17th AIAA Computational Fluid Dynamics Conference, June 6–9, 2005, Toronto, Ontario, Canada

Parallel AMR Scheme for Turbulent Multi-Phase Rocket Motor Core Flows

J. S. Sachdev[*], C. P. T. Groth[†] and J. J. Gottlieb[‡]

University of Toronto Institute for Aerospace Studies

4925 Dufferin Street, Toronto, ON, M3H 5T6, Canada

The development of a parallel adaptive mesh refinement (AMR) scheme is described for solving the governing equations for turbulent multi-phase (gas-particle) core flows in solid propellant rocket motors (SRMs). The Favre-Averaged Navier-Stokes equations are solved for the gas-phase. Turbulence closure is achieved by using a two equation turbulence model. An Eulerian formulation is used to describe the motion of the inert, dilute, and disperse particle-phase. A cell-centred upwind finite-volume discretization and the use of limited solution reconstruction, Riemann solver based flux functions to determine the inviscid flux for the gas and particle phases at cell interfaces. Green-Gauss integration over the diamond-path defined at cell interfaces is used to determine the primitive-variable gradients for evaluation of the viscous fluxes. A parallel multigrid method coupled with an explicit optimally-smoothing multi-stage time-stepping scheme is used to obtain steady state solutions. Unsteady calculations are achieved through the use of a dual time-stepping approach. The propagation of the propellant-core flow interface is tracked using the level set method and a mesh adjustment scheme is used to fit the computational mesh to the location of the burning interface. Application of block-based AMR accurately resolves the multiple solution scales of the fluid flow and enables efficient and scalable parallel implementations on distributed memory multi-processor architectures. High-scalability of the model has been achieved on a parallel cluster computer consisting of 276 processors. Various numerical test cases are presented to verify the validity of the scheme as well as demonstrate the capabilities of the approach for predicting SRM core flows.

I. Introduction

Modern numerical methods are a potential tool for studying the complex characteristics of solid propellant rocket motor (SRM) internal ballistics. High-fidelity simulations, in terms of numerical accuracy and physical phenomena, are required to gain an understanding of the fluid processes involved in SRM core flows. The flow of the propellant products from the combustion interface into the combustion chamber, through the nozzle, and into the plume of the rocket is a high temperature, multi-phase, chemically reactive, turbulent flow. The Mach number ranges from low subsonic values in the combustion chamber, to sonic at the throat of the nozzle, and supersonic at the nozzle exit. In addition, the combustion of the solid propellant at the combustion interface occurs in a thin, high temperature layer between the propellant grain and the main flow cavity. This topologically complex surface evolves as the propellant burns and the rate of regression is dependent on various steady and unsteady effects. Particles, reactive or non-reactive, are often added to the propellant to stabilize the internal flow due to possible combustion instabilities. Reactive particles also act as a fuel, injecting hot gas into the combustion chamber as they burn. The particles, however, can have detrimental effects on the rocket motor, causing excessive erosion of the throat and nozzle, and altering the effective thrust and choking of the rocket motor.

