ACADEMIC PRESS

Available online at www.sciencedirect.com

SCIENCE DIRECT®

Journal of Computational Physics 185 (2003) 120–138

JOURNAL OF COMPUTATIONAL PHYSICS

www.elsevier.com/locate/jcp

A flux-split algorithm applied to conservative models for multicomponent compressible flows

Antonio Marquina *, Pep Mulet

Department of Matemàtica Aplicada, Universitat de València, Burjassot, València 46100, Spain

Received 31 January 2002; received in revised form 13 August 2002; accepted 27 August 2002

Dedicated to S.J. Osher on the occasion of his 60th birthday

Abstract

In this paper we consider a conservative extension of the Euler equations for gas dynamics to describe a two-component compressible flow in Cartesian coordinates. It is well known that classical shock-capturing schemes applied to conservative models are oscillatory near the interface between the two gases. Several authors have addressed this problem proposing either a primitive consistent algorithm [J. Comput. Phys. 112 (1994) 31] or Lagrangian ingredients (Ghost Fluid Method by Fedkiw et al. [J. Comput. Phys. 152 (1999) 452] and [J. Comput. Phys. 169 (2001) 594]). We solve directly this conservative model by a flux-split algorithm, due to the first author (see [J. Comput. Phys. 125 (1996) 42]), together with a high-order (WENO5) flux reconstruction [J. Comput. Phys. 115 (1994) 200; 83 (1989) 32]. This algorithm seems to reduce the oscillations near the interfaces in a way that does not affect the physics of the experiments. We validate our algorithm with the numerical simulation of the interaction of a Mach 1.22 shock wave impinging a helium bubble in air, under the same conditions studied by Haas and Sturtevant [J. Fluid Mech. 181 (1987) 41] and successfully simulated by Quirk and Karni [J. Fluid Mech. 318 (1996) 129].

Keywords: Multiphase gas dynamics; Richtmyer–Meshkov instability

1. Introduction

Richtmyer–Meshkov instabilities arise when an interface between two different gases is impulsively accelerated by acoustical waves (shock waves, large changes in density, etc.). These instabilities may induce a compressible turbulent regime with high Reynolds numbers. Our goal is the numerical simulation of those instabilities by means of a conservative and entropy satisfying scheme applied to the Euler equations for multicomponent gas dynamics. We know that vanishing viscosity solutions to problems involving Richtmyer–Meshkov instabilities might not exist, but we can obtain high-order numerical approximations with a

* Corresponding author. Tel.: +34-96-386-4358; fax: +34-96-386-4735.
E-mail addresses: marquina@uv.es (A. Marquina), mulet@uv.es (P. Mulet).

0021-9991/02/$ - see front matter
PII: S0021-9991(02)00050-5

resulting numerical viscosity (Reynolds number), for which the compressible turbulent regime is completely developed and shocks, contacts and rarefaction waves propagate with correct strength and speed.

In this paper we use a fifth-order accurate conservative scheme, based on a flux splitting due to the first author and the WENO5 reconstruction [11,15], to solve the multispecies Euler equations with an equation of state (EOS) that allows mixing of the species. It has been pointed out in [12] that some conservative shock-capturing schemes suffer from pressure and velocity oscillations near gas interfaces. However, we have found through our numerical experiments that pressure and velocity fluctuations near gas interfaces are small and do not seem to interfere with the physics of the simulation. The explanation of this apparent contradiction to Karni's analysis may be found in the mixing property of the EOS, which allows a smooth transition between gases, and in the use of a scheme that has shown the property of alleviating, if not avoiding, some pathologies, such as overheating at reflections, "carbuncle" and "kinked" Mach stems (cf. Section 3).

The organization of the paper is the following: In Section 2 we show the Euler equations for the dynamics of a mixture of two gases. In Section 3 we explain the scheme we use in our experiments. Next, in Section 4 we validate our one-dimensional algorithms, comparing their results with the exact solution of some selected tests. In Section 5 we use our two-dimensional algorithms for the simulation of a 1.22 Mach air shock impinging a cylindrical helium bubble, with identical setup as the one used in [20], obtaining similar shock velocities and an interface with very fine details where a complete turbulent regime is developed. In Section 6 we use some techniques borrowed from level set theory to analyze the growth of the instability. Finally, in Section 7 we present some issues on the parallel implementation of the algorithms and in Section 8 our conclusions.

2. Multicomponent flow equations

For simplicity of exposition we assume that we aim to model the dynamics of a mixture of two gases in two space dimensions, the extensions to more components or more dimensions being directly deduced. Let ρ denote the density of the mixture, ϕ, the mass fraction of the first component and, therefore, $1-\phi$, the mass fraction of the second component.

We assume that both components are in thermal equilibrium and are calorically perfect gases with specific heats at constant volume Cv_1, Cv_2, specific heats at constant pressure Cp_1, Cp_2 and ratios of specific heats γ_1, γ_2. By standard thermodynamic arguments (see [2] and references therein), the ratio of specific heats of the mixture of gases is

$$\gamma(\phi) = \frac{Cp}{Cv} = \frac{Cp_1\phi + Cp_2(1-\phi)}{Cv_1\phi + Cv_2(1-\phi)}. \tag{1}$$

The equation of state expresses the pressure P in terms of the density ρ, the specific internal energy ϵ and mass fraction ϕ and it reads as

$$P(\rho, \epsilon, \phi) = (\gamma(\phi) - 1)\rho\epsilon. \tag{2}$$

We model the dynamics of this mixture by the compressible Euler equations with an additional equation expressing conservation of the first component, which, coupled to the conservation of mass, implies conservation of the second component as well. In two dimensions, these equations respectively express the conservation of mass, momentum in x and y directions, total energy and mass of the first component and they read as

$$U_t + F(U)_x + G(U)_y = 0, \tag{3}$$

with the conserved variables U and fluxes F and G being given by

$$\begin{aligned} U &= [\rho \quad \rho u \quad \rho v \quad E \quad \rho\phi]^{\mathrm{T}}, \\ F(U) &= [\rho u \quad \rho u^2 + P \quad \rho uv \quad (E+P)u \quad \rho\phi u]^{\mathrm{T}}, \\ G(U) &= [\rho v \quad \rho uv \quad \rho v^2 + P \quad (E+P)v \quad \rho\phi v]^{\mathrm{T}}, \end{aligned} \tag{4}$$

where (u, v) is the velocity field of the mixture and E is the total energy per unit volume. The functional dependence of the fluxes F and G on the conserved variables can be obtained from (4), (2) and the usual relations

$$u = \frac{\rho u}{\rho}, \quad v = \frac{\rho v}{\rho}, \quad \epsilon = \frac{E}{\rho} - \frac{1}{2}(u^2 + v^2), \quad \phi = \frac{\rho\phi}{\rho}. \tag{5}$$

System (3) is hyperbolic: the eigenvalues of $F'(U)$ are $\lambda_1 = u - c$, $\lambda_{2,3,4} = u$, $\lambda_5 = u + c$ and the corresponding right eigenvectors r_i and left eigenvectors l_i, normalized so that $r_i \cdot l_j = \delta_{ij}$, are

$$\begin{aligned} r_1 &= [1 \quad u - c \quad v \quad H - uc \quad \phi]^{\mathrm{T}}, \\ r_2 &= \left[1 \quad u \quad v \quad \frac{u^2 + v^2}{2} \quad \phi\right]^{\mathrm{T}}, \\ r_3 &= [0 \quad 0 \quad 1 \quad v \quad 0]^{\mathrm{T}}, \\ r_4 &= \left[0 \quad 0 \quad 0 \quad -\frac{X}{\gamma - 1} \quad 1\right]^{\mathrm{T}}, \\ r_5 &= [1 \quad u + c \quad v \quad H + uc \quad \phi]^{\mathrm{T}}, \\ l_1 &= \left[\beta_2 + \frac{u}{2c} - \phi\beta_3 \quad -\beta_1 u - \frac{1}{2c} \quad -\beta_1 v \quad \beta_1 \quad \beta_3\right], \\ l_2 &= [1 - 2\beta_2 + 2\phi\beta_3 \quad 2\beta_1 u \quad 2\beta_1 v \quad -2\beta_1 \quad -2\beta_3], \\ l_3 &= [-v \quad 0 \quad 1 \quad 0 \quad 0], \\ l_4 &= [-\phi \quad 0 \quad 0 \quad 0 \quad 1], \\ l_5 &= \left[\beta_2 - \frac{u}{2c} - \phi\beta_3 \quad -\beta_1 u + \frac{1}{2c} \quad -\beta_1 v \quad \beta_1 \quad \beta_3\right], \end{aligned} \tag{6}$$

where

$$c = \sqrt{P(\rho, \epsilon, \phi)_\rho + \frac{P}{\rho^2} P(\rho, \epsilon, \phi)_\epsilon} = \sqrt{\frac{\gamma(\phi) P}{\rho}} \tag{7}$$

is the local sound speed,

$$H = \frac{E + P}{\rho} = \frac{c^2}{\gamma(\phi) - 1} + \frac{1}{2}(u^2 + v^2)$$

is the enthalpy and

$$X = \gamma'(\phi)\epsilon, \quad \beta_1 = \frac{\gamma - 1}{2c^2}, \quad \beta_2 = \beta_1 \frac{u^2 + v^2}{2}, \quad \beta_3 = \frac{\beta_1 X}{\gamma - 1}. \tag{8}$$

