AIAA Journal
Vol. 47, No. 12, December 2009

Ghost Fluid Method for Strong Shock Interactions Part 2: Immersed Solid Boundaries

Shiv Kumar Sambasivan* and H. S. UdayKumar†
The University of Iowa, Iowa City, Iowa 52242

DOI: 10.2514/1.43153

Numerical simulation of shock waves interacting with multimaterial interface is immensely challenging, particularly when the embedded interface is retained as a sharp entity. The challenge lies in accurately capturing and representing the interface dynamics and the wave patterns at the interface. In this regard, the ghost fluid method has been successfully used to capture the interface conditions for both fluid–fluid and solid–fluid interfaces. However, the ghost fluid method results in over/underheating errors when shocks impact interfaces, and hence must be supplemented with numerical corrective measures to mitigate these errors. Such corrections typically fail for strong shock applications. Variants and extensions of the ghost fluid method have been proposed to remedy its shortcomings with mixed success. In this paper, the performance of approaches based on the ghost fluid method, in the case of strong shocks impinging on immersed solid boundaries in compressible flows, is evaluated. It is found that (from the viewpoint of simplicity, robustness, and accuracy) a reflective boundary condition used in conjunction with a local Riemann solver at the interface proves to be a good choice. The method is found to be stable, accurate, and robust for wide range problems involving strong shocks interacting with embedded solid objects.

I. Introduction

THE interaction of shocks with multimaterial interfaces is important in several applications, including high-speed flows with droplets [1], bubbles, and particles [2,3], hypervelocity impact and penetration [4], and detonation diffraction [5,6]. In such problems, simulations encounter challenges associated with the treatment of material interfaces, particularly when shocks and detonation waves impinge upon them. The result of such shock-interface interactions may be partial reflections, refractions or transmission of the shocks, and the motion of the material interfaces under the influence of the shocked fields [7]. These motions can be large and may also lead to topological changes, such as shattering, reconnection, collapse, etc.

Recently, the use of embedded boundary methods for the representation of solid boundaries in incompressible [8–11] and compressible [4,12–16] flows has become popular. These methods are attractive for the treatment of moving boundaries in the flow domain because the complex process of grid generation and management can be dispensed with. Particularly, the ghost fluid method (GFM) [12] provides a simple framework to implicitly transmit the presence of an embedded interface to the flowfield. GFM requires the definition of a band of ghost points on either side of the interface, corresponding to each phase of the interacting media. The band of ghost fluid, when populated with suitable flow properties, together with the respective real fluid constitutes a continuous flowfield in which a single equation of state can be employed. Thus, discretization schemes with uniform order of accuracy can be applied throughout the computational domain without requiring any special treatment near the interface. The implementation of the numerical scheme and the interface treatment are decoupled, and the onus is shifted to populating the ghost points with physically consistent values.

When applied to shock interactions with stationary or moving solid objects embedded in compressible flows, the GFM was found to suffer from acute over/underheating numerical errors [14,17]. The GFM coupled with an isobaric fix proposed by Fedkiw et al. [12], although successful in handling solid–fluid interfaces for a limited range of shock strengths, failed for strong shocks interacting with a fluid–fluid interface. It was found that the failure of GFM to accurately resolve the interface could be alleviated by populating the ghost points by solving a local Riemann problem normal to the interface [14]. This was later adopted by several researchers, both in the GFM and non-GFM framework [14,18–22]. However, these methods were mostly developed to treat fluid–fluid interfaces, and hence it is not clear how one could extend such methods to represent embedded solid objects. A variant of the GFM, the characteristic-based matching (CBM) method proposed by Nourgaliev et al. [15], was found to be effective in reducing the shock-interface interaction errors significantly. In the present work, a unified formulation to treat both solid–fluid and fluid–fluid interfaces is presented.

Briefly, a simple reflective boundary condition (RBC) method is used to populate the ghost points and the resulting heating errors are corrected by augmenting the solutions by solving a Riemann problem between the real field and the reflected ghost field. The Riemann problem constructed normal to the interface is solved using an exact Riemann solver, and the resulting Riemann states are used to populate the ghost points. This approach of solving a Riemann problem is simpler to implement than the CBM approach and is also shown later to be computationally less expensive. The Riemann-solver-based interface treatment technique to represent embedded fluid–fluid interfaces is discussed in a separate work [23], and this paper focuses on extending this technique to treat the presence of solid objects in compressible flows. As demonstrated in the results from the current simulations, the proposed method is found to yield satisfactory solutions for several complex configurations and shock diffraction phenomena. Shocks interacting with multiple particles and complex shapes have been computed, and the method is currently being applied to study the dynamics of dense particulate compressible flows.

II. Governing Equations

The governing equations comprise a set of hyperbolic conservation laws. In Cartesian coordinates, the governing equations in two-dimensions take the following form:

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

where

Received 9 January 2009; revision received 20 July 2009; accepted for publication 17 August 2009.

*Graduate Student, Department of Mechanical and Industrial Engineering; ssambasi@engineering.uiowa.edu. Student Member AIAA.

†Professor, Department of Mechanical and Industrial Engineering; ush@engineering.uiowa.edu. Member AIAA (Corresponding Author).

$$\mathbf{U}=\begin{pmatrix}\rho\\ \rho u\\ \rho v\\ \rho E\end{pmatrix},\qquad \mathbf{F}=\begin{bmatrix}\rho u\\ \rho u^2+p\\ \rho uv\\ u(\rho E+p)\end{bmatrix},\qquad \mathbf{G}=\begin{bmatrix}\rho v\\ \rho uv\\ \rho v^2+p\\ v(\rho E+p)\end{bmatrix}$$

where $E=e+\frac{1}{2}(u^2+v^2)$ is the total internal energy and e is specific internal energy. For the purely inviscid flows of interest here, the source term **S** in Cartesian coordinates is set to zero. Extension of the solution methodology to Navier–Stokes equations is straightforward. Closure for the preceding set of governing equations is obtained by using a stiffened equation of state [24,25],

$$P=\rho e(\gamma-1)-\gamma P_\infty \tag{2}$$

where γ is the specific heat ratio (also called a Grüneisen exponent [1] for stiff fluids) and P_∞ is a material-dependent constant [26]. For the case of ideal gas, we have $\gamma=\frac{c_p}{c_v}$ and $P_\infty=0$. For stiff fluids like water, the Grüneisen exponent and the material-dependent constant take the value of 5.5 and 0.613 GPa, respectively. From the definition of sound speed, we have

$$c^2=\left(\frac{\partial P}{\partial\rho}\right)_{s=\text{constant}} \tag{3}$$

Using Eq. (2), the speed of sound then becomes

$$c=\sqrt{\frac{\gamma(P+P_\infty)}{\rho}} \tag{4}$$

III. Sharp Interface Treatment Methods

A. Implicit Interface Representation Using Level Sets

Sharp interface treatment methods require continuous tracking and representation of the interface. In this work, level sets introduced by Osher and Sethian [27] are used to represent the embedded interface. The level set field is advected using the level set advection equation [28,29]:

$$\frac{\partial\phi_l}{\partial t}+\mathbf{V}_l\bullet\nabla\phi_l=0 \tag{5}$$

where ϕ_l and $\mathbf{V}_l$ denote the level set and the level set velocity field for the lth embedded surface. A fourth-order essentially nonoscillatory (ENO) scheme for spatial discretization and a fourth-order Runge–Kutta (RK) time integration are used in solving the level set advection equation. The normal vector and the curvature at the interface can be computed from the level set field according to

$$\mathbf{n}_l=\frac{\nabla\phi_l}{\|\nabla\phi_l\|} \tag{6}$$

$$\kappa=-\nabla\bullet\mathbf{n} \tag{7}$$

For other relevant details regarding the implementation of level set methods, the reader is referred to the original literature [27,30].

B. Boundary Conditions

The tracked interface has to be coupled with the flow solver such that the jump in the mass, momentum, and energy fluxes (along with the material properties across the interface) are depicted accurately. In the GFM approach, this translates to suitably populating the ghost points. However, there is a significant difference between the treatments of fluid–fluid and solid–fluid interfaces in the GFM framework. In the case of fluid–fluid interfaces, the flow variables (such as velocity and pressure) that are continuous across the interface are injected into the ghost field directly. In the latter case, due to the absence of flowfields on one side of the interface (the solid side), it is not immediately clear how to populate the ghost points with the virtual flow properties.

At the interface of a solid body immersed in compressible flows, the following boundary conditions are to be applied for velocity, temperature, and pressure fields:

No-penetration condition for the normal velocity

$$v_n=U_n \tag{8}$$

where U_n is the velocity of the center of mass of the embedded rigid object.

Slip condition for the tangential velocity

$$\frac{\partial v_{t_1}}{\partial n}=0 \tag{9}$$

$$\frac{\partial v_{t_2}}{\partial n}=0 \tag{10}$$

Adiabatic condition for temperature

$$\frac{\partial T}{\partial n}=0 \tag{11}$$

Normal force balance on pressure

$$\frac{\partial p}{\partial n}=\frac{\rho_s v_{t_1}^2}{R}-\rho_s a_n \tag{12}$$

where

$$v_n=\mathbf{V}\bullet\hat{n} \tag{13}$$

$$v_{t_1}=\mathbf{V}\bullet\hat{t}_1 \tag{14}$$

$$v_{t_2}=\mathbf{V}\bullet\hat{t}_2 \tag{15}$$

are the normal and (contravariant) tangential velocities in the local body-fitted curvilinear coordinate; **V** is the velocity vector in the global Cartesian coordinate; $\hat{n}$, $\hat{t}_1$, and $\hat{t}_2$ are the normal and tangential vectors; R is the radius of curvature; and a_n is the acceleration of the interface.

In case of the Navier–Stokes equations, the slip condition is replaced by the no-slip condition at the interface. The set of boundary conditions (8–12) govern the behavior of the flow near the embedded solid body and must be enforced on the real fluid by suitably populating the ghost points. As outlined in this section, there are several approaches for applying these boundary conditions within the spirit of the GFM. However, as will be demonstrated in the results section of this paper, not all of them are successful in computing shock-interface interactions in a robust fashion.

C. Boundary Condition Type 1: Reflective Boundary Condition

The RBC approach is widely adopted to depict the presence of a rigid body in the fluid [5,31–34]. RBC, as the name indicates, reflects the real flowfields across the interface. In the GFM framework, the fields from the real fluid are reflected onto the ghost points in the solid phase. The reflection operation is carried out such that the boundary conditions (8–12) are enforced exactly at the interface. Because the boundary conditions involve the flow properties in the normal and tangential directions, the reflection process must be carried out in the local body-fitted curvilinear coordinates (v_n, v_{t_1}, and v_{t_2}) instead of the global Cartesian coordinates (u, v, and w).

To carry out the reflection process accurately, the reflected point on the real fluid side for each interfacial ghost point on the solid phase must be determined. As shown in Fig. 1, the reflected point IP1 on the real fluid side, corresponding to ghost point P, can be obtained by inserting a probe through point P. IP is the point of intersection of the probe with the interface. The absolute value of the level set function $|\phi_P|$ gives the normal distance of point P from the interface. This value ϕ_P can then be used to march into the real fluid to determine the coordinates of the reflected point IP1:

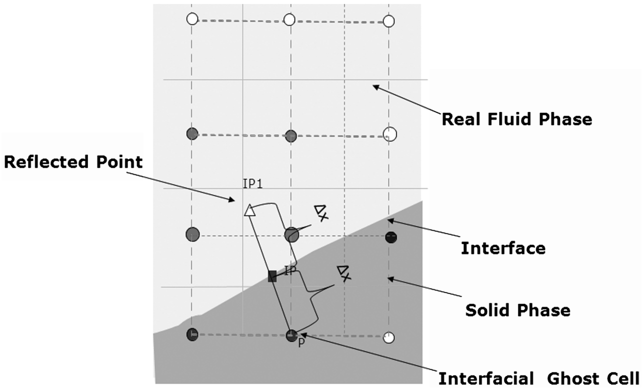


Fig. 1 Illustration of grid points used in RBC.

$$\mathbf{X}_{\mathrm{IP1}} = \mathbf{X}_P + 2 \mid \phi_P \mid \mathbf{N}_P \tag{16}$$

The flow properties at the reflected point can be obtained using a regular bilinear interpolation from the surrounding computational points, as shown in Fig. 1. It must be noted that one or more of these neighboring computational points may lie inside the solid, in which case the corresponding ghost fluid properties are used in the interpolation procedure. Once the flow properties at the reflected point are determined, it is straightforward to reflect the flow properties on to point P.

