Title

An embedded boundary method for viscous, conducting compressible flow

Permalink
https://escholarship.org/uc/item/07p982x3

Journal
Journal of Computational Physics, 216(1)

ISSN
0021-9991

Authors
Dragojlovic, Z
Najmabadi, F
Day, M

Publication Date
2006-07-01

Peer reviewed

An Embedded Boundary Method for Viscous, Conducting Compressible Flow

Zoran Dragojlovic[a,1], Farrokh Najmabadi[a] and Marcus Day[b]

[a] *Center for Energy Research, Department of Electrical & Computer Engineering,*
University of California in San Diego, 9500 Gilman Drive, La Jolla, CA 92093-0417

[b] *Center for Computational Sciences and Engineering, Lawrence Berkeley National Laboratory,*
One Cyclotron Road Berkeley, CA 94720-8142

Abstract

The evolution of an Inertial Fusion Energy (IFE) chamber involves a repetition of short, intense depositions of energy (from target ignition) into a reaction chamber, followed by the turbulent relaxation of that energy through shock waves and thermal conduction to the vessel walls. We present an algorithm for 2D simulations of the fluid inside an IFE chamber between fueling repetitions. Our finite-volume discretization for the Navier-Stokes equations incorporates a Cartesian grid treatment for irregularly-shaped domain boundaries. The discrete conservative update is based on a time-explicit Godunov method for advection, and a two-stage Runge-Kutta update for diffusion accommodating state-dependent transport properties. Conservation is enforced on cut cells along the embedded boundary interface using a local redistribution scheme so that the explicit time step for the combined approach is governed by the mesh spacing in the uniform grid. The test problems demonstrate second-order convergence of the algorithm on smooth solution profiles, and the robust treatment of discontinuous initial data in an IFE-relevant vessel geometry.

Key words:
embedded boundary, Godunov, convergence analysis, Cartesian grid, redistribution, inertial fusion energy

[1] Corresponding author. Tel.:+858-534-7842, *Email address: zoran@fusion.ucsd.edu*

1 Introduction

In inertial confinement fusion scenarios, a small target containing frozen deuterium and tritium (DT) is compressed and heated to fusion temperatures by powerful laser or particle beams. With the careful design of the target and beam pulses, a fusion burn can be initiated in the target, which subsequently releases large quantities of energy into its surrounding environment. The energy must then be removed from the chamber housing the target and beam lines, and the environment inside the chamber returned to a quiescent state so that a new fusion target may be positioned for the next cycle. In an inertial fusion energy (IFE) system, it is envisioned that this process will be repeated at a frequency of approximately 5-10 Hz. In order to analyze and design such systems, it is therefore necessary understand the time-dependent response of the chamber environment between target ignitions over these time scales.

There are two phases that characterize the IFE chamber evolution and relevant energy transfer. The IFE target is ignited essentially instantaneously (over less than a nanosecond) and its energy is released in the chamber in the form of X-rays (mainly bremsstrahlung radiation), neutrons (fusion products), and ions (both fusion products and target debris). The first phase of chamber evolution, lasting up to a few microseconds, includes the transport of this energy through the chamber fill gas toward the vessel walls. The fill gas absorbs a portion of the radiated energy, helping to reduce the peak energy flux transmitted to the vessel wall. During this process, peak temperatures in the chamber may reach 1 keV or more, but then quickly decay to a few eV predominantly through radiative transport processes. A significant fraction of the energy is dissipated as well through slower hydrodynamic processes, such as shock waves, which propagate throughout the chamber. This first phase after ignition is known as the "fast phase", and is immediately followed by a much longer period of time, the "slow phase" over which convection and shock hydrodynamics distribute energy throughout the chamber and to the vessel wall. The slow phase extends for 100-200 ms or longer until the next target is inserted and ignited to initiate the process over again, and is the focus of the present study.

Because of the vastly disparate time scales and dominant physical phenomena, studies of the heat transport problem for IFE target chambers may be similarly separated into the two phases discussed above. The environment in the slow-evolution phase has a sufficiently low temperature to allow reasonable modelling with the compressible Navier-Stokes equations. Due to the longer time scales involved however, multidimensional geometry effects become important in this phase as the fluid interacts with the vessel wall containing various beam access ports.

An extensive overview of approaches to modeling the IFE chamber physics is given in Ref. [1]. Several computer codes have been used to model the chamber environment behavior. BUCKY is a one-dimensional radiation-hydrodynamics code [2] that has been used to model blast-wave propagations through IFE chambers, laser-ablation-driven shocks in gases, and X-ray-driven shocks in solids. BUCKY results are used in this study as a model for the fast-phase physics, and to provide initial conditions for a longer-time simulation of slow phase chamber relaxation.

TSUNAMI (Transient Shockwave Upwind Numerical Analysis Method for Inertial Confinement Fusion) [3] is a two-dimensional gas dynamics code and has been used to model the slow phase of IFE chamber evolution using the Euler equations for polytropic gas dynamics. Because the code treats the fluid as an inviscid, nonconducting ideal gas with constant specific heat and no radiation transport, it is most useful for studying the early stages of the slow phase prior to the onset significant transport effects. Recently, researchers at Osaka University's Institute of Laser Engineering have used a DSMC (Direct Simulation of Monte Carlo) code to model the KOYO IFE power plant chamber [4]. This DSMC code was developed by NASA and integrates Boltzmann's equations directly, providing a mechanism to assess the effects transport in the system evolution. Ref. [4] includes a comparison of long-time DSMC simulations of the KOYO IFE chamber with those performed with TSUNAMI. The comparisons demonstrated that many key features of the full evolution of the slow-phase solution are not properly captured when conduction and viscous transport are neglected.

The simulation approach described in this paper represents the assembly of several well-documented algorithmic components to enable the study of a more broad range of timescales in the slow phase of IFE chamber evolution, while also affording a reasonably general capability for multidimensional geometries. The core numerical scheme here is based on the Cartesian-grid Godunov integration method for the compressible Euler equations detailed in Ref. [5] (hereafter referred to as the "base scheme"). In the Cartesian grid approach, irregular (cut) cells are formed at the intersection of the uniform mesh and the problem domain boundary, and state variables are defined at the geometric centers of the rectangular mesh. In cells away from the embedded boundary, the difference algorithm in the base scheme reduces to a time-explicit second-order algorithm for advection. Fluxes into the cut cells are treated in a post-processing step that ensures temporal stability, consistency and accuracy. Stability of the overall integration scheme is governed by a time step condition that is based on the cell size in the uniform mesh. Background literature justifying this approach for compressible gas dynamics, and details of implementation of the base scheme are discussed at length in the reference. We have improved the accuracy of the base scheme in the cut cells through the use of a flux interpolation procedure outlined in Ref. [6], and added the ability to treat diffusion terms (viscosity and conduction) using ideas which first appeared in Ref. [7] related to Cartesian grid methods for Poisson's equation. Variable transport properties in the evolution of the parabolic diffusion terms are incorporated with an iterative, time-explicit Runge-Kutta method.

Like the base method, implementation of our combined solution approach is well-suited for incorporation into a block-structured adaptive-grid algorithm, and is extensible to three dimensions. Detailed convergence analyses and a sample application relevant to IFE chamber scenarios are used to validate the assembled algorithm in a simplified two-dimensional, non-adaptive setting.

2 Model and Numerical Algorithm

The compressible Navier-Stokes equations may be written for an ideal polytropic gas with viscosity, μ, and conductivity, k:

$$\begin{aligned} \rho_t + \nabla \cdot \rho \vec{u} &= 0 \\ (\rho \vec{u})_t + \nabla \cdot (\rho \vec{u} \vec{u}) + \nabla p &= \nabla \cdot \tau \\ (\rho E)_t + \nabla \cdot \Big((\rho E + p)\, \vec{u} \Big) &= \nabla \cdot (\tau \cdot \vec{u} + k \nabla T) \\ \tau &= \mu \left(\nabla \vec{u} + \nabla^T \vec{u} - \frac{2}{3} (\nabla \cdot \vec{u})\, I \right) \end{aligned} \qquad (1)$$

Here, ρ is the mass density, $\vec{u} = u\hat{x} + v\hat{y}$ is the velocity, $E = C_v T + \vec{u} \cdot \vec{u}/2$ is the total energy, C_v is the specific heat of the gas at constant volume, $p = \rho R T$ is the pressure, and τ is the stress tensor.

A Cartesian grid finite volume discretization for hyperbolic flow was described in detail in Ref. [5]. We give an overview of this algorithm, discussing first the the algorithm on a uniform grid, and then outlining the modifications necessary to accommodate the cut cells at the embedded boundary. This summary provides context for the subsequent description of our extensions that improve the scheme's accuracy at the embedded boundary, and expand its applicability to diffusive systems.