[*] Ph.D. Candidate, Student Member AIAA, sachdev@utias.utoronto.ca

[†] Associate Professor, Senior Member AIAA, groth@utias.utoronto.ca

[‡] Professor, gottlieb@utias.utoronto.ca

Due to the complex nature of SRM internal ballistics, high-fidelity numerical prediction of these fluid flows are computationally expensive. Nevertheless, a number of engineers and scientists have successfully designed and applied numerical schemes to study these problems. Much of the recent numerical SRM research has been directed toward the investigation of the characteristics of the turbulent flow structure, the modelling of reactive and non-reactive particles (with and with-out particle-particle collision processes), and the interaction between the turbulent flow and the particle trajectories. In terms of turbulence modelling, solution of the Favre-averaged Navier-Stokes equations where turbulence closure is achieved using two-equation turbulence models[1,2,3,4,5,6,7] and large eddy simulation[8,9] have both been used. Numerical simulation of the particle-phase has been achieved using both Lagrangian[3,4,2,7] and Eulerian[10,11,5] methods. Lagrangian tracking methods more-easily allow for the inclusion of complex particle processes (such as combustion, break-up, and agglomeration); however, Eulerian techniques can be less computationally expensive. Dupays *et. al* have performed reactive multi-phase computations using both Lagrangian and Eulerian methods and their findings tend to support these statements. Najjar *et al.*[9] used a Lagrangian method to track larger particles/droplets and a *fast-Eulerian* method[12] to model smaller smoke particles. The latter takes advantage of the relatively small relaxation time-scale of the smoke particles. The work of Najjar *et al.*[9] was carried out as part of the whole-system simulation research program of the Center for Simulation of Advanced Rockets at the University of Illinois at Urbana-Champaign.[13] The propagation of the burning interface was included in this project using a grid deformation technique.[14] Karimian and Amoli also performed SRM internal ballistic simulations with a moving combustion interface using a grid deformation approach.[15] In their work, additional triangular cells were inserted when the elements became highly stretched.

The purpose of the current research is to develop a computational framework for predicting turbulent multi-phase (gas-particle) SRM core flows. The cornerstone of this numerical scheme is the use of a parallel block-based adaptive mesh refinement algorithm. This allows for accurate resolution of the multiple solution scales associated with SRM internal ballistics. This paper is an extension of our previous work involving the solution of inviscid multi-phase SRM core flows.[16,17] In addition, the algorithm is extended to allow for propagation of the propellant-core flow interface. This parallel algorithm has been devised with a view to enabling the computation of complex rocket motor flows on a more routine basis.

The Favre-Averaged Navier-Stokes equations are solved here for the gas-phase. Turbulence closure is achieved by using a two-layer two equation $k-\epsilon$ turbulence model.[18,19] An Eulerian formulation is used to describe the motion of the particle-phase which is assumed to be inert, dilute, and disperse. The governing partial differential equations are solved in conservation form using a cell-centred upwind finite-volume discretization procedure on multi-block quadrilateral mesh. Limited linear solution reconstruction and Riemann-solver based flux functions are used to evaluate the hyperbolic numerical fluxes for the gas and particle-phases. The elliptic (viscous) numerical fluxes require the calculation of the gradient of the primitive variable solution state at cell interfaces. This is determined by computing a Green-Gauss integration over the diamond-path defined at the cell face. Block-based adaptive mesh refinement (AMR) is used to accurately resolve the multiple solution scales associated with SRM flows. This AMR scheme has successfully been applied to laminar and turbulent diffusion flames,[20,21] micron-scale flows,[22] and inviscid multi-phase SRM core flows.[16,17] A flexible hierarchical data structure has implemented to facilitate automatic solution-directed mesh adaptation according to physics-based refinement criteria. This block-based data structure lends itself naturally to domain decomposition and thereby enabled efficient and scalable implementation of the algorithm on distributed-memory multi-processor architectures. For steady state problems, a parallel multigrid method coupled with an explicit optimally-smoothing multi-stage temporal discretization scheme.[23] Unsteady calculations are achieved through the use of a dual time-stepping approach.[24] The motion of the propellant-core flow interface is tracked by coupling the flow solver with the level set method[25,26] where the propagation speed of the burning propellant is determined from a pressure-dependent burning law. A scheme has been devised to adjust the mesh to location of the embedded burning interface to facilitate the application of boundary conditions and the finite volume scheme in a straight-forward manner.[27]

The governing equations and the numerical algorithms outlined above are described in the next section. Various numerical results are presented in the subsequent sections that verify the validity of the proposed scheme and that illustrate the capabilities of the approach for predicting SRM core flows. The reference rocket motor chosen is based on the CRV-7 rocket system, which was developed for military applications. The characteristics of the reference SRM and the propellant are summarized in Table 1. The CRV-7 rocket system contains a non-aluminized propellant composed of 80% oxidizer (ammonium perchlorate, AP) and 20% fuel (hydroxyl terminated polybutadiene, HTPB). As is typical in some tactical rocketry, inert aluminium oxide particles account for 2-3% of the solid propellant by mass. Therefore, the modelling of burning particles, particle-particle collision processes (such as agglomeration), and the transport of smoke are not required for the present work and are left for future research. In addition, the propellant ignition is assumed to be instantaneous and the burning rate dependent only on the local pressure. Erosive burning[28,29,30] and transient effects[31,32] are left for future consideration.