1. Embedding Boundary Conditions with the Interpolation Procedure

Sometimes, an interfacial ghost point, like point P in Fig. 2, may lie very close to the interface. In such cases, using a direct bilinear interpolation procedure will involve the (as yet unknown) ghost point. To avoid interpolating from the ghost point in consideration, the procedure discussed by Ye et al. [35] is employed as follows.

For any reflected point IP1, the surrounding interpolating points 1, 2, 3, and P are determined. If point P corresponds to the interfacial ghost point for which the flow properties of the reflected point are to be determined, then point P should not be used in the interpolation procedure. Instead point 4 (or IP) on the interface can be used, for which either the value of the flow variables (Dirichlet conditions) or the flow gradient (Neumann-type conditions) is available. Thus, it is necessary to embed the appropriate boundary conditions to complete the interpolation procedure. Accordingly, the value of the flow variables at the reflected point IP1 can be obtained using a bilinear interpolation of the form:

$$\psi = a_1 + a_2 x + a_3 y + a_4 xy \tag{17}$$

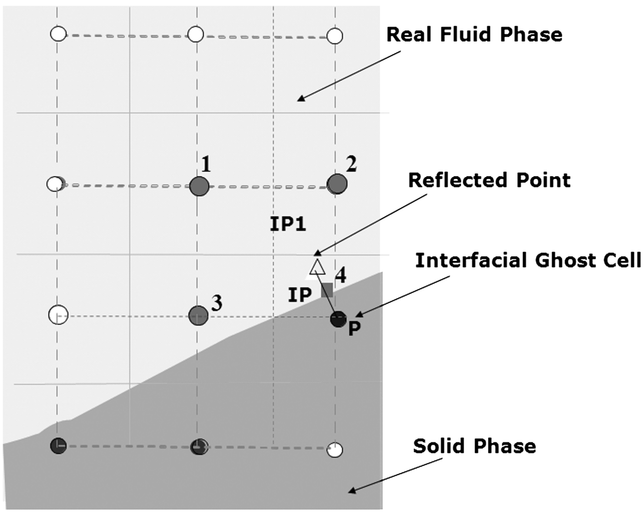


Fig. 2 Embedding the boundary conditions with the interpolation procedure.

where (x, y) are the coordinates of the reflected point IP1 and ψ corresponds to the flow variables. To determine the coefficients $a_1 - a_4$, Eq. (17) is applied to the surrounding points chosen for the interpolation procedure. Hence, for Dirichlet type boundary conditions, Eq. (17) reduces to

$$\begin{pmatrix} 1 & x_1 & y_1 & x_1 y_1 \\ 1 & x_2 & y_2 & x_2 y_2 \\ 1 & x_3 & y_3 & x_3 y_3 \\ 1 & x_{\mathrm{IP}} & y_{\mathrm{IP}} & x_{\mathrm{IP}} y_{\mathrm{IP}} \end{pmatrix} \begin{pmatrix} a_1 \\ a_2 \\ a_3 \\ a_4 \end{pmatrix} = \begin{pmatrix} \psi_1 \\ \psi_2 \\ \psi_3 \\ \psi_4 \end{pmatrix} \tag{18}$$

and for Neumann-type conditions, we have

$$\begin{pmatrix} 1 & x_1 & y_1 & x_1 y_1 \\ 1 & x_2 & y_2 & x_2 y_2 \\ 1 & x_3 & y_3 & x_3 y_3 \\ 0 & n_x & n_y & n_x y_{\mathrm{IP}} + n_y x_{\mathrm{IP}} \end{pmatrix} \begin{pmatrix} a_1 \\ a_2 \\ a_3 \\ a_4 \end{pmatrix} = \begin{pmatrix} \psi_1 \\ \psi_2 \\ \psi_3 \\ S_4 \end{pmatrix} \tag{19}$$

The last row of the coefficient matrix in Eq. (19) is obtained by differentiating Eq. (17), noting that

$$\frac{\partial \psi}{\partial n} = \frac{\partial \psi}{\partial x} n_x + \frac{\partial \psi}{\partial y} n_y \tag{20}$$

where n_x and n_y are the components of the normal vector at point P, and S_4 corresponds to the value of the normal gradient at the interface. After determining the coefficients, the flow properties at the reflected point can be determined using Eq. (17).

2. Determining the Flow Properties at the Interfacial Ghost Points

Once the flow properties at the reflected point IP1 are determined, the next step is to incorporate the boundary conditions (8–12) at the interface. In the RBC framework, the boundary conditions take the following form.

The no-penetration condition is satisfied by reflecting the normal velocity, as given by

$$v_{n_P} = 2U_{n_{\mathrm{IP}}} - v_{n_{\mathrm{IP1}}} \tag{21}$$

where $U_{n_{\mathrm{IP}}}$ is the velocity of the center of mass of the solid body. For the tangential velocities, Eq. (9) must be satisfied at the interface. The Neumann condition for the tangential velocities in the RBC framework then becomes

$$v_{t_{1_P}} = 2U_{t_{1_{\mathrm{IP}}}} - v_{t_{1_{\mathrm{IP1}}}} \tag{22}$$

$$v_{t_{2_P}} = 2U_{t_{2_{\mathrm{IP}}}} - v_{t_{2_{\mathrm{IP1}}}} \tag{23}$$

where, for slip

$$U_{t_{1_{\mathrm{IP}}}} = v_{t_{1_{\mathrm{IP1}}}} \tag{24}$$

$$U_{t_{2_{\mathrm{IP}}}} = v_{t_{2_{\mathrm{IP1}}}} \tag{25}$$

and for no slip

$$U_{t_{1_{\mathrm{IP}}}} = 0 \tag{26}$$

$$U_{t_{2_{\mathrm{IP}}}} = 0 \tag{27}$$

where $U_{t_{1_{\mathrm{IP}}}}$ and $U_{t_{2_{\mathrm{IP}}}}$ are the tangential velocities at IP. Along the same lines, the pressure field can be obtained from Eq. (12),

$$P_P = P_{\mathrm{IP1}} - \left(\frac{\rho_p v_{t_{1_P}}^2}{R_P} - \rho_p a_{n_P} \right) 2 \mid \phi_P \mid \tag{28}$$

where a_{n_P} is the acceleration of the solid. The density ρ_p used in Eq. (28), is obtained by reflecting the value at IP1; that is,

$$\rho_P = \rho_{\mathrm{IP1}} \tag{29}$$

Once the flow properties corresponding to the interfacial ghost points in the solid phase are determined, these properties are then extended to populate the interior ghost points using a suitable multidimensional extrapolation procedure. Linear and other higher-order extrapolation procedures have been discussed by Aslam in [36]. However, the numerical experiments carried out in this work indicate that the order of the extrapolation procedure has little or no significant influence over the accuracy of the solution, and hence a simple constant extrapolation procedure was found to be sufficient for the problems attempted in this paper. With the reflected ghost field, the problem is reduced effectively to a single fluid (real fluid–ghost fluid) problem, which can be solved with known (fluid–fluid) interface treatment techniques.

3. *Correction for Over/Underheating Problems*

The reflection of strong shocks from moving or stationary walls can cause numerical artifacts, such as excessive over/underheating errors [37,38]. Donat and Marquina [39] developed a flux splitting scheme that had a built-in mechanism to dissipate the excessive heat generated at the wall. Fedkiw et al. [17] proposed an isobaric fix, which was a simple extrapolation of entropy from the real fluid side onto the corresponding ghost field, to suppress the over/underheating errors.

Several interface treatment techniques developed in the past [12,15–17] were prone to these heating errors, and RBC is no exception. As mentioned previously, with RBC, the solid–fluid interface is converted to a relatively simple (real) fluid–(ghost) fluid interface problem. The interaction between the real and the corresponding ghost fields resulted in strong shock or rarefaction waves or both, which travel into the real and ghost fields. The problem becomes accentuated for a strong shock impacting the interface due to the following three reasons.

1) The RBC approach employs a bilinear interpolation procedure to obtain the flow properties at the reflected point. Using any form of interpolation technique across strong discontinuities will result in interpolation errors. These errors, although generated locally near the interface, accrue over time and afflict the solution away from the interface. It is important to point out that the errors arising from the interpolation procedure can be significantly reduced by employing a lower-/higher-order MUSCL [40]/ENO-based [41] adaptive polynomial interpolation procedure. However, such schemes require wider stencils and gradient evaluation, which becomes complicated to compute and computationally expensive, particularly when the mesh becomes highly unstructured [i.e., when additional mesh enrichment techniques like adaptive mesh refinement (AMR) [42] or tree-based local mesh refinement [43] is employed].

2) The errors arising from shock-interface interaction phenomena contribute significantly to the over-/underheating errors.

3) The characteristic analysis of the governing hyperbolic equations (1) dictates that the variables corresponding to incoming characteristic waves must be supplied with flow conditions independent of the flowfield, and the variables corresponding to the outgoing characteristic waves must be evolved in time (based on the flow properties within the computational domain [15,16]). Because the RBC approach does not conform to the restriction imposed by the characteristic waves at the interface (except for supersonic inflow condition), it results in over/underspecification of the boundary conditions, which in turn causes spurious waves to enter the computational domain.

The isobaric fix [17] was one approach to annihilate the growth of over/underheating shock interaction errors. However, it was established in Liu et al. [14] that the isobaric fix does not perform adequately for strong shocks interacting with a fluid–fluid interface. Furthermore, it was shown that the most effective way of dynamically correcting such errors is to accurately resolve the wave interactions occurring at the interface [14]. To this effect, a Riemann problem was solved locally, normal to the interface [14,21], to decompose the singularities in wave patterns and material properties. The same methodology could be extended to correct the heating errors arising due to shocks interacting with the solid–fluid interfaces. Moreover, the Riemann solver constructed locally at the interface essentially solves the mass, momentum, and energy balances at the interface (Rankine–Hugoniot jump conditions) and implicitly adheres to the nature of the waves arriving/leaving the interface. Hence, the Riemann states obtained from the Riemann solver are devoid of shock interaction errors and could be used to mitigate the growth of such errors.