2.1 Base Uniform Grid Discretization

The equations (1) may be written in conservation form

$$U_t + F(U)_x + G(U)_y = 0 \qquad (2)$$

where $U = \{\rho, \rho u, \rho v, \rho E\}$ is the state vector, and F and G are the fluxes in the $\hat{x}$ and $\hat{y}$ directions, respectively. Each component direction of the fluxes may be written as a sum of advection and diffusion components. For example, $F = F^A + F^D$, where $F^A = \{\rho u, \rho u^2 + p, \rho u v, \rho u E + up\}$. If $\mu = k = 0$, the system is hyperbolic and we recover the model discretized in Ref. [5].

Following Ref. [5], the hyperbolic component of equation (1) is integrated in time with a second-order time-explicit Godunov method from $t = t^n$ to $t^{n+1} = t^n + \Delta t$ on a uniform grid, spaced $(\Delta x, \Delta y)$ in the $(\hat{x}, \hat{y})$ directions, respectively. The advection fluxes are centered in time at $t^{n+1/2}$, and in space at the geometric center of the faces bounding each cell. The fluxes are evaluated using data extrapolated in space and time from the centers to the faces of each cell. Prior to extrapolation, the state vector, U, is transformed into primitive variables, $Q = \{\rho, u, v, p\}$ using a polytropic (constant ratio of specific heats, $\gamma = 1.4$) equation of state. The extrapolations from cell centers to face centers are evaluated in terms of derivatives only in space at time t^n by substituting the model equations (1) to replace the time derivative. Double-valued face states are generated by extrapolating data from cell centers on both sides of the interface, and are resolved with the approximate Riemann solver discussed in Ref. [8]. The resulting primitive edge state is used to construct second-order conservative advection fluxes, which are scaled by the respective face area and time increment, Δt, and then differenced across each cell to form a volume-weighted advective forcing, S^A, for the state update for each cell:

$$U(t^{n+1}) = U(t^n) - \frac{\Delta t}{V} S^A \tag{3}$$
$$S^A = \left(F_r^A A_r - F_\ell^A A_\ell\right) + \left(G_t^A A_t - G_b^A A_b\right)$$

The subscripts r, ℓ, t, b refer to the right, left, top and bottem face of the cell, respectively, $A_\#$ is the area of the $\#^{th}$ face, and V is the volume of the cell ($= \Delta x \times \Delta y$). This algorithm is time-explicit, and stable only for $\Delta t < \min\left(\Delta x/\lambda_x, \Delta y/\lambda_y\right)$, where $\lambda_\# = \max\left(|\lambda_{\#,j}|\right)$ and $\lambda_{\#,j}$ are the eigenvalues of the flux Jacobian for the advective transport in the $\#^{th}$-direction.

2.2 Cut Cell Modifications in the Base Scheme

In order to extend the second-order Godunov algorithm to the Cartesian grid case, geometrical information about the cut cells must be included in the area and volume terms in equation (3). This includes partial cell volumes and edge areas, specified as fractions of the respective quantities from the underlying uniform grid. We must also accomodate fluxes through the interface in cut cells separating the fluid and body regions.

The volume fraction, Λ, at each mesh location is the ratio of cell volume inside the flow domain to total cell volume (*e.g.,* $\Lambda = 1$ for cells entirely inside the domain, $\Lambda = 0$ for cells in the uniform mesh but completely outside the fluid domain). The area fraction a indicates the portion of a cell face inside the flow domain. The quantities Λ and a may be extracted for realistic flow geometries by tools, such as Cart3D [9], that can process the output of computer-aided drawing packages. However, the geometries used in the examples here are simple enough that the requisite data may be generated with a few lines of computer code.

A straighforward application of equation (3) to the case of partial cell leads to an update for U which involves a division by Λ, and is numerically unstable for fixed Δt and $\Lambda \rightarrow 0$. A method to circumvent this stability issue, as suggested in Ref. [5], involves the calculation of an alternative update, the so-called "reference" state, by using equation (3) without accounting specifically for the reduction in face areas and cell volume. The reference state will be stable and mathematically consistent with equations (1), but not discretely conservative. The computed difference between the reference state and the update via equation (3) with proper area and volume reductions provides the degree to which the former update violates conservation in each field quantity. A stable, consistent and conservative update is formed by redistributing this error over nearby cells in the fluid (see Ref. [10]).

Computation of the reference state requires that face-centered fluxes be available on all four regular faces of any cell with $\Lambda > 0$. However, some of these faces may be entirely outside the fluid (*i.e.,* $a = 0$). In these cases cell-centered data may not be readily available for constructing the extrapolation that generates the double-valued face flux data. Since there is no valid state data at positions beyond edges outside the domain, a heuristic is used to create so-called "extended-states" from a simple low-order extrapolation of flow field from just inside the domain. The simple prescription detailed in Ref. [5] is sufficient for generating these values for the algorithm performance tests discussed in the next section.

The final modification in the base scheme is to accommodate the pressure on the fluid exerted by the embedded boundary interface. For the hyperbolic component of the model, this is the only non-trivial component of the conservation flux that is communicated through the embedded boundary. The wall pressure is obtained by constructing a special Riemann problem at the interface that represents a reflecting boundary condition for hyperbolic waves. The update, equation (3), is augemented to include this flux modification (suitably scaled by the area of the embedded boundary interface of that cell).

2.3 Extensions to the Base Scheme

The base scheme for integration of the compressible Navier-Stokes equations in the Cartesian grid setting is extended in three ways in order to improve the algorithm's accuracy in the cut cells, and to incorporate diffusion terms arising when $\mu \neq 0$ or $k \neq 0$. We outline a procedure for computing diffusion fluxes that is based on a centered difference approximation to the relevant differential operators. Once computed, the diffusion fluxes are treated very similar to the hyperbolic fluxes above, in terms of "reference state" computation and the local redistribution procedure for conservation. We then discuss a simple improvement that increases the accuracy of the conservation fluxes used to compute the conservative update. The correction is applied to both the parabolic and hyperbolic fluxes. Finally, a procedure is outlined for the construction of diffusion fluxes at the embedded boundary due to viscous shear and heat conduction.

2.3.1 Runge-Kutta Scheme for Diffusion

In order to incorporate diffusion fluxes into the discretization of equation (1), we construct an algorithm based on centered differences that reverts to a second order, symmetric stable and consistent approximation in the interior of the domain away from the cut cells along the embedded boundary. It will also remain stable for $\Lambda \rightarrow 0$, and will consistently treat the case where the transport coefficients depend on the state (*i.e.,* nonlinear diffusion). The procedure involves computing a flux divergence analagous to that for the advection fluxes already discussed. In particular, we define a diffusion flux divergence based on data at t^n:

$$S^{D,n} = \frac{1}{V}\left[\left(F_r^{D,n}A_r - F_\ell^{D,n}A_\ell\right) + \left(G_t^{D,n}A_t - G_b^{D,n}A_b\right) + F_{EB}^{D,n}A_{EB}\right] \qquad (4)$$

where A_{EB} is the surface area of embedded boundary interface, and $F_{EB}^{D,n}$ represents the diffusion flux through that interface. The boundary flux term will be discussed in the next section. Dropping the time index, n, for simplicity and using F^D as an example, we parameterize the diffusive flux as an arbitrary sum of terms involving $\nabla^\perp U$, the gradient of U perpendicular to a face, and $\nabla^\parallel U$, the gradient of U tangential to the face, $F^D = D\left(\alpha\nabla^\perp U + \beta\nabla^\parallel U\right)$. What remains then is to define the $\nabla^\perp$ and $\nabla^\parallel$ operators, and a procedure for computing D.

The perpendicular gradient approximation at the r, ℓ faces of cell (i,j) are evaluated using centered differences:

$$\nabla_r^\perp U_{i,j} = \left(U_{i+1,j} - U_{i,j}\right)/\Delta x \qquad \nabla_\ell^\perp U_{i,j} = \left(U_{i,j} - U_{i-1,j}\right)/\Delta x$$

Analagous expressions apply for $\nabla_t^\perp$ and $\nabla_b^\perp$ in the $\hat{y}$-direction.

The tangential griadient approximation at the r, ℓ faces of cell (i,j) are evaluated using a centered average of centered differences:

$$\nabla_r^\parallel U_{i,j} = 0.5\left(U_{i+1,j+1} - U_{i+1,j-1} + U_{i,j+1} - U_{i,j-1}\right)/\Delta x$$
$$\nabla_\ell^\parallel U_{i,j} = 0.5\left(U_{i,j+1} - U_{i,j-1} + U_{i-1,j+1} - U_{i-1,j-1}\right)/\Delta x$$

Analagous expressions apply for $\nabla_t^\parallel$ and $\nabla_b^\parallel$.