The eigenstructure of $G'(U)$ is obtained in the usual manner, by interchanging the roles of u and v and the second and third components of each left and right eigenvector. We point out that these linearizations only depend on four variables, for instance, c, u, v, ϕ. We refer to [7,12,18] for further details.

3. Marquina's flux splitting formula

We describe our flux splitting formulation in the two-dimensional case. We follow Shu–Osher's flux formulation [22] to obtain a numerical flux $\hat{F}$ (respectively $\hat{G}$) that approximates F (respectively G) to an order r. We then use the method of lines to approximate system (3) by applying the TVD third-order Runge–Kutta ODE solver (see [22]) to the following system, obtained by spatial discretization:

$$\frac{\mathrm{d}U_{i,j}}{\mathrm{d}t} + \frac{1}{h}\left(\hat{F}_{i+(1/2),j} - \hat{F}_{i-(1/2),j}\right) + \frac{1}{h}\left(\hat{G}_{i,j+(1/2)} - \hat{G}_{i,j-(1/2)}\right) = 0, \tag{9}$$

where $U_{i,j}(t)$ is an approximation to $U(x_i, y_j, t)$ at a point (x_i, y_j) of a bidimensional equispaced Cartesian mesh, where $x_i = ih$, $y_j = jh$ and h is the mesh spacing. In this expression, $\hat{F}, \hat{G}$ are the *numerical fluxes*, that should approximate the fluxes in the following sense:

$$F(U)_x(x_i, y_j) = \frac{1}{h}\left(\hat{F}_{i+(1/2),j} - \hat{F}_{i-(1/2),j}\right) + \mathcal{O}(h^r), \tag{10}$$

$$G(U)_y(x_i, y_j) = \frac{1}{h}\left(\hat{G}_{i,j+(1/2)} - \hat{G}_{i,j-(1/2)}\right) + \mathcal{O}(h^r), \tag{11}$$

for smooth U. This whole setup gives a conservative scheme which is third-order accurate in time and rth-order accurate in space.

Shu and Osher's idea (see [22]) to obtain the approximation in (10) ((11) is obtained by analogy) is to apply the following idea to each characteristic flux, obtained from F by the local linearizations explained below. Let then $u_t + g(u)_x = 0$ be a scalar one dimensional conservation law. We view the flux $g(u)$ as a sliding average of some "primitive" function ϕ_h:

$$g(u)(x) = \frac{1}{h}\int_{x-(h/2)}^{x+(h/2)} \phi_h(s)\,\mathrm{d}s, \tag{12}$$

for which the partial $g(u)_x$ is given by finite differencing

$$g(u)_x(x) = \frac{\phi_h\left(x + \frac{h}{2}\right) - \phi_h\left(x - \frac{h}{2}\right)}{h}. \tag{13}$$

For smooth enough ϕ_h, one can achieve the desired approximation in (10) by a *reconstruction via primitive* approach, that consists in the construction of a suitable approximation to a primitive Φ_h of ϕ_h and the definition of the approximation to ϕ_h at $x_{i+(h/2)}$ as the derivative at this point of this approximation to Φ_h. Namely, one can define the primitive of ϕ_h as

$$\Phi_h(x) = \int_0^x \phi_h(s)\,\mathrm{d}s, \tag{14}$$

and notice that

$$\frac{\Phi_h\left(x_{i+(1/2)}\right)-\Phi_h\left(x_{i-(1/2)}\right)}{h}=\frac{1}{h}\int_{x_{i-(1/2)}}^{x_{i+(1/2)}}\phi_h(s)\,\mathrm{d}s=g(u)(x_i) \tag{15}$$

by (12), where $x_{i\pm(1/2)}=x_i\pm(h/2)$. This observation is the key to construct approximations to Φ_h from (known) generalized differences of tables of values of $(g(u)(x_l))_l$, for a generalized difference of order q of this table corresponds to a generalized difference of order $q+1$ of the table $(\Phi_h(x_{l+(1/2)}))_l$ and order 0 differences (i.e., values) of this table vanish when differentiating.

In general, a reconstruction procedure R defined at $x_{p+(1/2)}$ $(p=i-1,i)$ from cell averages $\overline{a}_l$ $(l=i-s,\ldots,i+s)$, is required to satisfy

$$R(\overline{a}_{i-s},\ldots,\overline{a}_{i+s};x_{p+(1/2)})=a\left(x_{p+(1/2)}\right)+e\left(x_{p+(1/2)}\right)h^r+\mathcal{O}(h^{r+1}), \tag{16}$$

with

$$\overline{a}_j=\int_{x_{j-(1/2)}}^{x_{j+(1/2)}}a(s)\,\mathrm{d}s,$$

for a smooth function a. The positive integer r is the order of the reconstruction.

The use of a reconstruction for $a=\phi_h$ and Eqs. (14) and (15) yield cell averages

$$\overline{a}_l=\frac{\Phi_h\left(x_{l+(1/2)}\right)-\Phi_h\left(x_{l-(1/2)}\right)}{h}=g(u)(x_l),$$

and, from (16), we deduce

$$\begin{aligned}&R\left(\overline{a}_{i-s},\ldots,\overline{a}_{i+s};x_{i+(1/2)}\right)-R\left(\overline{a}_{i-s-1},\ldots,\overline{a}_{i+s-1};x_{i-(1/2)}\right)\\&\quad=\phi_h\left(x_{i+(1/2)}\right)-\phi_h\left(x_{i-(1/2)}\right)+\left(e\left(x_{i+(1/2)}\right)-e\left(x_{i-(1/2)}\right)\right)h^r+\mathcal{O}(h^{r+1}),\end{aligned}$$

from which one obtains an rth-order numerical flux

$$\hat{g}_{i+(1/2)}=R\left(g(u)(x_{i-s}),\ldots,g(u)(x_{i+s});x_{i+(1/2)}\right). \tag{17}$$

Popular reconstructions are based on polynomials (ENO [10], PPM [25]) or hyperbolas (PHM [16]). We have used the WENO5 [11] reconstruction, which achieves fifth-order accuracy using the same five points stencil ($s=2$) as the ENO3 reconstruction. The WENO5 reconstruction is a nonlinear convex combination of the three interpolating parabolas based on the stencils $x_{i-2+j},x_{i-1+j},x_{i+j}$, $j=0,1,2$, that appear in the ENO3 reconstruction. The nonlinear weights are based on smoothness indicators, judiciously designed so as to attain fifth-order accuracy at smooth regions while degenerating to "digital" ENO3 at discontinuities. We include the precise formulae in an Appendix A for the sake of completeness.

In [6] it has been experimentally established that Marquina's flux splitting has a built-in heat conduction mechanism that alleviates certain pathologies that other solvers exhibit in some experiments, such as overheating at shock reflections, kinked Mach stem (cf. [6]) and carbuncle (cf. [5]). It has been used very successfully in astrophysical simulations at high relativistic regimes (cf. [5,14]). Although, as we will see below, this formulation requires the use of two linearizations for each cell interface, instead of one, as Roe solver does, this increase of computational cost in our implementation is only about 11%, for the time spent in the solver is much lower than the time for the reconstructions.