4. *Constructing the Riemann Problem at the Interface*

In this section, a brief discussion on the procedure to construct the Riemann problem at the interface is outlined. From each interfacial ghost point, such as point *P* in Fig. 3, a probe is inserted in the normal direction to find the nearest point of intersection on the interface (point IP in Fig. 3). The coordinates of the point IP on the interface are given by

$$\mathbf{X}_{\mathrm{IP}} = \mathbf{X}_P + \mid \phi_P \mid \mathbf{N}_P \tag{30}$$

The left and the right states required for assembling the Riemann problem are obtained by advancing a distance of $1.5\Delta x$ on either side from IP. The coordinates of the points corresponding to the left and right states are given by

$$\mathbf{X}_L = \mathbf{X}_{\mathrm{IP}} - 1.5\Delta x \mathbf{N}_P \tag{31}$$

$$\mathbf{X}_R = \mathbf{X}_{\mathrm{IP}} + 1.5\Delta x \mathbf{N}_P \tag{32}$$

Once the coordinates of the left and the right states are determined, the flow properties at these points can be obtained using a simple bilinear interpolation procedure, as shown in Fig. 3. The Riemann problem is solved and the Riemann (intermediate $*$) states obtained from the Riemann solver are then used to correct the flow properties at the ghost point *P*:

$$\rho_P = \rho_L^* \tag{33}$$

$$v_{n_P} = v_{n_L}^* \tag{34}$$

$$P_P = P_L^* \tag{35}$$

Once the flow properties at the interfacial ghost points are determined, the flowfield corresponding to the interior ghost points (located inside the solid) can be obtained by extending the flow properties, using a suitable multidimensional extrapolation procedure [36].

D. Boundary Condition Type 2: Simple Boundary Condition

The simple boundary condition (SBC) approach is the most direct way to enforce the boundary conditions (8–12) accurately at the

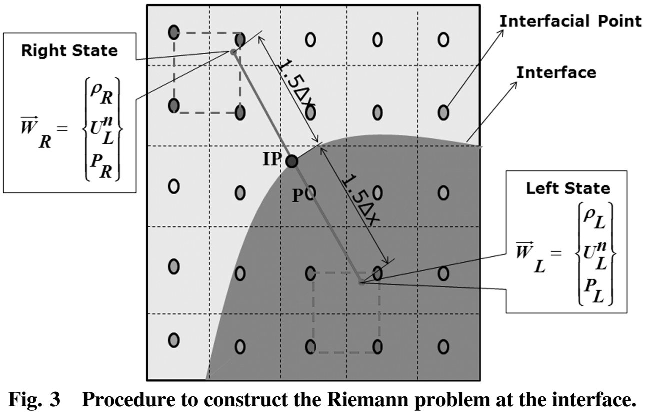


Fig. 3 Procedure to construct the Riemann problem at the interface.

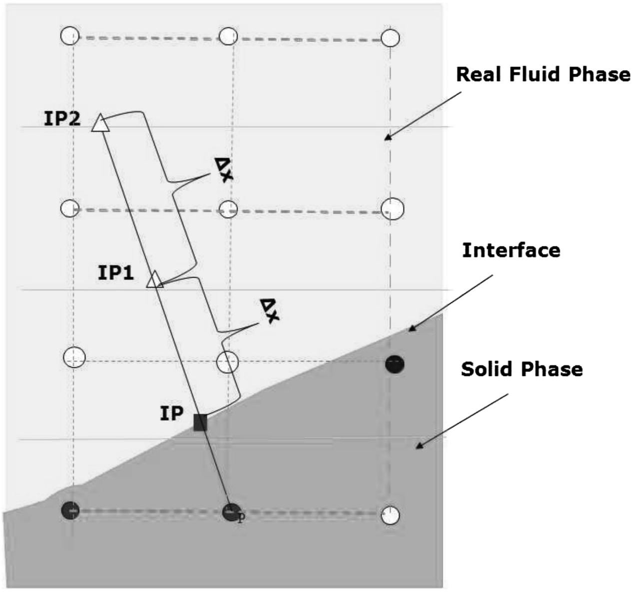


Fig. 4 SBC procedure: evaluating the normal derivatives at the interface.

interface. This is a simple extension of the approach discussed in Marella et al. [8], for which a methodology was developed to treat the presence of embedded objects in an incompressible flow. A version of this approach was also discussed by Arienti et al. [31]. As shown in Fig. 4, a probe is inserted through each interfacial ghost point P in the solid phase. Coordinates of IP, the point of intersection of the probe with the interface, are determined from Eq. (30). Along the same lines, the coordinates of points IP1 and IP2 are determined as follows:

$$\mathbf{X}_{\mathrm{IP1}} = \mathbf{X}_{\mathrm{IP}} + \Delta x \mathbf{N}_P \tag{36}$$

$$\mathbf{X}_{\mathrm{IP2}} = \mathbf{X}_{\mathrm{IP}} + 2\Delta x \mathbf{N}_P \tag{37}$$

Flow properties at points IP1 and IP2 are obtained using bilinear interpolation from the surrounding computational points. The normal derivatives required in Eqs. (8–12) can now be easily computed as

$$\frac{\partial \psi}{\partial n} = \frac{\psi_{\mathrm{IP}} - 4\psi_{\mathrm{IP1}} + \psi_{\mathrm{IP2}}}{3\Delta x} \tag{38}$$

where ψ corresponds to temperature T, tangential velocities v_{t_1} and v_{t_2}, or pressure P. The density at the interface can be obtained from the equation of state. The velocity of the center of mass is specified as the normal velocity at the point IP. After determining the flow properties at the interface, the values are extended to define the ghost field in the solid phase.

It is important to note that this method suffers from severe over/underheating errors. In particular, these errors are amplified during the transient shock-interface interaction process. Evaluation of derivatives with the use of probes, especially during the initial stages of the shock interaction with the interface, result in severe interpolation errors. If these startup errors are left uncorrected, they are fed back into the real fluid through the ghost field. It is found that these errors cannot be corrected by augmenting this approach with a local Riemann solver. This is because, populating the ghost points with SBC results in an abrupt change in flow properties. Employing a Riemann solver would yield intermediate states that, together with the real fluid, would not satisfy the boundary conditions (8–12) at the interface. Therefore, for strong shock interactions with the embedded solid boundaries, the SBC approach is ill-advised.

E. Boundary Condition Type 3: Simple Boundary Condition Using Coordinate Transformation

Another method for treating the presence of embedded solid bodies is to use the coordinate transformation (CT)/rotation technique [15,16]. In this method, the discretization of Eqs. (8–12) is carried out in global Cartesian coordinates for each interfacial ghost point in the solid phase and then transformed to the local (body-fitted) curvilinear coordinates by means of a suitable transformation matrix. Thus, the normal and tangential derivatives can be computed as follows:

$$\begin{pmatrix} \frac{\partial \psi}{\partial n} \\ \frac{\partial \psi}{\partial t_1} \\ \frac{\partial \psi}{\partial t_2} \end{pmatrix} = \mathbb{J} \begin{pmatrix} \frac{\partial \psi}{\partial x} \\ \frac{\partial \psi}{\partial y} \\ \frac{\partial \psi}{\partial z} \end{pmatrix} \tag{39}$$

$$\mathbb{J} = \begin{pmatrix} n_x & n_y & n_z \\ t_{1x} & t_{1y} & t_{1z} \\ t_{2x} & t_{2y} & t_{2z} \end{pmatrix} \tag{40}$$

where $\mathbb{J}$ is the Jacobian matrix, and the unit vectors $\hat{n}$, $\hat{t}_1$, and $\hat{t}_2$ are the normal and tangential vectors computed from the level set field [15,16,44]. This method minimizes interpolation across the

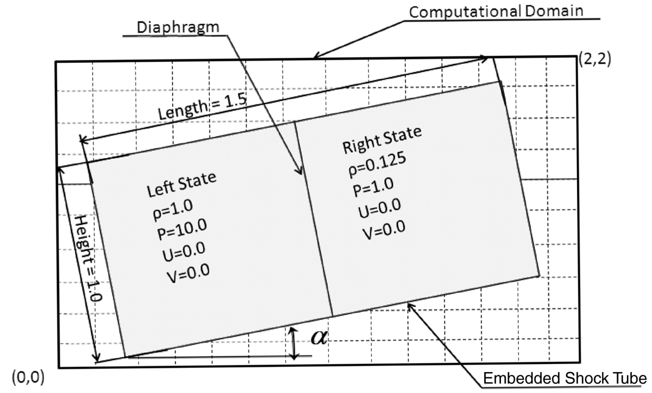
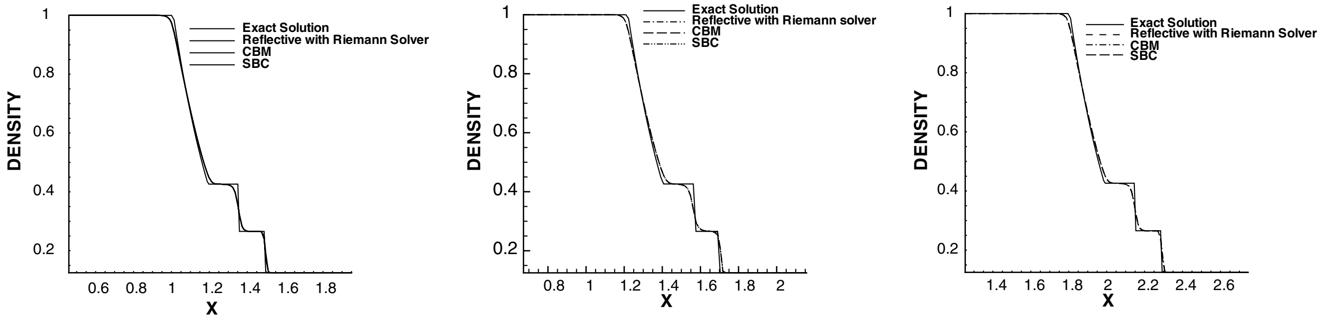


Fig. 5 Configuration of a rotating shock tube embedded in the computational domain.



a) $\alpha = 30^{\circ}$ **b)** $\alpha = 45^{\circ}$ **c)** $\alpha = 60^{\circ}$

Fig. 6 Plots of density along the centerline of the shock tube placed at different orientations.

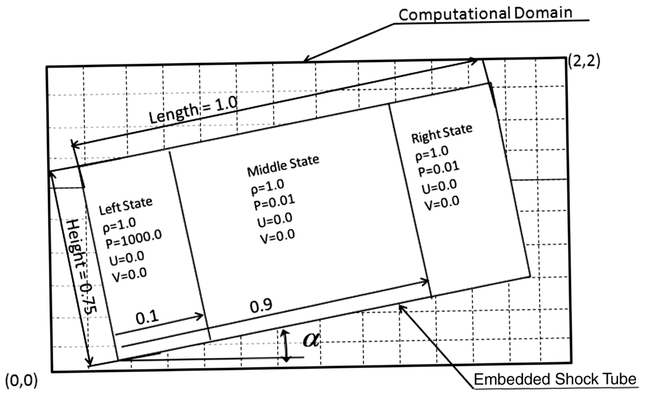
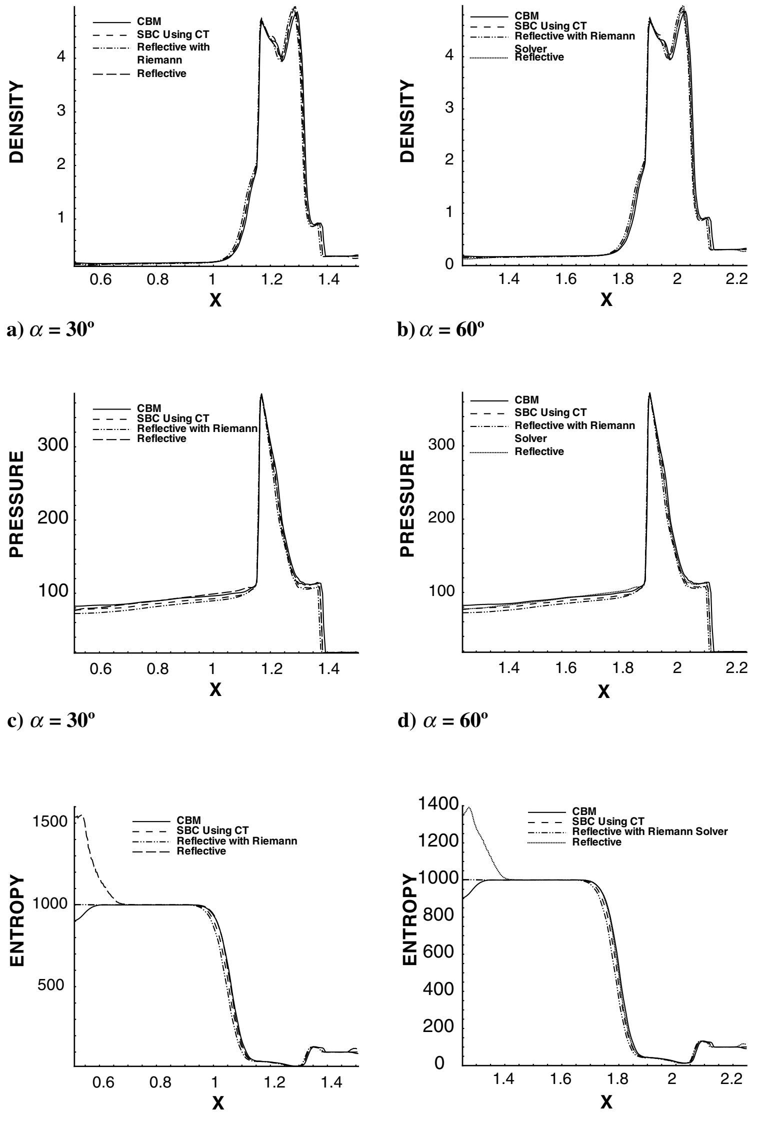


