Combust. Theory Modelling **1** (1997) 113–142. Printed in the UK PII: S1364-7830(97)80249-5

# High resolution numerical simulation of ideal and non-ideal compressible reacting flows with embedded internal boundaries

Shaojie Xu, Tariq Aslam and D Scott Stewart†
Theoretical and Applied Mechanics, University of Illinois, Urbana, IL, USA

Received 6 December 1996

**Abstract.** This paper explains the methodology used to develop a high-resolution, multi-dimensional Euler solver that is capable of handling non-ideal equation of state and stiff chemical source terms. We have developed a pointwise implementation that has computational advantages for our intended applications, as opposed to a finite volume implementation. Our solver allows for the placement of internal reflective boundaries and the standard inflow and outflow and reflective boundaries at the edge of the domain. We discuss the spatial discretization and the temporal integration schemes, upwinding and flux splitting and the combined use of the Lax–Friedrichs and Roe schemes to solve for the required fluxes. A complete description of the pointwise internal boundary method is given. An overall summary of a representative code structure is given. We provide details on the verification of our integrated set of algorithms that resulted in an application code. We demonstrate the order of convergence for test problems. Two example applications from measurement of detonation shock dynamics and deflagration to detonation transition in porous energetic materials are presented.

## 1. Introduction

The dynamics of reactive compressible flow is difficult to compute numerically because of fine scales associated with the chemical reaction zone. Accuracy requires fine resolution of these zones even if they are orders of magnitude smaller than scales imposed by the global geometric constraints or initial conditions. Further, the reaction rate chemistry for energetic materials that can support rapid combustion or detonation is usually modelled with state-sensitive reaction rates. The accuracy of underlying schemes used to simulate flows is often adversely affected by this strong rate sensitivity.

Engineering applications of great interest involve multi-dimensional and time-dependent flows, often in complex geometries and with material boundary interactions. Condensed explosives, rocket propellants and other pyrotechnic materials such as explosive powders are modelled as compressible flows with non-ideal (stiff) equations of state (EOS) with highly sensitive reaction rate sub-models as well. Thus the larger engineering requirements of modeling energetic materials mandate the use of continuum constitutive theory that is very non-ideal in the classical sense of materials. The capability to deal with non-ideal equation of state, sensitive reaction rates and interaction of flows with internal boundaries are essential requirements for engineering codes that can handle real material engineering problems of significance.

† To whom correspondence should be addressed. E-mail: dss@uiuc.edu

1364-7830/97/010113+30$19.50 © 1997 IOP Publishing Ltd 

Our own physical interests are in two related but distinct areas, both of which require a quality simulation with the capabilities described above. Deflagration-to-detonation transition (DDT) simulations in porous energetic materials is an important problem that arises in the safe handling of explosives. The impact of a bed of granular explosive powder by impactors of various sizes and shapes is a specific problem where the size and speed of the impactor can be used to identify ignition criteria. The EOS used in modeling this problem is highly non-ideal. We refer the reader to our recent efforts documented in [1] and [2]. The ideal EOS application we are also interested in arises in a natural way if one simply asks the question: Can one measure the dynamics of the motion the lead detonation shock in a detonation flow? In particular can one make a direct measurement of the normal velocity of the shock, the curvature and related intrinsic geometric quantities of the detonation shock, or an inert shock for that matter? To do so requires a high resolution simulation, since determination of the normal shock velocity, for example, involves a derivative of the shock locus. For more background the reader is referred to [3] and[4].

This paper explains the methodology that we used to develop a high-resolution, multi-dimensional Euler solver that is capable of handling non-ideal equation of state and stiff chemical reaction rate terms. In addition our solver allows for the placement of internal reflective boundaries in its current formulation, as well as the standard inflow and outflow and reflective boundaries at the edge of the domain. In constructing our solver we have drawn heavily on the excellent work of Harten, Osher, Enquist, Shu, Leveque, Quirk, Berger, Rogerson, Meiburg and Roe, and others [5, 6, 8, 9, 10, 11]. Our motivation for this work was more or less exclusively derived from the applications listed above. However we found that there was not a direct path to integration and implementation of the modern high resolution methods for these applications. In particular we found it necessary to develop and implement a pointwise method on a Cartesian grid that would allow a (non-moving) body of arbitrary shape in two dimension to be embedded in the flow field. Distinct advantages were apparent for our applications with this strategy, compared with Cartesian boundary method for finite volume scheme [6]: The time step restriction due to small cells was removed and it was easier to implement high-order boundary conditions on the (internal) body, We found that our strategy was reasonably computationally efficient. For our stated purposes, in the process of verification, we found that our scheme exhibited close to fifth-order convergence rates.

The philosophy of the paper is to (hopefully) present the methods we used in a clear way so that they can be duplicated if desired. Section 2 describes the scheme that allows for embedded boundaries. Specifically we discuss the spatial discretization and the temporal integration schemes, upwinding and flux splitting and the combined use of the Lax–Friedrichs and Roe scheme to solve for the required fluxes. A complete description of the pointwise internal boundary method is given. Finally an overall summary of a representative code structure is given. In section 3 we provide details on the verification of our integrated set of algorithms that resulted in an application code. We give results for the order of convergence and show results for a number of test problems. In section 4 we present two example applications and the related details of the formulation that were required to do the computations shown. The first example is drawn from the direct measurement of detonation shock dynamics, as computed from a direct numerical simulation of the Euler equation for a ideal equation of state and an Arrehnius rate law. The second example application is taken from the DDT problem described above, where issues of non-ideal equation of state are pre-eminent.

## 2. A high-order hyperbolic (Euler) solver for non-ideal flows with embedded boundaries

The general conservation laws we are solving can be expressed as

$$\boldsymbol{u}_{,t} + [\boldsymbol{f}(\boldsymbol{u})]_{,x} + [\boldsymbol{g}(\boldsymbol{u})]_{,y} = \boldsymbol{s}(\boldsymbol{u}) \,, \tag{1}$$

where $\boldsymbol{u} \in R^m$ is a vector whose components are the independent variables, $\boldsymbol{f}, \boldsymbol{g} \in R^m$ are flux functions in $x$ and $y$ directions respectively, and $\boldsymbol{s} \in R^m$ is the source term. For 2-D reactive Euler equations, $m \geqslant 5$.

### *2.1. Discretization and temporal integration*

Equation (1) can be rewritten as

$$\boldsymbol{u}_{,t} = \mathcal{L}(\boldsymbol{u}) \equiv \boldsymbol{s}(\boldsymbol{u}) - \boldsymbol{f}(\boldsymbol{u})_{,x} - \boldsymbol{g}(\boldsymbol{u})_{,y} \,, \tag{2}$$

where $\mathcal{L}(\boldsymbol{u})$ is a spatial differential operator on $\boldsymbol{u}$. If the spatial dependence of $\boldsymbol{u}$ on $x$ and $y$ is known, then (2) can be regarded as an ordinary differential equation (ODE) in time. If an $r$th-order numerical approximation to $\mathcal{L}(\boldsymbol{u})$ is given for all time by

$$\mathcal{L}(\boldsymbol{u}) = L(\boldsymbol{u}) + O((\Delta x)^r, (\Delta y)^r) \,, \tag{3}$$

then the truncated version of equation (2) can be solved strictly as a system of ODEs ,and by numerical means with the use of a Runge–Kutta scheme. Such a spatial truncation followed by a numerical integration is commonly referred to as the methods-of-lines approach, which allows for independent temporal and spatial approximations.

Shu and Osher [7] derived the standard Runge–Kutta scheme to solve (2) such that if the discrete approximation of $\mathcal{L}(\boldsymbol{u})$ has the property of total variation diminishing (TVD), then the time integration of the corresponding system of ODEs is also TVD. The second-order time integration scheme is written as

$$\boldsymbol{u}^{(1)} = \boldsymbol{u}^n + \Delta t \mathcal{L}(\boldsymbol{u}^n) \,, \tag{4}$$

$$\boldsymbol{u}^{n+1} = \frac{1}{2}\boldsymbol{u}^n + \frac{1}{2}\mathcal{L}(\boldsymbol{u}^{(1)}) + \frac{\Delta t}{2}\mathcal{L}(\boldsymbol{u}^{(1)}) \tag{5}$$

where the superscript $n$ denotes time step, and $\boldsymbol{u}^{(1)}$ is an intermediate variable. Their third-order scheme is given by

$$\boldsymbol{u}^{(1)} = \boldsymbol{u}^n + \Delta t \mathcal{L}(\boldsymbol{u}^n) \,, \tag{6}$$

$$\boldsymbol{u}^{(2)} = \boldsymbol{u}^{(1)} + \Delta t \left[ \frac{1}{4}\mathcal{L}(\boldsymbol{u}^{(1)}) + \frac{1}{4}\mathcal{L}(\boldsymbol{u}^n) \right] , \tag{7}$$

$$\boldsymbol{u}^{n+1} = \boldsymbol{u}^{(2)} + \Delta t \left[ \frac{1}{6}\mathcal{L}(\boldsymbol{u}^{(2)}) + \frac{1}{6}\mathcal{L}(\boldsymbol{u}^{(1)}) + \frac{2}{3}\mathcal{L}(\boldsymbol{u}^n) \right] . \tag{8}$$

Next we discuss the spatial discretization method.

### *2.2. Upwinding and flux splitting*

For the discretization of a computational domain $D$, let

$x_i = i \cdot \Delta x \,, \quad y_j = j \cdot \Delta y \,, \quad$ where $\quad i = 0, 1, ..., N \,, \quad j = 0, 1, ..., M \,.$

The numerical approximation to the spatial operator $\mathcal{L}(\boldsymbol{u})$ is taken to be

$$L(\boldsymbol{u}) = \boldsymbol{s}_{i,j} - \left[ \frac{1}{\Delta x}\left(\widetilde{\boldsymbol{f}}_{i+1/2,j} - \widetilde{\boldsymbol{f}}_{i-1/2,j}\right) + \frac{1}{\Delta y}\left(\widetilde{\boldsymbol{g}}_{i,j+1/2} - \widetilde{\boldsymbol{g}}_{i,j-1/2}\right) \right] , \tag{9}$$

where $\boldsymbol{s}_{i,j}$ is the source term and $\widetilde{\boldsymbol{f}}_{i+1/2,j}$ and $\widetilde{\boldsymbol{g}}_{i,j+1/2}$ denote interface fluxes in the $x$ and $y$ direction respectively. Notice that the flux calculations are independent in $x$ and $y$ directions. For example, $\widetilde{\boldsymbol{f}}_{i+1/2,j}$ is calculated component-wise holding index $j$ fixed, i.e. along a slice of data in $x$-direction.

It is important to point out that since we employ a pointwise method the source terms $\boldsymbol{s}_{i,j}$ are evaluated exactly on the grid points. In a finite volume method one must reconstruct the source terms through additional interpolation. This property of direct evaluation of source terms is particularly useful in the engineering applications that will be presented in sections 3 and 4, where the source terms are somewhat complicated and special treatments are required.

Calculation of the interface fluxes is a key element of the spatial integration, and varies from scheme to scheme. In this section, two interpolation schemes are introduced along with the implementation details. The Lax–Friedrichs (LF) scheme uses information solely from the flux function, while the Roe scheme linearizes the hyperbolic system and represents the fluxes in terms of independent, characteristic waves. Both schemes utilize the upwinding technique and high-order spatial interpolation. For clarity, we discuss the Lax–Friedrichs scheme and illustrate upwinding idea using a scalar equation in one dimension,

$$u_{,t} + f(u)_{,x} = 0\,. \tag{10}$$

The idea of upwind differencing originates from the property of hyperbolic PDEs that a disturbance (or a signal) travels only along the characteristics of the PDEs. For numerical approximation to derivatives, we want to use upstream information to construct difference schemes, or to formulate high-order approximation to the flux functions. In equation (10) for example, a disturbance will travel to the right if the derivative of the flux $f' > 0$. Thus on a 1-D grid, $f_i \equiv f(u(x_i, t))$ is considered an upstream value of $f_{i+1}$, and is used to approximate the interface value $\hat{f}_{i+1/2}$ in a first-order scheme. Similarly, $f_{i+1}$ is used to approximate $\hat{f}_{i+1/2}$, if $f_i' < 0$. The importance of the upwinding is to increase the stability of a scheme, especially at discontinuities.

High-order spatial integration accuracy is achieved with high-order approximation of the flux functions at cell interfaces. The distribution of interpolation points (stencil) is chosen according to the upwinding strategy. More points in upstream direction are used in a stencil to construct interpolants. For the ENO stencil, the first interpolation point is always chosen from the upstream site.

