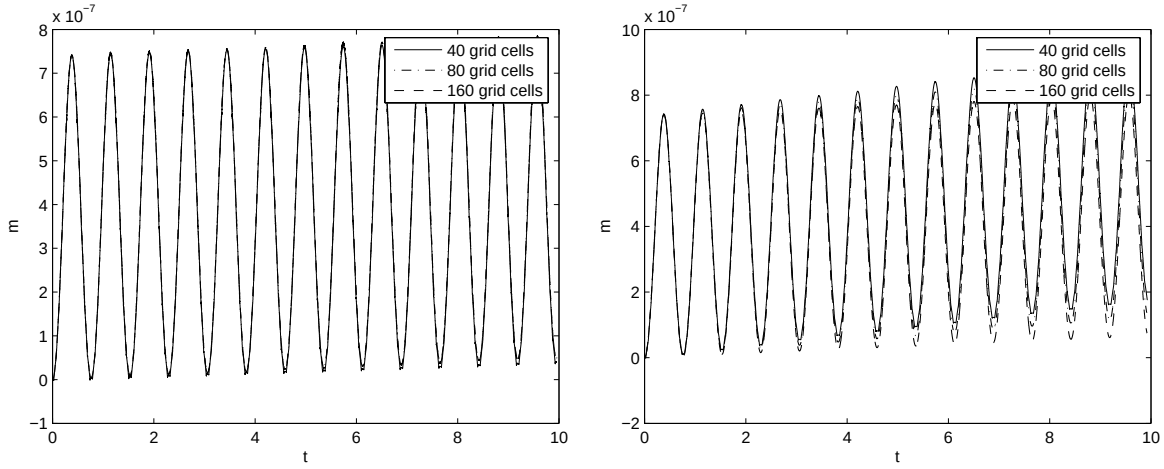

Figure 11: Time evolution of relative error in total mass of air. Left figure is generated by using the conservative explicit building block, and right figure is generated by using the primitive explicit building block. CFL=0.8 on left and CFL=3.0 on right.

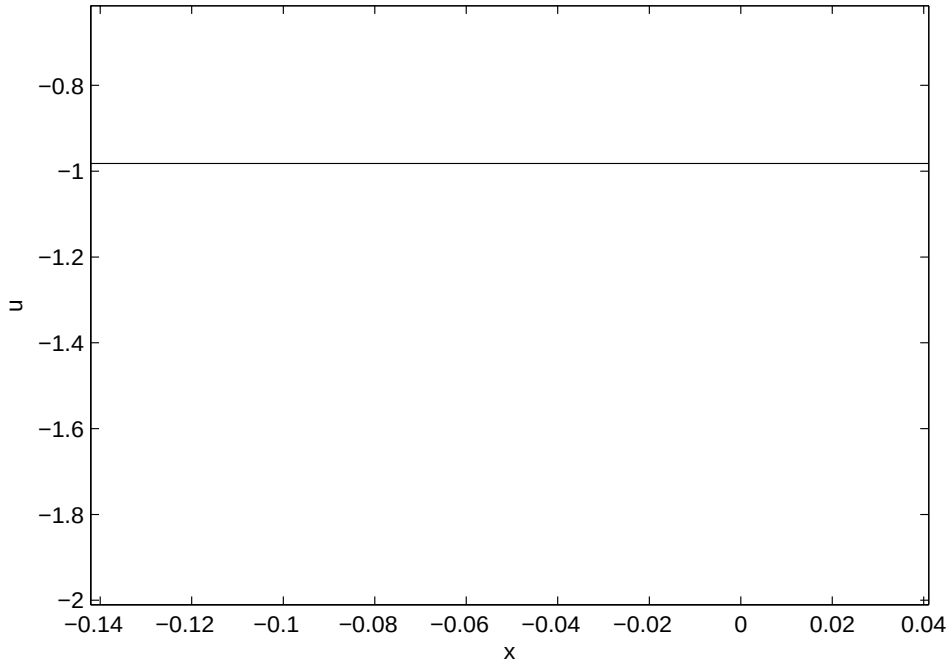

Figure 12: Water velocity for oscillating water column problem at $t = 10$ with $\Delta x = \frac{1}{80}$.