The temperature-dependent transport coefficients, μ and k, are evaluated with state data at the cell centers, and interpolated to cell edges using a harmonic averaging procedure. In particular, for the $\hat{x}$–direction faces of the cell (i,j),

$$D_r\left(U_{i,j}\right) = \frac{1}{1/D(U_{i,j}) + 1/D(U_{i+1,j})}$$
$$D_\ell\left(U_{i,j}\right) = \frac{1}{1/D(U_{i-1,j}) + 1/D(U_{i,j})}$$

Analagous expressions apply for $D_t\left(U_{i,j}\right)$ and $D_b\left(U_{i,j}\right)$.

We note that the existence "extended states", computed for the purposes of constructing the nonconservative "reference state" update discussed in Section 2.2 above guarantees that we will have sufficient data defined "inside" the embedded body for computing the above differences and averages.

A two-stage second-order Runge-Kutta method for integrating the diffusion terms with a constant advection forcing, S^A (taken directly from base scheme, as discussed above), may now be defined over a time interval, Δt:

$$\begin{aligned} U^* &= U^n - \Delta t \left(S^{D,n} + S^A \right) \\ U^{n+1} &= U^n - \frac{\Delta t}{2} \left(S^{D,n} + S^{D,*} \right) - S^A \Delta t \end{aligned} \tag{5}$$

The iteration defines a predictor-corrector procedure for a trapezoidal time-integration of diffusion terms, and a midpoint integration for hyperbolic terms. The combined integration is second-order accurate in time and space, as demonstrated in the next section. This time-explicit integration algorithm for advection and hyperbolic terms is stable only for numerical time step sizes satisfying the CFL and diffusional stability criteria:

$$\Delta t < \min \left[f_1 \frac{\Delta x}{(|\vec{v}| + c)}, \frac{f_2 \Delta x^2}{2 \max \left(\frac{\mu}{\rho}, \frac{k}{\rho C_p} \right)} \right] \tag{6}$$

where $c = \sqrt{\gamma p / \rho}$ is the local sound speed, and the min and max operations are performed over the entire computational domain. We take the "safety" factors, $f_1 = f_2 = 0.9$ for the examples presented below, and operate the algorithm in a regime where the advection-based stability condition is most restrictive, unless otherwise noted.

2.3.2 Flux Interpolation

In order to improve the accuracy of the treatment of the cut cells in the present context, we apply the advection flux interpolation procedure discussed in Ref. [6]. The procedure properly centers the evaluation of the conservative fluxes at the centroid of the partial cell faces, which are partially obscured by the embedded boundary. That is, since the cell-centered data is extrapolated along coordinate axes, the fluxes are second order accurate only at the centers of the underlying *uniform* grid. Flux values accurate to second order at the centroid of the *partial* faces may be constructed simply by linearly interpolating between full-face-centered fluxes at adjacent faces, as discussed in Refs. [7,6]. The procedure is simple to implement as a post-processing step after computing fluxes from the Riemann solution and the diffusion fluxes in equation (4), dramatically improves the accuracy of the solution along the embedded boundary, and has no advese affects on the time step-size limitation.

2.4 Fluxes at the Embedded Boundary

To complete the specification of our algorithm for diffusive transport in the presence of an embedded boundary, we need to define a procedure for computing the parallel and normal components of shear stress, and energy conduction at the wall. The expressions for the stress are particularly simple because the no-slip condition requires that both components of velocity, and therefore all gradients of velocity along the boundary are zero identically. The nontrivial component of the velocity derivatives relevant to boundary fluxes lead to the parallel, $\tau^{EB}_{\|,i,j}$, and normal, $\tau^{EB}_{\perp,i,j}$ components of shear stress:

$$\begin{aligned} \tau^{EB}_{\|,i,j} &= \tfrac{4}{3}\mu_{i,j}\left(\tfrac{\partial u_{\|}}{\partial n}\right)^{EB}_{i,j} \\ \tau^{EB}_{\perp,i,j} &= \mu_{i,j}\left(\tfrac{\partial u_{\perp}}{\partial n}\right)^{EB}_{i,j} \end{aligned} \tag{7}$$

Here, $u_{\|}$ and $u_{\perp}$ are the components of the velocity locally parallel and perpendicular to the embedded interface, and $\partial\phi/\partial n$ is the normal derivative of ϕ evaluated at the centroid of the embedded interface. The viscosity is evaluated with the state in the partial cell containing the interface. The heat conduction, $q^{EB}_{i,j}$, into the embedded boundary is:

$$q^{EB}_{i,j} = -k_{i,j}\left(\frac{\partial T}{\partial n}\right)^{EB}_{i,j} \tag{8}$$

.

The normal derivatives in equations (7) and (8) are estimated using a three-point interpolation formula, which is second order accurate and stable for fixed $\Delta x, \Delta y$ and $\Lambda \rightarrow 0$. A vector which is normal to the boundary is located with its origin at the centroid of the embedded boundary interface. This vector extends into the fluid at least three layers of cells, as shown in Figure 1. The state values at the origin of the normal are prescribed from the physical boundary conditions. The remaining values are computed using parabolic interpolations of the nearby state data, according to the prescription in Ref. [7]. These three points are sufficient to define a well-behaved parabola, from which a slope is evaluated at the embedded interface.

3 Algorithm performance and convergence

We illustrate the performance of our algorithm with two test cases. The first case focuses on the convergence of the algorithm with an isothermal condition implemented at the embedded boundary. The analysis is based on the propagation of a smooth temperature distribution in a channel not aligned with the underlying uniform grid. The second case demonstrates the performance of the algorithm in modeling conditions more specifically relevant to the intended application–simulations of the evolution of an IFE chamber environment.

3.1 Propagation of a Smooth Isobaric Disturbance in a Straight Channel

The numerical quality of our embedded boundary algorithm may be evaluated by monitoring the evolution of gas with a nontrivial temperature distribution as it passes through a bounded channel with no-slip, isothermal walls. The calculation is performed over the square region shown in Figure 2, 1.6 m on a side. Within this area, a straight-wall channel 1 m wide is oriented 30° counter-clockwise from the horizontal axis. This choice of embedded boundary shape has the advantage of providing a range of partial volumes between 0 and 1 within the domain, while being analytically smooth so as not to pollute the convergence analysis.

Xenon is used as the transport medium, and is treated as an ideal gas with viscosity and conductivity specified by the Sutherland Law [11]:

$$\eta(T) = \eta_o \left(\frac{T}{T_{o,\eta}}\right)^{1.5} \frac{T_{o,\eta} + T_{s,\eta}}{T + T_{s,\eta}} \tag{9}$$

where viscosity, μ, or conductivity, k should be substituted for the general property η. The conductivity-related constants in this model $T_{s,k} = 320.93$ K, $T_{o,k} = 800$ K and $k_o = 0.0132$ W/(mK) were determined by fitting the experimental data reported in Ref. [12] on the range 800-2000 K. The equivalent constants for viscosity, $T_{s,\mu} = 320.26$ K, $T_{o,\mu} = 800$ K and $\mu_o = 4.88 \cdot 10^{-5}$ Ns/m^2 were based on the experimental data found in Ref. [13]. The initial flow velocity is fixed at 120 m/s, and is aligned with the walls of the channel. To avoid discontinuity between the wall and the fluid, the wall is set to move at the same speed as the initial flow. The initial pressure is set constant at 100 Pa throughout the domain, and the initial temperature distribution is prescribed as

$$T(x,y,0) = \begin{cases} T_\infty + 50\left(1.0 + \cos(2\pi\sqrt{x^2+y^2})\right), \ for\ x^2+y^2 \leq (L/2)^2 \\ T_\infty, \ elsewhere \end{cases} \tag{10}$$

In order to approximate the span of conductivity and viscosity in the IFE chamber environment, the temperature of the surrounding gas and channel walls was given the value $T_\infty = 10^5$ K. As the system evolves, the peak in the temperature profile advects downstream, and heat is conducted through the embedded boundary. The system was integrated in time long enough to effect sufficient changes to the solution to allow a reasonable convergence analysis.

The simulation was performed using six consecutive uniform mesh resolutions (the cell size was halfed consecutively). The coarsest simulation was performed with $\Delta x = 50$ mm, corresponding to a 32×32 uniform grid. We define the discretization error, $E_{i,j}$ for any state variable $U_{i,j}$ as

$$E_{i,j} \ = \ U_{i,j} - U_{i,j}^{exact} \tag{11}$$

where $U_{i,j}^{exact}$ represents the "exact" solution, which we take here from 2D quadratic interpolations of the solution on our finest grid (1024×1024). The convergence rate of the algorithm was estimated using the norms L_∞, L_1, L_2 as defined in Table I. As the performance of the algorithm near the embedded boundary will be different from that of the uniform grid far away from the boundary, a special norm, $L_{2,EB}$, was also included (see Table I). For each norm, the rate of convergence, p, for a grid spacing $\Delta x = \Delta y = h$ is calculated from:

$$p(h) \ = \ log_2 \left(\frac{\| E_{i,j} \|_h}{\left\| E_{i,j}^n \right\|_{h/2}} \right) \tag{12}$$

In this measure, if the boundary cells show a first-order local convergence rate in grid spacing, while the interior cells converge at second order, we expect that errors in the L_1 norm would show a second-order overall convergence rate, while the L_2 norm would exhibit a rate between first and second order, and the L_∞ norm would be first order. However, if the boundary cells converge to second order, the L_1 and L_2 norm measures should indicate a convergence rate $2 < p < 3$. Rates based on the L_∞ norm indicates the convergence performance of the domain's worst regions, and would show a value of 1 in the former example, but a value of 2 in the latter.