*2.2.1. The Lax–Friedrichs scheme.* The numerical flux (which is different from the exact flux which would be given by solving a Riemann problem) used in the Lax–Friedrichs (LF) scheme has the form

$$\hat{f}(u) = f^+(u) + f^-(u)\,, \tag{11}$$

where $f^+$ and $f^-$ satisfy

$$\frac{\mathrm{d}f^+}{\mathrm{d}u} \geqslant 0\,, \qquad \text{and} \qquad \frac{\mathrm{d}f^-}{\mathrm{d}u} \leqslant 0\,.$$

One choice for the plus and minus fluxes ($f^+$ and $f^-$) is

$$f^{\pm}(u) = \frac{1}{2}(f(u) \pm \alpha u)\,, \qquad \text{where} \qquad \alpha = \max|f'(u)|\,,\ u \in D\,. \tag{12}$$

According to upwinding, the flux at an interface is defined by

$$\hat{f}_{i+1/2} = (f_i^+ + f_{i+1}^-)\,. \tag{13}$$

Variation of the definition of $\alpha$ in (12), which controls the amount of viscosity or diffusion of a scheme and can result in different variations of the LF method. For the global method, the maximum of the derivative is searched in the whole domain $D$, the maximum can also be defined for a smaller set, such as a line of data, or even a pair of neighbouring points of the interface.

The global LF scheme is computationally economical as compared with the Roe scheme for larger systems, since no decomposition into characteristic variables is needed. However, it does require twice as much interpolation since $f^+$ and $f^-$ must both be computed.

*2.2.2. The Roe scheme.* The Roe flux splitting scheme for the 1-D scalar equation (10) is relatively simple to state. At an interface $x_{i+1/2}$, the wave speed is given by $\lambda_i = f'(u_i)$, if $f(u)$ is smooth, and by

$$\lambda_i = \frac{f_{i+1} - f_i}{u_{i+1} - u_i}\,, \tag{14}$$

if $f(u)$ is nonsmooth. The speed so defined agrees with that obtained from the solution of the Riemann problem at the interface, with the initial data $u_l = u_i$ and $u_r = u_{i+1}$. Once the wave structure is known, the interface flux is calculated by an interpolation routine using upwinding.

For a nonlinear system of equations in one dimension,

$$\boldsymbol{u}_{,t} + \boldsymbol{f}(\boldsymbol{u})_{,x} = 0\,,$$

wave structures are determined by the characteristics of the local Jacobian $\boldsymbol{J}$ of the flux functions. For example, the eigenvalues of $\boldsymbol{J}$ give the local wave speeds. Thus the scalar flux calculation can be applied to each characteristic field in the characteristic space, or the eigen-space. The local mapping operator from the physical space to the characteristic space is the left orthogonal eigenvector matrix of $\boldsymbol{J}$ evaluated at an averaged state, or the Roe state, which can be obtained by solving a locally linearized Riemann problem with initial data being the right and left neighboring states. The expression of the Roe state depends on the equation of state (EOS). For the ideal EOS, Roe's method [11] works out simply. For a general nonlinear EOS, Glaister [12] proposed an efficient method to solve the linearized Riemann problem. We use Glaister's method to express the Roe state for a non-ideal EOS that is suitable for porous energetic material–HMX, (see section 4.2.2).

Next we give the procedure of the Roe scheme in one dimension. The implementation in two dimension will be discussed in section 4.2. Let

$$\boldsymbol{J} = \nabla \boldsymbol{f}(\boldsymbol{u}) = \boldsymbol{L}\Lambda\boldsymbol{R}, \qquad \text{where} \qquad \boldsymbol{L}\boldsymbol{R} = \boldsymbol{I}, \tag{15}$$

be an eigenvalue-eigenvector decomposition of $\boldsymbol{J}$, evaluated at the interface with the Roe averaged states. Local linear decompostion into $m$ independent equations of the system (1) can be obtained at each interface by left multiplying (1) by matrix $\boldsymbol{L}$. Hence the characteristic flux $\boldsymbol{f}^c$ is

$$\boldsymbol{f}^c = \boldsymbol{L} \cdot \boldsymbol{f}\,. \tag{16}$$

Each component of $\boldsymbol{f}^c$ is then interpolated at the cell interface (as in the scalar equation case), and the interface flux $\boldsymbol{f}^c_{i+1/2}$ is determined by upwinding. Then the interface flux in the characteristic space is mapped back to physical space by

$$\boldsymbol{f}_{i+1/2} = \boldsymbol{R} \cdot \boldsymbol{f}^c_{i+1/2}\,, \tag{17}$$

which will be used in the spatial operator $L(\boldsymbol{u})$ in equation (9).

Numerical solutions obtained with the Roe scheme usually exhibit sharper shocks, (i.e. are less diffusive) than for the LF scheme. Interpolation of the characteristic fluxes minimizes the effect of nonlinearity of the system on numerical solutions. One shortcoming of Roe's scheme is that the weak solutions are not guaranteed to be entropy satisfying ones, i.e. the solutions may violate the entropy condition, thus not be physical [13]. A cure for this problem, called an entropy fix, is added to the Roe scheme in our solver. If the entropy violation is detected to occur, (e.g. eigenvalues of $\boldsymbol{J}$ are close to zero,) then the flux calculation of Roe's scheme is replaced by that of the LF scheme locally. For example, if the second eigenvalue, $\lambda_2$, of $\boldsymbol{J}$ is zero, then the flux in the second characteristic field, or the second component of $\boldsymbol{f}^c$, $f_i^c(2)$, is calculated by

$$f_i^c(2) = f_i^{c+}(2) + f_{i+1}^{c-}(2)\,, \tag{18}$$

where the plus and minus fluxes are defined according to (12). Numerical tests shows that the entropy fix scheme works well. It effectively stabilizes the solution near shocks and suppresses the disturbances otherwise would grow.

### *2.3. High-order spatial approximation*

In this section, we describe the interpolation used to obtain the high-order spatial approximation, which is used in both the LF and Roe schemes. Recall from (3) that we want to formulate $L(\boldsymbol{u})$ such that

$$\mathcal{L}(\boldsymbol{u}) = L(\boldsymbol{u}) + O((\Delta x)^r, (\Delta y)^r)\,.$$

Polynomial interpolants of $f_i^+$ and $f_i^-$ in the LF scheme, and of components of $\boldsymbol{f}_i^c$ in the Roe scheme are constructed locally to achieve high-order accuracy. We use weighted ENO scheme (WENO), first introduced in [14], and later improved in [15]. The WENO scheme improves on the ENO scheme by achieving high-order interpolation through a convex combination of several interpolants (see [14] and [15] for details). The advantage of the WENO scheme, as compared with the ENO scheme, is that it provides a continuous data dependency, and convergence for nonlinear systems can be proven (under certain restrictions) [15]. The results presented here use the fifth-order scheme of Jiang and Shu [15] and for completness, the fifth-order scheme is reviewed next.

For the spatial operator, (9), an approximation to the interface flux is needed, i.e. $\tilde{f}_{i+1/2}$, etc. If $\partial f/\partial u > 0$, then the first-order scheme would be $\tilde{f}_{i+1/2} = f_i$. For the WENO scheme, $\tilde{f}_{i+1/2} = W(q_{i-1}, q_i, q_{i+1})$, where the function $W$ is a convex combination of the three quadratic interpolants, $q_{i-1}, q_i, q_{i+1}$, evaluated at $x_{i+1/2}$. Each quadratic interpolant in turn is a function of three nodes, and these evaluated at $x_{i+1/2}$ are:

$$q_{i-1} = 1/3 f_{i-2} - 7/6 f_{i-1} + 11/6 f_i\,,$$

$$q_i = -1/6 f_{i-1} + 5/6 f_i + 1/3 f_{i+1}\,,$$

$$q_{i+1} = 1/3 f_i + 5/6 f_{i-1} - 1/6 f_{i+2}\,.$$

The function $W$ is evaluated as follows:

$$W = w_{i-1} q_{i-1} + w_i q_i + w_{i+1} q_{i+1}$$

where

$$w_{i-1} = \frac{\alpha_{i-1}}{\alpha_{i-1} + \alpha_i + \alpha_{i+1}},\ w_i = \frac{\alpha_i}{\alpha_{i-1} + \alpha_i + \alpha_{i+1}},\ w_{i+1} = \frac{\alpha_{i+1}}{\alpha_{i-1} + \alpha_i + \alpha_{i+1}}$$

and

$$\alpha_{i-1} = \frac{1}{(\epsilon + IS_{i-1})^2},\ \alpha_i = \frac{6}{(\epsilon + IS_i)^2},\ \alpha_{i+1} = \frac{3}{(\epsilon + IS_{i+1})^2}$$

and

$$IS_{i-1} = \frac{13}{12}(f_{i-2} - 2f_{i-1} + f_i)^2 + \frac{1}{4}(f_{i-2} - 4f_{i-1} + 3f_i)^2\,,$$
$$IS_i = \frac{13}{12}(f_{i-1} - 2f_i + f_{i+1})^2 + \frac{1}{4}(f_{i-1} - f_{i+1})^2\,,$$
$$IS_{i+1} = \frac{13}{12}(f_i - 2f_{i-1} + f_{i+2})^2 + \frac{1}{4}(3f_i - 4f_{i+1} + f_{i+2})^2\,.$$

We point out that the data structure of the scheme is suitable for parallel computing using the technique of domain decomposition. If we combine the third-order Runge–Kutta scheme (equations (6) to (8)) with the fifth-order WENO interpolation, it is easily seen that largest amount of computational time is spent on the spatial integration which has perfect data independence, and can be run concurrently on multiple processors. Independence means that an interpolation at point $(i, j)$ is independent of the interpolation at point $(k, l)$ for $i \neq k$, $j \neq l$ so that they can be performed simultaneously.

### *2.4. The internal boundary method for the Euler equations in two dimensions*

Engineering applications usually require that simulations be carried out for complex geometries. The computational domains can be multiply connected regions of arbitrary shapes. The method we have developed for these applications is a pointwise method implemented on Cartesian grid that allows a (non-moving) body of arbitrary shape in two dimension to be embedded in the flow field. Comparing with the Cartesian boundary method for a finite volume scheme [6], our method has the following advantages: (1) It removes time step restriction due to small cells; (2) it is easy to implement high-order boundary conditions on the (internal) body; and (3) it is computationally efficient, i.e. it does not require extra computation, such as solving extra Riemann problems due to distorted cells near the boundary. Therefore it is simple to apply to any pointwise scheme.

*2.4.1. Internal boundary representation.* An internal boundary is described by a level-set function $\psi(x, y)$, such that contour $\psi = 0$ is the (closed) boundary. The inside region of the boundary curve is denoted by $B$ for body, and the outside region (inside computational domain $D$) is denoted by $F$ for flow field. The level-set function is assumed to satisfy

$$\psi(x, y) < 0, \qquad \text{for all } (x, y) \in F\,, \tag{19}$$
$$\psi(x, y) > 0, \qquad \text{for all } (x, y) \in B\,, \tag{20}$$
$$|\vec{\nabla}\psi(x, y)| = 1, \qquad \text{for all } (x, y) \in D = F \cup B, \tag{21}$$

which makes $\psi$ the distance function. It can be proven that the requirement of condition (21) guarantees that the normal function (a vector field function on $D$)

$$\boldsymbol{n}(x, y) = \frac{\vec{\nabla}\psi(x, y)}{|\vec{\nabla}\psi(x, y)|} \tag{22}$$

is a constant along a trajectory perpendicular to any $\psi$ = constant surface. Hence one can calculate $\boldsymbol{n}$ at a point which does not necessarily lie on the internal boundary.

Expressing the distance function analytically for an arbitrary boundary is very difficult. For the problems we discuss later as applications of this integrated methodology, the function

$\psi(x, y)$ can be expressed analytically. A numerical approximation can be used for more complicated boundaries.

*2.4.2. Reflective boundary condition.* It is common to use a non-penetrating, or a reflective boundary condition, on a non-moving objects and that condition is used later in our presentations of the applications. The reflective boundary condition requires that velocity component normal to the boundary be zero at the boundary. Let $X^B$ denote an internal boundary point, and $X^F$ denote a point in flow field, which is a mirror image of $X^B$ along normal direction (see figure 1). Then the reflective boundary conditions are

$$\begin{aligned}
&\text{Density :} && \rho|_{X^B} = \rho|_{X^F}\,,\\
&\text{X-momentum :} && (\rho u_n)|_{X^B} = -(\rho u_n)|_{X^F}\,,\quad (\rho u_t)|_{X^B} = (\rho u_t)|_{X^F}\,,\\
&\text{Y-momentum :} && (\rho v_n)|_{X^B} = -(\rho v_n)|_{X^F}\,,\quad (\rho v_t)|_{X^B} = (\rho v_t)|_{X^F}\,,\\
&\text{Total energy :} && E_t|_{X^B} = E_t|_{X^F}\\
&\text{Prog. var. :} && \lambda|_{X^B} = \lambda|_{X^F}
\end{aligned}$$

where $u_n$ and $v_n$ are velocity components that are normal to the boundary; and $u_t$ and $v_t$ are velocity components that are tangential to the boundary. The point $X^F$ does not usually lie on a grid point, so a 2-D interpolation scheme should be employed to determine the state from the neighboring four grid points, $X_1^F$, $X_2^F$, $X_3^F$ and $X_4^F$ as shown in figure 1. In our solver, a bilinear interpolation,

$$\mathcal{B}(x, y) = a + b\,x + c\,y + d\,xy\,, \tag{23}$$

is used. We calculate the coefficients $a, b, c, d$ in the polynomial by fitting it to the four states corresponding to the four neighbouring points. The state of $X^F$ is given by $\mathcal{B}(X^F)$ the bilinear interpolation function. Then the reflective boundary condition applies. Furthermore, if the neighboring four points do not include the internal boundary points, the calculation of the coefficients in (23) is explicit, otherwise the calculation is implicit. In the case of implicit calculation, a relaxation method is used to determine the coefficients iteratively.