Fig. 7 Configuration of a rotating shock tube embedded in the computational domain.

discontinuity and hence localizes the over/underheating errors to few points close to the interface. Although this method appears to be very simple and robust, there is a slight smearing of the interface due to the following reason: Eqs. (8–12) are valid at the interface, and hence the normal and tangential derivatives must be evaluated at the interface itself. Evaluating the derivatives at the interfacial ghost points results in diffusing the interface over few grid points, and hence using Eq. (39) would not ensure the application of boundary conditions strictly at the interface.

F. Boundary Condition Type 4: Characteristic-Based Matching [15,16]

Another promising approach to treat the presence of embedded objects in compressible flows is to use the characteristic wave analysis at the interface [15,16,45]. As the governing equations are a set of hyperbolic conservation laws, Eq. (1) can be recast in characteristic form using a suitable transformation matrix. The sign of the eigenvalues determines whether the waves are incoming or outgoing and hence the direction of propagation of information. The incoming waves carry information into the flowfield and hence correspond to the physical boundary conditions that need to be supplied independent of the flowfield. The outgoing waves carry information from within the flowfield and hence correspond to the numerical boundary conditions (obtained via extrapolation) that need to computed from within the flowfield. The characteristic analysis was carried out by Thomson [45] for the domain boundaries. Nourgaliev et al. [15,16] extended this approach to populate the ghost points in the GFM framework. Thus, with the characteristic wave analysis, the ghost points are always supplied with physically consistent flow values. In addition, the characteristic wave analysis prevents over/underspecification of boundary conditions that, in turn, inhibits spurious disturbances from entering the flowfield [45].



a) $\alpha = 30^\circ$

b) $\alpha = 60^\circ$

c) $\alpha = 30^\circ$

d) $\alpha = 60^\circ$

e) $\alpha = 30^\circ$

f) $\alpha = 60^\circ$

Fig. 8 Plots of density, pressure, and entropy along the centerline of the shock tube for the Woodward Colella (bang–bang) problem.

For additional details pertaining to the mathematical background and implementation of the CBM approach, the reader may refer to Nourgaliev et al. [15,16].

G. Other Methods

Other methods for applying interface conditions on embedded solid boundaries have been discussed at length by Arienti et al. [31]. The methods include the mirroring technique, the reflection method, and the direct injection method. In the mirroring technique [33], the flow properties on the real fluid side are mirrored onto the ghost fluid by multiplying with a suitable rotation/reflection matrix. The reflection method discussed in Arienti et al. [31] simply extends the flow properties at the real fluid interfacial nodes but with an inverted sign of the normal velocity to satisfy the no-penetration boundary condition. The injection method formulated by Arienti et al. [31] is a variation of SBC. In this approach, flow variables at the real fluid interfacial points are extended to the ghost points, but now the normal velocity is set to zero, preserving the tangential velocities to depict the slip condition at the interface. These methods are mentioned here only for the sake of completeness and will not be considered in the numerical examples to be discussed in the next section.

IV. Numerical Examples

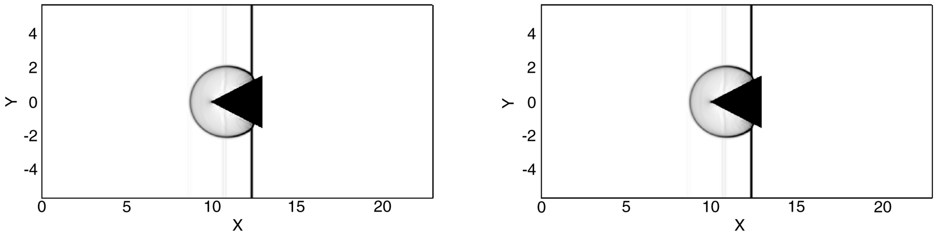
In this section, several numerical examples will be shown to evaluate the performance of the interface treatment methods outlined in the previous section. In the numerical examples shown here, the Euler equations (1) were solved using a third-order total-variation-diminishing-based RK scheme for time integration and a third-order convex ENO scheme [46] for spatial discretization. Details pertaining to the implementation of the convex ENO scheme can be found in Liu et al. [46]. Unless otherwise specified, a Courant–Friedrichs–Lewy number of 0.6 was used in the simulations performed. Numerical Schlieren images shown in this section were generated using the method outlined by Quirk and Karni [3].

A. One-Dimensional Examples

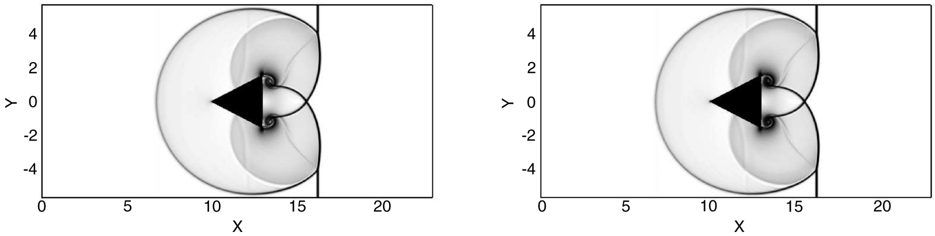
The one-dimensional examples shown here demonstrate the performance of various (interface treatment) methods in handling regular shock reflections. It is found that the choice of method impacts significantly on the accuracy and stability of the computations.

1. Rotated Shock Tube

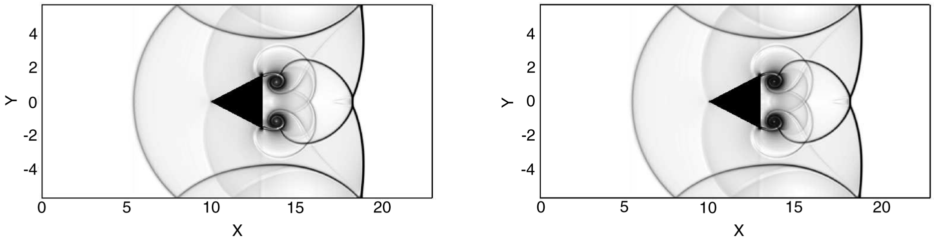
A variant of the one-dimensional shock tube problem demonstrated by Nourgaliev et al. [16] is considered first. As shown in



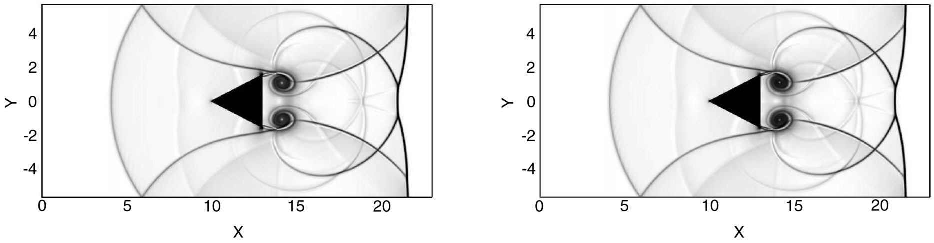
a) RBC-RS; t=1.9 **b) CBM; t = 1.9**



c) RBC-RS; t = 4.9 **d) CBM; t = 4.9**



e) RBC-RS; t = 6.9 **f) CBM; t = 6.9**



g) RBC-RS; t = 8.9 **h) CBM; t = 8.9**

Fig. 9 Snapshots of the numerical Schlieren image of a Mach 1.3 shock diffracting a wedge.

Fig. 5, the walls of the shock tube are embedded in the computational domain by defining a suitable level set field. The shock tube is oriented at an angle to the regular Cartesian mesh so that a two-dimensional problem is solved, but the problem is one-dimensional along the centerline of the shock tube. The discontinuity at $t = 0$ is at the center of the shock tube. The shock tube is placed at different orientations relative to the grid lines. Rotation angles corresponding to $\alpha_{\mathrm{rot}} = 30$ deg, $\alpha_{\mathrm{rot}} = 45$ deg, and $\alpha_{\mathrm{rot}} = 60$ deg (with respect to the horizontal) have been considered. The size of the computational domain is 2×2 units. The length and the height of the shock tube were chosen to be 1.5 and 1.0 units, respectively. The simulations were carried out to $T = 0.164$. The results obtained from the current simulations are shown in Fig. 6. RBC augmented with local Riemann solver (henceforth RBC–RS), SBC, and CBM approaches were used to simulate this problem. Figure 6 shows good agreement between these methods, and with the exact solution, in capturing the shock and the contact discontinuity. The simulations were stopped to prevent the shock from reflecting off the wall, so that the resulting numerical solution could be compared with the exact solution.

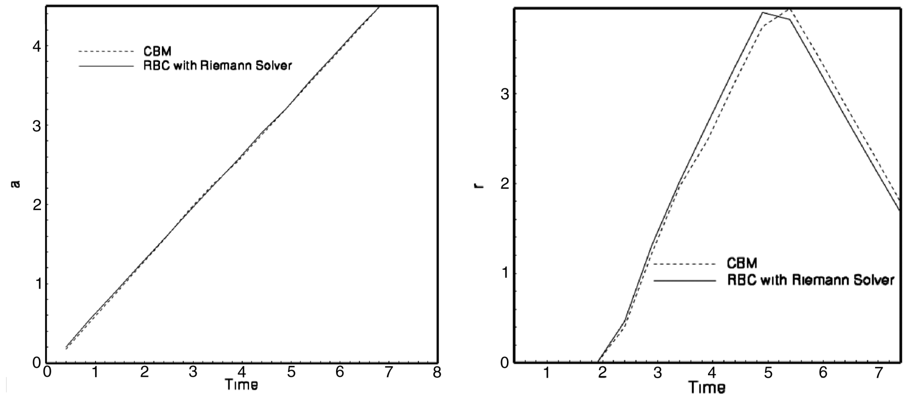
2. *Woodward Colella (Bang–Bang) Problem*

Unlike the previous case, this example involves strong shock reflections and shock collisions. The initial conditions along with the geometrical setup are given in Fig. 7. As in the previous example, the shock tube is placed at different orientations ($\alpha_{\mathrm{rot}} = 30$ deg and $\alpha_{\mathrm{rot}} = 60$ deg) so that the problem is not strictly one-dimensional and the impact of the interface treatment and mesh orientation can be examined. The simulations were run to time $T = 0.038$ units, and the plots of density, pressure, and entropy along the centerline of the shock tube are displayed in Fig. 8. Note that the SBC, unable to suppress the initial transient shock development errors, failed in the first few times steps of the shock reflection process.