When applied to the one-dimensional helium bubble simulation (cf. Section 4), our implementation of Shu–Osher's scheme based on Roe's linearization and WENO5 reconstruction works fine for coarse grids, but crashes on medium grids (800 points), when gives negative pressure values, due to dramatic pressure spikes. On the other hand, Marquina's flux splitting seems to naturally prescribe enough viscosity to dissipate spurious oscillations but not so much to smear excessively shocks and contacts. Our implementation is extremely robust and works perfectly in very fine grids with high CFL conditions.

We now describe the construction of the numerical flux $\hat{F}_{i+(1/2),j}$; obtaining $\hat{G}$ is similar. We denote by $\mathbf{u}_{ij}$ an approximation to $U_{i,j}$.

In what follows R stands for a reconstruction procedure; we have mainly used the WENO5 reconstruction, which has a 5-points stencil and is fifth-order accurate, and also in one experiment the PHM reconstruction, which has a 3-point stencil and is third-order accurate. We choose c, u, v, ϕ as the four variables the eigenstructure of the Jacobian $F'(U)$ depends on. We then perform two-sided reconstructions of each of these quantities at the interface to obtain quantities

$$c^{\pm}_{i+(1/2),j}, \quad u^{\pm}_{i+(1/2),j}, \quad v^{\pm}_{i+(1/2),j}, \quad \phi^{\pm}_{i+(1/2),j}, \tag{18}$$

where

$$\begin{aligned} Z^{+}_{i+(1/2),j} &= R\big(Z_{i-s,j}, \ldots, Z_{i+s,j}; x_{i+(1/2)}\big), \\ Z^{-}_{i+(1/2),j} &= R\big(Z_{i-s+1,j}, \ldots, Z_{i+s+1,j}; x_{i+(1/2)}\big), \end{aligned} \tag{19}$$

and Z stands for c, u, v or ϕ. These quantities define two local linearizations at the point $(x_{i+(1/2)}, y_j)$, with corresponding left and right eigenvectors $l^{\pm}_k, r^{\pm}_k$ (we obviate the dependence of these eigenvectors on i, j). We denote

$$f^{\pm}_{m,k} = F(\mathbf{u}_{m,j}) \cdot l^{\pm}_k, \quad \omega^{\pm}_{m,k} = \mathbf{u}_{m,j} \cdot l^{\pm}_k \tag{20}$$

the characteristic fluxes and variables, according to each local linearization and then define the high-order *upwind characteristic fluxes* $\psi^{\pm}_k$ $(k = 1, \ldots, 5)$ by the following algorithm:

if $\lambda_k(\mathbf{u}_{i,j}) > 0$ and $\lambda_k(\mathbf{u}_{i+1,j}) > 0$
  $\psi^{+}_k = R(f^{+}_{i-s,k}, \ldots, f^{+}_{i+s,k}; x_{j+(1/2)})$
  $\psi^{-}_k = 0$
else if $\lambda_k(\mathbf{u}_{i,j}) < 0$ and $\lambda_k(\mathbf{u}_{i+1,j}) < 0$
  $\psi^{+}_k = 0$
  $\psi^{-}_k = R(f^{-}_{i-s+1,k}, \ldots, f^{-}_{i+s+1,k}; x_{j+(1/2)})$
else
  $\alpha = \max(|\lambda(\mathbf{u}_{i,j})|, |\lambda(\mathbf{u}_{i+1,j})|)$
  $\psi^{+}_k = R(\frac{1}{2}(f^{+}_{i-s,k} + \alpha\omega^{+}_{i-s,k}), \ldots, \frac{1}{2}(f^{+}_{i+s,k} + \alpha\omega^{+}_{i+s,k}); x_{j+(1/2)})$
  $\psi^{-}_k = R(\frac{1}{2}(f^{-}_{i-s+1,k} - \alpha\omega^{-}_{i-s+1,k}), \ldots, \frac{1}{2}(f^{-}_{i+s+1,k} - \alpha\omega^{-}_{i+s+1,k}); x_{j+(1/2)})$
end

Then the numerical flux is defined by the flux-split formula

$$\hat{F}_{i+(1/2),j} = \sum_{k=1}^{5} \big(\psi^{+}_k r^{+}_k + \psi^{-}_k r^{-}_k\big). \tag{21}$$

Note that, as usual, this splitting is defined according to the characteristic speeds, but with two "upwind linearizations". At sonic points a local Lax–Friedrichs splitting is used.

We point out that, for general systems, the conditions that "$\lambda(\mathbf{u}_{i,j})$ and $\lambda(\mathbf{u}_{i+1,j})$ have the same sign" and "$\alpha = \max(|\lambda(\mathbf{u}_{i,j})|, |\lambda(\mathbf{u}_{i+1,j})|)$" should be replaced by "if $\lambda(u)$ does not change sign in a path joining $\mathbf{u}_i$ and $\mathbf{u}_{i+1}$" and "$\alpha = \max(|\lambda(u)|)$, u in a path joining $\mathbf{u}_i$ and $\mathbf{u}_{i+1}$". This simplification is permitted by the fact that the characteristic fields for system (3) and (4) are either genuinely nonlinear or linearly degenerate.

4. One-dimensional numerical experiments

We consider the two-component Riemann problem for the 1D Euler equations corresponding to the Sod tube test with a change in the adiabatic exponent

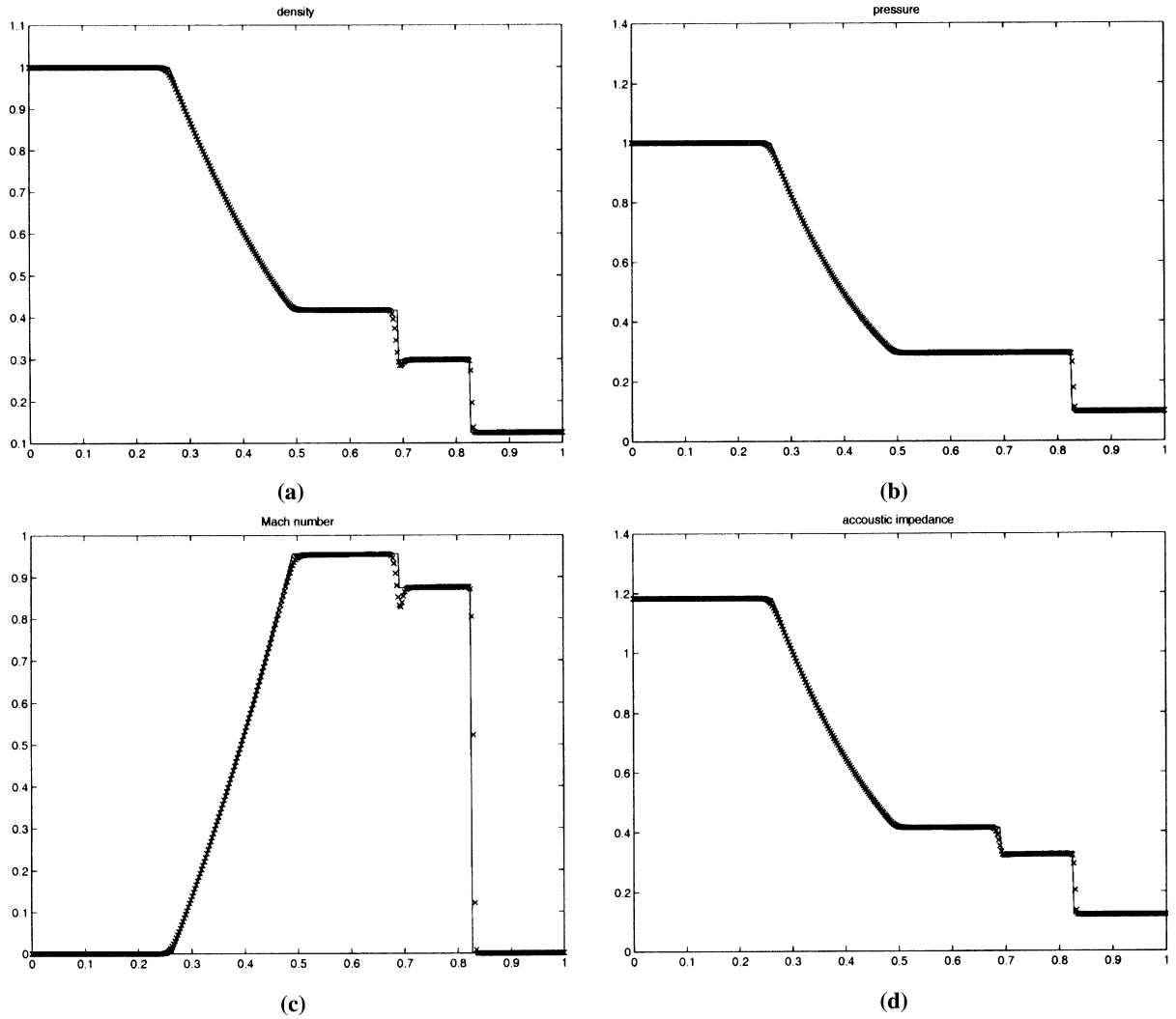


Fig. 1. Comparison of the results of our algorithm (400 points) versus the exact solution of the Sod tube problem with change of adiabatic exponents, $t = 0.2$. (a) Density, (b) pressure, (c) Mach number and (d) acoustic impedance.