*2.4.3. Implementation of the internal boundary method.* The computational module developed for the implementation of internal boundary method executes the following two steps:

(a) *Preprocessing*. In this step, the distance function $\psi(x, y)$ is calculated for a given internal boundary curve, using either an analytical formula or a numerical approximation. The data generated includes an index set of the internal boundary points, their coordinates and the associated normal directions, $\boldsymbol{n}$ of $\psi(x, y)$ at the points. It also includes an index set of mirror images of the internal boundary points and their surrounding (four) grid points. For the non-moving boundary, the bilinear interpolation coefficients are fixed for a given geometry, they are included in the preprocessing as well.

(b) *Boundary update*. The reflective boundary conditions are updated at the boundary points for each time step in the main hydrodynamic loop.

To determine the index set in step (a), one uses the property of the distance function (19). For $(x,\ y) \in D$ and $\psi_{i,j} > 0$, one checks for

$$\psi_{i+k,j} < 0 \text{ or } \psi_{i,j+k} < 0\,, \qquad \text{for } k = -r, ...0, ...r \tag{24}$$

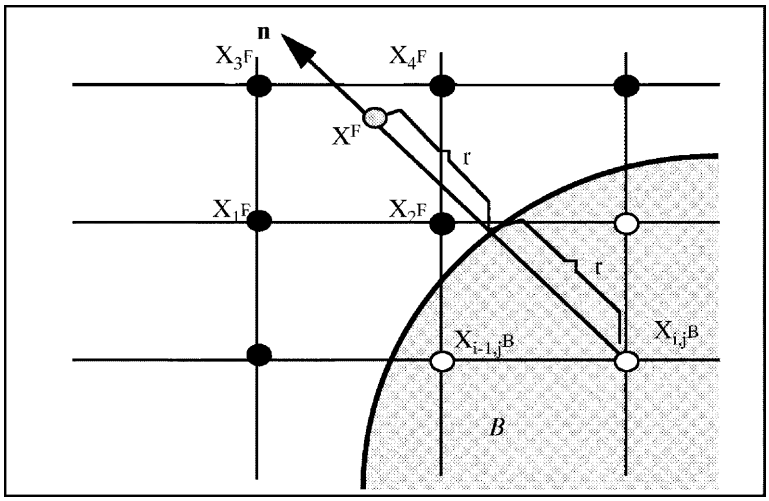


**Figure 1.** Implementation of internal boundary conditions. The state at the internal boundary point $X^B_{i,j}$ is the reflected state of mirror point $X^F$.

where $r$ is the order of accuracy of the scheme. If any of the inequalities holds, then point $(i, j)$ is an internal boundary point, or a point on the internal boundary.

Most information needed in the step (b) is generated in the step (a), therefore the internal boundary calculation in the main update is efficient.

### *2.5. Summary of the integrated algorithm*

We summarize our scheme by a flow chart displayed in figure 2, in which each block represents a functional module. For completeness we also mention two more minor issues in the implementation. The first issue is the time step calculation. Time step is calculated using $CFL$ condition,

$$\Delta t = \frac{CFL}{\text{Dim}} \min\left\{\frac{\Delta x}{|u+c|}, \frac{\Delta y}{|v+c|}\right\} \tag{25}$$

where $CFL$ is the $CFL$ number, Dim $(= 2)$ is the dimension of the problem and $c$, $u$, and $v$ are sound speed, velocity in $x$ and $y$ direction respectively. This formula is an extension of the 1-D formula, and is evaluated at every grid point in the domain prior to time integration. The $CFL$ number is in a range from 0.2 to 1.0 depending on the order of the spatial integration algorithm and the problem, e.g. the stiffness of the source terms.

The second issue is the implementation of the external boundary condition. For the problems presented in this paper, nonreflective and reflective boundary conditions are required. To implement the boundary conditions, the computational domain $D$ is first augmented by adding external boundary points, called the ghost points, outside $D$. See figure 3. The number of the ghost points needed depends on spatial integration schemes. The reflective boundary condition is implemented in the same way as used in the internal boundary algorithm. The states of the ghost nodes are the reflected state of the symmetry points—the mirror image about the boundary—in the flow field. In the implementation of the nonreflective boundary condition, the states of the ghost nodes are the same as that of the boundary points.

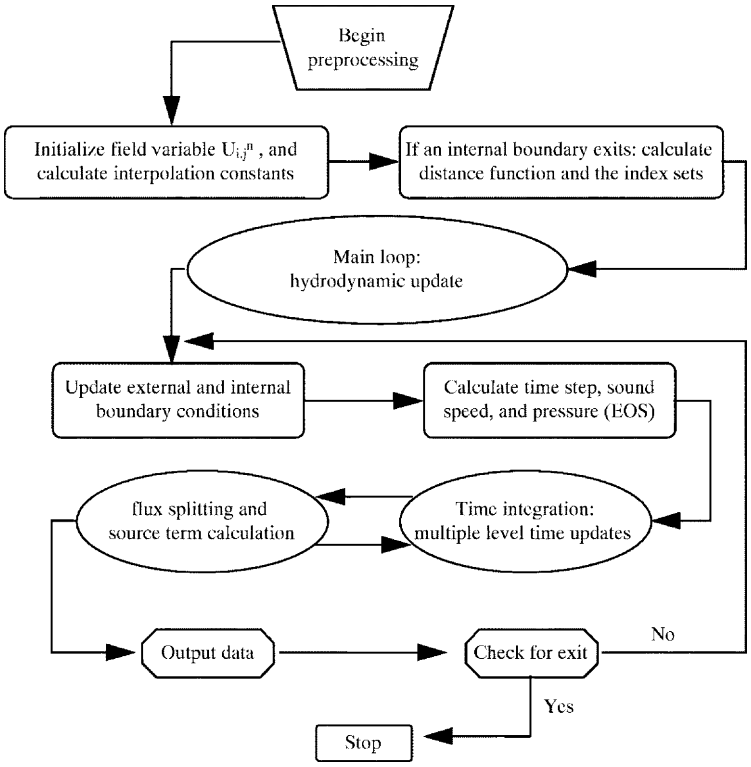


**Figure 2.** Flow chart of the algorithm. Each block is a functional module. Time integration and the flux function interpolation are described in detail in section 2.

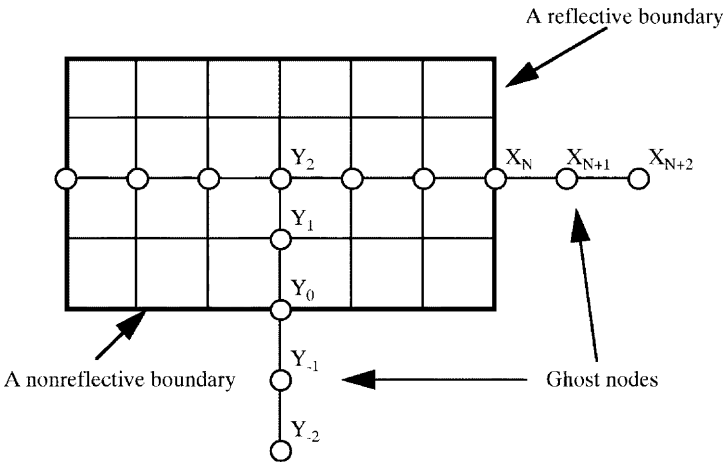


**Figure 3.** Implementation of external boundary conditions. For nonreflective boundary condition, states at point $X_{N+1}$, $X_{N+2}$ are the same as that at $X_N$. For reflective boundary condition, state of $Y_{-2}$ and $Y_{-1}$ are reflected state of $Y_1$ and $Y_2$ respectively.

## 3. Verification of the code

Here we show the results obtained when the reactive Euler equations are coded in the integrated manner discussed in section 2 and we discuss the verification of the code with

the use of different test problems. The reactive Euler equations are coded and solutions are verified by different test problems. The reactive Euler equations are generally assumed to be of the form

$$\boldsymbol{w},_t + (\boldsymbol{f}(\boldsymbol{w})),_x + (\boldsymbol{g}(\boldsymbol{w})),_y = \boldsymbol{s}(\boldsymbol{u})\,, \tag{26}$$

where

$$\boldsymbol{w} = \begin{pmatrix} \rho \\ \rho u \\ \rho v \\ \rho E_t \\ \rho\lambda \end{pmatrix}, \quad \boldsymbol{f}(\boldsymbol{w}) = \begin{pmatrix} \rho u \\ \rho u^2 + p \\ \rho uv \\ u(\rho E_t + p) \\ \rho u\lambda \end{pmatrix},$$

$$\boldsymbol{g}(\boldsymbol{w}) = \begin{pmatrix} \rho v \\ \rho uv \\ \rho v^2 + p \\ v(\rho E_t + p) \\ \rho v\lambda \end{pmatrix}, \quad \boldsymbol{s}(\boldsymbol{u}) = \begin{pmatrix} 0 \\ 0 \\ 0 \\ 0 \\ \rho r_\lambda \end{pmatrix}, \tag{27}$$

In the equations, $E_t = e + (u^2 + v^2)/2$ is the total energy, and $e$ denotes the internal energy, $p$ is the pressure, $\lambda$ is a reaction progress variable ($\lambda = 0$ for unreacted, $\lambda = 1$ for complete reacted material).

In our first set of test problems we show results for an ideal EOS,

$$e = \frac{p}{\rho(\gamma - 1)} - Q\lambda\,, \tag{28}$$

where $\gamma$ is the polytropic exponent and $Q$ is the heat of reaction. We take the rate law to be an Arrhenius form where $r_\lambda$ is given by

$$r_\lambda = k(1 - \lambda)\mathrm{e}^{-E/(p/\rho)}\,, \tag{29}$$

where $k$ is the rate constant, and $E$ is the activation energy.

Two problems, detonation initiation and shock diffraction from a cylinder, are tested. We first present the numerical verification of order of convergence, and then show comparisons by contour plots and line graphs. Analytical convergence results are not available since the system of PDEs are nonlinear.

### *3.1. Numerical verification of order of convergence*

The problem we tested is 1-D detonation initiation, with the activation energy $E = 10$, the heat release $Q = 50$, $\gamma = 1.4$, and rate constant, $k = 7$. Initially, there is a prescribed density $\rho = 1/(1 + 3\mathrm{e}^{-x^2})$, and $u = 0$, $p = 1$, $\lambda = 0$ everywhere. Note that this initial state has an initial temperature which is 4 times as large at the origin than for $|x| \to \infty$. Since the reaction term is exponentially sensitive to the temperature, the reaction rate at the origin is roughly 1800 times that of the $|x| \to \infty$ region. The scheme described in section 2 is used to solve numerically the above problem, on a domain $0 \leqslant x \leqslant 12$. The initial burning increases pressure in the system locally, and sends out acoustic waves. At roughly $t = 0.5$, these acoustic waves form into a shock, a reaction wave follows behind. Eventually the accelerating reaction wave catches up to the shock and coalesces to form a detonation at $t = 3.5$. See figure 4 for a pressure contour plot.

Since the exact solution is not available, we measure the relative error $E_1$ in the $L_1$ norm during the time integration to estimate the order of convergence. Let $r_c$ denote the

order of convergence. If a method is of $r$th order, then for a uniform mesh with $N$ grid points, the error should satisfy

$$E_1^N = O(\Delta x^{r_c}) .$$

When the uniform mesh is refined by doubling the grid points, we should have

$$E_1^{2N} = O\left(\left(\frac{\Delta x}{2}\right)^{r_c}\right) .$$

By assuming that the coefficients in the two formulae are the same, and solving for $r_c$, we obtain

$$r_c = \frac{\ln(E_1^N) - \ln(E_1^{2N})}{\ln(2)} . \tag{30}$$

Consequently, for two meshes of grid points $N$ and $2N$, one measures the relative errors at the *same* grid points, and thus determines $E_1^N$. Then the rate of convergence can be calculated according to (30).