From the entropy plots displayed in Figs. 8e and 8f, it is evident that RBC suffers from extensive overheating errors that appear to be problem dependent (over/underheating errors were not observed in the previous example). Clearly, the overheating errors associated with the RBC cannot be neglected and hence must be corrected in order to obtain better solutions in the presence of strong shocks. On the other hand, the RBC–RS shows no significant over/underheating errors. Both CBM and the SBC–CT produce underheating errors that are small. This example clearly illustrates the requirement of an additional correction procedure to suppress the errors associated with shock-interface interactions. Thus, it is clear from the one-dimensional examples presented in this section that the SBC approach suffered from severe over/underheating errors that had catastrophic consequences in the final outcome of the solution. The RBC approach incurred over/underheating errors that were problem dependent. In contrast, the RBC–RS and CBM methods resulted in lower levels of over/underheating errors and were consistent in predicting satisfactory solutions. This clearly favors the use of the CBM and RBC approaches with Riemann solver correction over other methods.

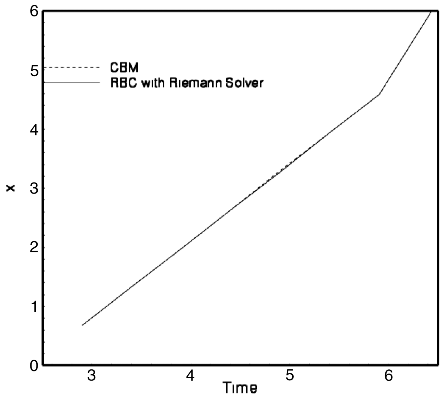
B. Two-Dimensional Examples

In the one-dimensional test cases considered in the preceding section, the shocks interacted with interfaces to produce regular reflections. Multidimensional shock diffraction patterns can be irregular (consisting of a Mach stem, slip lines emanating from a Mach stem, and eventual roll up of slip lines due to baroclinic vorticity generation, etc.). In this section, a series of multidimensional test



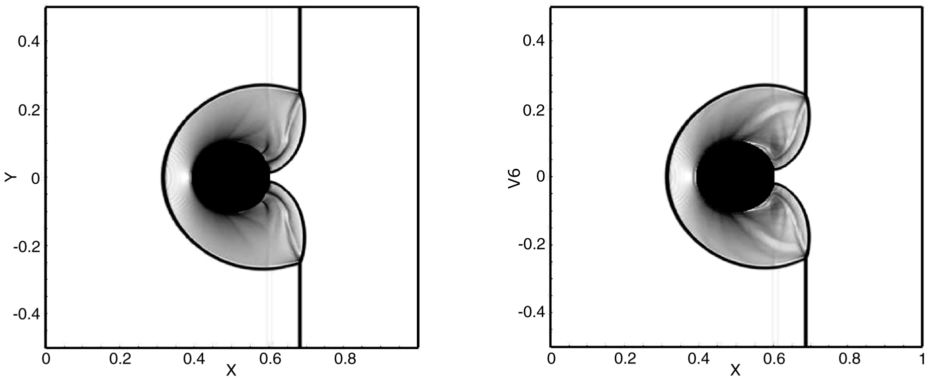
a) Horizontal shock detachment distance of the reflected bow shock from the foot of the wedge

b) Vertical distance from the tip of the wedge to the diffracting shock



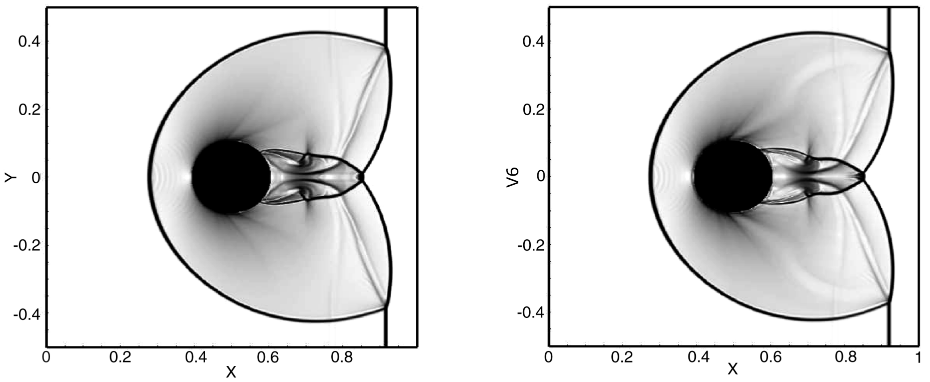
c) Horizontal distance between the tip of the wedge and the incident shock

Fig. 10 Comparison of the primary shock detachment distance over time between CBM and RBC–RS.



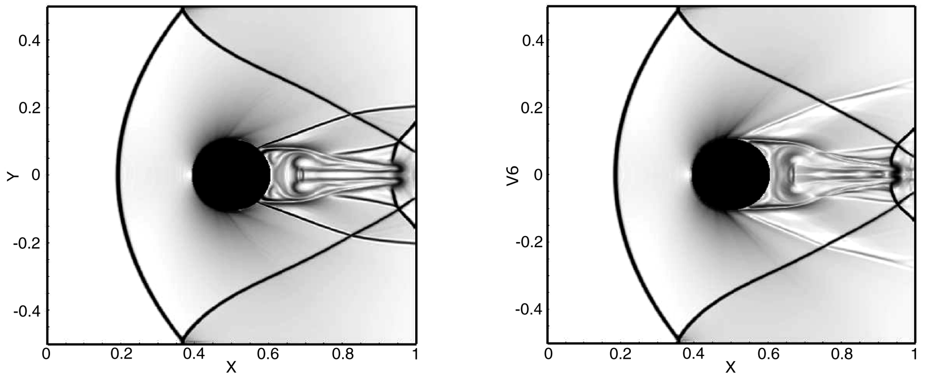
a) RBC-RS; t = 0.11

b) CBM; t = 0.11



c) RBC-RS; t = 0.2

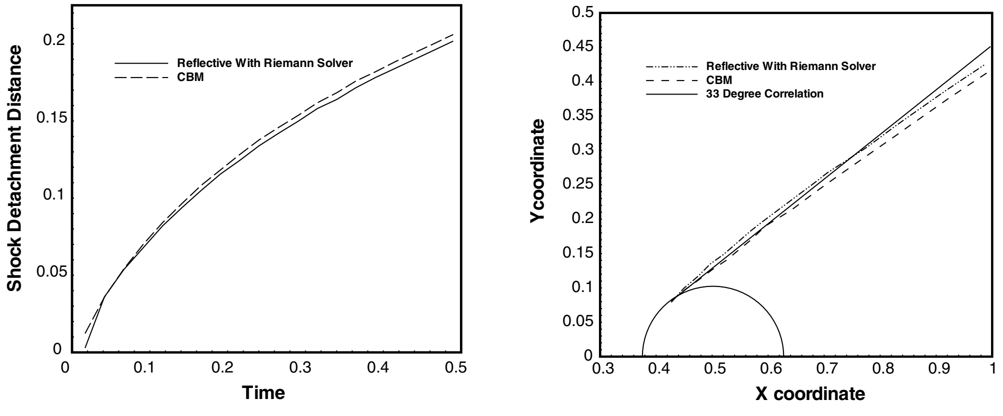
d) CBM; t = 0.2



e) RBC-RS; t = 0.5

f) CBM; t = 0.5

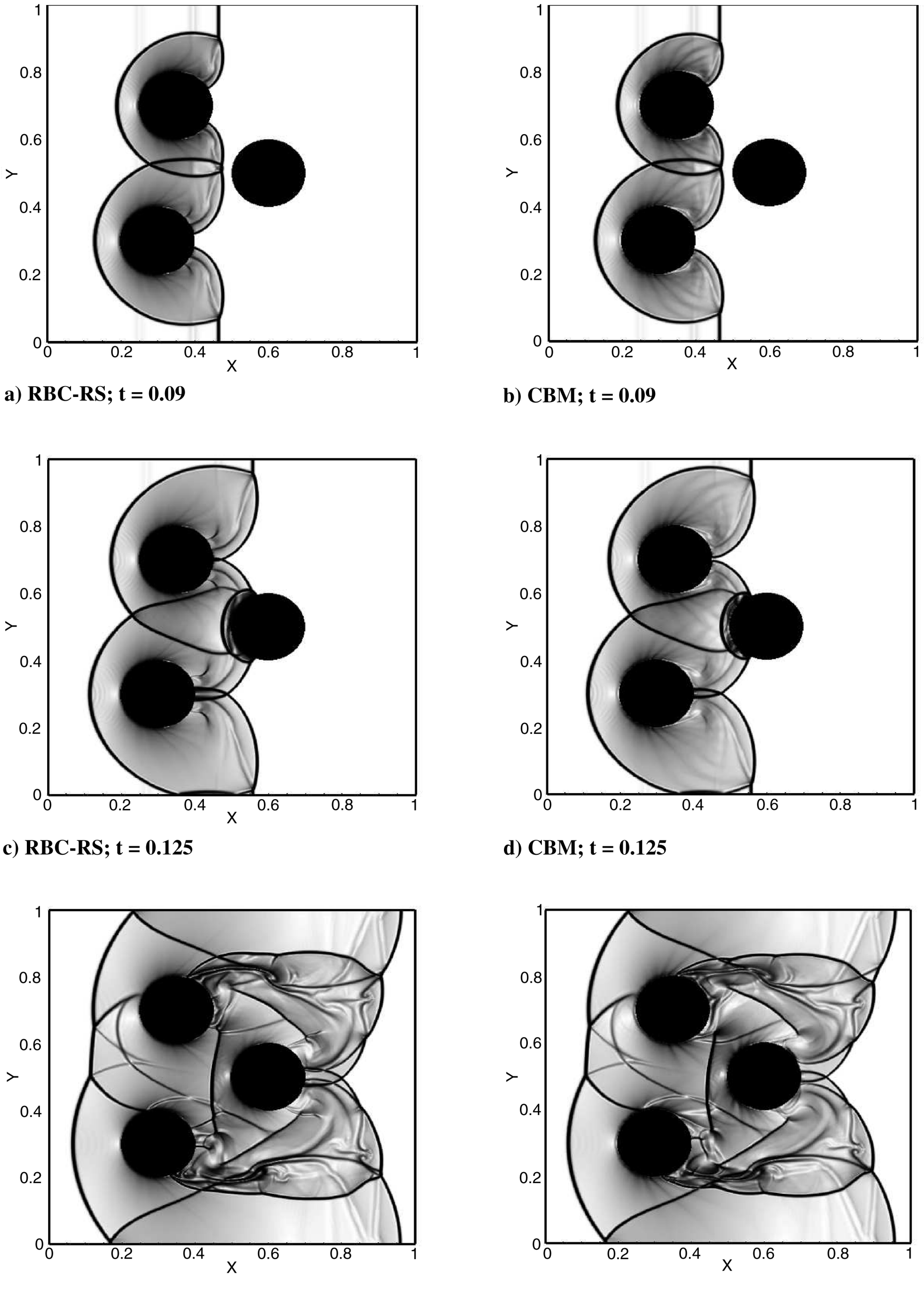
Fig. 11 Numerical Schlieren image at different instants in time.



a) Shock detachment distance as a function of time

b) locus of upper triple point

Fig. 12 Comparison between CBM and the RBC–RS: shock detachment distance vs the time (left) and locus of the upper triple point (right).



a) RBC-RS; t = 0.09 b) CBM; t = 0.09

c) RBC-RS; t = 0.125 d) CBM; t = 0.125

e) RBC-RS; t = 0.25 f) CBM; t = 0.25

Fig. 13 Snapshots of numerical Schlieren image for a Mach 3 shock interacting with multiple stationary cylinders at different instants in time.