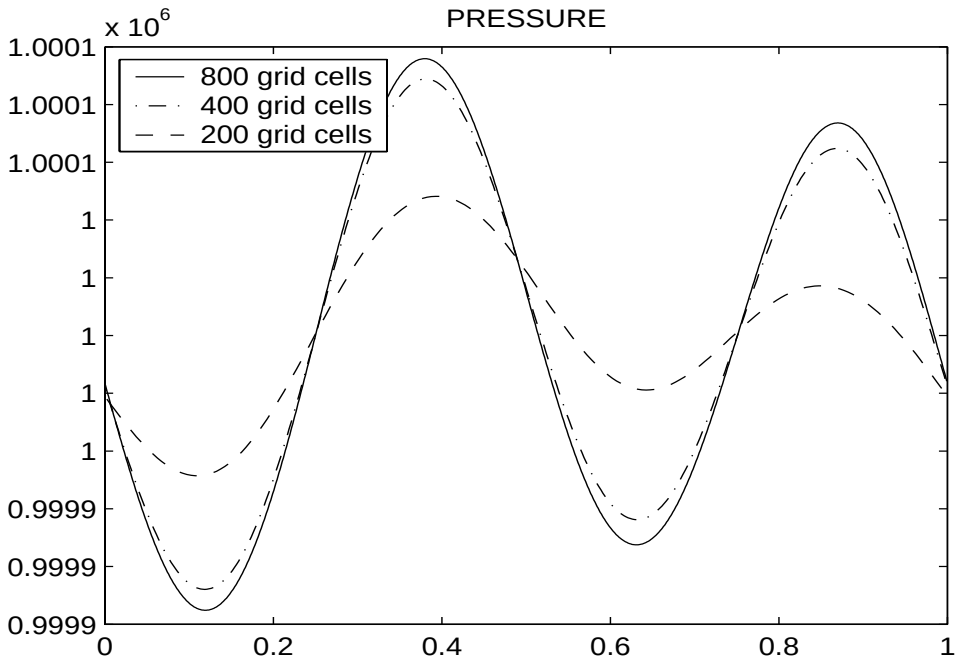

Figure 13: Numerical results of the smooth flow test with CFL=3 at zero Mach limit.

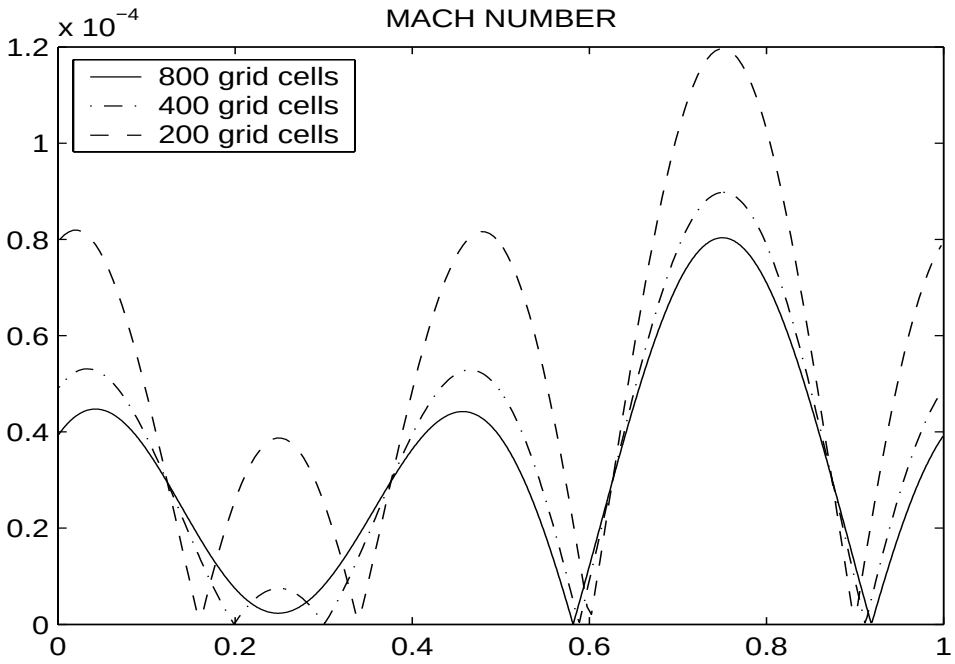

Figure 14: Numerical results of the smooth flow test with CFL=3 at zero Mach limit.

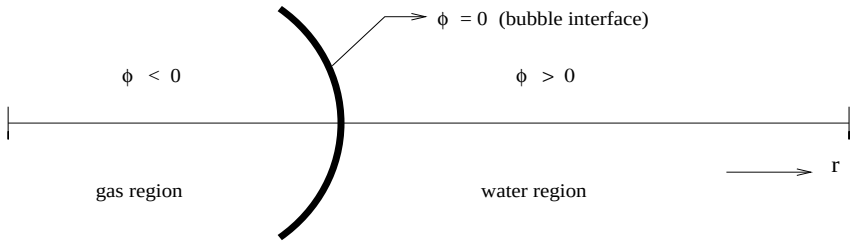

Figure 15: Bubble interface (the material surface) represented by the level set function.