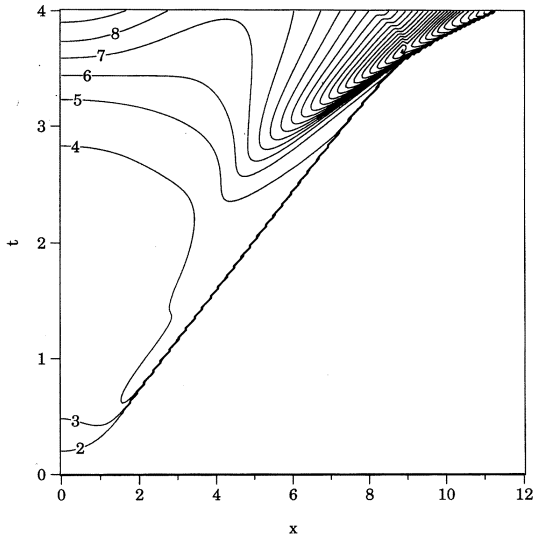


**Figure 4.** Numerically converged $x - t$ pressure contour plot (WENO scheme with $N = 3200$).

The $L_1$ error together with the rate of convergence are tabulated for density $\rho$, pressure $p$, velocity $u$, and mass fraction $\lambda$, as mesh is refined (at $t = 0.4$, before any shocks are formed). Both the Lax–Friedrichs (LF) and Roe (RF) flux splitting methods are tested with various order of interpolation methods. Table 1 shows that $r_c \to 2$ as $N \to \infty$ for the second-order (TVD) method with a $CFL = 0.8$, and second order Runge–Kutta scheme, and $r_c \to 5$ for the fifth-order (WENO5) scheme with a $CFL = 5.22\Delta x^{2/3}$, and a third order Runge–Kutta scheme. (Note that this $CFL$ is chosen to eliminate time integration errors since the time integration is only third-order, while the spatial scheme is fifth-order). Notice that our test is carried out for the *reactive* Euler equations with nonlinear source term, which is different from the previous similar convergence test (e.g. [7]), where *inert* Euler equations were solved.

**Table 1.** Results of numerical test of convergence. In the table, $N$ denotes grid points; $E_1$ is the $L_1$ norm of error measured between two consecutive meshes at $t = 0.4$; and $r_c$ is the numerical rate of convergence.

| Method | N | $E_1 - \rho$ | $r_c - \rho$ | $E_1 - p$ | $r_c - p$ | $E_1 - u$ | $r_c - u$ | $E_1 - \lambda$ | $r_c - \lambda$ |
|---|---|---|---|---|---|---|---|---|---|
| | 200 | 8.65e-3 | — | 2.65e-2 | — | 1.81e-2 | — | 5.32e-3 | — |
| MinMod | 400 | 4.12e-3 | 1.04 | 1.20e-2 | 1.14 | 6.65e-3 | 1.44 | 1.57e-3 | 1.76 |
| TVD-LF | 800 | 1.63e-3 | 1.36 | 4.36e-3 | 1.46 | 2.74e-3 | 1.28 | 4.39e-4 | 1.84 |
| | 1600 | 5.13e-4 | 1.67 | 1.41e-3 | 1.63 | 9.75e-4 | 1.49 | 1.19e-4 | 1.88 |
| | 200 | 4.52e-3 | — | 1.33e-2 | — | 1.24e-2 | — | 3.81e-3 | — |
| MinMod | 400 | 1.53e-3 | 1.57 | 4.79e-3 | 1.47 | 4.18e-3 | 1.57 | 1.05e-3 | 1.85 |
| TVD-RF | 800 | 4.47e-4 | 1.77 | 1.51e-3 | 1.67 | 1.18e-3 | 1.82 | 3.02e-4 | 1.80 |
| | 1600 | 1.26e-4 | 1.83 | 4.08e-4 | 1.89 | 3.05e-4 | 1.96 | 7.86e-5 | 1.94 |
| | 200 | 8.65e-3 | — | 6.61e-3 | — | 3.95e-3 | — | 2.89e-4 | — |
| WENO5- | 400 | 1.98e-4 | 3.51 | 4.76e-4 | 3.71 | 2.68e-4 | 3.88 | 1.32e-5 | 4.45 |
| LF | 800 | 1.05e-5 | 4.24 | 2.36e-5 | 4.34 | 1.42e-5 | 4.24 | 5.85e-7 | 4.50 |
| | 1600 | 3.90e-7 | 4.75 | 7.44e-7 | 4.98 | 4.73e-7 | 4.91 | 2.17e-8 | 4.75 |
| | 200 | 7.94e-4 | — | 2.48e-3 | — | 1.64e-3 | — | 2.04e-4 | — |
| WENO5- | 400 | 5.92e-5 | 3.75 | 1.65e-4 | 3.91 | 1.15e-4 | 3.84 | 9.10e-6 | 4.49 |
| RF | 800 | 2.34e-6 | 4.66 | 8.88e-6 | 4.22 | 6.48e-6 | 4.15 | 3.38e-7 | 4.75 |
| | 1600 | 8.94e-8 | 4.71 | 5.18e-7 | 4.10 | 3.61e-7 | 4.17 | 1.25e-8 | 4.76 |

Figure 4 shows the pressure contour plot versus $x$ and $t$ for the WENO scheme with $N = 3200$. This plot is very well converged. For comparison, figure 5 is the same pressure plot for the MinMod TVD scheme with $N = 200$, and figure 6 is for the WENO scheme with $N = 200$. The WENO scheme clearly resolves the flow features better. A set of line graphs comparing the converged solution (solid line, with N=3200) with the TVD scheme (N=200) and WENO scheme (N=200) at $t = 3.0$ show this even more dramatically. See figures 7 and 8. Notice that unlike inert shock capturing, detonation shocks must resolve the chemistry accurately to generate the correct shock locations and we believe this is a difficult and demanding test for any scheme.

### *3.2. Test problems*

A two-dimensional test problem is chosen to be that of a Mach number 2.81 shock diffraction around a circular cylinder. See Bryson and Gross [16] for the experimental results. Roe's flux splitting scheme (with the LF entropy fix) and WENO5 is used. The numerical grid used is $760 \times 530$, which puts 200 points in the radius of the cylinder. Figure 9 shows a contour plot of the density field which corresponds to roughly the time of figure 3 of (Bryson and Gross). A model Schlieren plot at the same time is shown in figure 10. Note that a numerically steady traveling shock wave was used as initial data to eliminate the start-up errors inherent to all shock capturing schemes. This example demonstrates that the internal boundary method works well, even for problems with strong shocks. Notice that there is a small amount of noise generated near the reflected shock near the front of the circular cylinder. This is most likely to do with the relatively slow motion of the reflected shock on the numerical grid, even a TVD scheme produced some noise for this problem (although it diffuses more quickly).

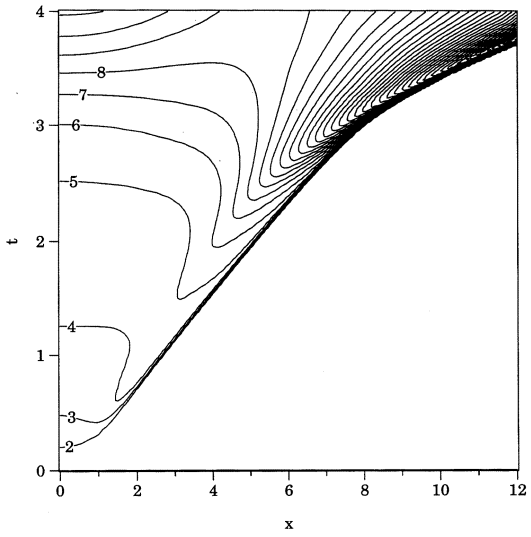


**Figure 5.** $x - t$ pressure contour plot (MinMod TVD scheme with $N = 200$).

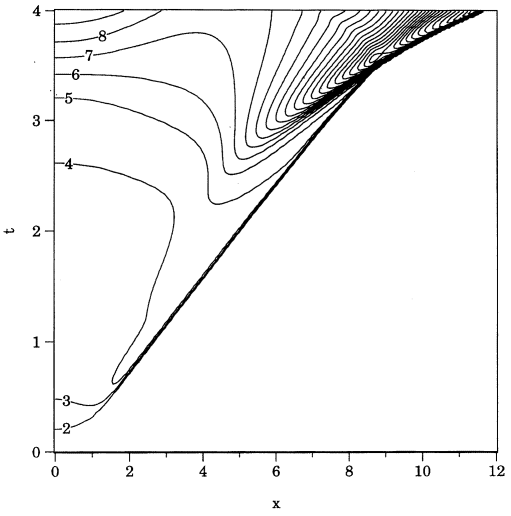


**Figure 6.** $x - t$ pressure contour plot (WENO scheme with $N = 200$).

## 4. Applications of the algorithm

In this section we present some practical examples of the application of these integrated algorithms to problems that are drawn from areas of research that are of current interest to us. In particular we present an application of the code to the measurement of detonation dynamics of the shock for an explosive with an ideal equation of state. The second example is drawn from the area of transition to detonation in non-ideal porous energetic materials.

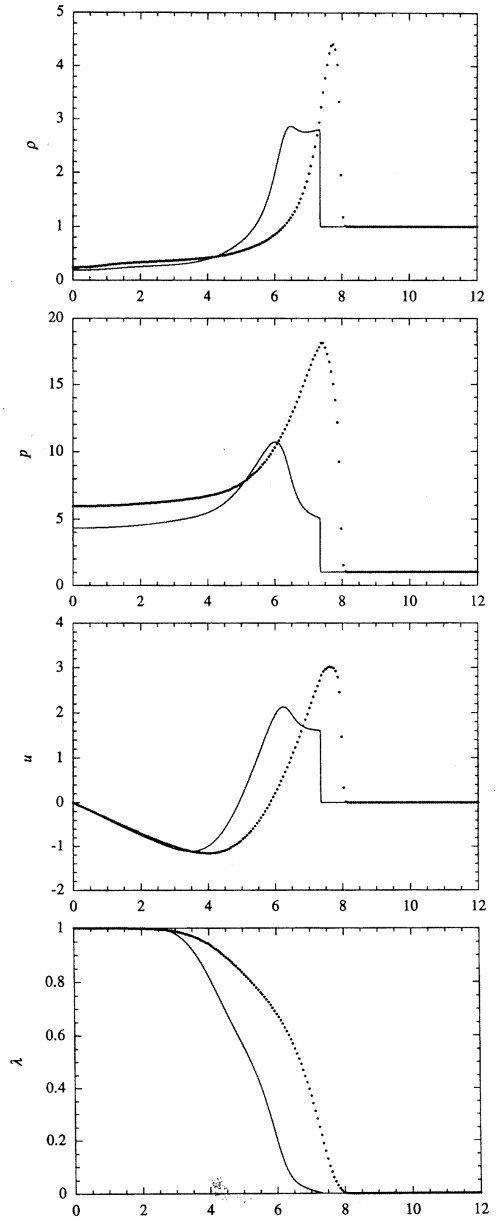


**Figure 7.** Line graphs at $t = 3.0$ (TVD scheme with $N = 200$).

### *4.1. Example one: measurement of detonation shock dynamics from direct numerical simulation*

Recent theoretical work [17, 18, 19, 20], on detonation shock dynamics (DSD), has suggested that under a wide range of flows that the motion of a detonation shock obeys an

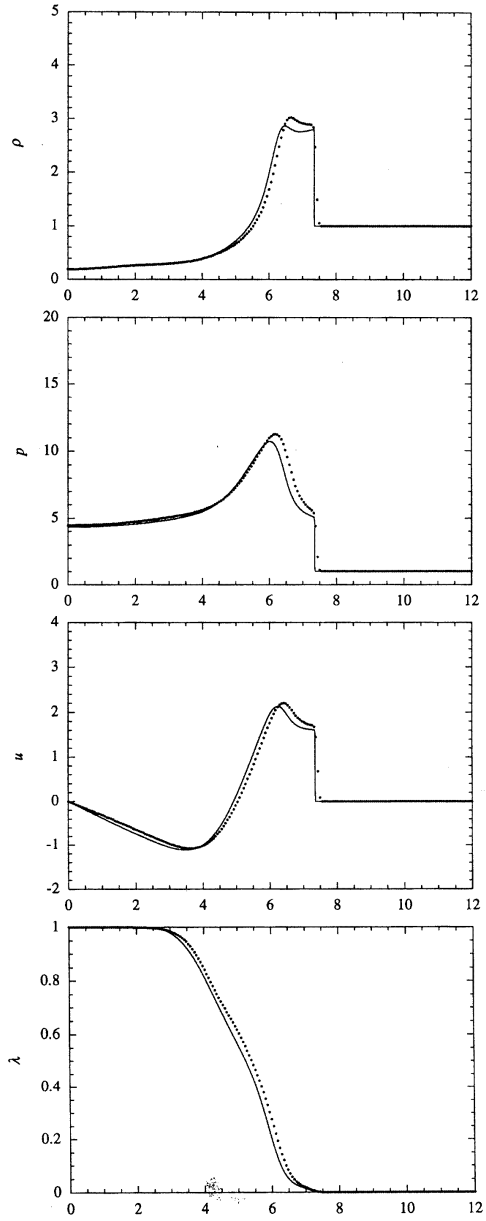


**Figure 8.** Line graphs at $t = 3.0$ (WENO scheme with $N = 200$).

intrinsic partial differential equation, where the normal velocity of the shock is depenendent only on other intrinsic quantities defined at the shock such as the curvature, and the first and second normal time derivatives of the shock velocity and the curvature, for example. To verify the quantitative accuracy of such theories, it is important to be able to measure these intrinsic quantities, such as the normal detonation velocity, $D_n$, the curvature of the