Convergence rates for the various norms are presented in Table II. Clearly, the convergence rates for the L_1 and L_2 measures are consistent with the interpretation that the boundary cells, like the interior cells, are converging at near the designed second-order rate. The values for the L_∞ norm confirm that all regions in the calculation are indeed converging near second-order. The values for $L_{2,EB}$ confirm these observations specifically for the cut cells, showing that with increasing resolution, the accuracy of the cut cells approaches second-order for nearly all the state components.

In this test problem, the energy equation exhibits the slowest convergence rates, particularly in measures concentrated at the embedded boundary. In fact, the second-order rate expected of the algorithm is not fully achieved for the energy equation even though the momentum and continuity equations are well-behaved at this resolution. Evidently, the asymptotic convergence rate is realized for the momentum equation on coarser grids than for the energy equation. The same test case is analyzed but with the conductivity uniformly decreased by a factor of four from the values given through equation (9). Convergence rates and error norms for this set of calculations are given in Table III. With this reduced conductivity, the energy equation (and all others) clearly exhibit second-order convergence.

Such a confirmation could alternatively have have been achieved by further refining the solution. However, for these parameters, the grid spacing would have been such that the diffusion-based criteria in equation (6) would have been more restrictive than the advective CFL condition. We note that this otherwise well-behaved system is likely to be an ideal candidate for a time-implicit variation of our algorithm. A time-implicit scheme would require solution of a linear equation set for the diffusive update corresponding to equation (5), but could be constructed as a sequential update to the velocity and energy fields separately, as discussed in Ref. [14], for example. However, the present algorithm will be perfectly adequate for a broad range of IFE-relevant applications where such fine grids along the embedded boundary would not be routinely required to achieve acceptable levels of absolute error. An example of such a problem is the subject of the next section.

Finally, the second-order convergence of our algorithm for this problem may come somewhat as a surprise, given the lack of error cancellations at the cut cells that is required for second-order accuracy of the centered differences in the interior of the domain. Johansen [15] presents a modified equation analysis for this diffusion operator showing that the discrete errors associated with Dirichlet boundary conditions at cut cells exhibit a dipole-like influence on the solution away from the boundary that decrease with Δx^3. This is consistent with our convergence analyses above, showing that in the asymptotic regime the solution errors become dominated by our second-order time-discretization and treatment of the advection terms.

3.2 A Neutral Gas Response to High Energy Target Blast for an Inertial Fusion Energy Chamber

We have demonstrated second order convergence both inside the domain and at the boundary for our algorithm on a smooth problem. In order to demonstrate the behavior of the algorithm for treating discontinuous profiles, we select a simplified 2D model of an IFE chamber (as described in Sec. 1). The chamber body is approximated by a cylinder of radius 6.5 m. Laser beam ports are represented as four beam lines, each 20 m long and 1 m wide (see Figure 3). The chamber gas is Xenon, with temperature-dependent viscosity and conductivity given by equation (9). Prior to injecting the target into the chamber, the quiescent gas is at $T = 298$ K and $p = 6.67$ Pa (50 mTorr). A "direct-drive" target yielding 160 MJ [16] is ignited at the center of the chamber, and heats the Xenon gas through direct X-rays and energetic ion absorption. The "fast" phase of chamber evolution is modeled prior to shock colleisions with the vessel boundary using the 1D radiation-hydrodynamics code BUCKY [2]. The BUCKY solution profiles at 500 μs for density, velocity and pressure are interpolated onto a uniform grid to form cylindrically symmetric 2D initial profiles. The 1D profiles of temperature and pressure shown in Figure 4 indicate that 500 μs after the initial X-ray heating (and prior to shock/wall impact), the hottest regions in the chamber have cooled to under 2 eV. We evolve the system for 0.1 s, corresponding to the time interval before the next fusion target insertion.

The boundary conditions at the vessel surface are no-slip wall, with the constant temperature $T_{wall} = 973$ K. The state gradients required for estimating the shear stress and heat flux at the wall is resolved by the parabolic interpolant described in Ref. [7] if the boundary layer of velocity and temperature field spans across at least several of the uniform grid cells. Such is the test problem described in Section 3.1. In the present case, however, a rather thin boundary layer in T and $\vec{u}$ suggests that the method from Ref. [7] cannot be used. Within the scope of this study, we adopt a coarse grid approach, which sets the embedded boundary a distance δ inside of the chamber wall. An unresolved heat and momentum transfer region is located between the embedded boundary and the physical wall, outside the computational domain. The normal state gradient at the embedded boundary is approximated as:

$$\left.\frac{\partial U}{\partial n}\right|_{EB} = \frac{1}{\delta}\left(U_{EB} - U_{wall}\right) \tag{13}$$

The state variables v_x, v_y and T should be substituted for U and δ is an arbitrary length scale taken here to be 1 cm. The state values denoted as U_{EB} are estimated by quadratic interpolations of the solution to the centroids of the fluid-body cell interfaces, while U_{wall} are the values of velocities and temperatures prescribed at the wall. Alternatively, in practical applications, adequate resolution local to the boundary layer may be achieved using adaptive mesh refinement.

The evolution of pressure and temperature is shown in Figure 5. At early times, the profiles are 4-fold symmetric, so 1/8 of the domain is sufficient to show the entire solution. The initial position of the shock wave is 40 cm away from the wall, as shown in Figure 5a. The elevated temperature ahead of the shock results from the X-ray and ion energy deposition during the first 500 μs, as computed from BUCKY. Figure 5b shows the chamber state 5 ms after the solution was advanced by our integration algorithm. The shock is about to hit the wall for the second time, after initially reflecting from the embedded boundary, converging to the center of the chamber and expanding outward again. The discontinuous profiles remain smooth and sharp. A secondary shock structure is apparent in the figure, and results from the reflection of the initial shock against the entrance to the beam channels. The core of the chamber is heated from the initial temperature of 5.37×10^4 K to above 10^5 K by the compression of the converging shock wave. At 13 ms, the temperature of the core is relatively hot compared to the near-wall region, as shown in Figure 5c. The temperature variation between the cold wall and the hot center causes large variations in viscosity, μ (ranging from 5.67×10^{-5} to 9.12×10^{-4} kg/m-s) and conductivity, k (ranging from 0.015 to 0.25 W/m-K) across the chamber. The geometry of the beam channels initiates large-scale vortical structures and a highly two-dimensional flow field in the late time solution, as seen in Figure 6.

Strict convergence of the algorithm is difficult to characterize on this problem because of the presence of discontinuities throughout the solution. However, we expect an integral quantity such as the total heat transferred to the wall to be a fairly smooth function of time and a useful indication of algorithm performance and convergence. The shear stress components ($\tau_{\parallel}$, $\tau_{\perp}$) at the wall can characterize the convergence of momenta and the heat flux q into the wall can be selected to characterize the convergence of the energy equation. The chosen quantities are integrated along the wall and in time, at every time step during the evolution. The results are shown in Figure 7 for three successive mesh resolutions and demonstrate a measurable energy and momentum loss of the chamber system to the environment. For reference, the total energy of the gas was initially $26.9 \cdot 10^4$ J, which implies an energy loss of roughly 30 %, for the duration of 0.1 s. The convergence rates of the boundary integrals from Figure 7 at a time t may be approximated as

$$p_*(t) \approx \log\left(\frac{|S_*^{1200\times1200}(t) - S_*^{300\times300}(t)|}{|S_*^{1200\times1200}(t) - S_*^{600\times600}(t)|}\right) \qquad (14)$$

where * denotes q, $\tau_{\parallel}$ or $\tau_{\perp}$ and $S_*^{grid\ size}(t)$ is the value of the boundary integral at t, for the given grid size. The average convergence rates based on the expression (14) and the plots in Figure 7 were estimated for the time interval of 0.1 s. Their values are $\bar{p}_{\tau_{\parallel}} = 1.639$, $\bar{p}_{\tau_{\perp}} = 1.321$ and $\bar{p}_q = 2.334$. This indicates that the scheme is converging to better than first-order for an initially discontinuous practically relevant scenario. In fact, Figure 7 indicates that the largest discrepancies arise while highly two-dimensional flow is interacting strongly with the vessel wall. Over most of the evolution, the profiles are smoothly varying and converge at second-order, consistent with the results of the first example.