II. Numerical Method

A. Semi-Discrete Form of the Governing Equations

An Eulerian formulation is used to describe the behaviour of the gas and particle phases in the rocket motor core flow.[33,34,35,16,36,37,17] The Favre-Averaged Navier-Stokes equations are used to describe the behaviour of the gas-phase and a set of equations similar to the Euler equations of inviscid compressible gas-dynamics are used to describe the behaviour of the particle-phase. The particle-phase is assumed to be inert (non-reacting), dilute (negligible volumetric fraction), and disperse (no particle-particle interactions).[38] Since the particle-phase is considered to be disperse, there can be no particle-pressure or particle-viscosity acting on the particle-phase. As a consequence of the assumptions of an inert and dilute particle phase, there is no interaction between the phases due to mass transfer or volume effects. There is, however, a strong interaction between the relatively heavy solid particles and the gas due to momentum transfer (drag) between the two phases. Typical particle density, ρ_p, to gas density, ρ, ratios are large for rocket motors ($\rho_p/\rho \sim 10^3$). Heat transfer between the phases also occurs for cases where the phases have different temperatures.

It has been well-established that the set of equations governing dilute and disperse particle flows is both hyperbolic and degenerate.[33,35,16] A discussion of the cause and implications of the degeneracy as well as a dispersion analysis of the inviscid form of the governing equations was presented in Ref. 16. Lagrangian formulations, which track and solve for the individual particle motions, can avoid the problems associated with degeneracy of the Eulerian formulations. However, Lagrangian methods generally require significantly more computing resources than an Eulerian approach, in terms of both memory and computing time. Efficient and scalable parallel implementations of Lagrangian particle tracking formulations are also difficult to achieve.

Although this study is restricted to mono-sized particles with uniform physical properties, the Eulerian framework adopted here can be readily modified to cope with particles having a distribution of sizes and characteristics. This can be accomplished by considering multiple families of different sized particles, each with there own mass, momentum, and energy. An additional set of particle-phase continuity, momentum, and energy equations for each additional particle family would then be incorporated into the mathematical formulation and solved along with the gas-phase equations.

For the gas-phase, the turbulent Reynolds stress tensor and heat flux vector are computed using a two-layer two equation k–ϵ turbulence model.[18,19] This requires the solution of two additional partial differential equations for the turbulent kinetic energy, k, and the dissipation rate, ϵ. In the two-layer approach, the standard $k-\epsilon$ turbulence model is used away from the wall and the one-equation model of Wolfshtein is used in the near wall region.[39] The use of the Wolfshtein model allows for solutions in the near-wall or low Reynolds number regions without resorting to wall function formulations. In terms of the influence of the particles on the turbulent flow, Peirano and Leckner discuss the classification of turbulent flow regimes for gas-particle flows and identified five mechanisms for turbulent modulation: (1) dissipation of turbulent kinetic energy by the particles, (2) increase of turbulent viscosity due to the presence of the particles, (3) shedding of vortices behind the particles (turbulent production), (4) fluid moving with the particles as added fluid mass, and (5) enhancement of the velocity gradient between two particles.[40] Mechanisms (2) and (5) can be neglected for a dilute and disperse particle-phase. Particle diameters are small for typical rocket motors ($d_p \sim 10^{-6}$ m) and, therefore, turbulent production mechanisms (3) and (4) can also be ignored. Due to the large relative mass of the particles the dissipation of turbulent kinetic energy by the particles may be important, however, this mechanism is neglected in this study.

