Simulation of strong shock solutions of Euler equations using the flux-corrected transport algorithm

S. V. G. Menon[1, a)] and Bishnupriya Nayak[2]

[1)] *ShivEnclave, 304-31-B, Tilak Nagar, Mumbai, 400 089, India.*[b)]

[2)] *High Pressure and Synchroton Radiation Physics Division,*
Bhabha Atomic Research Centre, Mumbai, 400 085, India.

Formation and propagation of strong shock waves, which are adequately described using the Euler equations, occur in many applications in several areas of high pressure science and technology. Applications to real problems need numerical methods because of the highly non-linear nature of Euler equations. A common technique used for solution is the finite-difference method, which requires the addition of certain amount of artificial viscous pressure to materiel pressure. This is to smooth out sharp discontinuities representing shocks. The flux-corrected transport algorithm treats the Euler equations as a set of three conservation equations; it generates positive, monotone and stable solutions, and does not need artificial viscous pressure. This method provides accurate solutions to several problems in fluid flow, using either the Lagrangian or Eulerian descriptions. The aim of the present chapter is to discuss this method and present simulation results of several strong shock problems. These include some benchmark problems as well as impact generated or explosive driven shock propagation problems. An accurate equation of state model for metals is used with the FCT algorithm to simulate experimental applications.

Keywords: Euler equations, Shock waves, Computational methods, Finite volume methods, Equation of state, Flux-corrected transport.

[a)] Corresponding author - Email address: menon.svg98@gmail.com

[b)] (Retired from Bhabha Atomic Research Centre, Mumbai, 400 089, India.)

I. INTRODUCTION

Euler equations express conservation laws of mass, momentum and energy in space and time[1]. In fact, these equations provide extensions of newton's laws to continuous medium. Basically, these are three coupled non-linear partial differential equations for the fluid density, velocity field and total energy field (kinetic energy and internal energy). Euler equations find applications in multidisciplinary fields of science and technology, particularly involving high energy density systems, which deal with energy densities about 10^5 J per gram, or equivalently about 100 GPa of pressure. These systems include deep interiors of giant planets, hot and dense plasma in stellar interiors, the core of nuclear weapons, etc., to name a few. An important topic of significant current investigations is the field of thermonuclear fusion systems - particularly employing inertial confinement concepts - studied in different laboratories world wide[2]. In all these systems, viscous forces and energy transfer by thermal conduction and convection processes are negligibly small. Euler equations, which describe inviscid fluid dynamics, adequately model such systems, and the more general set of Navier-Stokes equations are unnecessary[1].

Euler equations, being highly non-linear partial differential equations, require numerical methods of solution even for ideal problems. However, it is easy to obtain wave equations for sound propagation when one considers small perturbations around spatially uniform conditions[3]. This chapter is concerned with a specific numerical method and applications for cases of large perturbations, which lead to generation and propagation of shock waves. Shock propagation induces hydrodynamic motions in the medium, which in turn leads to space-time variation of the fluid variables such as density, pressure, internal energy, fluid velocity etc. Due to the omission of viscosity and heat transfer, shocks occur as discontinuities in the fluid variables, and these propagate with characteristic speeds called the shock velocity. All these aspects of fluid flow are of great importance in the high energy density systems mentioned earlier. Experimental techniques in shock wave physics are well developed for measuring pressure-time profiles, fluid velocity fields, shock velocity fields, free-surface velocity-time profiles, etc[4].

Several numerical methods based on the finite difference approximations[5] have been designed to solve Euler equations. Almost all, except those employing implicit methods, use a marching technique to advance in the time domain. The state variables are updated in each time slice over all spatial meshes using the numerical algorithm. To smooth out the discontinuous shock within the numerical treatment, certain amount of artificial viscosity is added to the discrete equations[6]. This is necessary because finite difference approximations to spatial and temporal derivatives breakdown whenever steep gradients are present in the solutions. Severe rounding-off of shock fronts is an unavoidable feature of such methods. Furthermore, they also suffer from problems related to monotone and positive properties of the fluid variables. Some of these aspects will be discussed later.

Euler equations have four dependent variables - mass density, fluid velocity, total energy and pressure. Therefore, to close the system of equations it is necessary to add an equation of state (EOS) of the material, which provides a relation - sometimes implicitly - between pressure, density and internal energy. While the numerical algorithm is used to update the spatial profiles of fluid variables, the EOS provides the pressure profile. EOS is determined by employing statistical mechanics methods and it must account for different states of the material that will be spanned during shock propagation. When the typical time scale of fluid flow is much larger than relaxation time to attain local thermodynamic equilibrium, EOS derived from statistical mechanics are useful. One approach is to generate tables of internal energy and pressure, over a fine grid of density and temperature, and integrate these with numerical algorithms.

One of the aims in this chapter is to discuss the flux-corrected transport (FCT) algorithm[7] and apply it to obtain solutions to some strong shock problems using Euler equations. The flux-corrected transport methodology is now well developed[8], and have already been applied to solve multidimensional hyperbolic transport equations[9]. An EOS model, applicable to compressed as well as expanded states of metals, that can be easily integrated with the method, is also discussed. Specifically, we discuss few problems in strong shock propagation in condensed media, which consist of benchmark problems as well as experimental cases. The FCT algorithm has several desirable features: second order accuracy in approximating time and space derivatives, minimal amount of residual diffusion, positive property of density fields and their monotonic variation, stability against error accumulation, etc. It is known to be very useful to resolve steep gradients occurring in fluid flow problems[10]. These are important features to be desired when numerical solutions are compared to analytical or experimental results. An added advantage of the method is that it does not need addition of artificial viscous pressure[6], which usually adds nonphysical fluctuations in profiles and smears steep gradients occurring in shock fronts.

Euler equations are described in Section II. The integral form of conservation laws, the EOS mentioned above and a brief outline of general aspects of numerical schemes are also included. The FCT algorithm is discussed in some detail in section III. Specific applications and results are covered and discussed in Section IV. Finally, a summary and conclusions are given in section V.

II. EULER EQUATIONS

As mentioned in the previous section, the Euler equations describing conservation laws of mass momentum and energy, are first order non-linear hyperbolic partial differential equations. These, expressed in three-dimensional space and time in a fixed (or Eulerian) reference frame, are given by[1,3]:

$$\begin{aligned} \frac{\partial \rho}{\partial t} + \vec{\nabla} \cdot (\rho\, \vec{u}) &= 0, \\ \frac{\partial (\rho u_k)}{\partial t} + \vec{\nabla} \cdot (\rho u_k\, \vec{u}) &= -\nabla_k P, \qquad k = 1,2,3, \\ \frac{\partial (\rho E_t)}{\partial t} + \vec{\nabla} \cdot (\rho E_t\, \vec{u}) &= -\vec{\nabla} \cdot (P\vec{u}). \end{aligned} \tag{1}$$

The space-time field variables in these equations ($\rho, \vec{u}, P$ and E_t) are, respectively, mass density, fluid velocity, material pressure and total specific energy of the substance. The subscript k used for velocity field $\vec{u}$ denotes its kth Cartesian component. Total specific energy is the sum of specific internal energy (E) and specific kinetic energy, that is, $E_t = E + 0.5\, \vec{u} \cdot \vec{u}$. The divergence terms on the left hand side in these equations account for net flow of the field variables out of a volume element, while the terms on the right hand side represent the driving sources. There are three equations but four unknowns. So the set is closed by specifying an EOS of the form, $P = P(\rho, E)$, which is specified externally.

All the three equations state conservation laws of a field G in a fixed volume element: *[rate of change of G + rate of outward flow of G = rate of source]*. So these can be expressed in concise form as

$$\frac{\partial G}{\partial t} + \vec{\nabla} \cdot (G\, \vec{u}) + S_{in} - Q = 0, \tag{2}$$

where G stands for ρ, ρu_k or ρE_t, and S_{in} denotes internal sources like $\nabla_k P$ in the momentum equations and $\vec{\nabla} \cdot (P\vec{u})$ in the energy equation. Finally, Q represents rate of external source, if any. There is an alternate way of expressing the conservation laws using a Lagrangian reference frame. Here the evolution of the field variables is followed in a volume element as it moves throw the fluid with the local fluid velocity. The volume element may change size (due to compression or expansion) and shape (because of space dependent velocity), however, the mass of the fluid within the element will remain unchanged. The Lagrangian conservation equation is given by:

$dG/dt + G\,\vec{\nabla}\cdot\vec{u} + S_{in} - Q = 0$. This equation follows on expressing the convective (or material) derivative as the total derivative, that is, $dG/dt = \partial G/\partial t + \vec{u}\cdot\vec{\nabla}G$. It is easy to transform the conservation equations from one scheme to the other using this definition.

For developing numerical methods, it is advantageous to express Euler equations in integral form, over a volume $\mathscr{V}$ bounded by surface $\partial\mathscr{V}$, as: $\int_{\mathscr{V}} \partial G/\partial t\; dV + \int_{\partial\mathscr{V}} G\,\vec{u}\cdot d\vec{A} + \int_{\mathscr{V}}(S_{in} - Q)\; dV = 0$. If the reference frame moves with local velocity $\vec{u}^g$, then the equation of motion for a volume element can be written as: $\partial V/\partial t = \vec{u}^{\,g}\cdot d\vec{A}$. On multiplying this equation with G, integrating over $\mathscr{V}$ and adding to the integral form Euler equation yields the general equation:

$$\frac{\partial}{\partial t}\int\limits_{\mathscr{V}(t)} G\,dV \;+ \int\limits_{\partial\mathscr{V}(t)} G\,(\vec{u} - \vec{u}^{\,g})\cdot d\vec{A} \;+ \int\limits_{\mathscr{V}(t)} (S_{in} - Q)\; dV = 0, \tag{3}$$

It is clear that in the Lagrangian scheme, as $\vec{u} = \vec{u}^{\,g}$ and so the flux through the surface element $\partial\mathscr{V}(t)$ is zero, only the sources contribute to total change in G. Integration of Eq.(3) over a time step makes it convenient to derive discrete form of equations as will be done later in the section on FCT algorithm.