4 Discussion and Conclusions

In this paper, we have presented an algorithm to solve the time-dependent compressible Navier Stokes equations in complex 2D geometries. The algorithm has been applied to study the behavior of a simple model for target chamber dynamics for inertial fusion energy (IFE) in the “slow phase” where hydrodynamic energy transport processes are important. The problem involves the evolution of discontinuous profiles (shocks), temperature-dependent diffusion transport, and isothermal, no-slip irregular boundaries. The computational method has been constructed using a Godunov method for advection, a Runge-Kutta method for diffusion, and an embedded boundary approach to incorporate a flexible geometry capability.

A convergence analysis was performed using a pair of test problems. The first, designed to evaluate the formal convergence properties of the algorithm, was based on smooth initial and boundary data. The results demonstrated uniform second-order convergence. In order to test the robustness of the algorithm in a context closer to the intended application, we simulated a two-dimensional model of an IFE chamber. The initial conditions for the test were generated with the BUCKY code, and included a strong radially-outward propagating shock wave. We simulated 100 ms of the ensuing interaction of this shock wave with the boundary and fluid environment. The resulting profiles exhibited complex multi-dimensional flow patterns and secondary shock structures due to interactions with the optical beam channels. A strict convergence analysis of this system was impossible due to the presence of discontinuities in the initial data and boundary shape. Instead, we monitored the behavior of the wall loads, such as the integral of shear stress components and heat flux along the boundary and in time. We found that these quantities behaved reasonably with increased grid resolution. This indicates that the algorithm is robust enough to be suitable for long time-scale simulations of IFE chambers.

The embedded boundary algorithm presented in this paper shares the geometrical limitations discussed in Ref. [5]. In particular, long, narrow fin-like (under-resolved) features in the embedded boundary shape are not correctly represented by the algorithm, which assumes a piecewise linear wall at the resolution of the uniform grid. The construction of "extended-states" in a data structure that is assumed logically rectangular requires that several full uniform "body" cells (with $\Lambda = 0$) separate distinct fluid regions, and the interpolation procedure for evaluating state gradients at Dirichlet boundaries requires that several "fluid" cells (with $\Lambda = 1$) separate body regions normal to the surface. These limitations notwithstanding, this embedded boundary implementation has exhibited the appropriate level of geometrical flexibility and accuracy to be used as the basis of a tool for investigating the slow phase of IFE chamber phyiscs.

From the point of view of investigating IFE chamber scenarios, many additional phenomena should be incorporated into the model. For example, the chamber fill gas likely includes many distinct chemical constituents, generated by vessel wall ablation for example, particularly during the fast phase and early parts of the slow phase. The algorithms presented here will extend rather directly to multi-fluid implementations, such as the one described in Ref. [8], or to the mass-fraction approach discussed in Ref. [14]. In addition, our second example was interesting in that the convergence of blast waves reflected from the chamber wall resulted in localized heating of the central core region to above 10 eV, suggesting that radiative energy transport may be an important physical process in the slow phase. As more detailed IFE chamber scenarios are developed, three-dimensional simulations will become more important, and will require concommitant extensions that allow local dynamic adaptive mesh refinement. All these improvements are under current development.

5 Acknowledgments

The research of ZD and FN was supported by the Naval Research Laboratory under grant N00173-02-1-6010. The research of MD was supported by the Applied Mathematical Sciences Program of the DOE Office of Mathematics, Information, and Computational Sciences, under contract DE-AC03-76SF00098.

References

[1] G. A. Moses, R. R. Peterson: ”Computer Modeling of ICF Target Chamber Phenomena”, Laser and Particle Beams (1994), volume 12, no. 2, pp.125-162.

[2] J. J. MacFarlane, G. A. Moses, R. R. Peterson, University of Wisconsin, Fusion Technology Institute report UWFDM-984, 1995.

[3] J. Liu, P. Peterson, and V. Schrock, “Blast Venting through Blanket Material in the HYLIFE ICF Reactor,” Fusion Technology, 21, pp. 1514-1518, 1992.

[4] Y. Kozaki *et al.,* “Simulation Study on Evacuation of Metal Vapor in Laser Fusion Liquid Wall Chamber,” Institute of Laser Engineering, Osaka University report, 2000.

[5] R.B. Pember, J.B. Bell, P. Colella, W.Y. Crutchfield, and M.L. Welcome, ”An Adaptive Cartesian Grid Method for Unsteady Compressible Flow in Irregular Regions”, Journal of Computational Physics, 120:2, pp. 278-304, Sept. 1995.

[6] D. Modiano, P. Colella: ”A higher-order embedded boundary method for time-dependent simulation of hyperbolic conservation laws”, Proceedings of FEDSM’00, 2000.

[7] Johansen, P. Colella: ”A Cartesian Grid Embedded Boundary Method for Poissons Equation on Irregular Domains”, Journal of Computational Physics, Volume 147(1), pp. 60-85, 1998.

[8] P. Colella, R. E. Ferguson and H. Glaz, ”Multifluid Algorithms for Eulerian Finite Difference Methods”, in preparation.

[9] AIAA 97-0196: ”Robust and Efficient Cartesian Mesh Generation for Component-Based Geometry”, 35th AIAA Aerospace Sciences Meeting, Reno NV Jan, 1997.

[10] I.L. Chern, P. Colella: ”A Conservative Front-Tracking Method for Hyperbolic Conservation Laws”, UCRL-97200, Lawrence Livermore National Laboratory, 1987.

[11] W. Sutherland: ”The Viscosity of Gases and Molecular Force”, Philosophical Magazine, Volume 36, 507-531, 1893.

[12] Jain, P.C.: ”The Prediction of the Thermal Conductivity of Xenon”, Journal of Physics, Division: Applied Physics, Volume 11, pp. 2371-2373., 1978.

[13] McCarty, R.D.: ”‘Correlations for the Thermophysical Properties of Xenon”, National Institute of Standards and Technology, Boulder, CO, 1989, 00.

[14] R. B. Pember, L. H. Howell, J. B. Bell, P. Colella, W. Y. Crutchfield, W. A. Fiveland, J. P. Jesse: ”An Adaptive Projection Method for Unsteady, Low-Mach Number Combustion”, Combust. Sci. Technol., Volume 140, pp. 123-168, 1998.

[15] H. S. Johansen: "Cartesian Grid Embedded Boundary Finite Difference Methods for Elliptic and Parabolic Partial Differential Equation on Irregular Domains", Ph.D. Thesis, University of California in Berkeley, 1997.

[16] S. E. Bodner, D. G. Colombant, A. J. Schmitt and M. Klapisch, *Phys. of Plasmas,* **7**, pp. 2298-2301, 2000.

Table I
Definition of Error Norms Used For Convergence Analysis

Error Norm	L_∞	L_1	L_2	$L_{2,wall}$
Definition	$max\,\lvert E\rvert$	$\frac{1}{V}\int\limits_V EdV$	$\sqrt{\frac{1}{V}\int\limits_V E^2dV}$	$\sqrt{\frac{1}{l_{EB}}\int\limits_{l_{EB}} E^2dl}$
Discrete Form	$max\left\lvert E^n_{i,j}\right\rvert$	$\frac{\sum\limits_{i,j} E^n_{i,j}\Lambda_{i,j}\Delta x\Delta y}{\sum\limits_{i,j}\Lambda_{i,j}\Delta x\Delta y}$	$\sqrt{\frac{\sum\limits_{i,j}(E^n_{i,j})^2\Lambda_{i,j}\Delta x\Delta y}{\sum\limits_{i,j}\Lambda_{i,j}\Delta x\Delta y}}$	$\sqrt{\frac{\sum\limits_{\Lambda_{i,j}<1}(E^n_{i,j})^2\Lambda_{i,j}\Delta x}{\sum\limits_{\Lambda_{i,j}<1}\Lambda_{i,j}\Delta x}}$