$$\begin{aligned} &\rho_l = 1, \quad u_l = 0, \quad P_l = 1, \quad \gamma_l = 1.4, \quad Cv_l = 1, \\ &\rho_r = 0.125, \quad u_r = 0, \quad P_r = 0.1, \quad \gamma_r = 1.2, \quad Cv_r = 1, \end{aligned} \tag{22}$$

where the first set of state variables initially apply to $x \in [0, 0.5]$ and the second to $x \in (0.5, 1]$. This problem has been studied before by several authors [1,12,13]. We compute the approximate solution of this problem with a grid of 400 points and time 0.2, using our fifth-order accurate one-dimensional algorithm. In Fig. 1 we compare the density, pressure, Mach number and acoustic impedance (ρc) obtained in this experiment versus the exact solution (which can be obtained via Rankine–Hugoniot relations, cf. [24]). We observe good agreement and accurate location of the waves. We also notice a slight overheating downstream the contact discontinuity.

Our second test is the one-dimensional version of the shock–bubble interaction computed in our main applications, studied in Section 5. We display in Fig. 2 the density, pressure, Mach number and acoustic

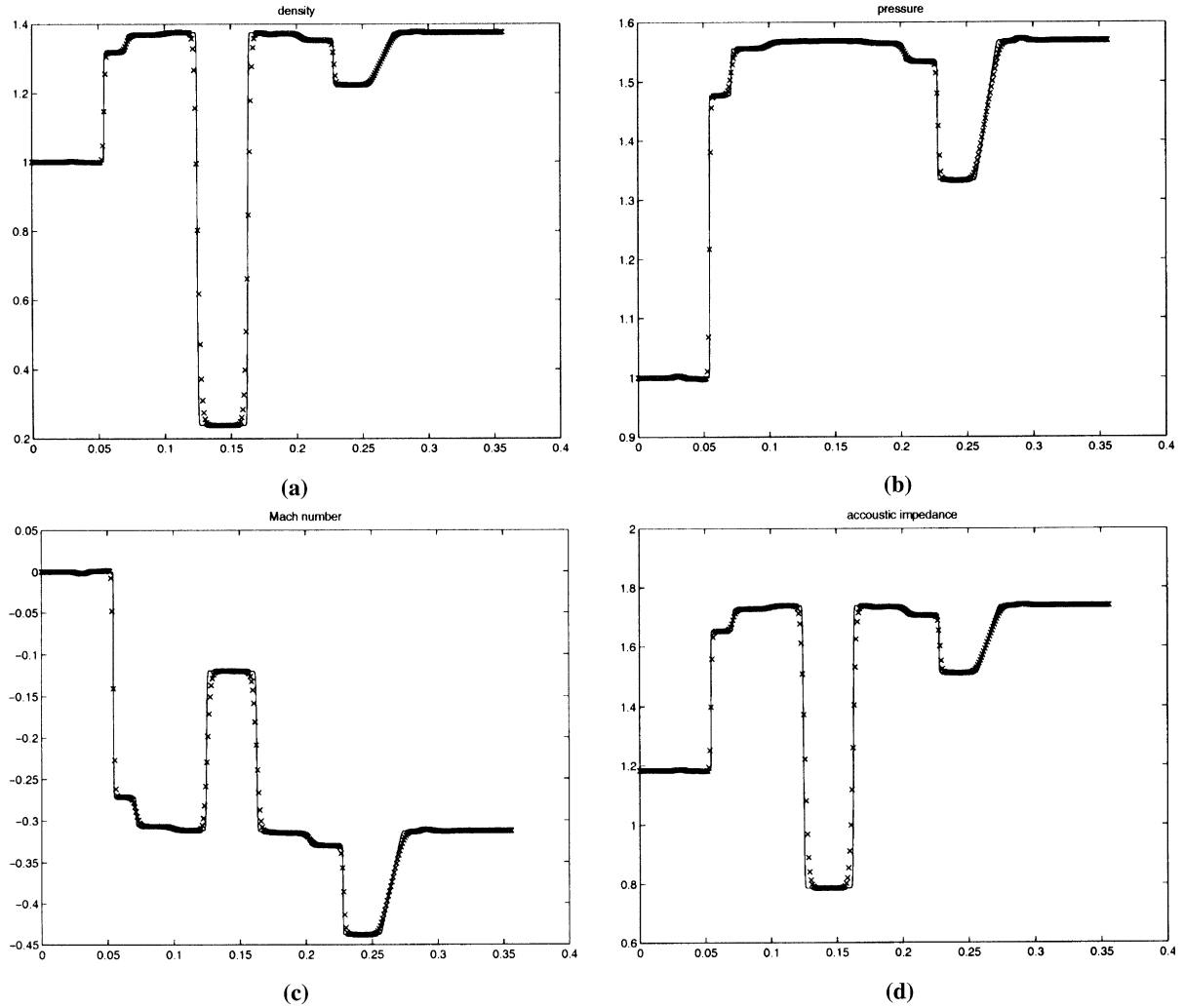


Fig. 2. Comparison of the results of our algorithm with 400 points versus 3200 points when applied to the one-dimensional helium bubble problem, $t = 290\,\mu s$. (a) Density, (b) pressure, (c) Mach number and (d) acoustic impedance.

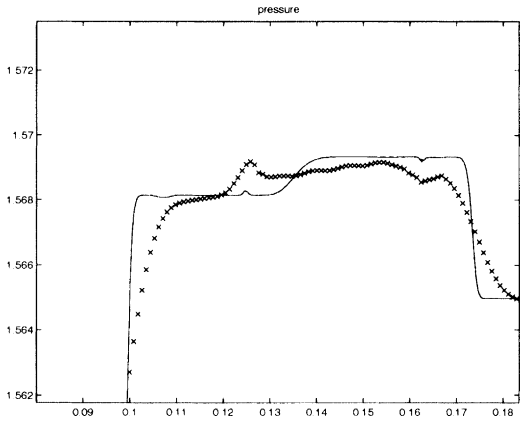


Fig. 3. Comparison of zoom of pressure for the results of our algorithm with 400 points versus 3200 points when applied to the one-dimensional helium bubble problem, $t = 290\,\mu s$.

impedance obtained by our algorithm using 400 points versus those quantities obtained by our algorithm with a grid of 3200 points. We observe that the refracted shock wave inside the bubble is travelling faster due to the fact that helium has a lower acoustic impedance than air. This also implies that the first reflected wave is a rarefaction wave (cf. [26] for a complete study of one-dimensional shock-contact Riemann problems). We depict in Fig. 3 the details of the pressure profile. We can see very mild oscillations at the edges of the bubble corresponding to the low resolution simulation (400 points) and that these oscillations are hard to discern in the high resolution plot (3200 points), that has a resolution close to that used in our main simulation, see Section 5. This is by no means a contradiction to the analysis in [12], that concerns the "startup" error in the pressure when only γ changes value through a contact, mainly because it does not apply at advanced stages of the simulation and for high-order schemes. We do not claim that our scheme does not have pressure oscillations near material interfaces, but that those are really startup errors that dissipate very early in the simulation. In fact, assuming a problem with an initial supersonic material interface (so that the analysis of our scheme is simple, for then the numerical fluxes are real fluxes), it can be seen that the first-order version of our scheme has exactly the same pressure oscillation as the one computed in [12], namely

$$\delta p = \lambda(1-\lambda)\frac{p}{(\gamma_l - 1)(\gamma_r - 1)}\delta\gamma^2, \tag{23}$$

where $\delta\gamma = \gamma_r - \gamma_l$, γ_l and γ_r are the quotients of specific heats at each side of the interface, $\lambda = (\mathrm{d}t/\mathrm{d}x)u$ and u is the (constant) velocity, which we assume to satisfy $u > \max(c_l, c_r)$, with $c_i = \sqrt{\gamma_i p/\rho}$ being the local sound speed at each corresponding side of the interface.