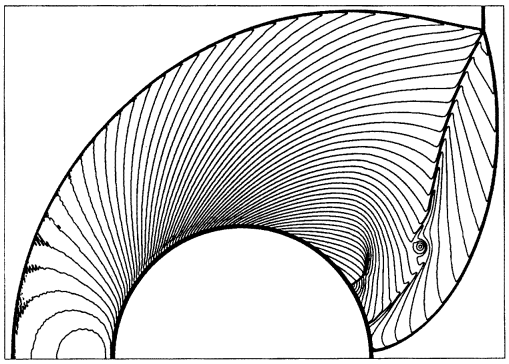

**Figure 9.** Density contours of a shock diffracting around a cylinder.

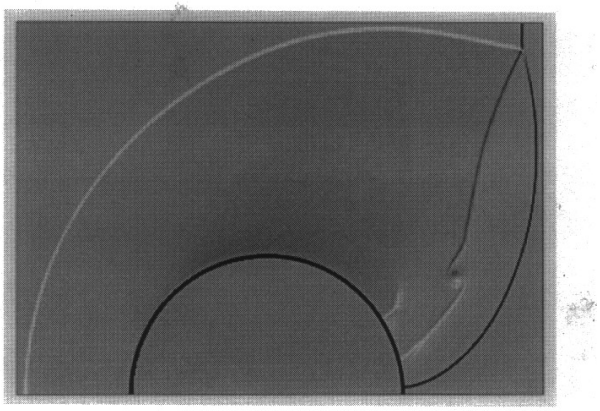

**Figure 10.** Model Schlieren plot of a shock diffracting around a cylinder.

shock front, $\kappa$, normal acceleration of the front, $\dot{D}_n$, etc directly from that found in a DNS of a detonating flow. This measurement, for comparison against theoretical prediction, is demonstrated below and is a main topic found in [3] and a forthcoming paper.

It is important to notice that for the purpose of the measurement of $D_n$ and $\dot{D}_n$ numerically, we have to take derivatives of the shock passage time found from the numerical solution of the DNS, which is a discrete function on a grid. This operation imposes a rigorous restriction on the smoothness and accuracy of the solution since the operation of differentiation amplifies the fluctuations in the solution, and decreases order of accuracy. Thus to make unambiguous comparisons of the DNS with the predictions of DSD theory, it is necessary to use a high order scheme for the DNS. Fifth order integration and small $CFL$ numbers (ranging from 0.1 to 0.2) were used in the calculation presented in this section.

*4.1.1. Governing equations and implementation of the Roe splitting scheme.* The governing equations we used for the shock dynamics measurements are the reactive Euler equations

with ideal EOS, given by (26), (27) and (28). To implement our scheme with Roe's splitting method, we need to know the mapping operators between the physical space and the characteristic space. For simplicity, we give an example in the $x$-direction. The Jacobian matrix $\boldsymbol{J}_f$ of the flux function $\boldsymbol{f}(\boldsymbol{w})$ has the form

$$\boldsymbol{J}_f = \begin{bmatrix} 0 & 1 & 0 & 0 & 0 \\ \frac{\beta}{2}\bar{q}^2 - \bar{u}^2 & (3-\gamma)\bar{u} & -\bar{v}\beta & \beta & Q\beta \\ -\bar{u}\bar{v} & \bar{v} & \bar{u} & 0 & 0 \\ -\beta\bar{u}(\bar{H} - \frac{\bar{q}^2}{2}) & \bar{H} - \beta\bar{u}^2 & -\bar{u}\bar{v}\beta & \gamma\bar{u} & \beta Q\bar{u} \\ -\bar{u}\bar{\lambda} & \bar{\lambda} & 0 & 0 & \bar{u} \end{bmatrix},$$

where $\beta = \gamma - 1$, and $\bar{q}^2 = \bar{u}^2 + \bar{v}^2$. The barred quatities are the Roe averaged values, i.e. variables or functions evaluated at the Roe states. For this model equation, the averaged state can be derived as follows [11]

$$\begin{aligned} \bar{\rho}_{i+1/2,j} &= \sqrt{\rho_{i,j}\rho_{i+1,j}} \\ \bar{u}_{i+1/2,j} &= \frac{u_{i,j}\sqrt{\rho_{i,j}} + u_{i+1,j}\sqrt{\rho_{i+1,j}}}{\sqrt{\rho_{i,j}\rho_{i+1,j}}} \\ \bar{v}_{i+1/2,j} &= \frac{v_{i,j}\sqrt{\rho_{i,j}} + v_{i+1,j}\sqrt{\rho_{i+1,j}}}{\sqrt{\rho_{i,j}\rho_{i+1,j}}} \\ \bar{\lambda}_{i+1/2,j} &= \frac{\lambda_{i,j}\sqrt{\rho_{i,j}} + \lambda_{i+1,j}\sqrt{\rho_{i+1,j}}}{\sqrt{\rho_{i,j}\rho_{i+1,j}}} \\ \bar{H}_{i+1/2,j} &= \frac{H_{i,j}\sqrt{\rho_{i,j}} + H_{i+1,j}\sqrt{\rho_{i+1,j}}}{\sqrt{\rho_{i,j}\rho_{i+1,j}}} \end{aligned}$$

where

$$H = \frac{E+p}{\rho}$$

is the enthalpy.

The eigenvalues of the Jacobian are

$$\lambda^{(f)}_{1,2,3,4,5,6} = (u-a),\ u,\ u,\ u,\ (u+a), \tag{31}$$

where $a = \sqrt{\frac{\gamma p}{\rho}}$ is the sound speed. Notice that all the variables and their functions (in (31) and in the expressions of $\boldsymbol{L}_f$ and $\boldsymbol{R}_f$ in the next paragraph) are evaluated at Roe's averaged states and we omit the bar for simplicity. The eigenvalues describe the local acoustic speeds of each characteristic field. The forward mapping operator $\boldsymbol{L}_f$ (from the physical space to the characteristic space) and the backward mapping operator $\boldsymbol{R}_f$ are then obtained as normalized left and right eigenvectors respectively. We have

$$\boldsymbol{L}_f = \frac{1}{2a^2}\begin{bmatrix} ua + \beta q^2 & -(a+\beta u) & -\beta v & \beta & \beta Q \\ va - \beta\frac{q^2}{2}\lambda & \beta u\lambda & -a + \beta v\lambda & -\beta\lambda & a^2 - \beta Q\lambda \\ 2a^2 - \beta(1-\lambda)q^2 & 2\beta(1-\lambda)u & 2\beta(1-\lambda)v & -2\beta(1-\lambda) & -2\beta(1-\lambda)Q - 2a^2 \\ -va - \beta\frac{q^2}{2}\lambda & \beta u\lambda & a + \beta v\lambda & -\beta\lambda & a^2 - \beta\lambda Q \\ -ua + \beta\frac{q^2}{2}\lambda & a - \beta u & -\beta v & \beta & \beta Q \end{bmatrix}$$

and

$$\boldsymbol{R}_f = \begin{bmatrix} 1 & 1 & 1 & 1 & 1 \\ u-a & u & u & u & u+a \\ v & v-a & v & v+a & v \\ H-ua & \frac{q^2}{2} - va - Q & \frac{q^2}{2} & \frac{q^2}{2} + va - Q & H+ua \\ \lambda & 1 & 0 & 1 & \lambda \end{bmatrix}$$

Once the mapping operators are available, the Roe's method can be implemented as detailed in section 2.

*4.1.2. Simulation of steady Chapman–Jouget detonation diffracting around a corner.* The application shown in this section specifically involves the diffraction of a steady planar Chapman–Jouget (CJ) detonation around a 90 degree corner. The relevant parameters are $E = 25$, $Q = 25$, $\gamma = 1.4$, and $k = 16.418$. The initial upstream pressure and density are taken to be unity. This results in a steady CJ detonation velocity of $D_{CJ} = 7.1247$, with a half reaction-zone length of unity. The computational domain $D$ (of size $0 \leqslant x \leqslant 120, 0 \leqslant y \leqslant 120$) is discretized by a numerical grid of $600 \times 600$, so that there are 5 points in the steady half-reaction zone. The detonation shock front is initially located at $x = 14$, and the corner is located at $x = 15$. Again, to avoid start-up errors, the initial states are chosen to be the numerically steady solution. The simulation is carried out until $t = 14$. A contour plot of the density at this time is given in figure 11. A plot showing the contours of the reaction progress variable is given in figure 12. From examination of the plots, it easy to see that there is a contact surface generated as the shock slows going around the corner. Behind the shock, and ahead of this contact surface, there is very little reaction, this is especially clear in figure 12.

As stated previously, one can measure the dynamics of the detonation shock by solving the compressible, reactive Euler equations using a DNS. Unfortunately, intrinsic shock-front dynamical quantities like the detonation shock speed, curvature of the shock front, etc. are not directly available from a the primitive quantities found in the formulation of the DNS. However if the fluid under goes a strong shock, the density jump at the shock is roughly a constant. So the detonation front may be approximated by some intermediate density (2.0 was used in these computations) between the undisturbed density, $\rho = 1$, and the shocked density, $\rho \approx 4$. For problems with quiescent upstream conditions, it can be safely assumed that that the detonation shock front will pass a fixed Eulerian point at most once. It is possible therefore to create a table of shock arrival times by sweeping over the computational grid and searching for grid points where the quantity $(\rho - 2)$ changes sign from one time level to the next. The first such occurrence will be when the shock passes over that fixed Eulerian point. Then quadratic interpolation in time can be used to get an accurate estimate of the arrival time, $t_a^{\mathrm{DNS}}(x, y)$. Once we have this DNS arrival table, important quantities such as shock speed, curvature, and other intrinsic quantities can be found. For example, the shock speed is given by $D_n = 1/|\vec{\nabla} t_a|$. The front locations are given simply as contours of $t_a^{\mathrm{DNS}}(x, y)$, etc. The contours of the DNS arrival times and instantaneous detonation velocities for the previous example is shown in figure 13.

### *4.2. Example two: simulation of DDT in porous energetic materials*

In this section, we present another example of an engineering application for deflagration-to-detonation transition (DDT) simulations in porous energetic materials, which is important for safety issues of high explosives.

A generic physical problem of interest concerns a block of porous explosive powder that is intially compacted to 70% of its full density. The block is suddenly impacted by an object (in this case an impenentrable bullet-shaped impactor) that drives into the porous explosive. From the physical DDT tube experiment [21], and 1-D simulation [2], we expected to observe the following sequence of events. The impact generates a compaction wave across which the density goes from 70% of full density to 100% full density or more and that travels at a steady speed normal to the compaction front. Burning near the impactor surface

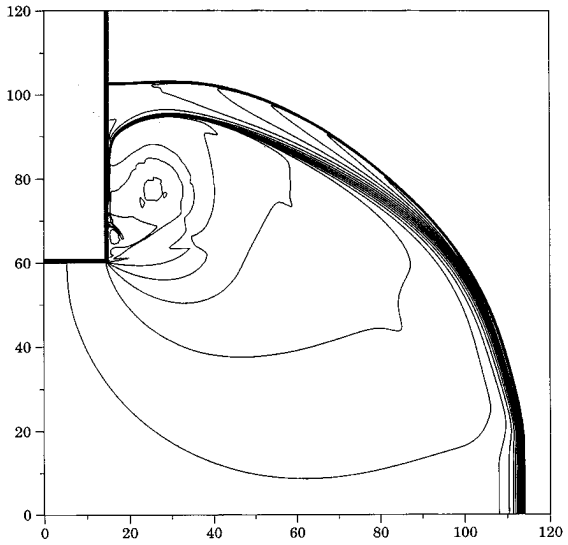


**Figure 11.** Density, $\rho$, contour plot at $t = 14$.

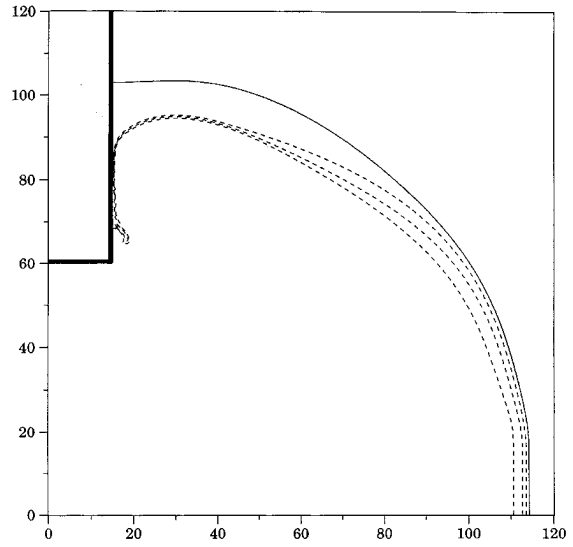


**Figure 12.** Reaction progress, $\lambda$, contour plot at $t = 14$. Dashed lines are $\lambda = 0.1, 0.5, 0.9$, solid line is shock location at $t = 14$.

starts to occurs, which pressurizes the region further and causes an even higher density region (a plug) to form slightly away from the impactor surface. After an further induction delay, vigorous detonation initiates in the dense plug region and travels at 6–7 km $s^{-1}$ downstream through all layers of the material.