Table II

Error norms L_* and their rate of convergence, p_* for Test case of Sec. 3.1

	density		x-momentum		y-momentum		energy	
grid size	L_*	p_*	L_*	p_*	L_*	p_*	L_*	p_*
L_∞ , p_∞								
32×32	3.855×10^{-6}	-	4.045×10^{-4}	-	2.312×10^{-4}	-	3.281×10^{-2}	-
64×64	1.493×10^{-6}	1.369	1.554×10^{-4}	1.380	8.957×10^{-5}	1.368	1.502×10^{-2}	1.128
128×128	5.182×10^{-7}	1.526	5.435×10^{-5}	1.515	3.060×10^{-5}	1.550	5.717×10^{-3}	1.393
256×256	1.908×10^{-7}	1.442	1.986×10^{-5}	1.453	1.125×10^{-5}	1.443	2.083×10^{-3}	1.457
512×512	5.674×10^{-8}	1.750	5.703×10^{-6}	1.800	3.176×10^{-6}	1.825	8.231×10^{-4}	1.340
L_1 , p_1								
32×32	3.195×10^{-7}	-	$3.311{\times}10^{-5}$	-	$1.917{\times}10^{-5}$	-	$2.809{\times}10^{-3}$	-
64×64	6.947×10^{-8}	2.201	$7.189{\times}10^{-6}$	2.204	$4.165{\times}10^{-6}$	2.203	$8.476{\times}10^{-4}$	1.729
128×128	1.379×10^{-8}	2.333	$1.430{\times}10^{-6}$	2.330	$8.386{\times}10^{-7}$	2.312	$3.100{\times}10^{-4}$	1.451
256×256	2.609×10^{-9}	2.402	$2.785{\times}10^{-7}$	2.360	$1.726{\times}10^{-7}$	2.281	$1.186{\times}10^{-4}$	1.386
512×512	4.774×10^{-10}	2.450	$5.481{\times}10^{-8}$	2.345	$4.186{\times}10^{-8}$	2.044	$3.680{\times}10^{-5}$	1.688
L_2 , p_2								
32×32	$6.124{\times}10^{-7}$	-	$6.361{\times}10^{-5}$	-	$3.671{\times}10^{-5}$	-	$5.094{\times}10^{-3}$	-
64×64	$1.589{\times}10^{-7}$	1.946	$1.651{\times}10^{-5}$	1.946	$9.530{\times}10^{-6}$	1.946	$1.561{\times}10^{-3}$	1.706
128×128	$3.694{\times}10^{-8}$	2.105	$3.842{\times}10^{-6}$	2.104	$2.220{\times}10^{-6}$	2.102	$5.104{\times}10^{-4}$	1.613
256×256	$7.233{\times}10^{-9}$	2.352	$7.643{\times}10^{-7}$	2.330	$4.506{\times}10^{-7}$	2.300	$1.771{\times}10^{-4}$	1.527
512×512	$1.164{\times}10^{-9}$	2.636	$1.343{\times}10^{-7}$	2.509	$8.526{\times}10^{-8}$	2.402	$5.399{\times}10^{-5}$	1.714
$L_{2,EB}$, $p_{2,EB}$								
32×32	$7.656{\times}10^{-7}$	-	$7.930{\times}10^{-5}$	-	$4.614{\times}10^{-5}$	-	$4.518{\times}10^{-3}$	-
64×64	$2.625{\times}10^{-7}$	1.544	$2.724{\times}10^{-5}$	1.542	$1.588{\times}10^{-5}$	1.539	$1.433{\times}10^{-3}$	1.657
128×128	$9.447{\times}10^{-8}$	1.475	$9.946{\times}10^{-6}$	1.453	$5.799{\times}10^{-6}$	1.453	$6.782{\times}10^{-4}$	1.079
256×256	$3.322{\times}10^{-8}$	1.508	$3.759{\times}10^{-6}$	1.404	$2.229{\times}10^{-6}$	1.380	$3.927{\times}10^{-4}$	0.788
512×512	$8.121{\times}10^{-9}$	2.032	$1.157{\times}10^{-6}$	1.700	$6.487{\times}10^{-7}$	1.781	$1.660{\times}10^{-4}$	1.242

Table III

Error norms and their rate of convergence (L, p) for Test case of Sec. 3.1 and modified gas properties[†]

	density		x-momentum		y-momentum		energy	
grid size	L_*	p_*	L_*	p_*	L_*	p_*	L_*	p_*
L_∞ , p_∞								
32×32	3.928×10^{-6}	-	4.092×10^{-4}	-	2.357×10^{-4}	-	2.818×10^{-2}	-
64×64	1.590×10^{-6}	1.305	1.653×10^{-4}	1.307	9.541×10^{-5}	1.305	1.257×10^{-2}	1.165
128×128	5.647×10^{-7}	1.493	5.876×10^{-5}	1.492	3.392×10^{-5}	1.492	4.712×10^{-3}	1.415
256×256	1.776×10^{-7}	1.669	1.848×10^{-5}	1.669	1.075×10^{-5}	1.658	1.582×10^{-3}	1.574
512×512	4.451×10^{-8}	1.997	4.615×10^{-6}	2.002	2.622×10^{-6}	2.035	3.964×10^{-4}	1.997
L_1 , p_1								
32×32	3.246×10^{-7}	-	3.370×10^{-5}	-	1.947×10^{-5}	-	2.437×10^{-3}	-
64×64	7.373×10^{-8}	2.138	7.649×10^{-6}	2.139	4.415×10^{-6}	2.141	5.902×10^{-4}	2.046
128×128	1.596×10^{-8}	2.208	1.652×10^{-6}	2.211	9.538×10^{-7}	2.211	1.496×10^{-4}	1.980
256×256	3.226×10^{-9}	2.307	3.332×10^{-7}	2.310	1.931×10^{-7}	2.305	4.260×10^{-5}	1.812
512×512	5.437×10^{-10}	2.569	5.600×10^{-8}	2.573	3.258×10^{-8}	2.567	1.149×10^{-5}	1.890
L_2 , p_2								
32×32	6.239×10^{-7}	-	6.482×10^{-5}	-	3.742×10^{-5}	-	4.613×10^{-3}	-
64×64	1.691×10^{-7}	1.884	1.757×10^{-5}	1.884	1.014×10^{-5}	1.884	1.280×10^{-3}	1.849
128×128	4.331×10^{-8}	1.965	4.499×10^{-6}	1.965	2.595×10^{-6}	1.966	3.450×10^{-4}	1.891
256×256	1.040×10^{-8}	2.059	1.081×10^{-6}	2.057	6.232×10^{-7}	2.058	9.098×10^{-5}	1.923
512×512	1.928×10^{-9}	2.431	2.007×10^{-7}	2.429	1.159×10^{-7}	2.426	2.093×10^{-5}	2.120
$L_{2,EB}$, $p_{2,EB}$								
32×32	7.663×10^{-7}	-	7.955×10^{-5}	-	4.604×10^{-5}	-	5.250×10^{-3}	-
64×64	2.506×10^{-7}	1.613	2.602×10^{-5}	1.612	1.507×10^{-5}	1.611	1.655×10^{-3}	1.665
128×128	7.953×10^{-8}	1.656	8.283×10^{-6}	1.651	4.795×10^{-6}	1.652	5.447×10^{-4}	1.604
256×256	2.387×10^{-8}	1.736	2.521×10^{-6}	1.716	1.466×10^{-6}	1.710	1.827×10^{-4}	1.576
512×512	5.671×10^{-9}	2.074	6.293×10^{-7}	2.002	3.615×10^{-7}	2.020	5.675×10^{-5}	1.687