In this work, a higher-order finite-volume scheme is used to solve the coupled gas-particle flow equations. The inviscid and viscous flux evaluation and time-integration techniques will be outlined in subsequent sections. In the proposed finite-volume approach, the governing equations are integrated over quadrilateral cells of a body-fitted multi-block quadrilateral mesh. When applied to cell (i, j), this results in the following semi-discrete form of the equations:

$$\Gamma \frac{\mathrm{d}\mathbf{U}}{\mathrm{d}\tau}\bigg|_{ij} + \frac{\mathrm{d}\mathbf{U}}{\mathrm{d}t}\bigg|_{ij} = -\frac{1}{A_{ij}} \sum_k \left(\vec{\mathbf{F}}_{ij_k} - \vec{\mathbf{w}}_k \mathbf{U}_{ij_k} - \vec{\mathbf{G}}_{ij_k}\right) \cdot \hat{\mathbf{n}}_{ij_k} \, \Delta\ell - \mathbf{S}_{ij} + \mathbf{T}_{ij} + \mathbf{P}_{ij} + \left(\frac{\mathbf{U}}{A}\frac{\mathrm{d}A}{\mathrm{d}t}\right)\bigg|_{ij}, \tag{1}$$

where A_{ij} is the area of cell (i, j), $\Delta\ell$ is the length of the cell face k, and $\hat{\mathbf{n}}$ is the unit vector normal to the cell face k. The vector $\mathbf{U}$ represents the conserved variable solution vector,

$$\mathbf{U} = \left[\, \rho,\ \rho v_r,\ \rho v_z,\ E,\ \rho k,\ \rho\epsilon,\ \sigma_p,\ \sigma_p u_r,\ \sigma_p u_z,\ E_p \,\right]^T, \tag{2}$$

where σ_p is the mass concentration of the solid particles, v_r and v_z are the radial and axial components of the gas velocity $\vec{\mathbf{v}}$, u_r and u_z are the radial and axial components of the particle velocity $\vec{\mathbf{u}}$, and E and E_p are the total energy per unit volume of the gas and particle-phases. The gas-phase is taken to be calorically perfect and the total energies per unit volume of the two phases are given by

$$E = \frac{p}{(\gamma - 1)} + \frac{\rho}{2}(\vec{\mathbf{v}}\cdot\vec{\mathbf{v}}) + \rho k = \rho c_v T + \frac{\rho}{2}(\vec{\mathbf{v}}\cdot\vec{\mathbf{v}}) + \rho k, \tag{3}$$

$$E_p = \sigma_p c_m T_p + \frac{\sigma_p}{2}(\vec{\mathbf{u}}\cdot\vec{\mathbf{u}}), \tag{4}$$

where $\gamma = c_p/c_v$ is the ratio of the specific heats for the gas, p is the gas-phase pressure, c_m is the specific heat of the particles, and T_p is the particle temperature. The ideal gas law, $p = \rho \mathrm{R} T = \rho a^2/\gamma$, provides a relationship between the gas pressure, p, density, ρ, and temperature, T, where $a = \sqrt{\gamma \mathrm{R} T}$ is the sound speed and R is the gas constant.

The dyadic quantities $\vec{\mathbf{F}} = [\mathbf{F_r}, \mathbf{F_z}]$ and $\vec{\mathbf{G}} = [\mathbf{G_r}, \mathbf{G_z}]$ are the inviscid and viscous flux dyads having components in the r and z directions

$$\mathbf{F_r} = \left[\rho v_r,\ \rho v_r^2 + p + \frac{2}{3}\rho k,\ \rho v_r v_z,\ v_r\left(E + p + \frac{2}{3}\rho k\right),\ \rho v_r k,\ \rho v_r \epsilon,\ \sigma_p u_r,\ \sigma_p u_r^2,\ \sigma_p u_r u_z,\ u_r E_p\right]^T, \tag{5}$$