A. Equation of state

As pointed above, EOS of materials is an inevitable ingredient in applications of Euler equations because it is needed to close the set of equations. In fact, EOS brings in the thermodynamic characteristics of the medium in simulating fluid flow. Of course, EOS is important in its own right in several fields of solid state science like geophysics, study of planetary and stellar structures, analysis of fast reactor accidental scenarios, analysis of nuclear weapon effects, etc. A complete specification of an EOS is effected by providing pressure and specific internal energy as functions of density (ρ) and temperature (T)[11]. Basically, there are three components in these functions, which account for (i) the zero-temperature isotherm, (ii) thermal ionic effects and (iii) thermal electronic effects. The effects of interaction between ionic and electronic motion are also present, however, these contribute only a few percent to pressure and energy and so may be neglected in modeling high energy density systems. Pressure and specific internal energy are, then, expressed as[11] :

$$P(\rho,T) = P_c(\rho) + P_{ti}(\rho,T) + P_{te}(\rho,T) \tag{4}$$

Here the terms represent, respectively, the three components mentioned above; the subscripts c, ti and te denote 'zero temperature', 'thermal-ion' and 'thermal-electron' components. Accurate representations of P_c for metals and compounds are available based on DFT computations. Details of ionic motion, starting from the Debye's model at low temperatures to melting transition and, finally, ideal gas behavior, are incorporated in P_{ti}. Electron EOS models usually make use of statistical approaches as in the Thomas-Fermi model. Exactly similar relation involving the three components also holds for specific internal energy $E(\rho,T)$. This form of EOS has been discussed in detail and validated against experimental data on shock-Hugoniot, critical point parameters, liquid-vapor phase diagram and isobaric expansion[11]. Temperature appearing in these expressions corresponds to thermodynamic equilibrium condition in the material, and can be eliminated using the expression for specific internal energy, thereby obtaining the required form $P = P(\rho,E)$. The model can also be employed to prepare extended tables for integrating with hydrodynamics calculations.

The more commom Mie-Grüneisen EOS directly relates pressure to internal energy and density using the Grüneisen parameter[12]. Starting with the thermodynamic definitions of this parameter and constant-volume specific heat, a generalized Mie-Grüneisen EOS can be derived as[13]

$$P = P_c + \frac{1}{V}\bar{\Gamma}(V,T)(E-E_c) + \frac{1}{V}[\Gamma_e(V) - \Gamma_i(V)]\ (z_i E_{te} - z_e E_{ti}) \tag{5}$$

Here thermal electron effects are represented explicitly with separate Grüneisen parameters (Γ_i and Γ_e), a weighted average $\bar{\Gamma}(V,T)$, and constant-volume specific heats (C_{Vi} and C_{Ve}) for ions and electrons[13]. This method allows smooth switch over to electron part at high temperature. Enthalpy-based EOS treats pressure as independent variable in lieu of density[14], and is more appropriate in modeling shock propagation in porous materials as well as binary and multi-component mixtures.

B. Aspects of numerical schemes

Before focusing on the FCT algorithm, the main aspects of numerical methods for hyperbolic conservation laws are considered in this section[15]. These can be brought out by considering a scalar hyperbolic conservation equation of the form: $\partial q/\partial t + \partial f(q)/\partial x = 0$ for the field $q(x,t)$ in one space dimension. Here the non-linear function $f(q)$ specifies the flux of the quantity represented by q. The objective of the method is to integrate the conservation equation over the spatial domain of interest $[a,b]$ and a certain time domain $[0,T]$, taking in to account the boundary conditions on $q(x,t)$ at the end points and the the initial condition $q(x,0)$ at time $t = 0$. The interval $[a,b]$ is divided in to N sub-intervals of width $h=(b-a)/N$, the i^{th} interval, (denoted by I_i) being bounded by $x_{i-1/2}$ and $x_{i+1/2}$. So the end points of the domain are $x_{1/2}=a$ and $x_{N+1/2}=b$, respectively, where the boundary conditions are specified. Integration of the equation over $[x_{i-1/2},\, x_{i+1/2}]$ and the time interval $\delta = t_n - t_{n-1}$ provides the discrete form of conservation law: $\bar{q}_i^{n+1} = \bar{q}_i^n - (\delta/h)\ [\bar{f}_{i+1/2}^n - \bar{f}_{i-1/2}^n]$, where $\bar{q}_i^{n+1} = (1/h)\int_{I_i} q(x,t_{n+1})dx$ is the average of q in I_i at t_{n+1} and $\bar{f}_{i+1/2}^n = (1/\delta)\int_{t_n}^{t_{n+1}} f(x_{i+1/2},t)dt$ is the time-average of the flux over the n^{th} time interval $\delta = t_{n+1} - t_n$ at location $x_{i+1/2}$. Addition of the contributions in all spatial intervals shows that integral of q over $[a,b]$ at successive time-points differ only by the net inward flow at the end points during that time interval, thereby confirming that the discrete version indeed conserves q. The algorithms obtained above for cell average quantities like $\bar{q}_i^{n+1}$ belong to the class of **finite volume methods**. However, it is also possible to derive the same formula by using **finite difference methods** by associating $\bar{q}_i^{n+1}$ to the value of q at the mesh-center. But it is clear that the former nomenclature is more appropriate for discrete conservation laws.

For practical use of the discrete conservation law, it is necessary to express $\bar{f}_{i+1/2}^n$ in terms of the cell average *q-values* at t_n and t_{n+1}. As this can not be done exactly, various numerical methods have been proposed to effect this relation approximately. Hyperbolic equations propagate information at finite speeds, and so it is intuitively reasonable to assume that $\bar{f}_{i+1/2}^n$ can be expressed in terms of $\bar{q}_i^n$ and $\bar{q}_{i+1}^n$, which correspond to the cells on both sides of the interface $x_{i+1/2}$. So defining a **numerical flux function**, $F_{i+1/2}^n(\bar{q}_i^n, \bar{q}_{i+1}^n) = (1/\delta)\int_{t_n}^{t_{n+1}} f(x_{i+1/2},t)dt$, the discrete conservation law can be rewritten as: $q_i^{n+1} = q_i^n - (\delta/h)[F(q_i^n, q_{i+1}^n) - F(q_{i-1}^n, q_i^n)]$. (From now on, the average symbol on q will be dropped to simplify notations. The indices on F are also dropped, thereby implying that the flux function is the same for all spatial and time intervals). Methods which provide q_i^{n+1} (at t_{n+1}) explicitly in terms of q_i^n (at t_n) are called **explicit methods** . The specific example: $F(q_i^n, q_{i+1}^n) = (1/2)[f(q_i^n) + f(q_{i+1}^n)] - (1/2)(q_{i+1}^n - q_i^n)$ generates the **Lax-Friedrichs method** which provides the recursive formula: $q_i^{n+1} = (1/2)(q_{i+1}^n + q_{i-1}^n) - (\delta/2h)[f(q_{i+1}^n) - f(q_{i-1}^n)]$. Note that q_i^{n+1} is coupled to itself and two neighbors when expressed in conservation form. Another example is the **upwind method**, which corresponds to the numerical flux function: $F(v,w) = f(v)$ if $f'(v) > 0$ but $F(v,w) = f(w)$ if $f'(v) < 0$. The resulting recursive formula can be expressed as: $q_i^{n+1} = q_i^n - (\delta/h)[f(q_i^n) - f(q_{i-1}^n)]$ when $f'(q_i^n) > 0$, and a similar formula for the case $f'(q_i^n) < 0$. This also is a three-point difference formula. An appealing derivation of the upwind algorithm follows from the application of the celebrated **Godunov's method**. The idea here is to construct the numerical flux function using the solutions to a series of **Riemann problems**, which is about solving the conservation equation with specified q-values (as initial condition) on two sides of an interface[16].

It is possible to choose the numerical flux function so that it also depends on time t_{n+1}. A possible choice is: $\delta^{-1} \int_{t_n}^{t_{n+1}} f(x_{i+1/2},t)dt = (1-\varepsilon)F(q_i^n, q_{i+1}^n) + \varepsilon F(q_i^{n+1}, q_{i+1}^{n+1})$, where $0 \le \varepsilon \le 1$ is an interpolation parameter. While the choice $\varepsilon = 0$ gives the explicit method, $\varepsilon = 1/2$ provides second order accuracy in approximating the time derivative. However, q_i^{n+1} is to be determined by solving a non-linear algebraic equation for each i and time step. Such methods are called **implicit methods**, which could be computationally more expensive. Generally, hyperbolic conservation laws are solved using explicit methods.

The accuracy of discrete time variable can also be improved by dividing the time step into two parts. Then, in the first half, the q-values are updated at cell edges, for example, by employing the recursion: $\overset{*}{q}{}_{i+1/2}^{n+1/2} = (1/2)[q_{i+1}^n + q_i^n] - (\delta/2h)[f(q_{i+1}^n) - f(q_i^n)]$. Note that the first term is an approximation to $q_{i+1/2}^n$. As the quantity $\overset{*}{q}{}_{i+1/2}^{n+1/2}$ corresponds to cell edge, the numerical flux is directly given in terms of the exact flux $f(q)$ in the conservation law. In the second half, the intermediate q-values are used in the original recursive formula: $q_i^{n+1} = q_i^n - (\delta/h)[f(\overset{*}{q}{}_{i+1/2}^{n+1/2}) - f(\overset{*}{q}{}_{i-1/2}^{n+1/2})]$. This algorithm, called **Ritchmeyer-Lax-Wendoff method**, has second order accuracy with respect to time step.

A numerical method is said to be **convergent** if the difference between the exact and approximate solutions vanish when mesh width h and time step δ approach zero. The method is said to be k^{th} order accurate if the error in the approximate solution is proportional to $O(h^k + \delta^k)$. Generally it is difficult to prove convergence of a method as the number of equations to be solved tend to infinity as $(h,\delta) \to 0$. A less stringent concept, called **consistency**, is based on local truncation error. For the cases of smooth solutions, it may be easy to establish, via Taylor's series expansion, that the local truncation error is $O(h^k + \delta^k)$ and so it vanishes $(h,\delta) \to 0$. Such a method is said to be locally k^{th} order accurate and consistent.

Another important concept in numerical computations is that of **stability**. As the solution is advanced to larger time, it is necessary that the inevitable errors originating from local truncation as well as numerical round off do not accumulate and spoil the numerical solution. The method is said to be stable if this requirement is realized. Even if the method is locally consistent, the generated solution would be useless if it is not stable. Analysis of the numerical algorithms provides conditions on h and δ for ensuring stability. For hyperbolic conservation laws, this is called the Courant-Friedrichs-Lewy (CFL) condition which is expressed as: $u_{max}^n(\delta/h) \le 1$.

Here $u^n_{max} = \max_{1\leq i\leq n} |f'(q^n_i)|$ is the maximum speed of propagation of information in the interval $[t_n, t_{n+1}]$. As unstable methods are useless, this condition is of fundamental importance in numerical computations. Physically it implies that the information contained in a mesh should not propagate outside during a time step. It can also be derived using the concept of **domain of influence**[16]. As information propagates at finite speed, the initial distribution $q(x, 0)$ within a domain $\mathscr{D}$, of the region $[a, b]$, alone contributes to the solution q^{n+1}_i during $[0, t_{n+1}]$. Note that $\mathscr{D} \equiv \mathscr{D}(x_i, t_{n+1})$ as size of this domain depends on x_i and t_{n+1}. An algorithm to compute q^{n+1}_i should at least cover this domain of influence; otherwise the computed q^{n+1}_i will be devoid of the salient features of $q(x, 0)$ in $\mathscr{D}(x_i, t_{n+1})$. This basic principle limits the allowed range of δ that can be used for a given mesh width h. Then the CFL condition follows because a smaller value of δ increases $\mathscr{D}(x_i, t_{n+1})$.