5. Application: shock–bubble interaction

We study the shock–bubble interaction of a 1.22 Mach shock wave with a helium cylindrical bubble in air, originally studied by Haas and Sturtevant in [9] and addressed by Quirk and Karni in [20], from the computational point of view by using a primitive consistent second-order algorithm.

The computational domain is sketched in Fig. 4. We point out that the dimension of the bubble is the same as in [20] and the longitude of the domain is shorter, but this does not affect the simulation. We label

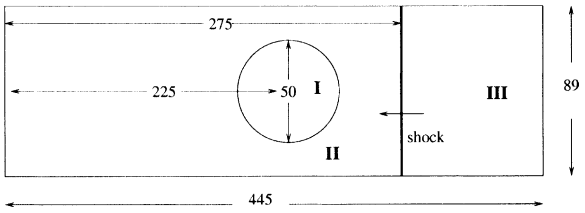


Fig. 4. Sketch of the computational domain (not to scale). Lengths in millimeters.

the interior of the bubble as region I, the pre-shock region outside the bubble as region II, the post-shock region as region III and we use these labels for identifying quantities in corresponding regions.

As in [20], we assume the bubble to be filled with helium contaminated with 28% of air and in thermodynamical and mechanical equilibrium with the surrounding air. We take the density of air at pressure $p_{II} = 101325\,\text{Pa}$ to be $\rho_{II} = 1225\,\text{g/m}^3$, so that its sound speed is 340.294 m/s. By using the relationship $p = R\rho T$ in regions I and II, $p_I = p_{II}$ and $T_I = T_{II}$, we deduce that $\rho_I = \rho_{II}(R_{II}/R_I)$ and obtain from this and Table 1(a) that $\rho_I = 222.8\,\text{g/m}^3$. A 1.22 Mach left traveling vertical shock separates regions II and III. From its Mach number and standard shock relationships (cf. [23, pp. 100–102]) we deduce the state variables in region III. In Table 1(b) we collect the constant state variables in each region.

We have used a computational grid of 8000×800 cells to discretize the upper half part of the domain, the lower have been obtained by symmetry with artificial reflecting boundary conditions. This results in a spatial resolution of 0.056 mm, the same used for the simulation in [20]. We impose reflecting boundary conditions on the top and bottom of the grid and outflow boundary conditions on the left and right sides of the grid. The reflective boundary conditions used in our WENO5 simulation were implemented by defining "ghost cells" where we extended the conserved quantities, changing the sign of the transversal momentum component, and leaving the boundary at the interface. Then, we use enough ghost cells to get defined all the reconstructed characteristic fluxes, using the WENO5 reconstruction procedure. This procedure is more complex than the one proposed by Jiang and Shu, but the resulting code is more consistent with Marquina's flux formula. We did not observe any loss of accuracy, nor overheating effect near the boundaries.

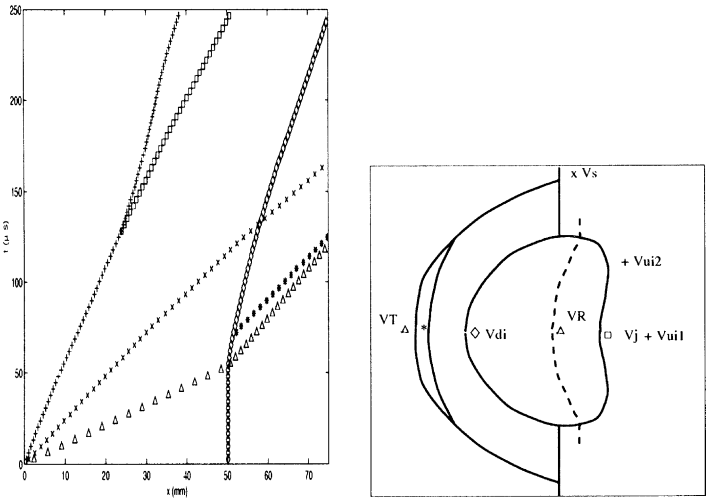
In Fig. 5(a) we show an x–t diagram of the position of the key features explained in Fig. 5(b). We obtain the position of these features by computing the zero crossings of the second difference (i.e., inflection points) of horizontal sections of the density at different times. We use sections taken at the axis of symmetry, except for the upstream bubble interface (for times $> 120\,\mu\text{s}$, we get a section at a height of 20 mm from the axis) and the incident shock (we use a section at 5 mm from the top wall). We compute the mean velocities of these features by using minimum squares line adjustment to approximate the (visually) straight segments of their trajectories displayed in Fig. 5.

In Table 2 we display these velocities, together with the time intervals involved in their computation, and compare them to those obtained by Haas and Sturtevant [9] and Quirk and Karni [20]. We note the good agreement with Quirk and Karni's results and with Haas and Sturtevant's experimental results within the estimated error bounds. The discrepancy of -5% in the velocity of the downstream interface (Vdi in the table) is the most notable. To explain it, we notice in Fig. 5(a) the curvature of the trajectory of the downstream bubble interface, just when the refracted shock hits it. This curvature suggests a positive acceleration of this interface, with an early stage velocity of about 90 m/s at 55 μs and a constant steady-state velocity of 153 m/s in the time interval [140, 240]. Therefore, we could have obtained any mean velocity of this interface, in the range [90, 153 m/s] when averaging; e.g., in the time interval [100, 240] we obtain a mean velocity of 145 m/s, thus matching the experiment. We have nevertheless preferred to obtain the mean velocities at the straight segments of the trajectories, whenever possible.

Table 1

	γ	Cv	R	
(a) *Specific heats and R constant for air and helium contaminated with 28% of air*[a]				
Air	1.4	0.72	0.287	
He + 28% air	1.648	2.44	1.578	
	ρ	u	v	P
(b) *State variables for regions I, II, and III*[b]				
I	222.8	0	0	101,325
II	1225	0	0	101,325
III	1686.1	−156.26	0	250,638

[a] The units for Cv and R are J/(g K).
[b] The units are g/m^3 for the density, m/s for the velocities and Pa for the pressure.



(a) **(b)**

Fig. 5. (a) x–t diagram of the key features explained in (b); (b) VS: incident shock, VR: refracted shock VT: transmitted shock, Vui: upstream border of the bubble, Vdi: downstream border of the bubble, Vj: air jet head.

Regarding the comparison of the velocities obtained in our numerical experiments to those obtained in [20], we point out the very close agreement of the mean velocities displayed in Table 2, above all the one for the refracted shock (exact match) and for the transmitted shock (1.1% of error).

Table 2
Velocities of the features explained in Fig. 5

	VS	VR	VT	Vui1	Vui2	Vdi	Vj
Marquina and Mulet	414	943	373	176	111	153	229
Haas and Sturtevant	410	900	393	170	113	145	230
Percentage error	−1.0	−4.6	5.4	−3.4	1.8	−5.2	0.4
Quirk and Karni	422	943	377	178		146	227
Percentage error	1.9	0	1.1	1.1		−4.6	−0.9

The time intervals for computing each velocity are: VS [0, 60], VR [0, 52], VT [52, 240], Vui1 [10, 52], Vui2 [140, 240], Vdi [140, 240], Vj [140, 240].

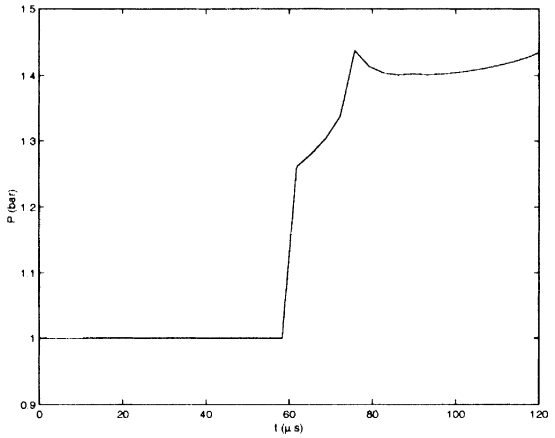


Fig. 6. Pressure profile at 3 mm downstream the initial bubble position.