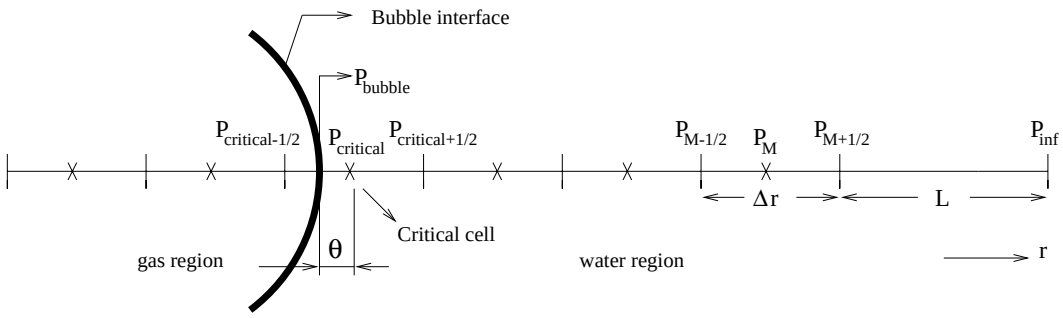

Figure 16: A moving boundary condition representation for pressure.

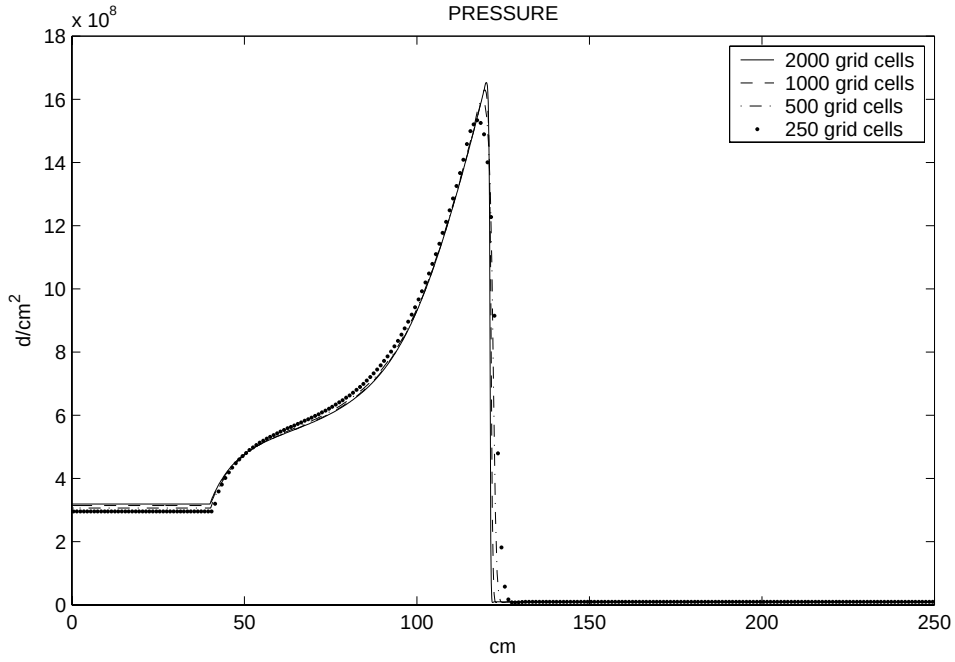

Figure 17: Demonstration of the convergence of the all speed primitive precondioner for different grid resolutions at $t = 0.5msec$.