III. FCT NUMERICAL ALGORITHM

Physical quantities like mass density are positive and the hyperbolic conservation laws preserve this property in the course of time. Another property preserved is the monotonic variation, which is related to the unchanging increase (or decrease) of the quantity in a certain spatial domain. These properties are more important in situations involving strong gradients in the flow like shocks or contact discontinuities. Preserving the monotonic feature avoids creation of maximum or minimum in spatial profiles due to truncation of convective derivative. It is easily verified that the upwind method maintains these properties, however, it has only first order accuracy in space derivative. So higher order methods are essential for improving accuracy, but Gudonov's theorem says that linear monotone numerical methods can at most be first-order accurate. This implies that higher order schemes have to be inherently non-linear. The FCT algorithm, developed by Boris, Book and co-workers[10], is one of such non-linear schemes, which is discussed briefly below. This algorithm has three features: a convective stage, a diffusive stage, and a flux-corrected anti-diffusive stage. While the first uses a lower order scheme, the unknowns are updated using a higher order scheme in the second stage. Certain amount of numerical diffusion is inevitable in these stages for ensuring numerical stability and positivity. In the final stage the algorithm adds anti-diffusion fluxes for reducing the smearing effects of numerical diffusion. The important aspect of the method is that the anti-diffusion fluxes are corrected such that no new maximum or minimum in the variables are generated. Advancing in time is improved using a two-step scheme wherein time-centered source terms are generated before completing the full time step. FCT algorithm is coded as FORTRAN subroutines[10] which can be easily used with a driver program and an EOS package. It also has the flexibility that either the fixed Eulerian frame or moving Lagrangian frame can be used in the simulation.

A. Features of FCT algorithm

To discuss the various features of the algorithm[17], it is enough to consider a planar conservation equation: $\partial q/\partial t + \partial(u\, q)/\partial x = 0$, where $u(x,t)$ denotes a specified velocity field. This equation belongs to the class of scalar conservation laws with $f(q) = u\, q$. The discrete version of the solution can be represented as[18]

$$q_i^{n+1} = q_i^n - \frac{\delta}{h}[F^{T\,n}_{i+1/2} - F^{T\,n}_{i-1/2}] - [F^{D\,n}_{i+1/2} - F^{D\,n}_{i-1/2}] - [F^{C\,n}_{C\,i+1/2} - F^{C\,n}_{i-1/2}]. \tag{6}$$

Average symbols on q and F are not shown even though they represent, respectively, spatial and temporal averages. The numerical fluxes F^T, F^D and F^C correspond to the convective (transport), numerical diffusion and corrected anti-diffusion contributions. Further, it is assumed that the diffusive and anti-diffusive fluxes have dimensions same as that of q, and hence the diffusion coefficients appearing in them are non-dimensional. A simple choice for convective part is $F^{T\,n}_{i+1/2} = u^n_{i+1/2}(q^n_{i+1} + q^n_i)/2$, which is just the flux at $x_{i+1/2}$ using the average value for the interface q. The diffusive contribution may be taken as the first order derivative of Fick's law: $F^{D\,n}_{i+1/2} = -\nu^n_{i+1/2}(q^n_{i+1} - q^n_i)$ where $\nu^n_{i+1/2}$ is a dimensionless diffusion coefficient. The intermediate approximation contributed by the transport flux is:

$$q_i^{t\,n} = \frac{1}{2}\varepsilon^n_{i-1/2} q^n_{i-1} + [1 - \frac{1}{2}\varepsilon^n_{i+1/2} + \frac{1}{2}\varepsilon^n_{i-1/2}]q^n_i - \frac{1}{2}\varepsilon^n_{i+1/2} q^n_{i+1}, \tag{7}$$

where $\varepsilon^n_{i+1/2} = (\delta/h)u^n_{i+1/2}$ is a dimensionless parameter. This part is written out explicitly for future reference. Now, addition of the diffusion component generates the next approximation:

$$q_i^{*\,n} = [\nu^n_{i-1/2} + \frac{1}{2}\varepsilon^n_{i-1/2}]q^n_{i-1} + [1 - \frac{1}{2}\varepsilon^n_{i+1/2} + \frac{1}{2}\varepsilon^n_{i-1/2} - \nu^n_{i+1/2} - \nu^n_{i-1/2}]q^n_i + [\nu^n_{i+1/2} - \frac{1}{2}\varepsilon^n_{i+1/2}]q^n_{i+1}. \tag{8}$$

In fact, if $\nu^n_{i+1/2} = 1/2$ for all i, this expression looks similar to that obtained in the Lax-Friedrichs method for the flux $f(q) = u\, q$. This three point formula, which is rewritten more compactly as: $q_i^{*\,n} = A^n_{i,i-1}q^n_{i-1} + A^n_{i,i}q^n_i + A^n_{i,i+1}q^n_{i+1}$, generally occurs in most of the methods. The column sum: $A^n_{i-1,i} + A^n_{i,i} + A^n_{i+1,i}$ for each i, must be unity for the method to be conservative. Further, the conditions: $A^n_{i,i-1} \geq 0$, $A^n_{i,i} \geq 0$ and $A^n_{i,i+1} \geq 0$ for each i, are sufficient for the algorithm to be positive. It is important to note that $\varepsilon^n_{i\pm1/2}$ can assume positive or negative values depending on the sign of the velocity field u. Therefore, $\pm\varepsilon^n_{i\pm1/2}$ is to be replaced by $-|\varepsilon^n_{i\pm1/2}|$ before obtaining the following bounds. Then, the first and last inequalities demand the sufficient condition: $(1/2)|\varepsilon^n_{i\pm1/2}| \leq \nu^n_{i\pm1/2}$. Next, replacing $|\varepsilon^n_{i\pm1/2}|$ with its minimum value zero in the second inequality gives the condition: $\nu^n_{i\pm1/2} \leq 1/2$. Similarly, replacing $\nu^n_{i\pm1/2}$ with its minimum value $(1/2)|\varepsilon^n_{i\pm1/2}|$ in the same inequality yields the limit: $|\varepsilon^n_{i\pm1/2}| \leq 1/2$. It is easily found that these inequities are satisfied within a triangle enclosed by the lines: $\nu = \varepsilon/2$, $\nu = (1-\varepsilon)/2$ and $\varepsilon = 0$ in the $\nu - \varepsilon$ plane. These inequalities bring out several important conclusions:

1. Non-zero values of $\nu^n_{i\pm1/2}$ are necessary for ensuring positivity, and so numerical diffusion is inevitable in methods preserving this property. Maximum amount of numerical diffusion occurs when $\nu^n_{i\pm1/2} = 1/2$ as in Lax-Friedrichs method. Minimum amount corresponding to $\nu^n_{i\pm1/2} = (1/2)|\varepsilon^n_{i\pm1/2}|$ arises in the spatially first order upwind method.

2. The condition on time step imposed via $|\varepsilon^n_{i\pm1/2}| \leq 1/2$ for positivity is twice stringent than the CFL-stability condition, namely, $|\varepsilon^n_{i\pm1/2}| \leq 1$.

3. Linear stability analysis (of Fourier modes) for constant ε and ν[17] show that the three point scheme is stable when $|\varepsilon| \leq 1$ (CFL condition) and also when $\varepsilon^2/2 < \nu$. Thus the allowed ranges of parameters: $\varepsilon^2/2 \leq |\varepsilon|/2 \leq \nu \leq 1/2$ guarantee stability and positivity.

4. One way to reduce numerical diffusion is to use $\nu \sim \varepsilon^2/2$ away from discontinuities, where positivity is not an issue, and $\nu \sim \varepsilon/2$ around strong gradients. Such a scheme is inherently non-linear as its depends on the spatial profile of q.

5. To reduce numerical diffusion further, however, with out destroying monotone property, a corrected anti-diffusion contribution is added to Eq.(6), which is discussed next.

The anti-diffusion flux is obtained using intermediate quantities $q^{t\,n}_{i+1}$, and may be approximated with anti-Fick's law, viz., $F^{A\,n}_{i+1/2} = +\mu^n_{i+1/2}(q^{t\,n}_{i+1} - q^{t\,n}_i)$. Note the difference in sign with respect to Fick's law; also that the μ's are positive anti-diffusion coefficients. This choice gives rise to flux in the positive x-direction if $q^{t\,n}_{i+1} > q^{t\,n}_i$, which is just the opposite of Fick's law. The largest value of $\mu^n_{i+1/2}$, which would not affect stability and positivity, is $\mu^n_{i+1/2} = \nu^n_{i+1/2} - (1/2)|\varepsilon^n_{i+1/2}|$.

This is simply the excess value of $\nu^n_{i+1/2}$ that will not hamper positivity. Note that $q^{n+1}_{i+1} \to q^n_{i+1}$ in regions where the flow velocity (and hence $|\varepsilon^n_{i\pm1/2}$) is negligible. By choosing the $\mu's$ and $\nu's$ same, the above definition of anti-diffusive flux ensures this limit in such regions.

To maintain monotonic property, addition of anti-diffusion contribution should not introduce additional maximum and minimum in the spatial profile . For example, if $q^{*\,n}_{i-1} < q^{*\,n}_i > q^{*\,n}_{i+1}$, that is, there is a maximum at i, anti-diffusion will increase $q^{*\,n}_i$ further. Similarly, if $q^{*\,n}_{i-1} > q^{*\,n}_i < q^{*\,n}_{i+1}$, so that there exists a minimum at i, anti-diffusion will reduce $q^{*\,n}_i$ and can make it even negative. To avoid these problem, $F^{A\,n}_{i+1/2}$ is corrected to obtain $F^{C\,n}_{i+1/2}$ which appear in Eq.(6). Thus, the final recursive formula is: $\mathbf{q^{n+1}_i} = \mathbf{q^{*\,n}_i} - [\mathbf{F^{C\,n}_{i+1/2}} - \mathbf{F^{C\,n}_{i-1/2}}]$, where the corrected flux is given by[17]

$$F^{C\,n}_{i+1/2} = S \times \max\left[0,\ \min\left(S \times (q^{*\,n}_i - q^{*\,n}_{i-1}),\ |F^{C\,n}_{i+1/2}|,\ S \times (q^{*\,n}_{i+2} - q^{*\,n}_{i+1})\right)\right] \geq 0\,, \tag{9}$$

where $S = sign(q^{*\,n}_{i+1} - q^{*\,n}_i)$. Therefore, $S = 1$ or -1, depending on $q^{*\,n}_{i+1} \geq q^{*\,n}_i$ or $q^{*\,n}_{i+1} < q^{*\,n}_i$, respectively. To understand this formula, take the case $q^{*\,n}_{i+1} > q^{*\,n}_i$ so that $S = 1$. Then there are four possibilities:

1. $q^{*\,n}_{i-1} < q^{*\,n}_i < q^{*\,n}_{i+1} < q^{*\,n}_{i+2} \Longleftrightarrow$ implies monotonic increase in $q \Longleftrightarrow$ so decrease $q^{*\,n}_i$ with out changing inequalities $\Longleftrightarrow$ now, the *'min'* operation in the above equation gives positive number $\Longleftrightarrow$ consequently $F^n_{C\,i+1/2} > 0$.