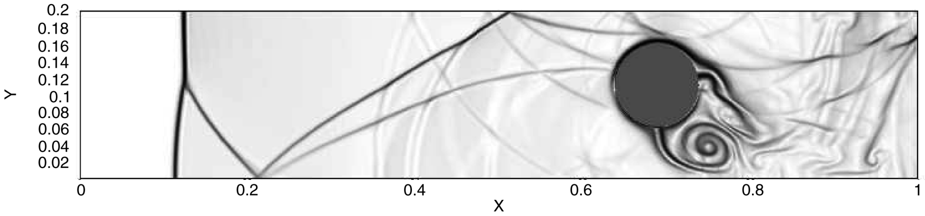
cases are presented to illustrate the performance of the GFM-based approaches in such situations.

1. Shock Diffraction Over Wedge

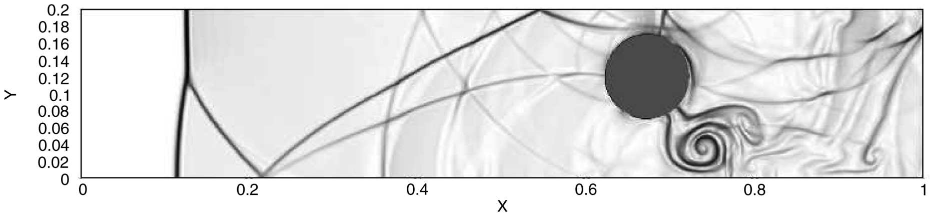
The first two-dimensional test case considered in this work is the diffraction of a Mach 1.3 shock wave past a wedge. The details of the initial configuration of the problem can be found in Sivier et al. [47]. This problem has received considerable attention in the past, and extensive numerical and experimental documentations are available for comparison [47]. Snapshots of the numerical Schlieren image, at different instants in time, obtained from using both the RBC–RS and the CBM approaches are displayed in Fig. 9. Although the contour plots generated using the two approaches look very similar, there are subtle differences that are worth noting. As mentioned in Sivier et al. [47], the planar shock (upon impacting the wedge) forms a cylindrical reflected bow shock centered at the wedge tip, with a Mach stem attached to the wedge surface (Figs. 9a and 9b). All typical features of Mach reflections are captured well by both (RBC–RS and CBM) methods. However, both methods fail to capture the slip line originating from the wedge surface, which is clearly visible in Sivier et al. But in Sivier et al., an AMR technique was employed that provides sufficient mesh enrichment to capture such fine details. Figures 9c and 9d correspond to the instant when the incident shock wave has passed over the wedge surface. As observed by Sivier et al., the roll up of slip layers to form vortices that engulfed the vertex of the entropy fan are clearly visible. At this instant, the opposite shock systems from the symmetrical side of the wedge have crossed

Table 1 CPU time taken for one complete (third-order) RK time iteration

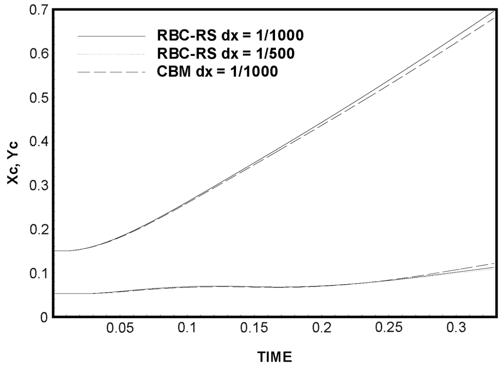
Interface treatment methods	CPU time ratio with respect to RBC–RS
RBC with Riemann solver	1.0
CBM without subiterations	1.0483
CBM with five subiterations	1.2276



a) RBC-RS



b) CBM



c) Locus of center of mass as a function of time

Fig. 14 Numerical Schlieren image for a) RBC–RS and b) CBM, and c) locus of center of mass for a cylinder subject to Mach 3 shock.

and have begun to interact with those vortices. The expansion fan, which is produced as a result of the diffraction of the Mach stem around the corner [47] and which subsequently moves down the wedge surface, is also visible in the figures. Figures 9e and 9h show that the vortices had shifted to the right, downstream of the wedge. The slip layer emanating from these vortices are now visible along with a new triple point formed at the vortex center. All of these features, typical of the shock diffraction phenomena past a wedge, are consistent with other numerical results [47] and with experimental observations [48].

Plots of the primary shock distance from the wedge surface are shown in Fig. 10. Shown in the figure, in accordance with Sivier et al. [47], are the horizontal shock detachment distance a of the reflected bow shock from the foot of the wedge (Fig. 10a), the vertical distance r from the tip of the wedge to the diffracting shock (Fig. 10b), and the horizontal distance x between the tip of the wedge and the incident shock (Fig. 10c). The plots show excellent agreement between the two methods in determining the position of the primary shocks and are consistent with the predictions made by Sivier et al. [47].

2. *Shock Diffraction Over a Circular Cylinder*

A Mach 2.81 planar shock wave negotiating a cylindrical obstacle in a shock tube is considered. As pointed out by Bazhenova et al. [49], the initial interaction of the planar shock with the cylinder is a regular reflection, with the shock reflecting normally along the axis of symmetry. The reflected shock begins traversing the cylinder before transitioning to a Mach reflection. The Mach reflection consists of the incident shock, the reflected shock, the Mach stem, and the slipstream. Ripley et al. [50] pointed out that this curved Mach stem traveled down the cylinder and interacts with the shock system from the symmetrical part of the cylinder to form a wake. Several papers documenting this phenomenon are available for comparison [5,6,49–52].

Because of the symmetry of the problem, only one-half of the problem was modeled using a symmetry condition at the bottom wall. RBC for the top wall and the Neumann condition for the outlet were imposed. At the inlet, a Dirichlet condition corresponding to a Mach 2.81 shock was enforced. The semicircular obstacle with a radius of 0.105 units was positioned at $x = 0.5$ and $y = 0.0$. The simulations were performed with 500×250 mesh points in a 1×0.5 computational domain. The initial discontinuity was located at $x = 0.385$ units so that the sharp discontinuity relaxes to a numerical shock profile before impacting the obstacle.

Results from the current simulations are shown in Fig. 11. Figures 11a and 11b show the curved Mach stem, with the reflected and incident shock, just before interacting with the opposite shock system. Both CBM and RBC–RS agree well with each other and also with the experimental results shown in Bazhenova et al. [49]. Figures 11c and 11d show the triple point and wake formed after the shock interaction. The contact discontinuities, upper and lower triple points, and the Mach systems are captured well. Although the numerical Schlieren image, displayed in Fig. 11f, predicts a wider wake for the CBM approach, there is no other discernible difference between the results. Comparing Figs. 11e and 11f, it is clear that the large scale features of the diffraction pattern, observed particularly in the wake region, appear more prominent for the RBC–RS method than the CBM method. For a quantitative comparison between the two methods, the bow shock detachment distance over time and the locus of the upper triple point were tracked [49]. Figure 12 shows the plots obtained from the current calculations. Figure 12a predicts a parabolic behavior for the shock detachment distance, which is consistent with the numerical results obtained by Ripley et al. [50]. The angle made by the locus of the upper triple point with the horizontal (predicted by the RBC–RS) is 31.3 deg, and that, obtained using CBM, is 30.5 deg. The correlation obtained from experimental observations [50,53] predicts an angle of 33 deg, which is in close agreement with the current numerical calculation.

3. *Shock Diffraction Over Multiple Cylinders*

This test case is considered to demonstrate the ability of the interface treatment methods to handle multiple embedded objects. This example is similar to the previous one except that, in this case, there are three asymmetrically placed cylindrical obstacles in the path of a planar shock wave. The cylinders of radius 0.1 units were placed at (0.3, 0.3), (0.35, 0.7), and (0.5, 0.1) in a 1×1 computational domain with 500 mesh points in each direction. A Mach 3 planar shock was initially positioned at $x = 0.19$. The results obtained from the current simulations are shown in Fig. 13. Table 1 shows a comparison of CPU time taken by the two methods for one complete time step. Because the simulations were carried out on a fine mesh, and due to the presence of multiple embedded objects, the CPU time for this example is a good representation of typical computation time for the problems attempted in the present work. Note that the transformation matrix computed in obtaining the compatibility relation is a function of time, and when subject to strong transient shocks, the inherent linearization process assumed in obtaining the characteristic equations does not hold true [15]. As pointed out by Nourgaliev et al. [15], these errors can be minimized by employing subiterations (i.e., by advancing in a series of small time steps for each RK time iteration). Although the CPU times taken by CBM without subiterations and RBC–RS are comparable, the CBM method with subiterations takes a larger amount of CPU time to converge. Therefore, the CBM and RBC–RS methods are comparable in terms of computational time. The choice of RBC–RS is, however, attractive due to the relative simplicity of implementation of the approach, because one can operate in the primitive variable Cartesian formulation. Despite the simplicity of the formulation and implementation of the RBC–RS approach, the results produced by these two methods are in good agreement.

4. *Dynamics of a Cylinder Subject to Mach 3 Shock*

The first two-dimensional moving boundary problem demonstrated in this work is the dynamics of a cylindrical object subject to a Mach 3 shock. This problem was considered by Falcovitz et al. [54] and later by several others [31,55]. The problem consists of a Mach 3 shock impacting a cylindrical object that is initially at rest and located at the bottom of the domain. The subsequent motion of the cylinder results in the cylinder lifting off the lower surface. The motion of the cylinder is computed by integrating the fluid forces acting on the cylinder at each instant. The numerical Schlieren image from the current calculations are displayed in Figs. 14a and 14b. The locus of the center of mass of the cylinder is shown in Fig. 14c for comparison. Thus, it is clear, from Fig. 14c, that the numerical predictions of the position of the cylinder made by the two approaches are in excellent agreement with each other. Furthermore, the numerical Schlieren image, displayed in Figs. 14a and 14b, closely follows the trend observed by Forrer and Berger [55].

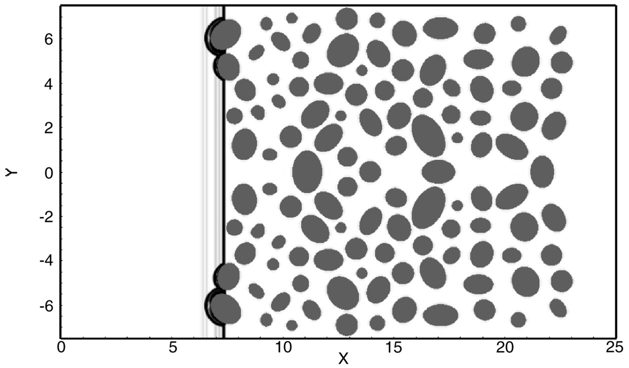
Thus, despite a few subtle differences, it can be inferred from the numerical examples shown previously that there are no significant discrepancies in the numerical results obtained using CBM and RBC–RS methods. Considering the relative ease of implementation and speed of computation, the RBC–RS method is preferred over the CBM method. In the following examples, the robustness and versatility of the RBC–RS approach are demonstrated for shocks of different strengths interacting with embedded (stationary and moving) objects of complex shapes and orientations.