$$\mathbf{F_z} = \left[\rho v_z,\ \rho v_r v_z,\ \rho v_z^2 + p + \frac{2}{3}\rho k,\ v_z\left(E + p + \frac{2}{3}\rho k\right),\ \rho v_z k,\ \rho v_z \epsilon,\ \sigma_p u_z,\ \sigma_p u_r u_z,\ \sigma_p u_z^2,\ u_z E_p\right]^T, \tag{6}$$

$$\mathbf{G_r} = \left[0,\ \tau_{rr},\ \tau_{rz},\ -q_r + v_r\tau_{rr} + v_z\tau_{rz} + (\mu + \sigma_k\mu_T)\frac{\partial k}{\partial r},\ (\mu + \sigma_k\mu_T)\frac{\partial k}{\partial r},\ (\mu + \sigma_\epsilon\mu_T)\frac{\partial \epsilon}{\partial r},\ 0,\ 0,\ 0,\ 0\right]^T, \tag{7}$$

$$\mathbf{G_z} = \left[0,\ \tau_{rz},\ \tau_{zz},\ -q_z + v_r\tau_{rz} + v_z\tau_{zz} + (\mu + \sigma_k\mu_T)\frac{\partial k}{\partial z},\ (\mu + \sigma_k\mu_T)\frac{\partial k}{\partial z},\ (\mu + \sigma_\epsilon\mu_T)\frac{\partial \epsilon}{\partial z},\ 0,\ 0,\ 0,\ 0\right]^T. \tag{8}$$

where τ_{ij} is the combined laminar and Reynolds stress tensor

$$\tau_{rr} = 2(\mu + \mu_T)\left(\frac{\partial u}{\partial r} - \frac{1}{3}\nabla\cdot\vec{\mathbf{v}}\right), \quad \tau_{rz} = (\mu + \mu_T)\left(\frac{\partial u}{\partial z} + \frac{\partial v}{\partial r}\right),$$
$$\tau_{zz} = 2(\mu + \mu_T)\left(\frac{\partial v}{\partial z} - \frac{1}{3}\nabla\cdot\vec{\mathbf{v}}\right), \quad \tau_{\theta\theta} = 2(\mu + \mu_T)\left(\frac{u}{r} - \frac{1}{3}\nabla\cdot\vec{\mathbf{v}}\right).$$

The kinetic energy term of the Reynolds stress tensor is included in the inviscid flux term since it must be solved in an upwind manner to avoid unstable solutions.[41,42,43] The standard $k-\epsilon$ turbulence model makes use of the Boussinesq approximation and relates the Reynolds stresses to the strain rates via turbulent eddy-viscosity. The eddy-viscosity, ν_T, and thermal conductivity, κ_T, are given by

$$\nu_T = \frac{\mu_T}{\rho} = c_\mu \frac{k^2}{\epsilon}, \qquad \kappa_T = \frac{\mu_T c_p}{\mathrm{Pr}_T}, \tag{9}$$

where c_μ is a model constant and Pr_T is the turbulent Prandtl number ($\mathrm{Pr}_T = 0.9$ is typical for many turbulent flows). The combined laminar and turbulent heat flux vector is given by

$$\vec{\mathbf{q}} = -(\kappa + \kappa_T)\left[\frac{\partial T}{\partial r}, \frac{\partial T}{\partial z}\right].$$

The vector, $\mathbf{S} = \big(\mathbf{S_{inv}} - \vec{\mathbf{w}}\mathbf{U} - \mathbf{S_{vis}}\big)/r$, represents inviscid and viscous sources associated with the axisymmetric geometry where $\mathbf{S_{inv}}$ and $\mathbf{S_{vis}}$ are given by

$$\mathbf{S_{inv}} = \left[\rho v_r,\, \rho v_r^2,\, \rho v_r v_z,\, v_r\left(E + p + \frac{2}{3}\rho k\right),\, \rho v_r k,\, \rho v_r \epsilon,\, \sigma_p u_r,\, \sigma_p u_r^2,\, \sigma_p u_r u_z,\, u_r E_p\right]^T, \tag{10}$$