2. $q^{*\,n}_{i-1} < q^{*\,n}_i < q^{*\,n}_{i+1} > q^{*\,n}_{i+2} \Longleftrightarrow$ implies a maximum $q^{*\,n}_{i+1} \Longleftrightarrow$ so no possibility of changing $q^{*\,n}_i \Longleftrightarrow$ the *'min'* operation now gives zero $\Longleftrightarrow$ consequently $F^n_{C\,i+1/2} = 0$.

3. $q^{*\,n}_{i-1} > q^{*\,n}_i < q^{*\,n}_{i+1} < q^{*\,n}_{i+2} \Longleftrightarrow$ implies a minimum $q^{*\,n}_i \Longleftrightarrow$ so no possibility of changing $q^{*\,n}_i \Longleftrightarrow$ the *'min'* operation now gives zero $\Longleftrightarrow$ consequently $F^n_{C\,i+1/2} = 0$.

4. $q^{*\,n}_{i-1} > q^{*\,n}_i < q^{*\,n}_{i+1} > q^{*\,n}_{i+2} \Longleftrightarrow$ implies a minimum $q^{*\,n}_i$ and maximum $q^{*\,n}_{i+1} \Longleftrightarrow$ so no possibility of changing $q^{*\,n}_i \Longleftrightarrow$ the *'min'* operation now gives zero $\Longleftrightarrow$ so $F^n_{C\,i+1/2} = 0$.

Other cases can be analyzed in a similar manner. Note that the correction algorithm does not affect existing maxima and minima. Other more sophisticated algorithms for correcting the fluxes are also developed[20].

Even though the FCT algorithm is explained above with reference to a specific method, it is quite general[17] and the steps involved can be summarized as follows: (i) update the cell average q^n_{i+1} using a method (which involves convective and diffusive transport effects) that preserves positive property, (ii) define the bare anti-diffusive cell boundary fluxes using the intermediate (transport) cell average values, (iii) define corrected anti-diffusive cell boundary fluxes that ensure monotonic property of the solution, and (iv) update all the cell average solutions employing the corrected anti-diffusive fluxes.

To complete the specification of the algorithm, the diffusion coefficient $(\nu^n_{i\pm1/2})$ and anti-diffusion coefficient $(\mu^n_{i\pm1/2})$ are to be specified. This is done by Fourier analysis of the algorithm for the constant velocity problem[18,19]. Marching through a time step introduces errors in the amplitude as well as phase of a Fourier mode. The choices $\nu^n_{i\pm1/2} = (1/6)+(1/3)(\varepsilon^n_{i\pm1/2})^2$ and $\mu^n_{i\pm1/2} = (1/6)-(1/6)(\varepsilon^n_{i\pm1/2})^2$ minimize these errors to fourth order in relative wave number.

Before concluding this discussion, it is necessary to mention a more general approach to designing FCT algorithms[20]. In regions where flow variables vary smoothly, it is natural to use a high order algorithm for the flux $F^h[q^n]$ to compute q^{n+1}. Positive and monotonic properties of solutions are not issues here. However, this would generate spurious oscillations and even negative solutions in regions of strong gradients like shocks. So one is constrained to use a lower order algorithm $F^l[q^n]$, which ensures the said properties, in this region of flow. The general approach to FCT defines the *anti-diffusive* flux as the difference between these two algorithms, that is, $F^{ad}[q^n] = F^h[q^n]-F^l[q^n]$. Using its *corrected* version $F^{adc}[q^n]$, the final update q^{n+1} is obtained with the flux $F^l[q^n]+F^{adc}[q^n]$, so that both types of regions are adequately attended to.

B. Euler equations in one-dimension

For the cases of all one-dimensional geometries (planar, cylindrical and spherical) the integral form of general conservation law in Eq.(3) can be written as:

$$\frac{\partial}{\partial t}\int\limits_{\mathscr{V}(t)} G\,dV + \int\limits_{\partial\mathscr{V}(t)} G\,(u-u^g)\cdot d\vec{A} + \int\limits_{\mathscr{V}(t)} \left[\frac{1}{r^{n-1}}\frac{\partial}{\partial r}(r^{n-1}B)-C\frac{\partial D}{\partial r}\right]dV - \int\limits_{\mathscr{V}(t)} Q\,dV = 0. \quad (10)$$

The spatial coordinate is r, and u and u^g represent the fluid velocity and grid velocity along the r-coordinate. Volume and surface elements in one-dimension are: $dV = 2^{n-1}\pi r^{n-1}dr$ and $d|\vec{A}| = 2^{n-1}\pi r^{n-1}$, respectively. The index n takes values 1, 2 and 3 in planar, cylindrical, and spherical geometry, respectively. The internal source term S_{in} is explicitly written in terms of coefficient terms B, C and D. These and source rate Q are functions of r and t, and they take values provided in Table I for the three equations. Thus it is sufficient to have an algorithm to treat Eq.(10) for solving Euler equations. In fact, this form is much more general and a large class of problems can

TABLE I. coefficient and source terms in generalized continuity equation

$Equation$	G	B	C	D	Q
Mass density	ρ	0	0	0	0
Momentum density	ρu	0	-1	P	0
Energy density	ρE_t	Pu	0	0	source power

be mapped into it by defining the coefficients appropriately.

Implementation of the FCT algorithm for Euler equations uses a two-step integration in time, mentioned earlier, to reduce truncation error in time derivative. Therefore, the density G is first computed at $t_{n+1/2}$, and it has first order accuracy. Then the coefficient terms B, C, D and source term Q are recomputed at time $t_{n+1/2}$. Using these time-centered driver and souce terms, the continuity equation is integrated to time t_{n+1}. The steps just mentioned can be summarized as follows:

1. Let all the densities ρ_i^n, $(\rho u)_i^n$, P_i^n and $(\rho E_t)_i^n$ for all i be known at time t_n.

2. Next evaluate $\rho_i^{n+1/2}$, $(\rho u)_i^{n+1/2}$, and $(\rho E_t)_i^{n+1/2}$ for all i at half the time step using FCT algorithm. This step can be done successively as the equations are uncoupled.

3. Next evaluate $P_i^{n+1/2}$ using the EOS package. Temperature field $T_i^{n+1/2}$ may also be needed.

4. Now, repeat steps 2 and 3 to advance the solutions from $t_{n+1/2}$ to t_{n+1}.

C. Discrete equations

The discrete version of the Euler equations is discussed in detail[10], so a brief outline is given below for completeness. The spatial region defined in the problem is divided into N cells or volumes, as was done earlier. Eulerian and Lagrangian schemes can be developed in a general way as moving grids velocity is already introduced. Note that the cell volumes change due to moving grinds in the Lagrangian scheme. With this objective in mind, the cell interfaces are denoted as $r_{i+1/2}^{\alpha}$ $(0 \leq i \leq N)$, where the index α denotes either o (old grid) or n (new grid) at beginning or end of time step δ. Time step index is not shown explicitly unless it is essential for clarity. Interfaces $r^{\alpha}_{1/2}$ and $r^{\alpha}_{N+1/2}$ are the first and last in the computational domain, and their time dependence (if any) is part of the problem specification. These define the velocities of end point interfaces, namely, $u^{g}_{1/2} = (r^{n}_{1/2} - r^{o}_{1/2})/\delta$ and $u^{g}_{N+1/2} = (r^{n}_{N+1/2} - r^{o}_{N+1/2})/\delta$. As the other grid velocities ($u^{g}_{i+1/2}$, $1 \leq i \leq N-1$) are known, the cell interfaces at the end of δ are given by $r^{n}_{i+1/2} = r^{o}_{i+1/2} + u^{g}_{i+1/2}\, \delta$ ($0 \leq i \leq N$). Coordinates of cell centers are averages defined as: $r^{\alpha}_{i} = (r^{\alpha}_{i+1/2} + r^{\alpha}_{i-1/2})/2$ ($1 \leq i \leq N$). The cells centered at r_0 and r_{N+1} - known as fictitious guard cells or ghost cells - lie outside the actual domain.

The interface areas $A_{i+1/2}$, which are averages over time, are defined as: 1, $\pi(r^{o}_{i+1/2} + r^{n}_{i+1/2})$ and $(4/3)\pi[(r^{o}_{i+1/2})^2 + r^{o}_{i+1/2}r^{n}_{i+1/2} + (r^{n}_{i+1/2})^2]$ in planar, cylindrical, and spherical geometries, respectively. This averaging ensures that a constant density profile remains unaltered in a static fluid. The corresponding volume elements V^{α}_{i} are given by: $(r^{\alpha}_{i+1/2} - r^{\alpha}_{i-1/2})$, $\pi[(r^{\alpha}_{i+1/2})^2 - (r^{\alpha}_{i-1/2})^2]$ and $(4\pi/3)[(r^{\alpha}_{i+1/2})^3 - (r^{\alpha}_{i-1/2})^3]$. At the beginning of the time step, the interface density $G^{o}_{i+1/2}$ and fluid velocity $u_{i+1/2}$ are defined as density-weighted averages: $G^{o}_{i+1/2} = (\rho^{o}_{i}G^{o}_{i+1} + \rho^{o}_{i+1}G^{o}_{i})/(\rho^{o}_{i} + \rho^{o}_{i+1})$ and $u_{i+1/2} = (\rho^{o}_{i}u^{o}_{i+1} + \rho^{o}_{i+1}u^{o}_{i})/(\rho^{0}_{i} + \rho^{o}_{i+1})$, respectively. Other weighting schemes, such as arithmetic averages, can also be used[10]. The endpoint fluid velocities $u_{1/2}$ and $u_{N+1/2}$ are specified as boundary conditions. Similarly, G^{o}_{0} and G^{o}_{N+1} for guard cells are also specified via boundary conditions. Relative velocity of the fluid with respect to the grid, defined by $U_{i+1/2} = u_{i+1/2} - u^{g}_{i+1/2}$ ($0 \leq i \leq N$), completes all definitions.