In Fig. 6 we show the pressure profile (t–P diagrams) at 3 mm downstream the initial bubble position. These compare qualitatively well with the corresponding pressure profile [9, Fig. 17(d), p. 61].

We borrow from [20] a schlieren-type technique for the visualization of weak flow features. Namely, we consider an *idealized schlieren* function on grid data (ρ, ϕ) as

$$s_{i,j} = \exp\left(- k(\phi_{i,j}) \frac{|\nabla \rho_{i,j}|}{\max_{i,j} |\nabla \rho_{i,j}|} \right), \quad k(\phi_{i,j}) = \begin{cases} 20 & \text{if } \phi_{i,j} > 0.25, \\ 100 & \text{if } \phi_{i,j} < 0.25, \end{cases} \tag{24}$$

and notice that $s_{i,j} \approx 1$ when $\rho_{i,j} \approx 0$ and, for large enough k, $s_{i,j} \approx 0$ for large $\rho_{i,j}$. When displayed on a screen with, for instance, MATLAB's commands `colormap gray(256); imagesc(s)`, the larger the density fluctuation the darker the corresponding pixel value.

We display in Fig. 7 our schlieren images that most closely resemble the shadowgraphs displayed in [9] at times: 32, 52, 62, 72, 82, 102, 245, 427, 674 and 983 μs. A close agreement between our images and the experimental shadowgraphs is obtained, but we warn the reader that the times at which these images have been obtained in our simulation are: 23, 43, 53, 67, 75, 102, 260, 445, 674 and 983 μs, which are slightly different from those above. This comes as no surprise, for the key features used to match our schlieren images with the shadowgraphs are the relative positions of some shocks (refracted, transmitted or reflected) with respect salient features of the interface, and those shocks in Haas and Sturtevant's experiment move at different velocities than in our simulation. In the following we expound plausible explanations for these time mismatches.

In [9], one clearly sees that the refracted shock, moving at an estimated velocity of 900 m/s, should be located at 900 m/s × 32 μs = 28 mm to the left of the original upstream (right) bubble edge at 32 μs, that is, it should have passed the center of the 50 mm diameter bubble. But it can be seen that the shock in the corresponding shadowgraph ([9, Fig. 7]) has barely traveled one third of the original bubble diameter.

From high resolution digitalization of the shadowgraphs corresponding to 32 and 52 μs, we measure that the distances traveled by the refracted shock from the initial upstream bubble position are 19.8 mm for 32 μs and 37.1 mm for 52 μs. These measurements give estimates for the mean velocities in the intervals $[0, 32\,\mu s]$ and $[32, 52\,\mu s]$ of 619 and 865 m/s, respectively. This second velocity nearly agrees with the refracted shock velocity estimated in [9] (900 m/s), but the first one is much lower than expected.

If we assume a uniform contamination with air inside the bubble, the refracted shock cannot accelerate once it has penetrated into it. Therefore, we conjecture a formula $s = v(t - t_0)$ for the distance s

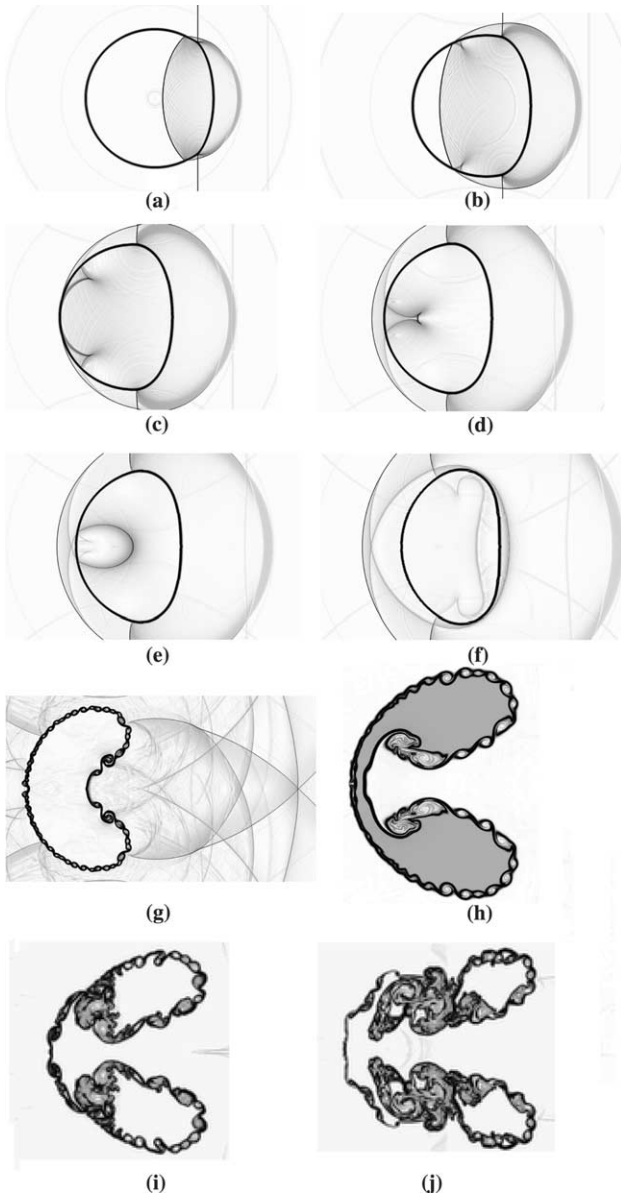


Fig. 7. Schlieren image of the bubble at times (a) 23 μs (32 μs), (b) 42 μs (52 μs, (c) 53 μs (62 μs), (d) 66 μs (72 μs), (e) 75 μs (82 μs), (f) 102 μs (102 μs), (g) 260 μs (245 μs), (h) 445 μs (427 μs), (i) 674 μs, (j) 983 μs (following each time we display in parentheses the corresponding times in the experiment).

traveled by the shock inside the bubble since time t, where v is the velocity and t_0 is a delay. From the measurements above we can estimate $t_0 \approx 9\,\mu s$. This delay is crucial in the early stages of the simulation, since the refracted shock travels so fast that crosses the 50 mm bubble in $50\,mm/943\,m/s = 53\,\mu s$ (see Table 2), so a discrepancy of 9 μs may account for a discrepancy in the shock location of 17% of the bubble diameter.

This delay explains the time mismatch in the pictures corresponding to 32, 52, 62 μs. After this, the gap between the times in the experiment and the simulations gets narrower, until it gets its sign reversed by 102 μs. Now, the match between the experiment and the simulations is basically governed by the transmitted shock, whose velocities in the experiment (393 m/s) and in our simulation (373 m/s) suffer a mismatch of 5.4% (and 4.2% with respect to the results in [20]). This velocity mismatch may cause proportional time mismatches in the simulation, from the time when the shock is transmitted outside the bubble (52 μs) to 260 (respectively 445 μs), when we record our image displayed in Fig. 7(g) (respectively (h)). This can be confirmed by the following calculations: $(260-52)/(245-52) = 1.077 \approx 1.054 = 393/373$, $(445-52)/(427-52) = 1.048 \approx 1.054 = 393/373$.

In Fig. 7(a) we display the bubble at several stages of the simulation:

- 23 μs (this corresponds to 32 μs in the experiment) after it is first hit by the shock: we can see the divergence of the refracted shock, which travels faster inside the bubble, due to the higher sound speed inside the bubble. The reflected wave is an expansion wave, because the acoustic impedance inside the bubble is lower than outside.
- 42 μs (52 μs in the experiment): we observe that the incident shock and the transmitted shock form a quadruple shock configuration, visible at the top; the refracted shock advances fast towards the left bubble interface; the internal reflection of the refracted shock appears as two little cusps behind the crossing of the refracted shock with the interface.
- 53 μs (62 μs in the experiment): the refracted shock hits the downstream interface; the internally reflected wave is clearly seen as two cusps moving towards the axis of symmetry. Meanwhile, the upstream bubble interface is flattening.
- 66 μs (72 μs in the experiment): we can see that the internally reflected wave has emerged from the downstream interface and its two branches appear clearly.
- 75 μs (82 μs in the experiment): now the internally reflected waves have crossed and look like a small oval inside the bubble; the two branches of the transmitted shock have crossed near the downstream bubble edge.
- 102 μs (102 μs in the experiment): at this point, we observe that the internally reflected wave hits the upstream interface, resulting in two very weak waves: the transmitted wave, that is noticed as a back-scattered wave, and a secondary internally reflected wave, that appears very faintly in the picture, but not in the experiment. The interface commences to appear slightly perturbed by the successive accelerations caused by the waves and almost flattened in its right side.
- 260 μs (245 μs in the experiment): now we clearly observe many shocks resulting from reflections of transmitted shocks with the top and bottom walls. There is considerable vorticity generation at the interface and a jet is forming at the upstream edge of the bubble, which begins to adopt a kidney shape.
- 445 μs (427 μs in the experiment): at this point, the jet is more clearly visible and the vorticity has increased considerably.
- 674 μs: we see the bubble spreading out laterally and forming two vortical structures, due to the impact of the jet head on the downstream interface.
- 983 μs: we observe that the two vortical structures are practically separated by a string of little bubbles.