[†] μ defined in eq. 9, k reduced by a factor of 4 from eq. 9

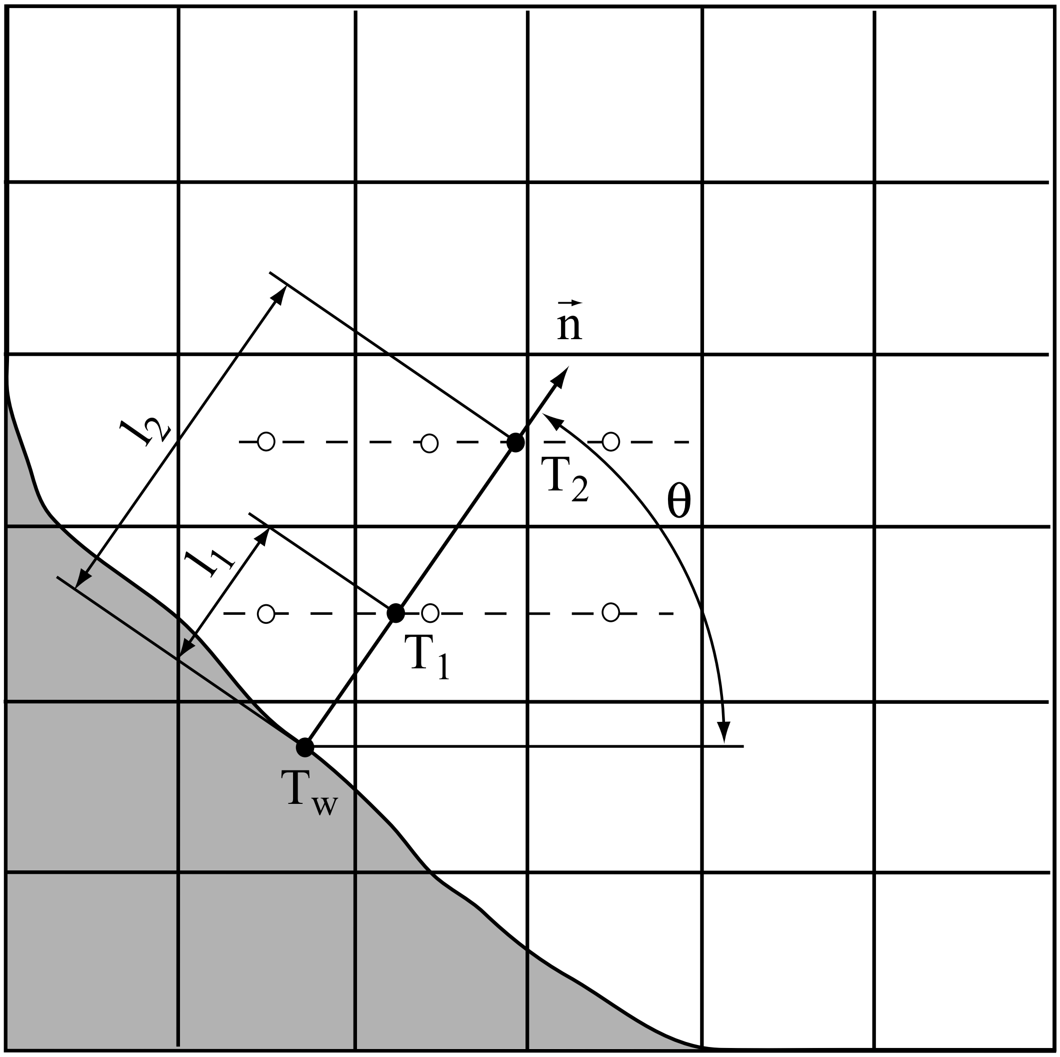

Fig. 1. Parabolic interpolation for estimating a second order temperature gradient normal to the boundary. Wall temperature T_w is prescribed. Temperatures T_1 and T_2 are estimated based on the second order interpolation from the values in the open circles. The stencil of temperatures T_w, T_1 and T_2 is used for estimating the temperature gradient.

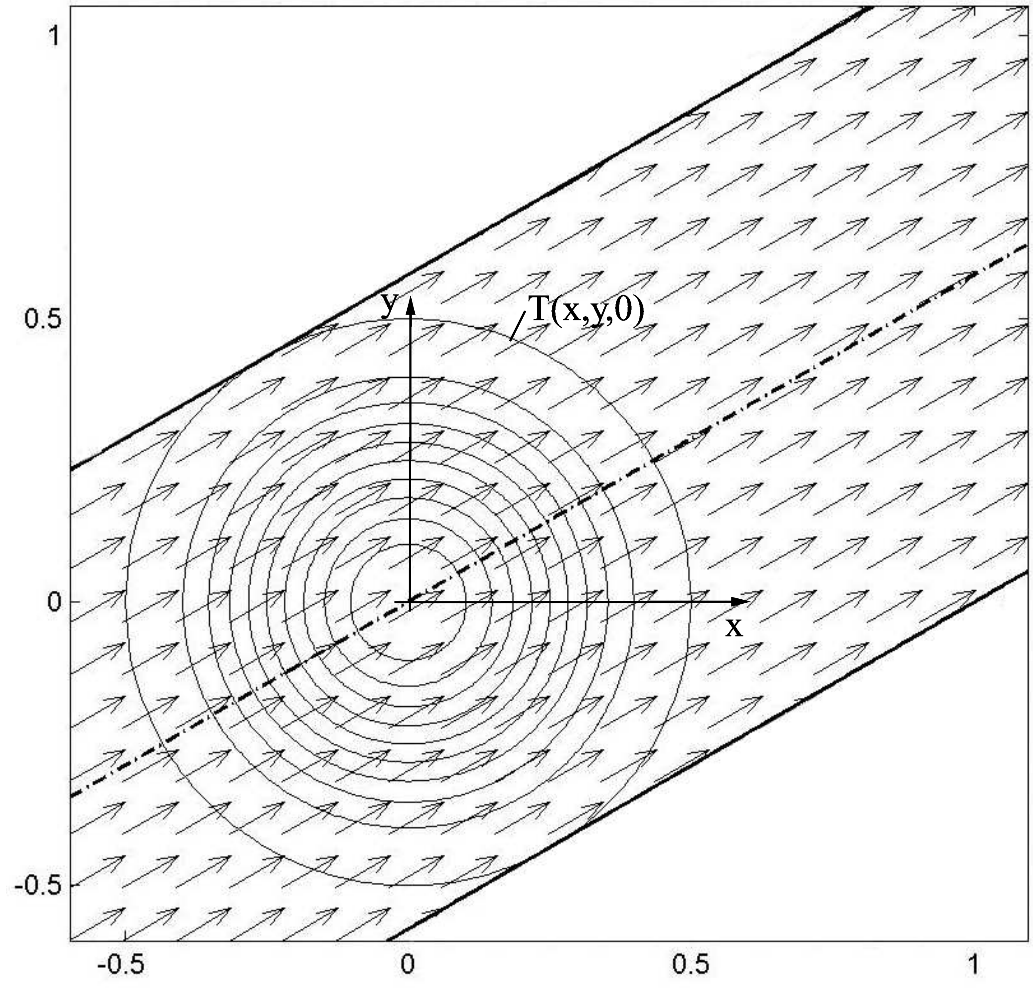

Fig. 2. Geometry and initial conditions for the straight channel flow. The temperature field is represented by isothermal lines while the velocity field is represented by arrows.

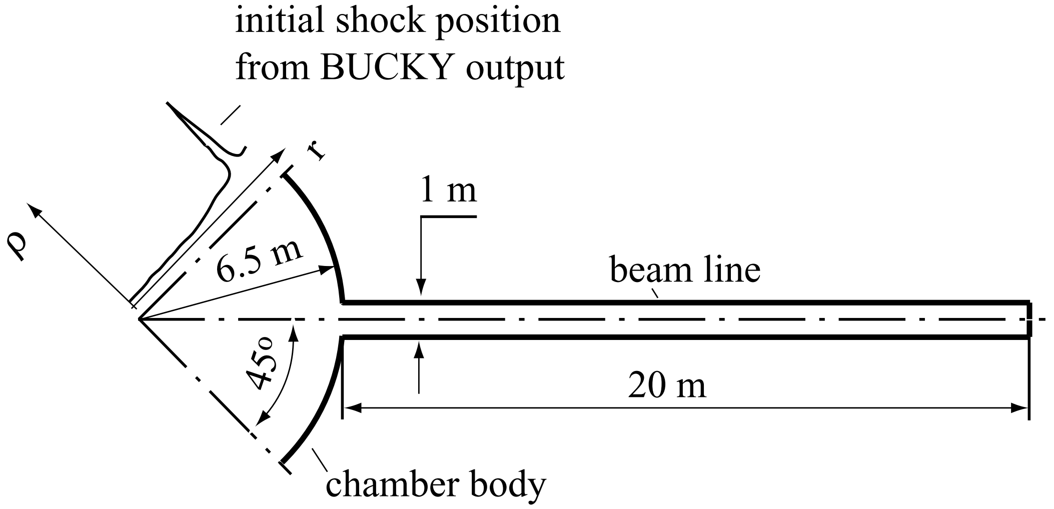

Fig. 3. Geometry of the four-fold symmetric IFE chamber. One of the four optical access ports is shown in the figure, as well as the initial position of the shockwave relative to the chamber wall.

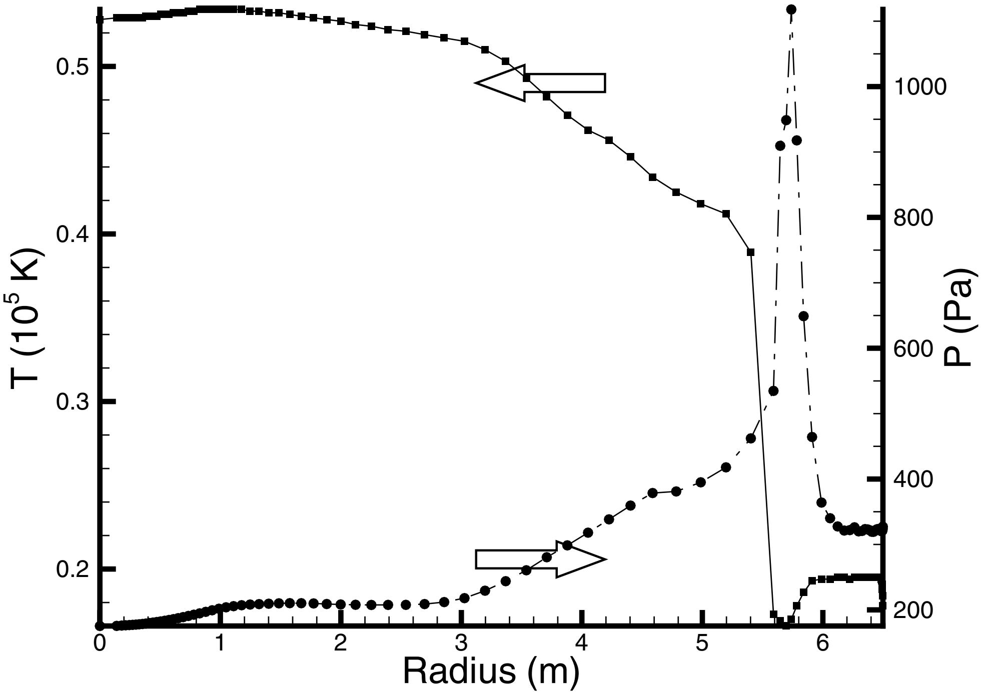

Fig. 4. Initial temperature and pressure for the IFE chamber model. The density, velocity and temperature profiles are obtained as a 1D solution from a Lagrangian code BUCKY. The solution is interpolated onto the grid before being advanced by the Godunov algorithm.