It is now straightforward to derive the discrete formulas from the integral form of conservation law in Eq.(10). Pure convective part of transport component G^{*}_{i} is given by:

$$V^{o}_{i}G^{*}_{i} = V^{o}_{i}G^{o}_{i} - \delta[A_{i+1/2}U_{i+1/2}G^{o}_{i+1/2} - A_{i-1/2}U_{i-1/2}G^{o}_{i-1/2}],\ 1 \leq i \leq N \tag{11}$$

Here first term is from beginning of time step and the remaining are due to transport across the interfaces. Note that the relative velocity $U_{i\pm 1/2}$ are used in the convective terms. Due to the occurrence of spatial derivatives, the internal source contributions are readily expressed in terms of interface areas. Addition of these terms provides the transported density G^{T}_{i}:

$$\begin{aligned} V^{o}_{i}G^{T}_{i} &= V^{o}_{i}G^{*}_{i} + \delta[A_{i+1/2}B_{i+1/2} - A_{i-1/2}B_{i-1/2}] + \\ &+ \frac{\delta}{4}C_i(A_{i+1/2} + A_{i-1/2})[D_{i+1} - D_{i-1}] + \delta(V^{0}_{i}Q_i). \qquad 1 \leq i \leq N \end{aligned} \tag{12}$$

Note that the change in cell volume is not introduced here. Values of pressure at end points, $P_{1/2}$ and $P_{N+1/2}$, which are needed in $B_{1/2}$ and $B_{N+1/2}$, are to be specified externally. Further, as the term $(\partial D/\partial r)_i$ is approximated using the difference formula $(D_{i+1} - D_{i-1})$, pressures P_0 and P_{N+1} corresponding to guard cells are needed for D_0 and D_{N+1}. Next contribution is from numerical diffusion flux defined as $F^D_{i+1/2} = -\nu_{i+1/2} V_{i+1/2}(G^T_{i+1} - G^T_i)$. The negative sign is as in Fick's law, and the dimensionless positive diffusion coefficient $\nu_{i+1/2}$ is to be specified. Average volume $V_{i+1/2}$ corresponding to the interface, which makes the diffusion coefficient dimensionless, is defined by: $V^n_{i+1/2} = (1/2)(V^n_{i+1} + V^n_i)$, and $V_{1/2} = V^n_1$ and $V_{N+1/2} = V^n_N$ at the end points. Addition of the net diffusion term, $-[F^D_{i+1/2} - F^D_{i-1/2}]$, to G^T_i provides the transported and diffused density $\bar{G}_i$ given by:

$$V^n_i \bar{G}_i = V^o_i G^T_i + [\nu_{i+1/2} V_{i+1/2}(G^o_{i+1} - G^o_i) - \nu_{i-1/2} V_{i-1/2}(G^o_i - G^o_{i-1})], \qquad 1 \le i \le N \tag{13}$$

Change of volume V^o_i to V^n_i in time step δ, due to grid motion, is accounted at this stage.
Next quantity is the anti-diffusion flux defined as $F^{AD}_{i+1/2} = \mu_{i+1/2} V_{i+1/2}(G^T_{i+1} - G^T_i)$, where $\mu_{i+1/2}$ are positive, dimensionless, anti-diffusion coefficients. By this definition, $F^{AD}_{i+1/2}$ is along the positive r-direction when $(G^T_{i+1} - G^T_i) > 0$, which is just the opposite of Fick's law. As discussed earlier, this flux can not be used directly, it has to be corrected to avoid the creation of new maximum and minimum. The corrected anti-diffusion flux is given by:

$$F^{CAD}_{i+1/2} = S \times \max\left[0,\, \min\left(S \times V^n_{i+1}(\bar{G}_{i+2} - \bar{G}_{i+1}),\, |F^{AD}_{i+1/2}|,\, S \times V^n_i(\bar{G}_i - \bar{G}_{i-1})\right)\right],\, 0 \le i \le N \tag{14}$$

where $S = sign(\bar{G}_{i+1} - \bar{G}_i)$. Terms proportional to $(\bar{G}_0 - \bar{G}_{-1})$ and $(\bar{G}_{N+2} - \bar{G}_{N+1})$ are to be dropped from $F^{CAD}_{1/2}$ and $F^{CAD}_{N+1/2}$, respectively. Finally, the density G^n_i at the end of δ is given by

$$V^n_i G^n_i = V^n_i \bar{G}_i - [F^{CAD}_{i+1/2} - F^{CAD}_{i-1/2}],\ 1 \le i \le N \tag{15}$$

It is necessary to generalize the definition of $\varepsilon_{i+1/2}$ to make use of the expressions for diffusion and anti-diffusion coefficients[10]. The discrete form of convective terms in Eq.(11) suggests that the mesh width around the interface at $r^n_{i+1/2}$ may be obtained as $(1/h)_{i+1/2} = (1/2)\, A_{i+1/2}/V^n_{i+1} + (1/2)\, A_{i+1/2}/V^n_i$. This leads to the definition $\varepsilon_{i+1/2} = \delta\, U_{i+1/2}(1/h)_{i+1/2}$, which in turn leads to the expressions:

$$\nu_{i+1/2} = \frac{1}{6} + \frac{1}{3}\,\varepsilon_{i+1/2}, \quad \mu_{i+1/2} = \frac{1}{6} - \frac{1}{6}\,\varepsilon_{i+1/2}, \tag{16}$$

$$\varepsilon_{i+1/2} = \delta\, U_{i+1/2}\, A_{i+1/2}\, (1/2)\left(1/V^n_{i+1} + 1/V^n_i\right), \qquad 0 \le i \le N. \tag{17}$$

The complete set of discrete equations, applicable with arbitrary grid velocity, and hence Eulerian and Lagrangian schemes, are covered in Eqs.(11) to (17).

D. Boundary conditions

Different types of boundary conditions, depending on the physics of the problem, are used to specify the nature of fluid flow at boundaries of the domain. These boundary conditions should specify expressions to compute G_0^o and G_{N+1}^o in the guard cells. Straightforward specifications for **free boundaries** are: $G_0^o = G_1^o$ and $G_{N+1}^o = G_N^o$, where G abbreviates ρ, (ρu) and (ρE_t). This automatically defines their end point interface values, and the fluid velocities $u_{1/2} = u_1^o$ and $u_{N+1/2} = u_N^o$. In addition, pressure values $P_0 = P_{1/2} = 0.1\ MPa$ and $P_{N+1} = P_{N+1/2} = 0.1\ MPa$ are also specified at free boundaries.

For the case of **rigid boundary** at $r_{1/2}^o$, like that at the center of a sphere, the specifications are: $\rho_0^o = \rho_1^o$, $(\rho E_t)_0^o = (\rho E_t)_1^o$, while $(\rho u)_0^o = -(\rho u)_1^o$ for momentum density. This leads to the physical requirement that $u_{1/2} = 0$. Further, the end point values of pressure are assigned as $P_0 = P_{1/2} = P_1$. Similar relations apply if $r_{N+1/2}^o$ is a rigid boundary. Much more general boundary conditions are also applicable within the FCT subroutine packages distributed by its authors[10].

IV. APPLICATIONS

A few strong shock problems are simulated using the the FCT algorithm and typical numerical results are discussed in this section. Two of these are benchmark problems while the remaining five are experimental cases involving projectile impact and explosive detonation.

Shock wave generation via projectile (impactor) impact is a common setup in shock wave physics. Such an impact generates two waves at the interface of impactor and target, a forward moving wave into the target and one traveling back to the projectile. Fluid in the target are accelerated and attain a steady velocity eventually. Shock wave moves in the target with its own speed, called shock velocity U_s, while fluid follows with velocity U_p. A typical experiment consists of measuring U_p and U_s, and the entire U_p - U_s curve is generated from a series of experiments. In other experiments pressure versus time profile is measured at some locations in the target. Measurement of release velocity-time profile, when shock breaks out at the rear surface of the target, is also done. Strong shocks are also generated by detonating explosive material in contact with the target[21]. Detonation process in explosives sustains the shock via releasing chemical energy at the shock front. The explosive-shock impinging on target acts like a piston at the target interface, thereby generating a shock that is characteristic of the medium.

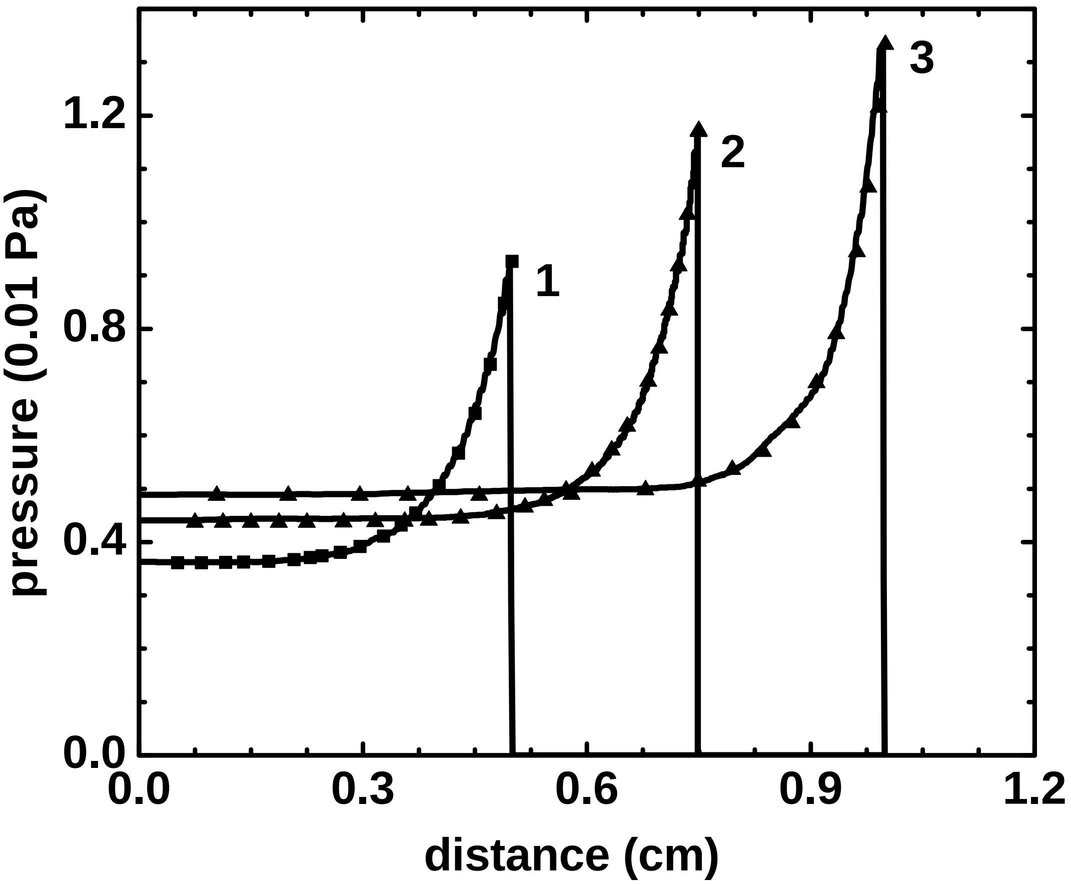

FIG. 1. Numerical and analytical results of pressure for the Sedov-von Neumann-Taylor blast wave problem. Curves 1, 2 and 3 correspond to planar, cylindrical and spherical geometries (see text). Spatial profiles of pressure are obtained at 1 s and symbols denote analytical results[22]. Good agreement found show the accuracy of FCT algorithm to resolve shock wave profiles.