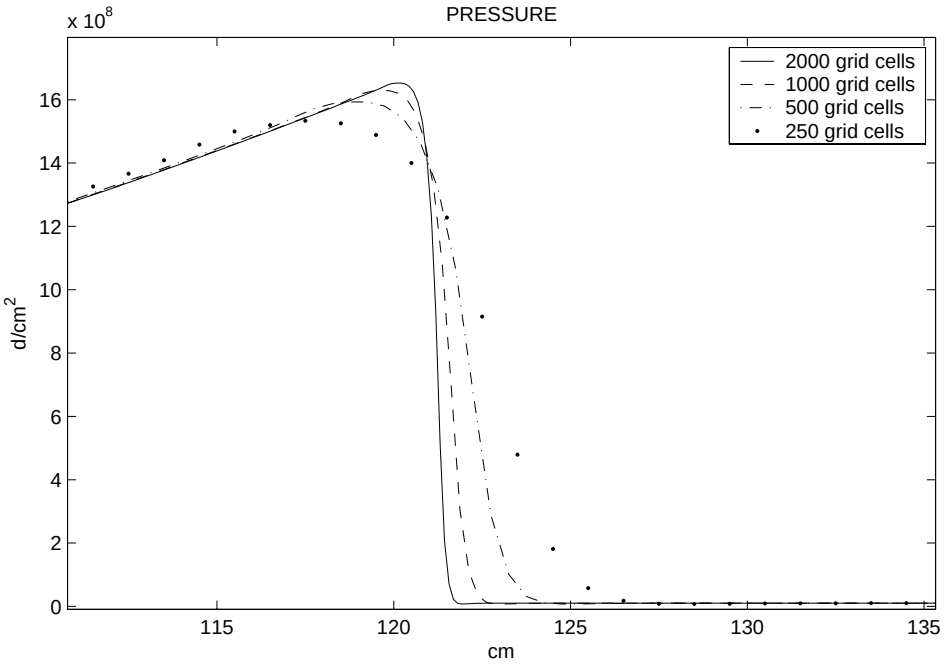

Figure 18: Demonstration of the convergence of the all speed primitive precondioner for different grid resolutions (enlarged part of the shock front) at $t = 0.5msec$.

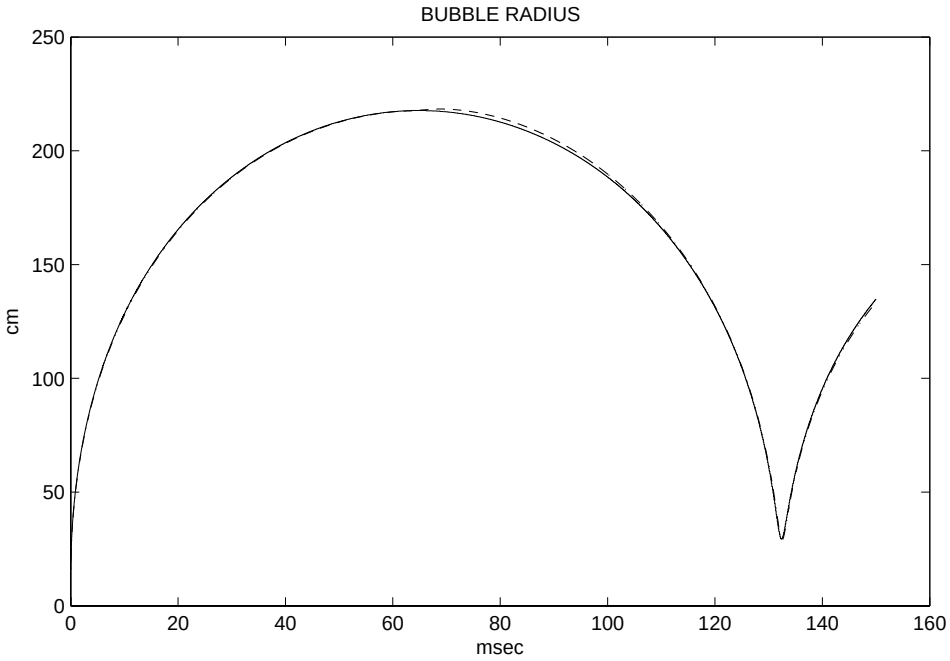

Figure 19: Comparing the bubble radius using grid size $\Delta r = 1cm$. The solid line represents the benchmark bubble radius and the dashed line represents the solution obtained by our all speed primitive preconditioner.

Table 1: Error analysis for the Sod's shock tube problem based on three levels of grid refinement.

$k = 2$ iterations			
Number of mesh points	100	200	400
Error in total mass	5.99×10^{-4}	3.54×10^{-4}	2.30×10^{-4}
Error in total energy	2.47×10^{-3}	1.75×10^{-3}	1.38×10^{-3}
$k = 5$ iterations			
Number of mesh points	100	200	400
Error in total mass	2.85×10^{-5}	2.46×10^{-5}	2.42×10^{-5}
Error in total energy	1.99×10^{-4}	1.88×10^{-4}	1.87×10^{-4}

Table 2: Error analysis for the zero Mach test problem based on three levels of grid refinement.

	Error in acoustic pressure from 200 to 400 mesh	Error in acoustic pressure from 400 to 800 mesh	Order of accuracy
l_1 norm	48.70	8.93	2.45
l_∞ norm	27.94	5.12	2.45