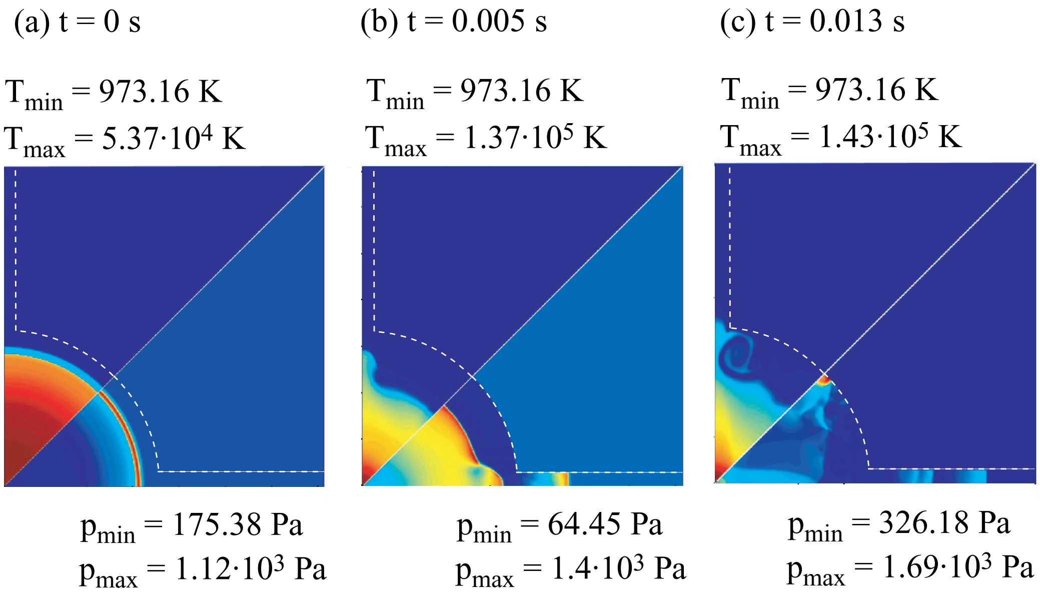

Fig. 5. Pressure (lower triangular portion of the figure) and temperature (upper portion of the figure) at three different times. A range of temperatures and pressures from minimum to maximum is provided for each individual case, as shown above.

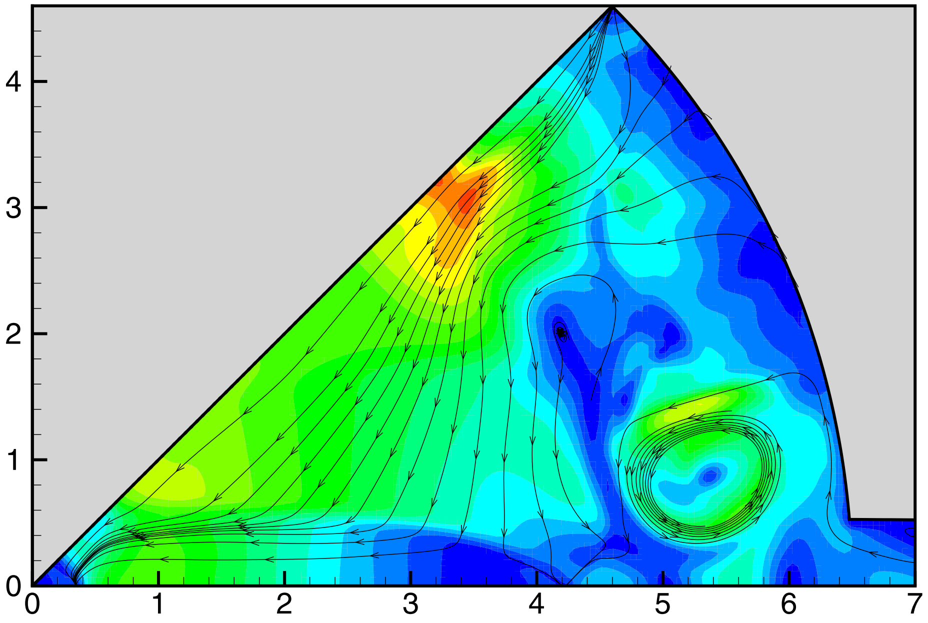

Fig. 6. Detail of velocity field in the cylindrical portion of the chamber at time t=0.013 s. The streamlines are accompanied by the color plot of velocity magnitude, which peaks at 821.28 m/s (red area).

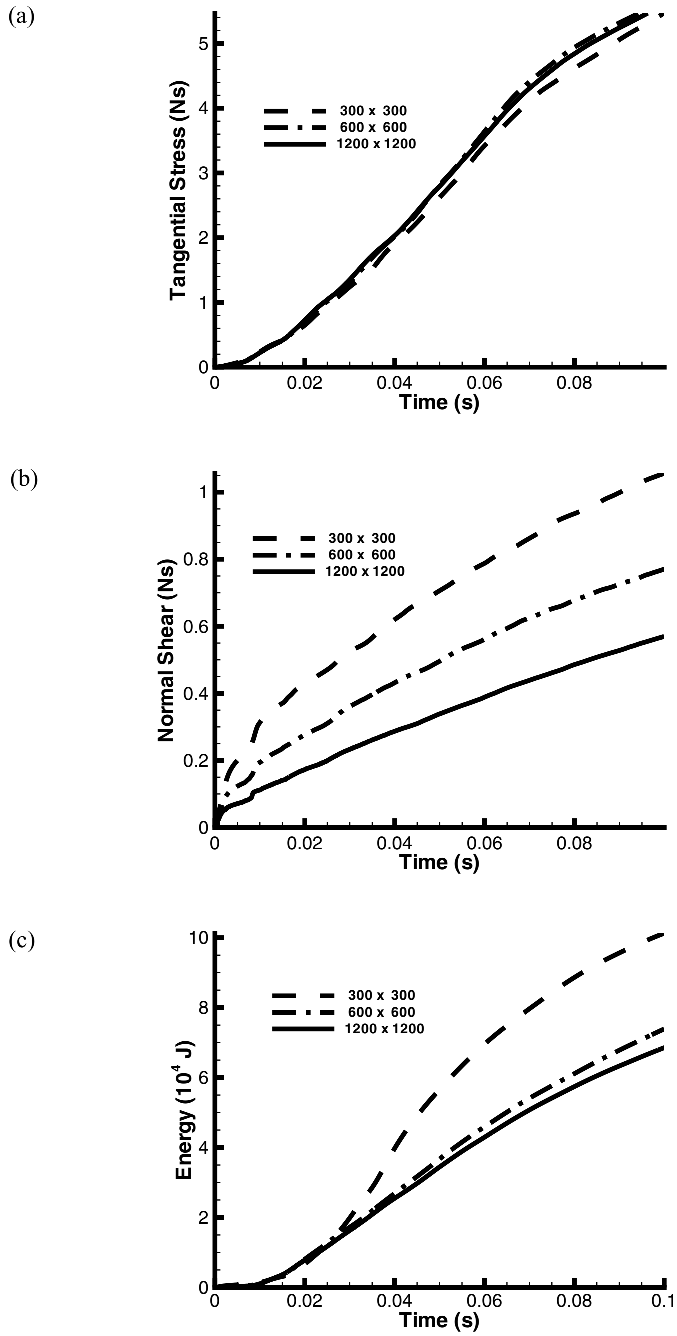

Fig. 7. Integral quantities used as performance indicators for the algorithm. Part (a) represents the integral of the parallel component of shear stress, $\tau_{\|}$ along the wall and in time. Similarly, part (b) is obtained by using the normal component of shear stress, $\tau_{\perp}$ as an integrand. Total energy conducted to the wall as a function of time is shown in part (c) of the figure. Each of the results is provided for the grids 300×300 (16 cm cell size), 600×600 (8 cm cell) and 1200×1200 (4 cm cell).