A. Sedov-von Neumann-Taylo benchmark

In a typical sweep over the meshes, the FCT algorithm generates spatial profiles of density, fluid velocity and internal energy. A new profile of pressure is then generated to advance to the next time step. An ideal gas EOS, with specific heat ratio $\gamma = 1.4$, is used for this benchmark problem which deals with outward shock propagation in either planar, cylindrical or spherical geometry. The process is initiated on depositing internal energies: 0.0673185, 0.311357 and 0.851072 (in ergs)

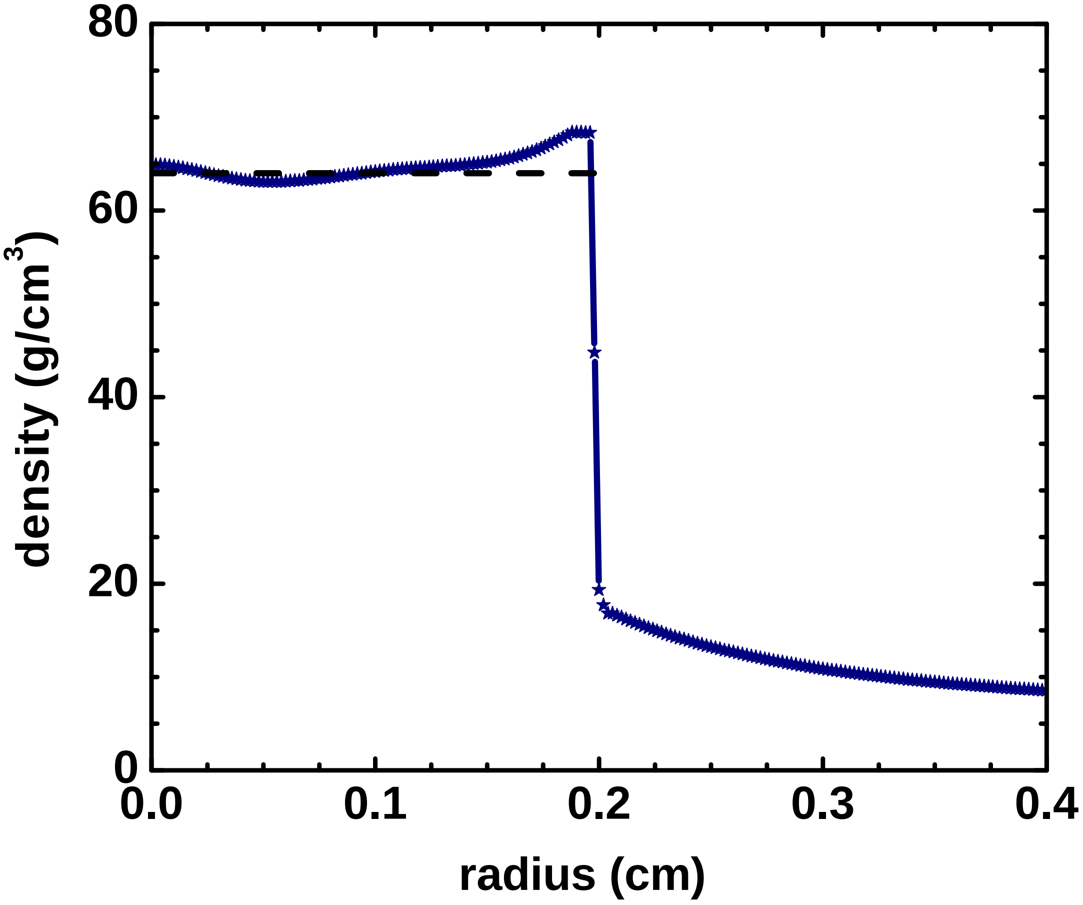

FIG. 2. Numerical and analytical results for density profile in the spherical Noh problem (see text). The density profile is obtained at = 0.6 μs and the dashed line represents the analytical result (64 g/cm^3)[23]. Slight discrepancy of the profile in the shocked region is noted, even though the shock speed compares well with analytical result $(0.333\ \mu s)$[23].

at $t = 0$ in a small volume - first mesh in the simulations - at the center in the three geometries, respectively[22]. Analytical solution of the problem shows that the shock travels distances: 0.55, 0.75 and 1.0 cm in 1 second in the three cases[22]. The simulations used the Eulerian reference frame with 300 meshes of equal volume within the above regions, and a constant time step of 10^{-6} second . Numerical results obtained are compared with the **analytical density profiles** at 1 second in Fig.1. Excellent agreement found brings out the accuracy of the algorithm in capturing shock profiles and shock speed. Similar comparisons are found for pressure and velocity profiles[13].

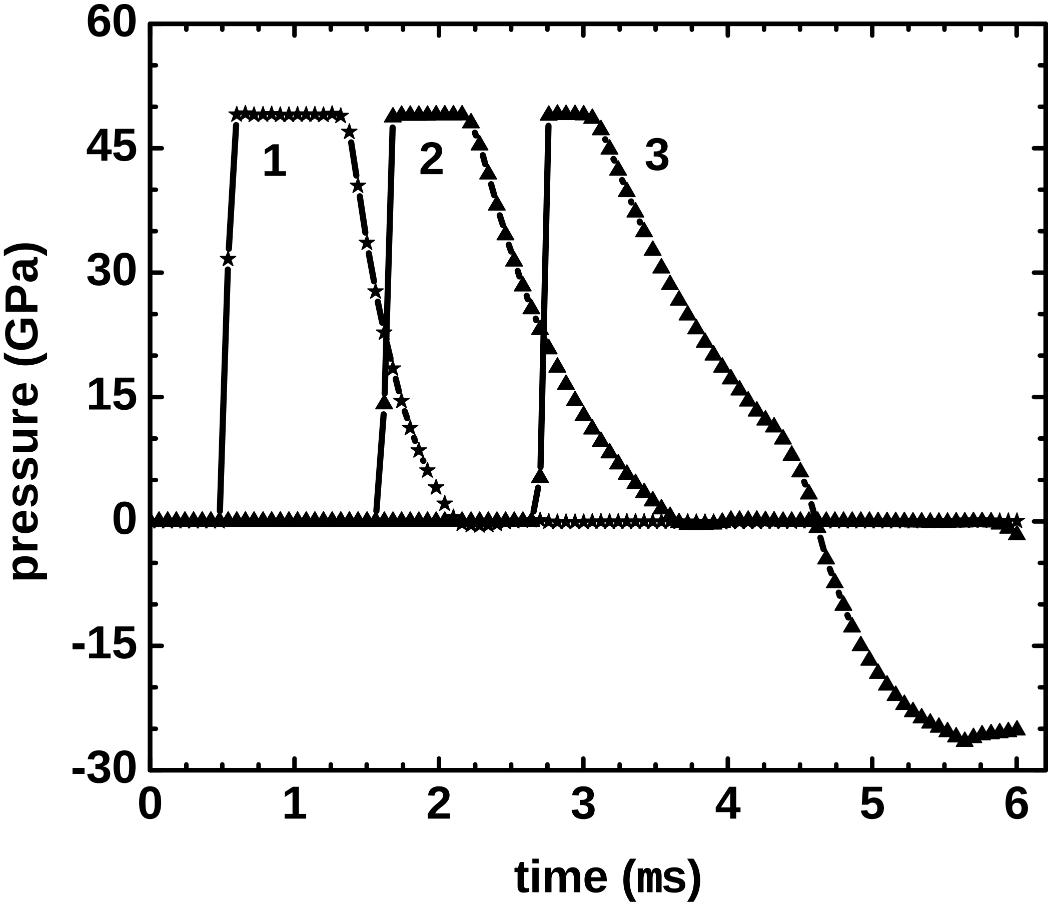

FIG. 3. Propagation of shock wave in solid Cu of 20 mm thickness, impacted with 3 mm thick Cu-impactor with velocity 2 km/s. Curves 1, 2 and 4 correspond to pressure-time profiles at 3, 9 and 15 mm from the interface. The peak pressure remains unaltered even up to 15 mm. Negative pressure occur at 15 mm at late times because of the release wave reaching back from the rear surface.

B. Noh benchmark problem

Noh benchmark problem deals with generation of shock at a rigid boundary due to an imploding fluid. The simulation starts with the specification of inward velocity of 1 cm/μs at all meshes in a spherical medium of radius 1 cm. The imploding material gets reflected at the center of the sphere and generates a shock moving outward into the imploding fluid. Solution to Noh problem corresponds to a stagnation shock where all the fluid variables are stationary behind the shock. Analysis of Euler equations show that these specifications are compatible only for certain special forms of EOS, like the ideal gas EOS[23]. Analytical expressions for the fluid variables, behind and ahead of the shock, are then obtained for such types of EOS. Using the ideal gas EOS, with $\gamma = 5/3$ and initial density 1 gm/cm^3, the simulation used 500 meshes over 1 cm radius and a time step of 10^{-6} second. The spatial density profile so generated at 0.6 μs is compared with analytical result (64 g/cm^3) in Fig.2. Slight discrepancy in the density-profile behind the shock is observed, even though the shock speed compares well with the analytical value 0.333 cm/μs.

C. Shock propagation in solid Cu

Numerical simulation of shock propagation in solid Cu is considered next. A 20 mm thick Cu target is impacted with 3 mm thick Cu impactor with 2 km/s velocity. This impact generates a fluid velocity U_p of 1 km/s in the target because the inelastic collision between impactor and target (which are same material) halves the fluid velocity. The hydrodynamic motion so initiated is then followed to capture pressure versus time profiles at 3, 9, and 15 mm from the target surface. Simulations used the EOS package discussed earlier, and the profiles displayed in Fig.3 show a steady shock propagating in the target. Note that the peak pressure of the pulses $\sim$ 49 GPa remain almost the same even after traveling 15 mm. However, width of the pulses decreases because of energy spend in compressing the shocked material. Sharp variations of pressure are accurately captured by the FCT algorithm. Shock speed of $\sim$ 5.5 km/s deduced from the position of pulses agrees very well with that given by the empirical relation $U_s = 4.14 + 1.408\ U_p$ for Cu. Magnitude of shock pressure, that can be computed using the Rankine-Hugoniot relation $P = \rho_0 U_s U_p$, also agrees with the simulation result. Negative values of pressure are observed at 15 mm at late time. This is due to the release wave reaching back from the rear surface of the target. FCT algorithm has also been used for simulating shock attenuation in porous materials[13].