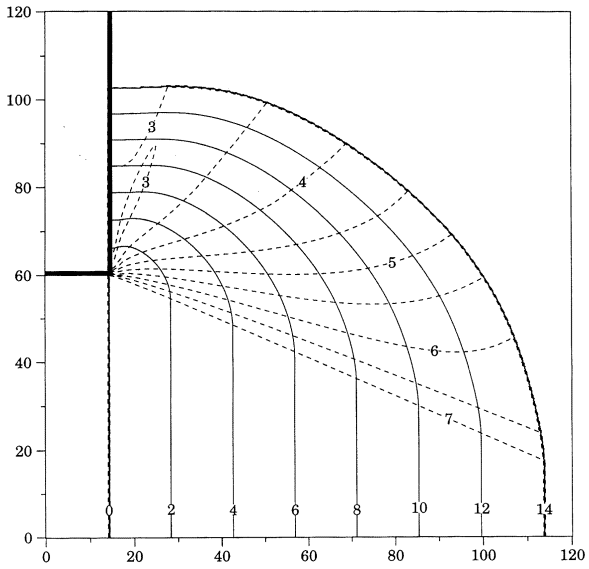


**Figure 13.** Shock fronts are given as solid numbered contours. Detonation velocities, $D_n$ are given as dashed numbered contours.

A single mixture-phase DDT model (GISPA), is used to simulate this scenerio described above [1]. The GISPA model has a conservative formulation that allows it to be cast in the general form of the reactive Euler equations expressed by (27), appended to include a compaction variable and a compaction rate law. In what follows, we introduce the specific governing equations and their constitutive relations of the GISPA model. Then we focus on the numerical implementation of the GISPA model using our solver, and show the simulation results. A hypothetical experimental geometry is shown in figure 14. The hard boundary bullet shown is approximated by a half circle, is embedded in a square body of explosive with the attributes of HMX powder and is subjected to an impulsive motion at 100 m $s^{-1}$.

*4.2.1. The GISPA model and governing equations.* The GISPA model embodies two phase properties for gas and solid phases, in its constitutive description that describes the mixture that comprises the porous energetic material. It has three conservation laws written for mass, momentum, and total energy, plus two evolution equations for independent rate variables, $\phi$—a compaction variable that measures the extent of compaction, and $\lambda$— a reaction variable that measures the extent of chemical reaction. Thus the governing equations of the GISPA model is a hyperbolic system of six PDEs:

$$\boldsymbol{w}_{,t} + (\boldsymbol{f}(\boldsymbol{w}))_{,x} + (\boldsymbol{g}(\boldsymbol{w}))_{,y} = \boldsymbol{s}(\boldsymbol{u}) \,, \tag{32}$$

where

$$\boldsymbol{w} = \begin{pmatrix} \rho \\ \rho u \\ \rho v \\ \rho E \\ \rho\lambda \\ \rho\phi \end{pmatrix}, \quad \boldsymbol{f}(\boldsymbol{w}) = \begin{pmatrix} \rho u \\ \rho u^2 + p \\ \rho uv \\ u(\rho E + p) \\ \rho u\lambda \\ \rho u\phi \end{pmatrix},$$

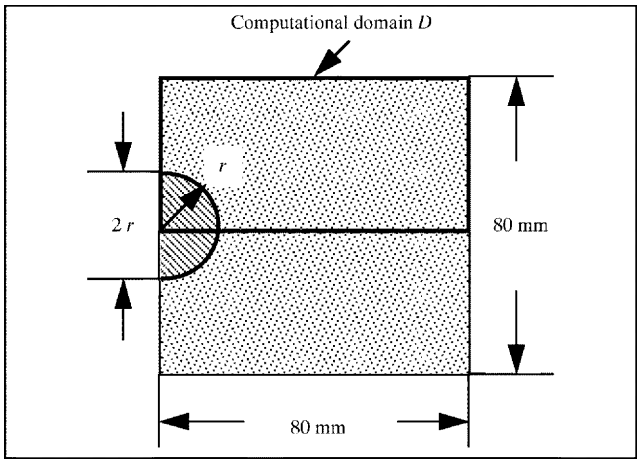


**Figure 14.** Setup of the virtual DDT experiment of a bullet impact on explosive material.

$$\boldsymbol{g}(\boldsymbol{w}) = \begin{pmatrix} \rho v \\ \rho u v \\ \rho v^2 + p \\ v(\rho E + p) \\ \rho v \lambda \\ \rho v \phi \end{pmatrix}, \quad \boldsymbol{s}(\boldsymbol{u}) = \begin{pmatrix} 0 \\ 0 \\ 0 \\ 0 \\ \rho r_\lambda \\ \rho r_\phi \end{pmatrix}, \tag{33}$$

and

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

The nonzero source term $r_\lambda$ and $r_\phi$ are rate functions for the compaction and the reaction processes respectively.

The equation of state (EOS) assumed in the GISPA model is a combination of the Hayes EOS (for solid phase $e_s(p_s, \rho_s)$ ) and the Jones–Wilkins–Lee (JWL) EOS (for gas phase $e_g(p_g, \rho_g)$ ) and takes the following general form

$$e(p, \rho, \phi, \lambda) = (1 - \lambda)e_s(\rho/\phi\,, p/\phi) + \lambda e_g(\rho, p). \tag{34}$$

The solid EOS has the form

$$e_s(p_s, \rho_s) = \frac{1}{\Gamma}(p_s - p_{s0}) - \left(t_3 - \frac{p_{s0}}{\rho_{s0}}\right)\left(1 - \frac{\rho_{s0}}{\rho_s}\right) + t_4\left\{\left(\frac{\rho_s}{\rho_{s0}}\right)^{n-1} - (n-1)\left[1 - \frac{\rho_{s0}}{\rho_s}\right] - 1\right\} \tag{35}$$

where $t_3, t_4$ are defined as

$$t_3 = C_{vs}T_{s0}\Gamma/\rho_{s0},\ \ t_4 = H_1/(\rho_{s0}n(n-1))\,,$$

where $\Gamma$, $H_1$ and $n$ are empirical constants $C_{vs}$ is the specific heat for pure solid and $\rho_{s0}$, $T_{s0}$ and $p_{s0}$ are the ambient density, temperature and pressure of the pure solid respectively.

The JWL EOS for the gas-phase can be expressed as

$$e_g(p_g, \rho_g) = \frac{1}{\omega\rho_g}\left[p_g - \left(Ae^{C_1/\rho_g} + Be^{C_2/\rho_g}\right)\right] - C_{vg}T_{g0}$$

$$-Q + \frac{1}{\phi_{s0}\rho_{s0}}\left[\frac{A}{R_1}e^{C_1/\rho_g} + \frac{B}{R_2}e^{C_2/\rho_g}\right], \tag{36}$$

where $\omega$, $A$, $B$, $R_1$, $R_2$ are empirical constants; $Q$ is a parameter to measure the detonation heat release, and $C_1$, $C_2$ are defined by $C_1 = -R_1\phi_{s0}\rho_{s0}$ , $C_2 = -R_2\phi_{s0}\rho_{s0}$.

The parameters used in the solid and gas EOS (35) and (36), together with initial conditions are listed in table 2.

**Table 2.** Parameters used in the Hayes (solid) and the JWL (gas) EOS and ambient properties.

| EOS parameter | Value |
|---|---|
| $H_1$ (N m$^{-2}$) | $1.3 \times 10^{10}$ |
| $n$ | 9.8 |
| $R_1$ | 4.2 |
| $R_2$ | 1.0 |
| $\omega$ | 0.25 |
| $A$ (N m$^{-2}$) | $[-8.005 + 21.39(\phi_{s0}\rho_0)10^{-3} - 16.23(\phi_{s0}\rho_0)10^{-6}$ $+4.595(\phi_{s0}\rho_0)10^{-9}]10^{11}$ |
| $B$ (N m$^{-2}$) | $[-0.014 + 0.0349(\phi_{s0}\rho_0)10^{-3} + 0.0156(\phi_{s0}\rho_0)10^{-6}$ $-0.026(\phi_{s0}\rho_0)10^{-9}]10^{11}$ |
| $C_{vs}$ (N.m kg$^{-1}$ K$^{-1}$) | $1.5 \times 10^3$ |
| $C_{vg}$ (N.m kg$^{-1}$ K$^{-1}$) | $[2.4 - 0.28((\phi_{s0}\rho_0)10^{-3} - 1.3)]10^3$ |
| $\Gamma$ (kg m$^{-3}$) | $2.1 \times 10^3$ |
| $Q$ (N.m) | $[7.91 - 4.33((\phi_{s0}\rho_0)10^{-3} - 1.3)^2$ $-0.934((\phi_{s0}\rho_0)10^{-3} - 1.3)]10^6$ |
| $p_{s0}$ (N m$^{-2}$) | $1.0 \times 10^5$ |
| $p_{g0}$ (N m$^{-2}$) | $1.0 \times 10^5$ |
| $T_{s0}$ (K) | 300 |
| $T_{g0}$ (K) | 300 |
| $\rho_s$ (kg m$^{-3}$) | $1.9 \times 10^3$ |

We adopted Hermann's quasi-static compaction (the $P - \alpha$) law, calibrated for HMX material by Sheffield and Gustavsen [22], in the rate function of the compaction process, which describes the equilibrium between compaction pressure and the deformation stress of the bed

$$\mathcal{P}(\phi, P_h) = \begin{cases} P_h\left(1 - \sqrt{\frac{\phi_0(1-\phi)}{\phi(1-\phi_0)}}\right), & p - p_0 \leqslant P_h \\ P_h, & p - p_0 > P_h. \end{cases} \tag{37}$$

The parameter $P_h$ is called hardening pressure or consolidation pressure, beyond which material will crush to solid density. The compaction rate function is then given by

$$r_\phi = k_\phi\left[p - p_0 - \mathcal{P}(\phi, P_h)\right]. \tag{38}$$

where $k_{\phi_v}$ is a rate constant, and $p$ is the bulk pressure.

The reaction rate law used in the GISPA model contains two parts, one for slow a rate in the porous compact and one for a fast rate in the fully dense material. The slow burn rate model uses autocatalytic kinetics to describe grain surface burning in the porous compact, and the fast rate model employs shock-to-detonation transition kinetics to describe detonation initiation in fully compacted solid material. (See [1] for details.) We then have

$$r_\lambda = H(p - P_{\text{ign}})\begin{cases} k_1(\lambda + \alpha_0)^b(1 - \lambda), & switch = 0 \\ k_2(1 - \lambda)\exp(-E\, p_0/P_{\text{shock}}), & switch = 1 \end{cases} \tag{39}$$

where $H(x)$ is heaviside function and $P_{\text{ign}}$ is a parameter. The parameters $k_1, k_2, b, \alpha_0, E$ are constants, and $p_0$ is the ambient pressure. In (39), $switch$ is a function of density used to detect high density region—the plug. The slow rate applies outside the plug ($switch = 0$), and the fast rate applies inside the plug ($switch = 1$). The threshold value of density is 95% of the theoretical maximum density of HMX material. The secondary shock pressure function $P_{\text{shock}}$, or $P_s$, is defined on the secondary shock front, and advected by the flow field. Such a shock pressure is a constant along a material particle path. Calculation of $P_s$ will be discussed shortly.

Parameters used in rate functions (38) and (39) are calibrated against the experimental data. For example, compaction rate constant in (38) is determined by calibrating the compaction wave thickness to be 2 mm, which is observed in the experiment. Similarly, we adjusted all the parameters and list them in table 3.