Our simulation shows a complete turbulent regime compared to the Quirk–Karni second-order accurate simulation. Indeed, we observe a very clear vortex structure, induced by the Kelvin–Helmholtz instabilities at the top and bottom interface of the bubble in our simulation, in contrast with the Quirk–Karni simulation where we see a noisy interface. The structure at the center appears to be better resolved for our simulation.

In Fig. 8(a) (respectively (b)) we depict a high resolution schlieren picture of the bubble 546 μs after its interaction with the shock computed with the WENO5 reconstruction, resp. the PHM reconstruction. Comparing those density schlieren images, we can observe that the PHM simulation has not developed all

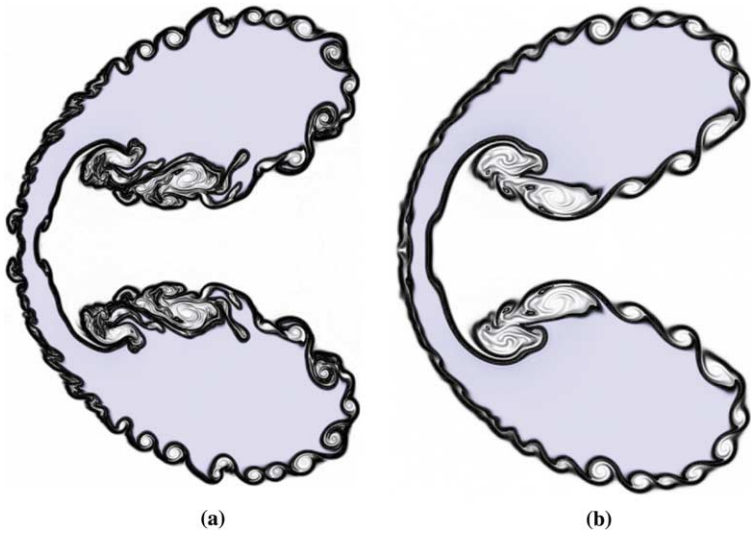


Fig. 8. Color Schlieren image at time 546 μs: (a) computed with WENO5 reconstruction; (b) computed with PHM reconstruction.

the turbulent structure at the interface, (see Kelvin–Helmholtz instabilities and vortex structure). This means that the resulting Reynolds number of the PHM experiment is neatly smaller than the one obtained for our WENO5 experiment. We can conclude that not only the grid size, but the order of accuracy of the reconstruction procedure used in the simulation determines the viscosity and a fortiori the resulting Reynolds number.

As a final remark in this section, we point out that the small discrepancies observed between the experimental data and our numerical simulation might come from the assumption of a uniform contamination of air inside the helium bubble (28% in our case). We therefore conjecture a nonuniform contamination profile, with a high concentration of air near the membrane that rapidly decreases towards the center.

We suggest a cost-effective procedure to adjust the initial data so that the speed and strength of the waves in the numerical simulation better match those in the experiment. Since the speed of propagation of shock waves depends on the acoustic impedance, it is enough to use a coarse grid to fit the initial contamination profile to get a satisfactory match for the velocity of the main waves. We can afford many trials on this coarse grid, for the computation of the simulation on it is cheap. Once we have adjusted the initial contamination profile, we can proceed to get a high resolution computation in a fine grid to get the compressible turbulent regime completely developed.

6. Level sets, morphology and convergence

The method described in this paper evolves two fluids, allowing some region of mixture, for which the conservation of partial masses is ensured by the conservative formulation. In general, for stable interfaces, we expect our algorithm to achieve convergence and conservation, since we use the inviscid Euler equations as model. However, if the interface is unstable, due to either Kelvin–Helmholtz [8] or Richtmyer–Meshkov [21] instabilities, the vanishing viscosity solution does not exist, thus we try to approximate a viscous solution depending on the grid resolution and degree of accuracy that represents the physical experiment with a resulting Reynolds number.

One way to measure the degree of instability of the interface consists in exploring the dependence of the length of the instability $L_t(h)$, for a fixed time t, on the spatial resolution h. If the interface is stable we expect the limit

$$\lim_{h\downarrow 0} L_t(h) = L_0 \tag{25}$$

to exists and to be finite.

However, if the interface is unstable, we conjecture a rate of growth of $L_t(h)$, namely, the existence of an exponent $0 < p < 1$ such that

$$L_t(h) = \mathcal{O}(h^{-p}). \tag{26}$$

Following [27], we denote by Γ the zero level set of a function ψ,

$$\Gamma = \{(x,y) : \psi(x,y) = 0\}, \tag{27}$$

and by $H(z)$ the Heaviside function ($H(z) = 1$ for $z \geqslant 0$, $H(z) = 0$ for $z < 0$). Then $H'(z)$ is the Dirac delta distribution and we have the following formula for the length of Γ:

$$L(\Gamma) = \int_\Omega |\nabla H(\psi(x,y))|\,\mathrm{d}x\,\mathrm{d}y = \int_\Omega \delta(\psi(x,y))|\nabla\psi|\,\mathrm{d}x\,\mathrm{d}y. \tag{28}$$

We implement this formula by numerical integration, using a smooth approximation of the Dirac delta

$$\tilde{H}(z) = \frac{1}{2}\left(\frac{z}{\sqrt{z^2+\beta}} + 1\right) \approx H(z), \tag{29}$$

$$\tilde{\delta}(z) = \tilde{H}'(z) = \frac{1}{2}\left(\frac{\beta}{(z^2+\beta)^{3/2}}\right) \approx \delta(z), \tag{30}$$

where $\beta = h$. That is, we approximate

$$\begin{aligned} L(\Gamma) &\approx \tilde{L}(\Gamma) = h^2 \sum_{i,j} \tilde{\delta}(\psi_{i,j}) \sqrt{\left(\nabla^x_{i,j}\psi\right)^2 + \left(\nabla^y_{i,j}\psi\right)^2}, \\ \nabla^x_{i,j}\psi &= \frac{1}{h}\mathrm{minmod}(\psi_{i+1,j} - \psi_{i,j}, \psi_{i,j} - \psi_{i-1,j}), \\ \nabla^y_{i,j}\psi &= \frac{1}{h}\mathrm{minmod}(\psi_{i,j+1} - \psi_{i,j}, \psi_{i,j} - \psi_{i,j-1}). \end{aligned} \tag{31}$$

We apply (31) to the function $\psi = 2\phi - 1$, where ϕ is the mass fraction (cf. Section 2) of the simulation at $t = 546\,\mu\mathrm{s}$ (a schlieren-type image of this simulation for the finest resolution is displayed in Fig. 8). The zero level set of ψ corresponds to the $\frac{1}{2}$ level set of ϕ, i.e., the points with a 50% of concentration of each species. The approximated lengths of this level set obtained from spatial resolutions of $h_1 = 2.2 \times 10^{-4}$, $h_2 = 8.90 \times 10^{-5}$ and $h_3 = 5.6 \times 10^{-5}$ are $l_1 = 0.2732$, $l_2 = 0.4047$ and $l_3 = 0.6573$, respectively. Regarding (26), we try to adjust these data to the formula

$$l_i = (1 + \epsilon_i) K h_i^{-p}, \tag{32}$$

where ϵ_i is some error that arises from errors in the simulation, in the approximation of the interface length and, above all, the chaotic nature of the instability and K is a fitting parameter. By taking logarithms, the formula (32) can be cast to a linear system, to be solved by least squares

$$\log l_i = \log K - p \log h_i. \tag{33}$$

From the least-squares fitting, we obtain $p = 0.61$ as an estimate for the exponent that governs the interface growth length.