5. *Mach 3 Shock Interacting with Multiple (120) Stationary Objects*

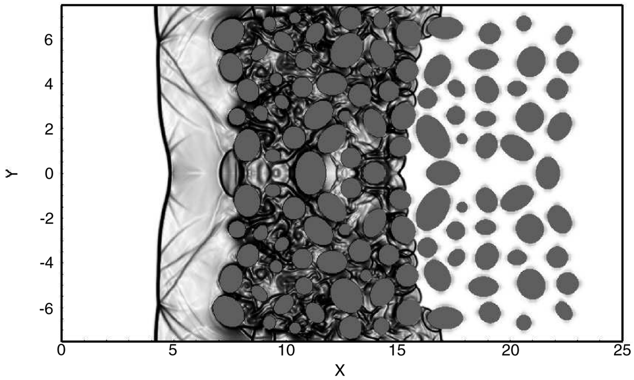
A Mach 3 shock wave interacting with randomly oriented stationary obstacles is considered. The problem consists of 120 arbitrarily oriented and randomly positioned elliptical and cylindrical obstacles in a 25×6 computational domain. Although the coordinates of the center of mass of the particles were generated randomly, care was taken in placing the particles particularly on the bottom boundary to depict a symmetrical problem of 60 particles that are distributed randomly in the x direction. These stationary particles, which are immersed in quiescent air, were subject to a Mach 3 shock. As seen from Fig. 15, the shock after negotiating through these (fixed) obstacles develops a complex diffraction pattern, in addition to the reflected and the transmitted shock waves that emerge from the diffraction pattern. This example demonstrates the potential of the present method for the application to state of the art direct numerical simulations of a shock wave traversing through a porous medium or a dusty gas [56,57].

6. *Mach 8 Spherical Shock Interacting with Multiple (24) Moving Objects*

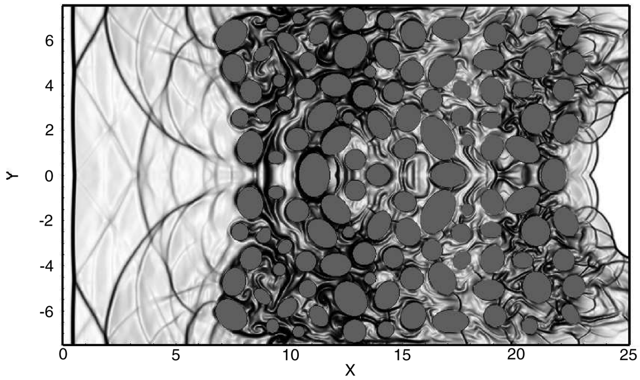
The second example considered in this section is an explosion corresponding to a Mach 8 cylindrical shock in the presence of 24 symmetrically positioned particles that are free to move under the influence of the shock wave. Again, due to the symmetry of the problem, only one quadrant of the problem was modeled. As in the example



a) Numerical Schlieren image at t = 0.36

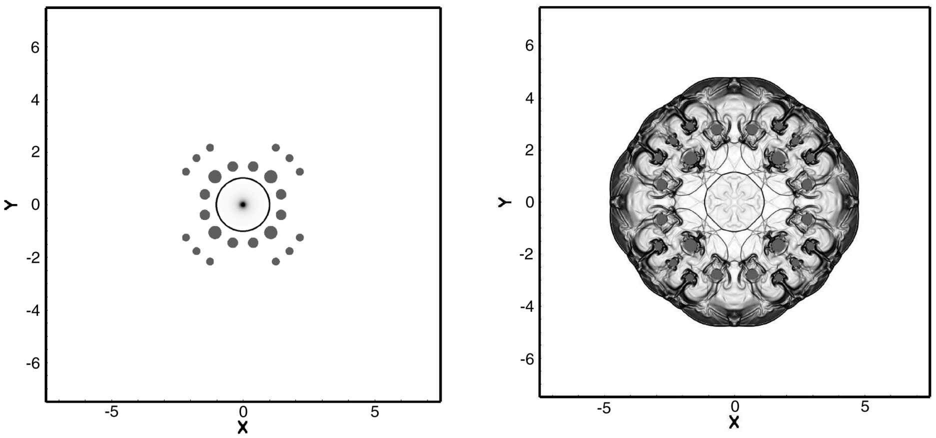


b) Numerical Schlieren image at t = 4.0



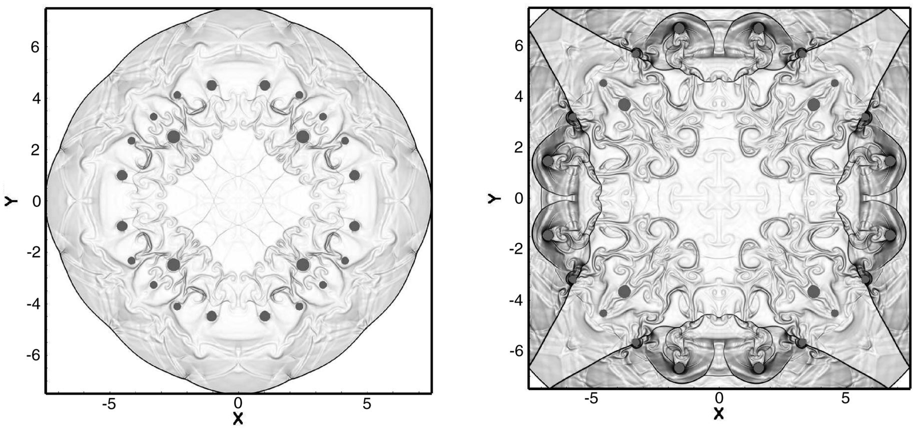
c) Numerical Schlieren image at t = 9.16

Fig. 15 Series of numerical Schlieren images for a Mach 3 shock traversing through 120 randomly oriented stationary particles.



a) Numerical schlieren image at t = 0

b) Numerical schlieren image at t = 0.725



c) Numerical schlieren image at t = 1.62

d) Numerical schlieren image at t = 3.0

Fig. 16 Series of numerical Schlieren images for a Mach 8 spherical shock dispersing randomly oriented particles.

considered in Sec. IV.B.4, the velocity of the center of mass of the particles is obtained from a simple force balance. The external fluid force acting on the particles was computed by integrating the pressure force around each particle. The density and size of the particles were (randomly) chosen so that the collisions between the moving particles and with the domain boundaries were averted. Hence, collision models, such as in Nourgaliev et al. [58], in the presence of shocks were not included in this calculation. Figure 16 shows a series of numerical Schlieren images for this simulation. The initial cylindrical shock, upon interacting with the particles, disperses the particles from their initial positions. The cylindrical shock front, after reflecting off the walls of the domain, interacts again but now in the opposite direction to the motion of the particles, thereby slowing the particles.

V. Conclusions

The issue of treatment of a solid–fluid interface interacting with strong shock waves was studied. Several approaches for numerical treatment of such interactions were considered and evaluated for a range of test problems. A Riemann-solver-based GFM approach formulated to resolve the interface dynamics for embedded fluid–fluid interfaces [23] was extended in conjunction with the RBC to treat the presence of solid objects in compressible flows. It was shown in the numerical examples that this method was robust and accurate, and it significantly reduced (but did not entirely eliminate) the over/underheating errors at the material interface. Both the CBM method and the RBC–RS approaches were found to be consistent in generating satisfactory solutions for several complex configurations and shock diffraction phenomena, and they significantly reduced the shock interaction errors. However, the associated complexity involved in carrying out the characteristic decomposition and the wave analysis in the characteristic space at the interface rendered the CBM approach less attractive than the RBC–RS approach. Shocks interacting with multiple particles and complex shapes have also been computed. The method is currently being applied to study the dynamics of dense particulate compressible flows.

Acknowledgments

This work was supported by grants from the U.S. Air Force Office of Scientific Research, Program for Computational Mathematics (F49620-02-1-0410, Fariba Fahroo) and from the U.S. Air Force Research Laboratory, Eglin Air Force Base, Florida (F08630-03-1-0007, Michael E. Nixon).

References

[1] Haller, K., Ventikos, Y., and Poulikakos, D., “Computational Study of High Speed Liquid Droplet Impact,” *Journal of Applied Physics*, Vol. 92, No. 5, 2002, pp. 2821–2828. doi:10.1063/1.1495533

[2] Perigaud, G., and Saurel, R., “A Compressible Flow Model with Capillary Effects,” *Journal of Computational Physics*, Vol. 209, No. 1, Oct. 2005, pp. 139–178. doi:10.1016/j.jcp.2005.03.018

[3] Quirk, J., and Karni, S., “On the Dynamics of Shock Bubble Interaction,” *Journal of Fluid Mechanics*, Vol. 318, 1996, pp. 129–163. doi:10.1017/S0022112096007069

[4] Tran, L. B., and UdayKumar, H., “A Particle Levelset Based Sharp Interface Cartesian Grid Method for Impact, Penetration, and Void Collapse,” *Journal of Computational Physics*, Vol. 193, No. 2, Jan. 2004, pp. 469–510. doi:10.1016/j.jcp.2003.07.023

[5] Xu, S., Aslam, T., and Scott Stewart, D., “High Resolution Numerical Simulation of Ideal and Non-Ideal Compressible Reacting Flows with Embedded Internal Boundaries,” *Combustion Theory and Modeling*,

Vol. 1, No. 1, Jan. 1997, pp. 113–142.
doi:10.1080/713665233

[6] Scott Stewart, D., Yoo, S., and Wescott, B. L. "High-Order Numerical Simulation And Modelling Of The Interaction Of Energetic And Inert Materials," *Combustion Theory and Modeling*, Vol. 11, No. 2, April 1997, pp. 305–332.
doi:10.1080/13647830600876629

[7] Grove, J., and Menikoff, R., "Anomalous Reflection of a ShockWave at a Fluid Interface," *Journal of Fluid Mechanics*, Vol. 219, Oct. 1990, pp. 313–336.
doi:10.1017/S0022112090002968

[8] Marella, S., Krishnan, S., Liu, H., and UdayKumar, H., "Sharp Interface Cartesian Grid Method 1: An Easily Implemented Technique for 3D Moving Boundary Computations," *Journal of Computational Physics*, Vol. 210, No. 1, Nov. 2005, pp. 1–31.
doi:10.1016/j.jcp.2005.03.031

[9] Hong, J., Shinar, T., Kang, M., and Fedkiw, R., "On Boundary Condition Capturing for Multiphase Interfaces," *Journal of Scientific Computing*, Vol. 31, 2007, pp. 99–125.
doi:10.1007/s10915-006-9120-x

[10] Mittal, R., and G. Iaccarino, "Immersed Boundary Methods," *Annual Review of Fluid Mechanics*, Vol. 37, Jan. 2005, pp. 239–261.
doi:10.1146/annurev.fluid.37.061903.175743

[11] Sussman, M., Smith, K., Hussaini, M. Y., Ohta, M., and Zhi-Wei, R., "A Sharp Interface Method for Incompressible Two-Phase Flows," *Journal of Computational Physics*, Vol. 221, No. 2, 2007, pp. 469–505.
doi:10.1016/j.jcp.2006.06.020

[12] Fedkiw, R., Aslam, T., Merriman, B., and Osher, S., "A Non Oscillatory Eulerian Approach to Interfaces in Multimaterial Flows (the Ghost Fluid Method)," *Journal of Computational Physics*, Vol. 152, No. 2, 1999, pp. 457–492.
doi:10.1006/jcph.1999.6236

[13] Fedkiw, R., "Coupling an Eulerian Fluid Calculation to a Lagrangian Solid Calculation with the Ghost Fluid Method," *Journal of Computational Physics*, Vol. 175, No. 1, Jan. 2002, pp. 200–224.
doi:10.1006/jcph.2001.6935

[14] Liu, T., Khoo, B., and Yeo, K., "Ghost Fluid Method for Strong Shock Impacting On Material Interface," *Journal of Computational Physics*, Vol. 190, No. 2, Sept. 2003, pp. 651–681.
doi:10.1016/S0021-9991(03)00301-2

[15] Nourgaliev, R., Dinh, T., and Theofanous, T., "The Characteristic Based Matching (CBM) Method for Compressible Flow in Complex Geometries," The 41st AIAA Aerospace Sciences Meeting and Exhibit, AIAA Paper 2003-0247, Jan. 2003.