**Table 3.** Calibrated compaction and reaction parameters of the GISPA model for DDT simulation using the new scheme.

| | |
|---|---|
| Porosity | $\phi = 0.7$ |
| Piston velocity | $u_p = 100$ (m s$^{-1}$) |
| Hardening pressure | $P_h = 1.2 \times 10^8$ (Pa) |
| Compaction rate constant | $k_\phi = 4.0 \times 10^{-3}$ (kg m$^{-1}$ s$^{-1}$) |
| Initial product concentration | $\alpha_0 = 3.0 \times 10^{-7}$ |
| Burn rate | $b = 1.0$ |
| Reaction rate constant | $k_1 = 1.2 \times 10^5,\ k_2 = 1.1 \times 10^6$ (kg m$^{-1}$ s$^{-3}$) |
| Activation energy-like constant | $E = 6.2 \times 10^4$ |
| Ignition pressure control | $P_{ign} = 1.0 \times 10^7$ (Pa) |

*4.2.2. Implementation details.* The first issue we discuss is the implementation of the Roe flux splitting scheme. As described in section 2, we need to find eigenvectors and the left orthogonal eigenvector matrices of the Jacobian of $\boldsymbol{f}$ and $\boldsymbol{g}$ in equation (33), in order to construct the mapping operator. Let $\boldsymbol{J}_f$ be the Jacobian of $\boldsymbol{f}$, and $\boldsymbol{J}_g$ be that of $\boldsymbol{g}$. Then the Jacobian $\boldsymbol{J}_f(\boldsymbol{w})$ has the form

$$\boldsymbol{J}_f = \begin{bmatrix} 0 & 1 & 0 & 0 & 0 & 0 \\ a^2 - u^2 - \beta & 2u - \dfrac{up_e}{\rho} & -\dfrac{vp_e}{\rho} & \dfrac{p_e}{\rho} & \dfrac{p_\lambda}{\rho} & \dfrac{p_\phi}{\rho} \\ -uv & -v & u & 0 & 0 & 0 \\ u(a^2 - H - \beta) & H - \dfrac{u^2 p_e}{\rho} & -u\dfrac{vp_e}{\rho} & u\left(1 + \dfrac{p_e}{\rho}\right) & \dfrac{up_\lambda}{\rho} & \dfrac{up_\phi}{\rho} \\ -u\lambda & \lambda & 0 & 0 & u & 0 \\ -u\phi & \phi & 0 & 0 & 0 & u \end{bmatrix}, \tag{40}$$

where subscripts denote partial derivatives of $p$, and

$$\beta = \frac{p_e}{\rho}(H - (u^2 + v^2)) + \frac{\lambda p_\lambda + \phi p_\phi}{\rho}, \tag{41}$$

$$H = e + \frac{1}{2}(u^2 + v^2) + \frac{p}{\rho}, \tag{42}$$

$$a^2 = p_\rho + \frac{p p_e}{\rho^2}. \tag{43}$$

The eigenvalues of the Jacobian are

$$\lambda^{(f)}_{1,2,3,4,5,6} = (u-a),\ u,\ u,\ u,\ u,\ (u+a)\,. \tag{44}$$

The right eigenvector matrix $\boldsymbol{R}_f$ can be chosen as

$$\boldsymbol{R}_f = \begin{bmatrix} 1 & 1 & 1 & 1 & 1 & 1 \\ u-a & u & u & u & u & u+a \\ v & v-a & v & v & v+a & v \\ H-ua & H-\eta_2-va & H+\eta_3 & H+\eta_4 & H-\eta_2+va & H+ua \\ \lambda & 1 & 1 & 0 & 1 & \lambda \\ \phi & 1 & 0 & 1 & 1 & \phi \end{bmatrix}, \tag{45}$$

where

$$\eta_2 = \frac{\rho a^2}{p_e} + \frac{1}{p_e}[(1-\lambda)p_\lambda + (1-\phi)p_\phi]\,, \tag{46}$$

$$\eta_3 = -\frac{\rho a^2}{p_e} + \frac{1}{p_e}[\phi p_\phi - (1-\lambda)p_\lambda]\,, \tag{47}$$

$$\eta_4 = -\frac{\rho a^2}{p_e} + \frac{1}{p_e}[\lambda p_\lambda - (1-\phi)p_\phi]\,. \tag{48}$$

There is a freedom of choice regarding the eigenvectors corresponding to the multiple eigenvalue $u$. A proper choice of the eigenvectors will ensure the invertability of $\boldsymbol{R}_f$ in a robust fashion. The determinant of $\boldsymbol{R}_f$ given by (45) is

$$\mathrm{Det}(\boldsymbol{R}_f) = -\frac{4a^4\rho}{p_e}\,,$$

and it is seen that the determinant will not vanish in a physically meaningful flow field, e.g. a non-vacuum state ($\rho > 0$). An improper choice of the eigenvectors will result in a matrix whose determinant contains such factors as $\lambda$, $\phi$ or $p_\lambda$, $p_\phi$, which will approach zero during the DDT process, and consequently, the invertability of $\boldsymbol{R}_f$ is violated, which can lead to a numerical instability. A singularity of this kind should be avoided.

Once the right eigenvector matrix $\boldsymbol{R}_f$ is available, the left eigenvector matrix $\boldsymbol{L}_f$ is the orthogonal inverse of $\boldsymbol{R}$, which can be found as

$$\boldsymbol{L}_f = \frac{1}{2\rho a^2}\begin{bmatrix} \boldsymbol{l}_1 \\ \boldsymbol{l}_2 \\ \boldsymbol{l}_3 \\ \boldsymbol{l}_4 \\ \boldsymbol{l}_5 \\ \boldsymbol{l}_6 \end{bmatrix}, \tag{49}$$

where the row vectors $\boldsymbol{l}_k$, $k = 1, 2, ..., 6$, are given as

$$\boldsymbol{l}_1 = \{z + \rho a u,\ -up_e - \rho a,\ -vp_e,\ p_e,\ p_\lambda,\ p_\phi\}, \tag{50}$$

$$\boldsymbol{l}_2 = \{cz - \rho(a^2 - av),\ -cup_e,\ -cvp_e - \rho a,\ cp_e,\ cp_\lambda + \rho a^2,\ cp_\phi + \rho a^2\}, \tag{51}$$

$$\boldsymbol{l}_3 = \{2(\rho a^2 - c_1 z),\ 2c_1 up_e,\ , 2c_1 vp_e,\ -2c_1 p_e,\ -2c_1 p_\lambda,\ -2c_1(p_\phi + \rho a^2)\}, \tag{52}$$

$$\boldsymbol{l}_4 = \{2(\rho a^2 - c_2 z),\ 2c_2 up_e,\ 2c_2 vp_e,\ -2c_2 p_e,\ -2(c_2 p_\lambda + \rho a^2),\ -2c_2 p_\phi\}, \tag{53}$$

$$\boldsymbol{l}_5 = \{cz - \rho(a^2 + av),\ -cup_e,\ -cvp_e + \rho a,\ cp_e,\ cp_\lambda + \rho a^2,\ cp_\phi + \rho a^2\}, \tag{54}$$

$$\boldsymbol{l}_6 = \{z - \rho a u,\ -up_e + \rho a,\ -vp_e,\ p_e,\ p_\lambda,\ p_\phi\}, \tag{55}$$

with intermeadiary variables

$$c = 1 - \lambda - \phi\,,\ c_1 = 1 - \phi\,,\ c_2 = 1 - \lambda\,, \tag{56}$$

$$z = \rho p_\rho - e p_e - \lambda p_\lambda - \phi p_\phi + p_e \left( \frac{u^2 + v^2}{2} \right). \tag{57}$$

Instead of going through the same lengthy algebraic excersices to find the mapping operator for $\boldsymbol{J}_g$, we use the following symmetric relation between $\boldsymbol{f}$ and $\boldsymbol{g}$

$$\boldsymbol{J}_g(\boldsymbol{w}) = [\boldsymbol{P}\boldsymbol{J}_f(\boldsymbol{w})\boldsymbol{P}]_{u \leftrightarrow v} \tag{58}$$

where $\boldsymbol{P}$ is the permutation matrix

$$\boldsymbol{P} = \begin{pmatrix} 1 & 0 & & & & \\ 0 & 0 & 1 & & & \\ & 1 & 0 & 0 & & \\ & & 0 & 1 & 0 & \\ & & & 0 & 1 & 0 \\ & & & & 0 & 1 \end{pmatrix} \tag{59}$$

and $[\boldsymbol{X}]_{u \leftrightarrow v}$ means to interchange $u$ and $v$ in matrix $\boldsymbol{X}$. Since $\boldsymbol{P}$ is a unitary matrix, i.e. $\boldsymbol{P} = \boldsymbol{P}^{-1}$, the eigenvalues of $\boldsymbol{J}_g(\boldsymbol{w})$ are the same as those of $\boldsymbol{J}_f(\boldsymbol{w})$ after interchanging $u$ and $v$

$$\lambda^{(g)}_{1,2,3,4,5,6} = (v - a),\ v,\ v,\ v,\ v,\ (v + a).$$

Similarly, the right eigenvectors can be obtained from

$$\boldsymbol{r}_i^{(g)} = [\boldsymbol{P}\,\boldsymbol{r}_i^{(f)}]_{u \leftrightarrow v}\,,\ i = 1, 2, ..., 6\,.$$

Hence, we obtain

$$\boldsymbol{R}_g = \{r_1^{(g)}, r_2^{(g)}, r_3^{(g)}, r_4^{(g)}, r_5^{(g)}, r_6^{(g)}\}\,, \tag{60}$$

and $\boldsymbol{L}_g$ can be calculated in the same way as $\boldsymbol{L}_f$.

All the eigenvalues and vectors (the mapping operators) are evaluated at the Roe averaged states:

$$\begin{aligned} \tilde{\rho} &= \sqrt{\rho_r \rho_l}\,, \\ \tilde{e} &= \frac{\sqrt{\rho_l} e_l + \sqrt{\rho_r} e_r}{\sqrt{\rho_l} + \sqrt{\rho_r}}\,, \\ \tilde{H} &= \frac{\sqrt{\rho_l} H_l + \sqrt{\rho_r} H_r}{\sqrt{\rho_l} + \sqrt{\rho_r}}\,, \\ \tilde{u} &= \frac{\sqrt{\rho_l} u_l + \sqrt{\rho_r} u_r}{\sqrt{\rho_l} + \sqrt{\rho_r}}\,, \\ \tilde{v} &= \frac{\sqrt{\rho_l} v_l + \sqrt{\rho_r} v_r}{\sqrt{\rho_l} + \sqrt{\rho_r}}\,, \end{aligned}$$

which are derived using Glaister's method.

The second detailed issue of the implementation for this application is the calculation of the secondary shock function $P_s$ for reaction rate function. Initially, $P_s = p_0$, the ambient pressure. After the secondary compaction shock forms, $P_s$ records the pressure values along the shock front at each time step. Such shock front pressures are advected (once $P_s$ is defined) in the flow field by solving the equation (see [1])

$$(P_s)_{,t} + u(P_s)_{,x} + v(P_s)_{,y} = 0\,. \tag{61}$$

To reduce the computational cost, the advection equation is solved separately from the model governing equations. From the simplicity of the equation, we expect that (61) can be solved by a simple numerical scheme. The numerical solution of this equation should have as little diffusion as possible to maintain the strength of the pressure at the shock front. Common finite-difference methods—forward Euler and McCormack's method for example—do not meet this requirement. The peak values are either diffused as the pressure function is advected, or oscillations near the shock front.

A new solver is developed for the advection equation using a second-order Runge–Kutta method for time integration and an upwind-second-order difference with a minmod-slope limiter for spatial discretization. The scheme is TVD in both time and space. Following the notation in section 5.1, we can express the scheme as

$$P_s^{(1)} = P_s^{(0)} + \Delta t \mathcal{D}(P_s^{(0)}) \,, \tag{62}$$

$$P_s^{(2)} = P_s^{(0)} + \frac{1}{4}\Delta t \mathcal{D}(P_s^{(0)}) + \frac{1}{4}\Delta t \mathcal{D}(P_s^{(1)}) \,, \tag{63}$$

where the discrete spatial operator $\mathcal{D}$ is

$$\mathcal{D}(P_s^{(0)}) = P_{s\,i,j}^{(0)} + u_{i,j}\frac{\Delta t}{\Delta x}P_{sx} + v_{i,j}\frac{\Delta t}{\Delta x}P_{sy} \,, \tag{64}$$

where the second-order approximation to the derivative is given by

$$P_{sx} = (P_s)_{i^u,j} - (P_s)_{i^u-1,j} + \sigma_i \,, \tag{65}$$

$$P_{sy} = (P_s)_{i,j^u} - (P_s)_{i,j^u-1} + \sigma_j \,, \tag{66}$$

where $i^u$ and $j^u$ means that the indices $i$ and $j$ are determined in the upwind fashion. The minmod limiters, $\sigma_i$ and $\sigma_j$, are obtained from the second derivatives. For example, $\sigma_i$ is determined by

$$\sigma_i = \frac{1}{\Delta x}\mathrm{minmod}((P_{s,xx})_{i+1}, (P_{s,xx})_i) \,, \tag{67}$$

where

$$(P_{s,xx})_{i+1} = (P_s)_{i+1,j} - 2(P_s)_{i,j} + (P_s)_{i-1,j} \,, \quad \text{and} \tag{68}$$

$$\mathrm{minmod}(a,b) = \frac{1}{2}(\mathrm{sign}(a) + \mathrm{sign}(b))\min(|a|, |b|) \,. \tag{69}$$

The simple scheme turns out to be efficient and robust, and can be applied to solve advection equations which admit discontinuities in their solutions. A numerical test has shown that for the DDT problem the solution maintains the shock strength very well with little diffusion or oscillation.