7. Computational issues: MPI

We have used the MPI standard for message passing (as implemented in the MPICH library [17]) for our two dimensional simulations. We have run our implementations on a Linux Beowulf cluster [3], consisting of ten i386-compatible processors, each capable of performing floating point operations at a rate of about 600 MFlops/s and connected by a fast ethernet switch, with a bisection bandwidth of about 50 Mbytes/s. These processors are arranged as a logical linear computing array and global communications is seldom needed.

The relatively large communication latency is hidden by our coarse grain parallel implementation, that assigns patches of 800×800 cells to each of the 10 processors and only needs to interchange a narrow vertical band (3×800 for our WENO5 implementation) with each of its two neighbours, to keep the "ghost cells" (those adjacent to the boundary of the "local" computational domain) updated. The load balance is very satisfactory. We have measured a parallel efficiency of 89%, i.e.,

$$\frac{\text{Time(1 proc)}}{10 \times \text{Time(10 proc)}} \approx 0.89.$$

Our brute force effort employs 378 wall-clock hours (or 3780 cpu hours for the 10 CPUs) for the simulation of the shock–bubble interaction. We plan to implement our algorithms using the AMR technique [4,19,20].

8. Conclusions

We have proposed in this paper a flux-split algorithm that resolves a fully conservative model based on the inviscid Euler equations for a fluid flow that consists of a mixture of ideal gases in thermal equilibrium. This algorithm uses the WENO5 spatial reconstruction to achieve fifth-order spatial accuracy. We have shown the robustness, numerical accuracy and essential lack of important oscillations near interfaces through the simulation of an air–helium shock–bubble interaction. We have found very good agreement with Haas and Sturtevant's experimental data as well.

Acknowledgement

Research supported by DGICYT project BFM2001-2814.

Appendix A

The values for the WENO5 reconstruction (see [11]) that are needed in (19) and in the algorithm in Section 3 are defined by

$$\begin{aligned} Z^+_{i+(1/2),j} &= R\left(Z_{i-2,j}, \ldots, Z_{i+2,j}; x_{i+(1/2),j}\right) = R_{\text{weno5}}(Z_{i-2,j}, \ldots, Z_{i+2,j}), \\ Z^-_{i+(1/2),j} &= R\left(Z_{i-1,j}, \ldots, Z_{i+3,j}; x_{i+(1/2),j}\right) = R_{\text{weno5}}(Z_{i+3,j}, \ldots, Z_{i-1,j}), \end{aligned} \tag{A.1}$$

where the function R_{weno5} is given by

$$a_0 = \frac{1}{10}\left(\varepsilon + \frac{13}{12}(f_{-2} - 2f_{-1} + f_0)^2 + \frac{1}{4}(f_{-2} - 4f_{-1} + 3f_0)^2\right)^{-2},$$

$$a_1 = \frac{6}{10}\left(\varepsilon + \frac{13}{12}(f_{-1} - 2f_0 + f_1)^2 + \frac{1}{4}(f_{-1} - f_1)^2\right)^{-2},$$

$$a_2 = \frac{3}{10}\left(\varepsilon + \frac{13}{12}(f_0 - 2f_1 + f_2)^2 + \frac{1}{4}(3f_0 - 4f_1 + f_2)^2\right)^{-2},$$

$$S = a_0 + a_1 + a_2, w_i = a_i/S,$$

$$R_{\text{weno5}}(f_{-2}, \ldots, f_2) = w_0\left(\frac{1}{3}f_{-2} - \frac{7}{6}f_{-1} + \frac{11}{6}f_0\right) + w_1\left(-\frac{1}{6}f_{-1} + \frac{5}{6}f_0 + \frac{1}{3}f_1\right) + w_2\left(\frac{1}{3}f_0 + \frac{5}{6}f_1 - \frac{1}{6}f_2\right).$$

The value of ε should be small. In this paper we have chosen $\varepsilon = 10^{-6}$.

References

[1] R. Abgrall, Generalization of the Roe scheme for the computation of mixture of perfect gases, La Recherche Aérospatiale 6 (1988) 31–43.
[2] R. Abgrall, S. Karni, Computations of compressible multifluids, J. Comput. Phys. 169 (2001) 594–623.
[3] Beowulf Linux clusters. Available from <http://www.beowulf.org/>.
[4] M. Berger, J. Oliger, Adaptive mesh refinement for hyperbolic partial differential equations, J. Comput. Phys. 53 (1984) 482–512.
[5] R. Donat, J. Font, J.-M. Ibáñez, A. Marquina, A flux-split algorithm applied to relativistic flows, J. Comput. Phys. 146 (1998) 58–81.
[6] R. Donat, A. Marquina, Capturing shock reflections: an improved flux formula, J. Comput. Phys. 125 (1996) 42–58.
[7] R.P. Fedkiw, T. Aslam, B. Merriman, S.J. Osher, An Eulerian approach to Interfaces in multimaterial compressible flows, (The Ghost Fluid Method), J. Comput. Phys. 152 (1999) 452–492.
[8] R. Gerwin, Stability of interface between fluids in relative motion, Rev. Mod. Phys. 40 (1968) 652–658.
[9] J.-F. Haas, B. Sturtevant, Interaction of weak shock waves with cylindrical and spherical gas inhomogeneities, J. Fluid Mech. 181 (1987) 41–76.
[10] A. Harten, B. Engquist, S. Osher, S. Chakravarthy, Uniformly high-order accurate essentially nonoscillatory schemes. III, J. Comput. Phys. 71 (1987) 231–303.
[11] G. Jiang, C. Shu, Efficient implementation of weighted ENO schemes, J. Comput. Phys. 126 (1996) 202–228.
[12] S. Karni, Multicomponent flow calculations by a consistent primitive algorithm, J. Comput. Phys. 112 (1994) 31–43.
[13] B. Larrouturou, How to preserve the mass fractions positive when computing compressible multi-component flows?, J. Comput. Phys. 95 (1991) 59–84.
[14] R.J. LeVeque, D. Mihalas, E. Dorfi, E. Mueller, Computational methods for astrophysical fluid flow, Lecture Notes, vol. 1997, Springer, Berlin, 1998.
[15] X.-D. Liu, S. Osher, T. Chan, Weighted essentially non-oscillatory schemes, J. Comput. Phys. 115 (1994) 200–212.
[16] A. Marquina, Local piecewise hyperbolic reconstructions for nonlinear scalar conservation laws, SIAM J. Sci. Comput. 15 (1994) 892–915.
[17] MPICH-A portable implementation of MPI. Available from <http://www-unix.mcs.anl.gov/mpi/mpich/>.
[18] W. Mulder, S.J. Osher, J. Sethian, Computing interface motion in compressible gas dynamics, J. Comput. Phys. 100 (1992) 209–228.
[19] J. Quirk, An adaptive mesh refinement algorithm for computational shock hydrodynamics, Ph.D. Thesis, Cranfield Institute of Technology, 1991.
[20] J. Quirk, S. Karni, On the dynamics of a shock-bubble interaction, J. Fluid Mech. 318 (1996) 129–163.

[21] R. Richtmyer, Taylor instability in shock acceleration of compressible fluids, Commun. Pure Appl. Math. 23 (1960) 297–319.
[22] C.-W. Shu, S.J. Osher, Efficient implementation of essentially nonoscillatory shock-capturing schemes. II, J. Comput. Phys. 83 (1989) 32–78.
[23] E. Toro, Riemann Solvers and Numerical Methods for Fluid Dynamics, Springer, Berlin, 1999.
[24] G. Whitham, Linear and Nonlinear Waves, Wiley, New York, 1974.
[25] P. Woodward, P. Colella, The numerical simulation of two-dimensional fluid flow with strong shocks, J. Comput. Phys. 54 (1984) 115–173.
[26] Q.Z.Y. Yang, D. Sharp, Small amplitude theory of Richtmyer–Meshkov instability, Phys. Fluids A 5 (1994) 1856–1873.
[27] H. Zhao, T. Chan, B. Merriman, S. Osher, A variational level set approach to multiphase motion, J. Comput. Phys. 127 (1996) 179–195.