D. Pressure-time profile in soild Cu

Pressure versus time profiles at different locations in the target are usually measured in impact experiments using impedance matching technique[12]. The experiment considered here employs Cu-impactor (thickness 2.034 cm) with impact velocity 0.645 km/s and a Cu-target (thickness 18.28 cm). Pressure-time profile is measured at the middle of the target[24]. This type of comparison of experimental data with simulation results is useful to check the accuracy of the FCT algorithm

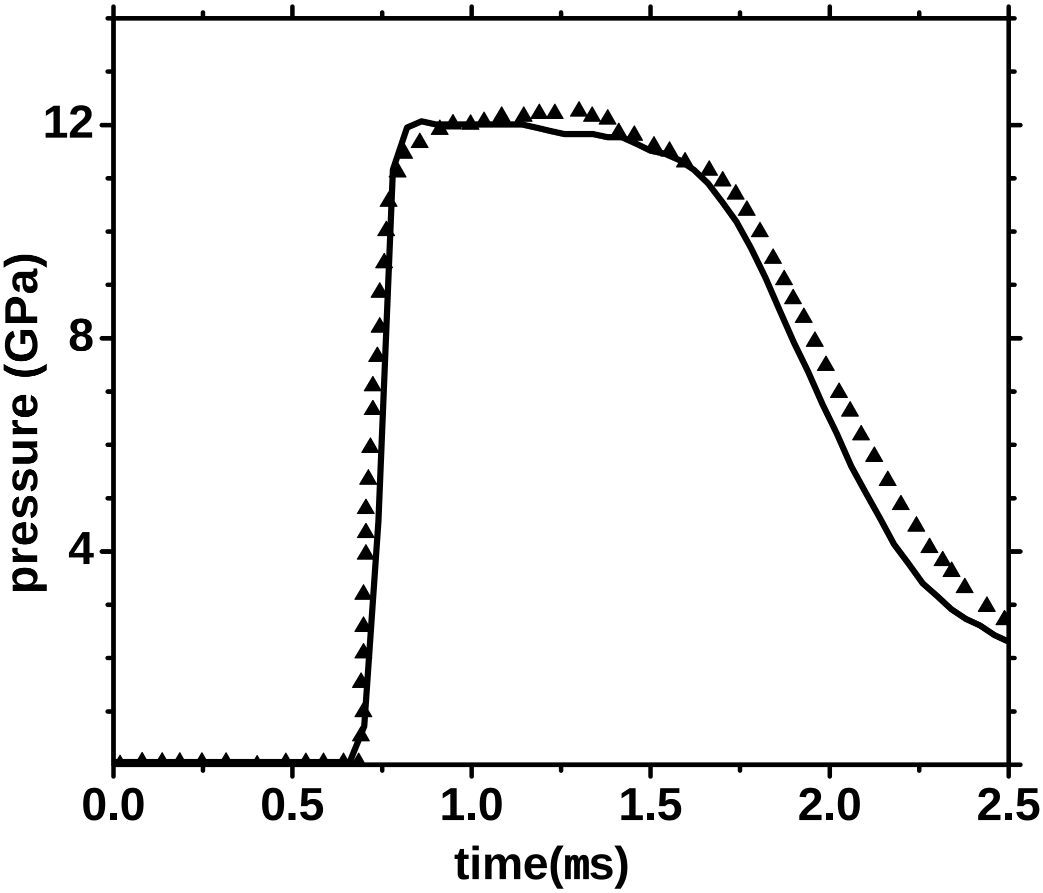

FIG. 4. Pressure versus time for Cu-impactor (thickness 2.034 mm and impact velocity 0.645 km/s) on Cu-target (thickness 18.38 mm). Solid line shows simulation results at the center of the target while the symbols denote experimental data[24]. Agreement is good, however, the slight disagreement at later times is due to the omission of material strength effects

.

as well the EOS model. Simulation starts with a configuration where the impactor and target plate are kept side by side; with the impact velocity specified in all meshes within the impactor. The resulting hydrodynamic flow is followed and the pressure-time profile given in Fig.4 show excellent agreement. The algorithm is able to capture the sharp initial pressure increase as well as subsequent variations in time. Peak pressure and shock duration are accurately modeled, although slight differences are noted towards the end. This discrepancy could be attributed to the omission

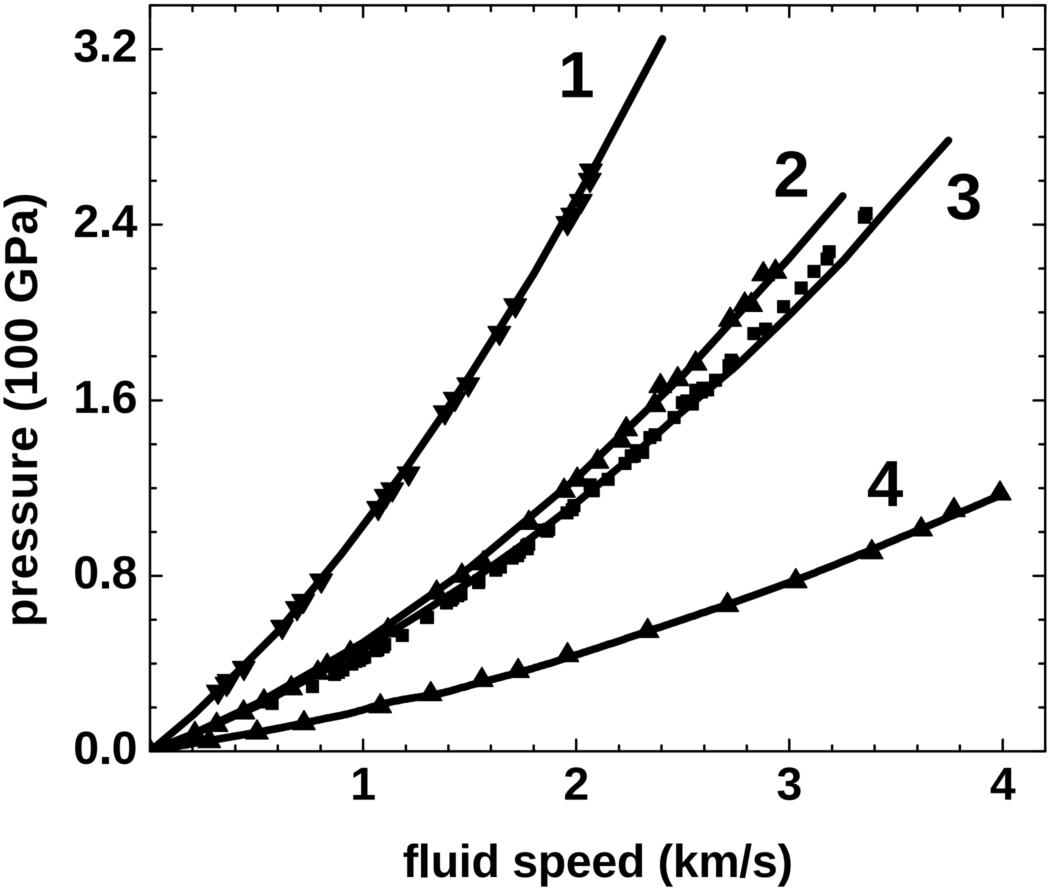

FIG. 5. Shock pressure versus fluid speed for W (curve-1), Cu (curve-2), Fe (curve-3) and Al (curve-4). Solid lines are simulation results while symbols denote experimental data[25]. Excellent agreement obtained, over this wide range of fluid velocities and materials, show the accuracy of the FCT algorithm and EOS model.

of material strength effects in the EOS model, which are essential at low pressures.

E. Shock pressure versus fluid speed

A more stringent check on the simulation method and EOS can be done by comparing experimental data on fluid speed (U_p) versus shock pressure (P). This type of database is extensive; it covers a wide of metals, compounds, mixtures and explosives[24]. As discussed earlier, simulations of impact experiments can be done at different impact velocities to generate U_s versus U_p curves

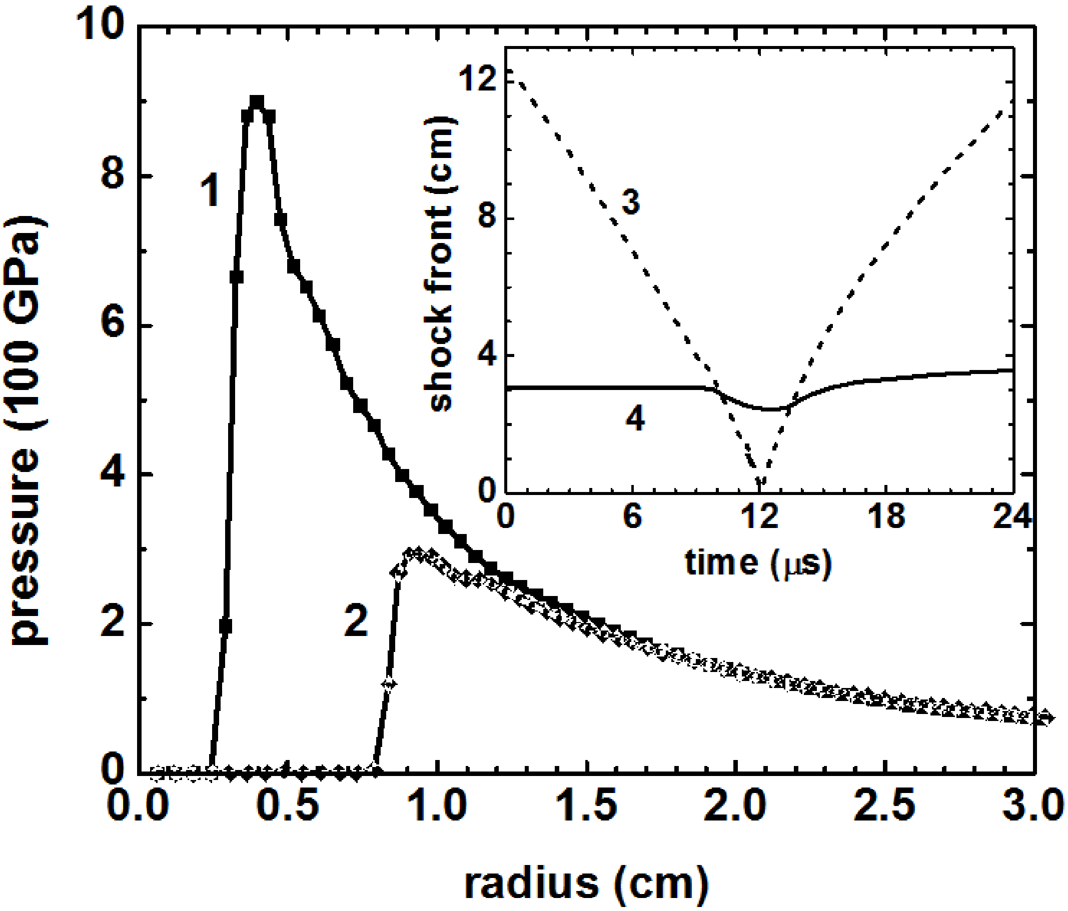

FIG. 6. Explosive PBX-9404 (outer radius 12.319 cm) driven imploding shock in Al-sphere (radius 3.048 cm)[21]. Spatial pressure-profiles are shown at time 12.8 μs (curve-1) and 12.5 μs (curve-2) after initiation. The imploding shock builds up due to spherical convergence effects. The insert graph shows the incomming shock front (curve-3) and Al-outer radius (curve-4) versus time.

or, more easily, P versus U_p curves. By choosing same material for impactor and target, it is easy to verify the relation $U_p = (1/2)U_{impact}$ within the simulation. It is essential to employ sufficiently small time step and mesh width so that a steady shock is established in the target, and peak pressure in the shock and the corresponding fluid velocity can be read off. Numerical results generated for four materials (W, Cu, Fe, Al), with impact velocities $\sim$ 8 km/s, are compared with experimental data in Fig.5. Excellent agreement obtained, once again, demonstrates the the accuracy of FCT algorithm as well as the EOS. Similar results have also been obtained in porous materials[13].