[16] Nourgaliev, R., Dinh, T. N., and Theofanous, T., "The Characteristics-Based Matching (CBM) Method for Compressible Flow With Moving Boundaries and Interfaces," *Journal of Fluids Engineering*, Vol. 126, No. 4, 2004, pp. 586–604.
doi:10.1115/1.1778713

[17] Fedkiw, R., Marquina, A., and Merriman, B., "An Isobaric Fix For the Overheating Problem in Multimaterial Compressible Flows," *Journal of Computational Physics*, Vol. 148, No. 2, Jan. 1999, pp. 545–578.
doi:10.1006/jcph.1998.6129

[18] Davis, S. F., "An Interface Tracking Method for Hyperbolic Systems of Conservation Laws," *Applied Numerical Mathematics*, Vol. 10, No. 6, 1992, pp. 447–472.
doi:10.1016/S0168-9274(06)80001-2

[19] Cocchi, J., and Saurel, R., "A Riemann Problem Based Method for the Resolution of Compressible Multimaterial Flows," *Journal of Computational Physics*, Vol. 137, No. 2, Nov. 1997, pp. 265–298.
doi:10.1006/jcph.1997.5768

[20] Aslam, T. D., "A Level Set Algorithm for Tracking Discontinuities in Hyperbolic Conservation Laws 1: Scalar Equations," *Journal of Computational Physics*, Vol. 167, No. 2, March 2001, pp. 413–438.
doi:10.1006/jcph.2000.6686

[21] Aslam, T. D., "A Level Set Algorithm for Tracking Discontinuities in Hyperbolic Conservation Laws 2: Systems of Equations," *Journal of Scientific Computing*, Vol. 19, Nos. 1–3, 2003, pp. 37–62.
doi:10.1023/A:1025387405273

[22] Hu, X. Y., and Khoo, B. C., "An Interface Interaction Method for Compressible Multi-Fluids," *Journal of Computational Physics*, Vol. 198, No. 1, July 2004, pp. 35–64.
doi:10.1016/j.jcp.2003.12.018

[23] Sambasivan, S., and UdayKumar, H. S., "Ghost Fluid Method for Strong Shock Interactions Part 1: Fluid–Fluid Interfaces," Vol. 47, No. 12, *AIAA Journal*, 2009, pp. 2907–2922.
doi:10.2514/1.43148

[24] Menikoff, R., and Plohr, B. J., "The Riemann Problem for Fluid Flow of Real Materials," *Reviews of Modern Physics*, Vol. 61, No. 1, Jan. 1989, p. 75.
doi:10.1103/RevModPhys.61.75

[25] Harlow, F., and Amsden, A., "Fluid Dynamics," Los Alamos National Lab., TR LA-4700, Los Alamos, NM, 1971.

[26] Cocchi, J., Saurel, R., and Loraud, J., "Treatment of Interface Problems with Godunov-Type Schemes," *Shock Waves*, Vol. 5, No. 6, 1996, pp. 347–357.
doi:10.1007/BF02434010

[27] Osher, S., and Sethian, J., "Fronts Propagating with Curvature Dependent Speed Algorithms Based on Hamilton–Jacobi," *Journal of Computational Physics*, Vol. 79, No. 1, Nov. 1988, pp. 12–49.
doi:10.1016/0021-9991(88)90002-2

[28] Sethian, J., "Evolution, Implementation, Application of Levelsets and Fast Marching Methods for Advancing Fronts," *Journal of Computational Physics*, Vol. 169, No. 2, 2001, pp. 503–555.
doi:10.1006/jcph.2000.6657

[29] Sethian, J., and Smereka, P., "Levelset Methods for Fluid Interfaces," *Annual Review of Fluid Mechanics*, Vol. 35, Jan. 2003, pp. 341–372.
doi:10.1146/annurev.fluid.35.101101.161105

[30] Sethian, J. A., *Level Set Methods and Fast Marching Methods: Evolving Interfaces in Computational Geometry, Fluid Mechanics, Computer Vision, and Materials Science*, 2nd ed., Cambridge Univ. Press, New York, 1999.

[31] Arienti, M., Hung, P., Morano, E., and Shepherd, J. E., "A Level Set Approach to Eulerian–Lagrangian Coupling," *Journal of Computational Physics*, Vol. 185, No. 1, 2003, pp. 213–251.
doi:10.1016/S0021-9991(02)00055-4

[32] Krivodonova, L., and Berger, M. J., "High-Order Accurate Implementation of Solid Wall Boundary Conditions in Curved Geometries," *Journal of Computational Physics*, Vol. 211, No. 2, Jan. 2006, pp. 492–512.
doi:10.1016/j.jcp.2005.05.029

[33] Forrer, H., and Jeltsch, R., "A Higher Order Boundary Treatment for Cartesian Grid Methods," *Journal of Computational Physics*, Vol. 140, No. 2, March 1998, pp. 259–277.
doi:10.1006/jcph.1998.5891

[34] Dadone, A., and Grossman, B., "Ghost Cell Method for Inviscid Three Dimensional Flows on Cartesian Grids," The 43rd AIAA Aerospace Sciences Meeting and Exhibit, AIAA Paper 2005-0874, Jan. 2005.

[35] Ye, T., Mittal, R., Udaykumar, H., and Shyy, W., "An Accurate Cartesian Grid Method for Viscous Incompressible Flows with Complex Immersed Boundaries," *Journal of Computational Physics*, Vol. 156, No. 2, Dec. 1999, pp. 209–240.
doi:10.1006/jcph.1999.6356

[36] Aslam, T., "A Partial Differential Equation Approach to Multidimensional Extrapolation," *Journal of Computational Physics*, Vol. 193, No. 1, Jan. 2004, pp. 349–355.
doi:10.1016/j.jcp.2003.08.001

[37] Noh, W., "Errors for Calculations of Strong Shocks Using an Artificial Viscosity and an Artificial Heat Flux," *Journal of Computational Physics*, Vol. 72, No. 1, Sept. 1987, pp. 78–120.
doi:10.1016/0021-9991(87)90074-X

[38] Menikoff, R., "Errors When Shock Waves Interact Due to Numerical Shock Width," *SIAM Journal on Scientific Computing*, Vol. 15, No. 5, Sept. 1994, pp. 1227–1242.
doi:10.1137/0915075

[39] Donat, R., and Marquina, A., "Capturing Shock Reflections: An Improved Flux Formula," *Journal of Computational Physics*, Vol. 125, No. 1, April 1996, pp. 42–58.
doi:10.1006/jcph.1996.0078

[40] van Leer, B., "Towards the Ultimate Conservative Difference Scheme, 5: A Second-Order Sequel to Godunov's Method," *Journal of Computational Physics*, Vol. 32, Jul. 1979, pp. 101–136.
doi:10.1016/0021-9991(79)90145-1

[41] Shu, C., and Osher, S., "Efficient Implementation of Essentially Non-Oscillatory Shock Capturing Schemes 2," *Journal of Computational Physics*, Vol. 83, No. 1, July 1989, pp. 32–78.
doi:10.1016/0021-9991(89)90222-2

[42] Berger, M. J., and Colella, P., "Local Adaptive Mesh Refinement for Shock Hydrodynamics," *Journal of Computational Physics*, Vol. 82, No. 1, May 1989, pp. 64–84.
doi:10.1016/0021-9991(89)90035-1

[43] Krishnan, S., "An Adaptively Refined Cartesian Grid Method for Moving Boundary Problems Applied to Biomedical Systems," Ph.D. Thesis, Univ. of Iowa, Iowa City, IA, 2006.

[44] Kang, M., Fedkiw, R., and Liu, X., "Boundary Condition Capturing Method for Multiphase Incompressible Flow," *Journal of Scientific Computing*, Vol. 15, No. 3, 2000, pp. 323–360.
doi:10.1023/A:1011178417620

[45] Thomson, K., “Time Dependent Boundary Conditions for Hyperbolic Systems 2,” *Journal of Computational Physics*, Vol. 89, No. 2, Aug. 1990, pp. 439–461.
doi:10.1016/0021-9991(90)90152-Q

[46] Liu, X., and Osher, S., “Convex ENO High Order Multidimensional Schemes Without Field by Field Decomposition or Staggered Grids,” *Journal of Computational Physics*, Vol. 142, No. 2, May 1998, pp. 304–330.
doi:10.1006/jcph.1998.5937

[47] Sivier, S., Loth, E., Baum, J., and Lohner, R., “Vorticity Produced by ShockWave Diffraction,” *Shock Waves*, Vol. 2, No. 1, March 1992, pp. 31–41.
doi:10.1007/BF01414419

[48] Schardin, H., *Stossrohre [Shock Tubes]*, Springer–Verlag, New York, 1966.

[49] Bazhenova, T., Gvozdeva, L., and Nettleton, M., “Unsteady Interactions of Shock Waves,” *Progress in Aerospace Sciences*, Vol. 21, No. 4, 1984, pp. 249–331.
doi:10.1016/0376-0421(84)90007-1

[50] Ripley, R., Lien, F., and Yovanovich, M., “Numerical Simulation of Shock Diffraction on Unstructured Meshes,” *Computers and Fluids*, Vol. 35, No. 10, Dec. 2006, pp. 1420–1431.
doi:10.1016/j.compfluid.2005.05.001

[51] Bryson, A., and Gross, R., “Diffraction of Strong Shocks by Cones, Cylinders and Spheres,” *Journal of Fluid Mechanics*, Vol. 10, 1961, pp. 1–16.
doi:10.1017/S0022112061000019

[52] Sun, M., “Numerical and Experimental Investigation of Shock Wave Interactions with Bodies,” Ph.D. Thesis, Tohoku Univ., Sendai, Japan, 1998.

[53] Kaca, J., “An Interferometric Investigation of the Diffraction of a Planar ShockWave Over a Semicircular Cylinder,” Univ. of Toronto, Inst. for Aerospace Studies, TR 269, 1988.

[54] Falcovitz, J., Alfandary, G., and Hanoch, G., “A Two Dimensional Conservation Laws Scheme for Compressible Flows with Moving Boundaries,” *Journal of Computational Physics*, Vol. 138, No. 1, Nov. 1997, pp. 83–102.
doi:10.1006/jcph.1997.5808

[55] Forrer, H., and Berger, M., “Flow Simulations on Cartesian Grids Involving Complex Moving Geometries Flows,” Vol. 129, No. 315, International Series of Numerical Mathematics, Birkhä́user, Boston, 1998, pp. 139–178, .

[56] Sommerfield, M., “The Unsteadiness of Shock Waves Propagating Through Gas-Particle Mixtures,” *Experiments in Fluids*, Vol. 3, No. 4, July 1985, pp. 197–206.
doi:10.1007/BF00265101

[57] Saito, T., Mauramoto, M., and Takayama, K., “Numerical Investigation of Shock Waves Through Gas-Particle Mixtures,” *Shock Waves*, Vol. 13, 2003, pp. 299–322.
doi:10.1007/s00193-003-0217-y

[58] Nourgaliev, R., Dinh, T., and Theofanous, T., “The Direct Numerical Simulation of Disperse Multiphase High- Speed Flows,” The 42nd AIAA Aerospace Sciences Meeting and Exhibit, AIAA Paper 2004-1204, Jan. 2004.

K. Anderson
Associate Editor