The third issue of implementation of the GISPA model is the calculation of the distance function of the internal boundary—the bullet. Since the geometry of the bullet is relatively simple an analytical formula can be found

$$\psi(x,y) = \begin{cases} r - \sqrt{(x-l)^2 + (y-l)^2} \,, & x > l \,; \\ r - y \,, & x \leqslant l \,, \end{cases} \tag{70}$$

where $r$ is the radius of the bullet, and $l$ is the length of the cylinder, (see figure 14). Once $\psi$ is calculated, the index set can be determined by the internal boundary method.

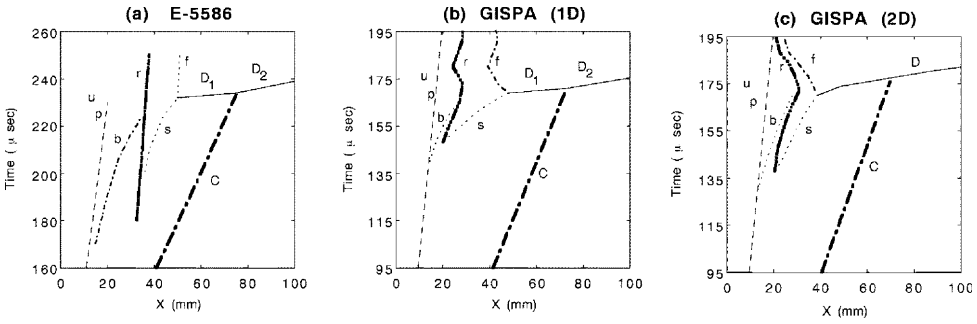


**Figure 15.** Comparison of the 2-D simulation, (c), with the DDT tube test, (a), and the 1-D simulation, (b). Data of graph (c) in $x$–$t$ plane is taken at the vertical symmetry line of the bullet.

*4.2.3. Results.* The simulation of the impact problem was run using both the Roe and Lax–Friedrichs flux splitting method with various order of accuracy. The results presented here are from the LF scheme with third-order Runge–Kutta in time and fifth-order WENO in space. Compared with the Roe scheme, the LF scheme is faster, and exhibits sharper shocks in the corresponding solution.

Taking advantage of the symmetry of the problem, we define the computational domain $D$ to be half of the physical domain shown in figure 14, which has physical dimension of 80 mm by 40 mm. The bullet is located in the lower left corner of $D$ with radius $r$=12.5 mm. The domain $D$ is discretized by 400 points in $x$ direction and 200 points in $y$ direction. A reflective boundary condition is applied to the internal boundary around the bullet and the lower horizontal edge of $D$—the symmetry line, and nonreflective boundary condition is applied to the rest of the boundaries of $D$. The $CFL$ number used in the solver is 0.6.

The simulation exhibits all of the major features that are observed experimentally. Figure 15(a) shows the results of a DDT physical experimental test displayed in an $x$–$t$ plane, [21]. The locus labelled $u_p$ is the trajectory of the supporting piston. The locus $b$ is a burn front where reaction can be observed to generate light. The locus labelled $C$ is the primary compaction wave that is initially created by the piston impact. The region between the labels $s$ and $r$ is the high density plug region, where $s$ now refers to the location of the secondary compaction wave that sits behind the primary compaction wave, and $r$ is the rear of the plug region. The labels $D_1$ and $D_2$ show the locations of the emergence and further propagation of a detonation shock as it evolves in the flow.

Figures 15(b) and 15(c) show similar diagrams taken from our simulation of the GISPA model. Since the relevant experimental data is not available, we compare a 1-D slice of data from the 2-D simulation with the DDT tube (the piston impact) experiments and its 1-D simulation [2]. Figure 15(b) shows the results of a 1–D simulation, and figure 15(c) shows the results of $x - t$ data is taken at the lower boundary of the computational domain (i.e. on the centerline of the bullet impactor) $D$, and the plot is shifted along the $x$-axis such that the bullet head is located at 0. The comparison shows that the transition process in the bullet impact is similar to that of the piston impact. The compaction structures are almost identical in the time period shown. However, the effects of two dimensionality is prominent. Due to the curvature effect of the bullet and the transverse flow, the plug region of the 2-D simulation is thinner than that of the 1-D simulation. Also, the retonation observed in the 2-D simulation is stronger than observed in the 1-D simulation.

In figure 16, we show three snap shots of the density and (log) pressure fields in the

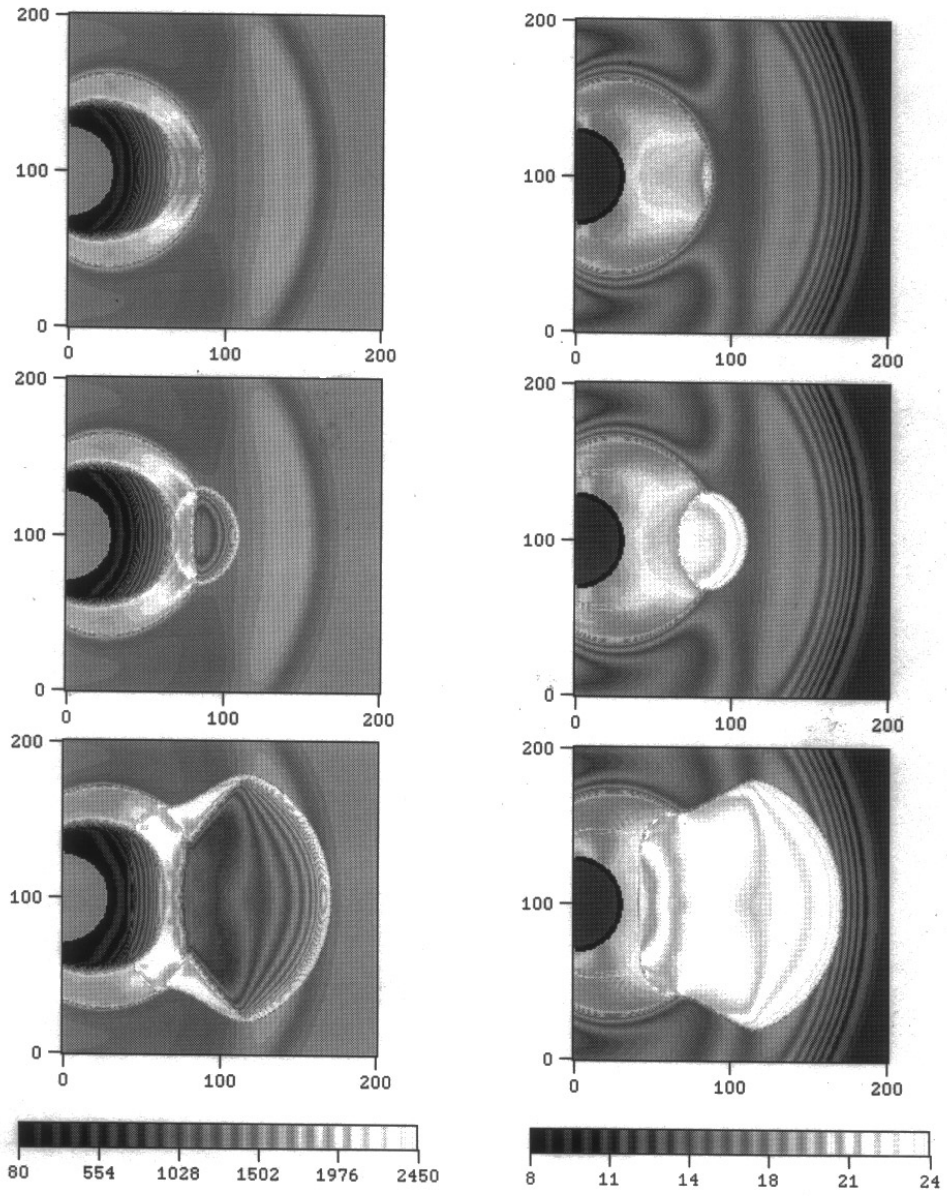


**Figure 16.** Snap shots of density (kg $m^{-3}$) (left) and (log) pressure (Pa) (right) fields from the GISPA 2-D simulation: $T = 171\,\mu$ s (top); $T = 174\,\mu$s (middle); and $T = 178\,\mu$s (bottom).

detonation initiation. It is seen that long time-scale phenomenon—compaction, and small time-scale features—detonation are well resolved. The simulation predicts sharp and stable wave fronts. As compared with the (second-order) McCormack's scheme, which is used for the 1-D DDT tube simulation, our solver uses only one fifth of the grid points, and yet there are no observable oscillation of any kind in the solution.

## Acknowledgments

This work was carried out with support from the US Air Force Wright Laboratories Armament Directorate, Eglin Air Force Bases F08630-95-K-0004, US Air Force Office of Scientific Research, Mathematics ASSERT grant F49620-93-1-0532 and the Department of Energy, Los Alamos National Laboratory, DOE/LANL 6730M0014-9.

## References

[1] Xu S 1996 Modeling and numerical simulation of deflagration and detonation transition in porous energetic materials *PhD thesis* Theoretical and Applied Mechanics, University of Illinois at Urbana-Champaign, May 1996

[2] Xu S and Stewart D S 1997 Deflagration to detonation transition in porous energetic material: a comparative model study *J. Eng. Math.* in press

[3] Aslam T 1996 Investigations on detonation shock dynamics *PhD thesis* Theoretical and Applied Mechanics, University of Illinois at Urbana-Champaign, May 1996

[4] Aslam T, Bdzil J and Stewart D S 1996 Level-set methods applied to modeling detonation shock dynamics *J. Comput. Phys.* **126** 390–409

[5] Harten A, Osher S, Engquist B and Chakravarthy S 1996 Some results on uniformly high order accurate essentially nonoscillatory schemes *J. Appl. Num. Math.* **2** 347–77

[6] LeVeque R J 1992 it Numerical Methods for Conservation Laws (Lecture Notes in Mathematics) (Boston: Birkhauser)

[7] Shu C and Osher S 1988 Efficient implementation of essentially non-oscillatory shock-capturing schemes *J. Comput. Phys.* **77** 439–71

[8] Quirk J 1994 An alternative to unstructured grids for computing gas dynamic flows around arbitrarily complex two-dimensional bodies *Computers and Fluids* **23** 125–42

[9] Berger M and LeVeque R 1989 An adaptive Cartesian mesh algorithm for the Euler equations in arbitrary geometries *AIAA Paper 89-1930-CP*

[10] Rogerson A and Meiburg E 1990 A numerical study of the convergence properties of ENO schemes *J. Sci. Computing* **5** 151–67

[11] Roe P 1981 Approximate Riemann solvers, parameter vectors and difference schemes *J. Comput. Phys.* **43** 357–72

[12] Glaister P 1988 An approximation linearised Riemann solver for the Euler equation for real gases *J. Comput. Phys.* **74** 382–408

[13] Quirk J J 1994 A contribution to the great Riemann solver debate *Int. J. Num. Methods* **18** 555–74

[14] Liu X-D, Osher S and Chan T 1994 Weighted essentially non-oscillatory schemes *J. Comput. Phys.* **115** 200–12

[15] Jiang G-S and Shu C 1996 Efficient implementation of weighted ENO schemesm *J. Comput. Phys.* submitted

[16] Bryson A E and Gross R W F 1961 Diffraction of strong shocks by cones, cylinders, and spheres *J. Fluid Mech.* **10** 1–16

[17] Stewart D S and Bdzil J B 1988 The shock dynamics of stable multidimensional detonation *Combustion and Flame* **72** 311–23

[18] Stewart D S and Bdzil J B 1988 A lecture on detonation shock dynamics *Mathematical Modeling in Combustion Science* (Lecture Notes in Physics 249) (Berlin: Springer) pp 17–30

[19] Bdzil J B and Stewart D S 1988 Modeling of two-dimensional detonation with detonation shock dynamics *Phys. Fluids* A **1** 1261–7

[20] Yao J and Stewart D S 1996 On the dynamics of multi-dimensional detonation *J. Fluid Mech.* **309** 225–75

[21] McAfee J M, Asay B, Campbell W and Ramsay J B 1989 Deflagration to detonation in granular HMX *Proc. 9th Int. Symp. on Detonation* vol 1 pp 265–79

[22] Sheffield S A and Gustavsen R L 1993 Unreacted Hugoniots for porous and liquid explosives *Proc. of the Joint AirAPT/APS Conf.*