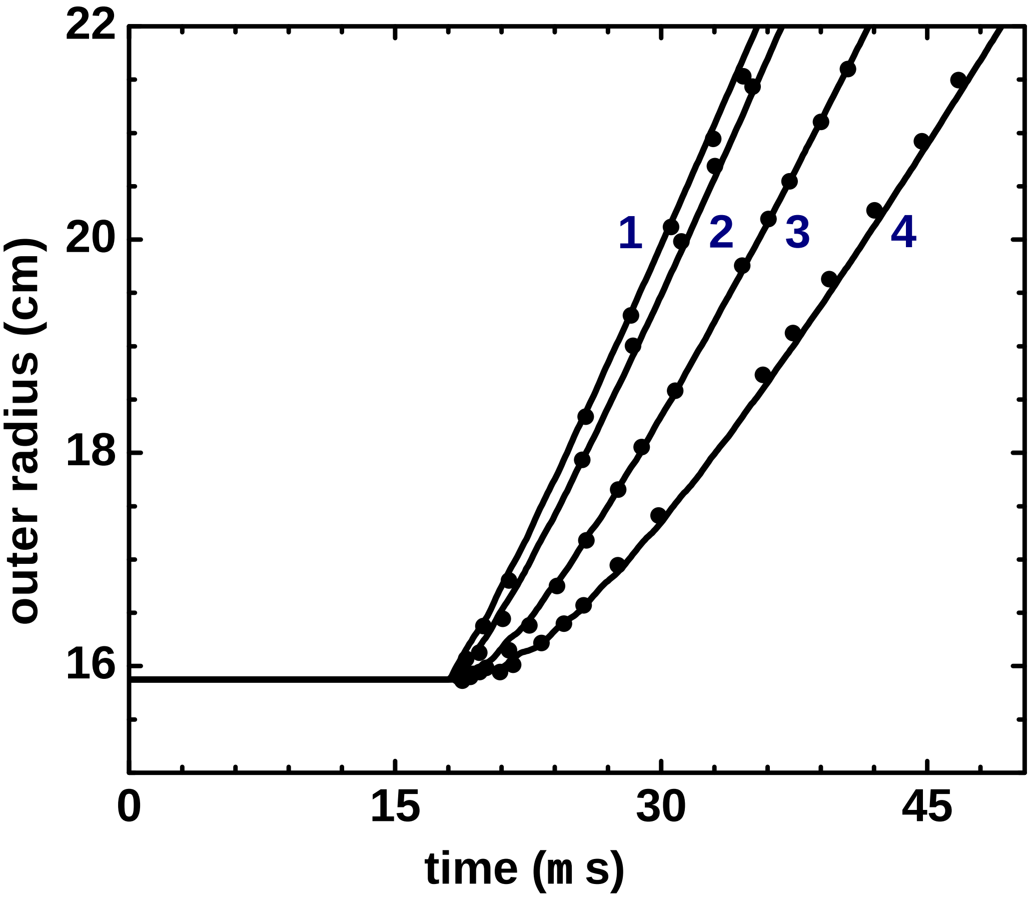

FIG. 7. Simulation results for explosive (PBX-9404) driven outward moving shock in Al-shell (inner radius=15.24 cm, outer radius=15.875 cm). Outer shell-radii versus time are shown for four experimental configurations corresponding to different gap thicknesses: 0.0 (curve-1), 0.5 (curve-2), 2.0 (curve-3) and 3.81 cm (curve-4), between explosive surface and Al-shell (see text). Lines are simulation results while symbols denote experimental data[26].

F. Imploding shock in Al sphere

Next experiment considered for simulation involves an explosive driven imploding shock generated in Al[21]. The arrangement consists of an Al-sphere of radius 3.048 cm surrounded by explosive PBX-9404 of outer radius 12.319 cm. Detonation of the explosive at its outer surface generates a converging shock which impinges on Al-sphere. This imploding shock compresses Al, and also gets amplified as it moves inward due to spherical convergence effects. Position of the shock front and Al-explosive interface have been measured at few time points. Simulation of the set up is done using the JWL model for the burned explosive products[26]. Spatial profiles of pressure at time 12.5 μs and 12.8 μs after initiation are shown in Fig.6. These demonstrate the build up of the shock due to spherical convergence. The incomming shock front and Al-outer radius versus time are also shown (inset graph) in this figure.

G. Exploding shock in Al shell

The last experiment chosen for comparison involves an outgoing spherical shock in Al-shell (inner radius 15.24 cm and outer radius 15.875 cm). Explosive PBX-9404 is now a sphere at the center of the shell, and four experimental configurations used correspond to different explosive radii, and hence gap thicknesses (0, 0.5, 2.0 and 3.81 cm) between the explosive surface and Al-shell inner surface[26]. On initiating the detonation at the center, first a shock wave moves outward in the explosive which subsequently breaks out at its surface. The burned explosive products at high pressure expand and impinge on the Al-shell, thereby launching another shock wave in the shell. The shell is compressed but also expands when the shock breaks out at its outer surface. Experimental data on the shell (outer) radius versus time and simulation results for the four cases are shown in Fig.7. Excellent comparison found shows that the FCT algorithm, together with the EOS models, generate accurate radius versus time profiles. Delayed expansion for larger gaps and consequent variations in the breakout velocity are all accurately captured in the simulations.

V. SUMMARY

The main aim in this chapter has been to discuss the FCT algorithm for solving one dimensional Euler equations. General aspects of numerical schemes for conservation laws and the discrete set of equations have been covered in some detail. An EOS model suitable for integration with the algorithm is also outlined. The FCT algorithm has several desirable features: it has second order accuracy in approximating space and time derivatives; preserves monotone and positivity properties of fluid variables; also reduces numerical diffusion thereby minimizing smearing of shock wave profiles. Further, there is no need to add artificial viscosity to accommodate steep gradients in flow variables. Results of several simulations using the algorithm are also discussed. These include two benchmark problems and five experimental cases involving plate impact and explosive driven shocks.

Acknowledgment

One of us (B N) thank Dr. P. Modak (High Pressure and Synchroton Radiation Physics Division, BARC) for his encouragement and support for this work.

REFERENCES

[1] Zel'ovich Ya B and Raizer Yu P, *Physics of Shock waves and High-Temperature Hydrodynamic Phenomena, volume-I* (Academic press, New York) (1966).

[2] https://lasers.llnl.gov/about, *About NIF and Photon Science*, Lawrence Livermore National Laboratory.

[3] Davison L, *Fundamentals of shock wave propagation in solids* (Springer-Verlag, Berlin, Heidelberg), (2008).

[4] Isbell W M, *Shock waves: Measuring the dynamic response of materials, volume-I* (London: Imperial College press) (2005).

[5] Sod G A , *A servey of several finite difference methods for systems on nonlinear hyperbolic conservation laws*, J. Comput. Phys. **27**, 1-31 (1978).

[6] Von Neumann J and Richtmyer R D, *A method for numerical calculation of hydrodynamic shocks* , J. Appl. Phys. **21**, 232-237 (1950).

[7] J. P. Boris and D. L. Book, *Flux-corrected transport I: SHASTA, a fluid transport algorithm that works*, J. Comput. Phys. **11**, 38-69 (1973).

[8] Book D L, Boris J P and Hain K , *Flux-corrected transport II: Generalizations of the method* , J. Comput. Phys. **18** 248-293 (1975).

[9] D. Kuzmin, R. Lӧhner, S.Turek, (eds.) *Flux-Corrected Transport: Principles, Algorithms and Applications*, (Scientific Computation, Springer Science+Business Media, Dordrecht) (2012)

[10] Boris J P, Landsberg A M, Oran E S and Gardner J H, *LCPFCT – Flux-Corrected-Transport Algorithm for solving generalized continuity equations*, NRL/MR/6410-93-7192 (1993)

[11] Menon S V G and Bishnupriya Nayak, *An equation of state for metals at high temperature and pressure in compressed and expanded volume regions*, Condensed Matter Journal, MDPI **4**, 71 (2019).

[12]Meyers M A, *Dynamic behavior of material* (Wiley-Inter Science Publication, New York) (1994).

[13]Nayak B and Menon S V G, *Numerical solution of Euler equations employing enthalpy-based equation of state for simulating shock propagation in porous materials* , Mater. Res. Express **6**, 055514 (2019).

[14]Nayak B and Menon S V G, *Enthalpy-based equation of state for highly porous materials employing modified soft sphere model*, Physica B: Phys Cond. Matter **529**, 66-72 (2017)

[15]LeVeque R J, Mihalas D, Dorfi E A and Mller, Computational methods for astrophysical fluid flow, Eds. Steiner O and Gautschy A (Springer-Verlag, Berlin, Heidelberg, New York), (1998).

[16]LeVeque R J, *Finite volume methods for hyperbolic problems* (Cambridge University Press, UK) (2002).

[17]Oran E S and Boris J P, *Numerical simulation of reactive flow* (Cambridge University Press, UK) (2001).

[18]Odstrcil D, *A new optimized FCT algorithm for shock wave problems*, J. Comput. Phys. **91**, 71-93 (1990).

[19]J. P. Boris and D. L. Book, *Flux-corrected transport. III. Minimal-error FCT algorithms*, J. Comput. Phys. **20**, 397-431 (1976).

[20]Zalesak S T, *Fully multidimensional flux-corrected transport algorithms for fluids*, J. Comput. Phys. **31** 335-362 (1979).

[21]Mader C L, *Numerical modeling of explosives and propellants*, p.107, p.378 (CRC Press, Taylor and Francis group, Boca Raton, London, New York) (2008).

[22]Kamm J R, *Evaluation of Sedov-von Neumann-Taylor blast wave solution* , LA-UR-00-6055 (2000).

[23]Ramsey S D, Boyd Z M and Brunett S C , *Solution of the Noh problem using universal symmetry of gas dynamics equations*, Shock Waves **27**, 477-485 (2017).

[24]Morris C E (Ed.) *Los Almos Shock Wave Profile Data* (University of California Press, London) (1982).

[25]Marsh S P, *LASL Shock Hugoniot Data* (University of California Press, California) (1980)

[26]Wilkins M K, *Computer simulation of dynamic phenomena*, p.78 (Springer-Verlag, Berlin, heidelberg, New York), (1999).