$$\mathbf{S_{vis}} = \left[0,\, \tau_{rr} - \tau_{\theta\theta},\, \tau_{rz},\, -q_r + v_r\tau_{rr} + v_z\tau_{rz} + (\mu + \sigma_k\mu_T)\frac{\partial k}{\partial r},\, (\mu + \sigma_k\mu_T)\frac{\partial k}{\partial r},\, (\mu + \sigma_\epsilon\mu_T)\frac{\partial \epsilon}{\partial r},\, 0,\, 0,\, 0,\, 0\right]^T \tag{11}$$

The source vector $\mathbf{T}$ contains the turbulent source terms and has the form

$$\mathbf{T} = \left[0,\, 0,\, 0,\, 0,\, \vec{\vec{\lambda}} : \nabla\vec{\mathbf{v}} - \rho\epsilon,\, c_1\frac{\epsilon}{k}(\vec{\vec{\lambda}} : \nabla\vec{\mathbf{v}}) - c_2\rho\frac{\epsilon^2}{k},\, 0,\ \ 0,\ \ 0,\ \ 0\right]^T, \tag{12}$$

where $\vec{\vec{\lambda}} : \nabla\mathbf{v}$ is the turbulent production term, $\vec{\vec{\lambda}}$ is the Reynolds stress tensor, and the closure coefficients for the turbulence model are $(\sigma_k, \sigma_\epsilon, c_\mu, c_1, c_2) = (1.0, 10/13, 0.09, 1.44, 1.92)$.

In the near-wall region, the Wolfshtein one-equation turbulence model only requires the solution of the kinetic energy equation. The dissipation rate, ϵ, is determined using the algebraic relation

$$\epsilon = \frac{k^{3/2}}{\ell_\epsilon}\,. \tag{13}$$

Near-wall damping effects are modelled by introducing the damping functions f_μ and f_ϵ which can be expressed in terms of the length scales ℓ_μ and ℓ_ϵ as follows:

$$f_\mu = \frac{\ell_\mu}{c_\ell\, y_w} = 1 - \exp\left(-\frac{\mathrm{Re_t}}{A_\mu}\frac{25}{A^+}\right), \qquad f_\epsilon = \frac{\ell_\epsilon}{c_\ell\, y_w} = 1 - \exp\left(-\frac{\mathrm{Re_t}}{A_\epsilon}\right), \tag{14}$$

where the turbulent Reynolds number, $\mathrm{Re_t}$, is given by $\mathrm{Re_t} = k^2 y_w/\nu$. The parameter y_w is the distance to the nearest wall and the function A^+ is a correction used when the nearest wall is a burning surface with mass injection. The turbulent viscosity is then determined using

$$\nu_T = \frac{\mu_T}{\rho} = c_\mu\frac{f_\mu}{f_\epsilon}\frac{k^2}{\epsilon}\,. \tag{15}$$

The additional closure coefficients are given by $(c_\ell, \kappa, A_\mu, A_\epsilon) = (\kappa c_\mu^{-3/4}, 0.41, 70.0, 2c_\ell)$ where κ is the von Kármán constant. Typically, the near-wall region is defined to be region for which $\mathrm{Re_t} < 200$.

Note that in the two-layer approach described above, the one-equation Wolfshtein model can be integrated right to the wall. Generally, this requires mesh spacing in the near wall region such that the first mesh point is located within $y^+ \leq$ 2-2.5 and there are a number of points within $y^+ = 10$, where $y^+ = u_\tau y_w/\nu$, $u_\tau^2 = \tau_w/\rho$, and τ_w is the wall shear stress. In situations where the mesh refinement does not permit direct integration of the k equation to the wall, the standard wall functions are used. In this case, the expressions $u/u_\tau = (1/\kappa)\ln(y^+) + C$, $k = u_\tau^2/\sqrt{c_\mu}$, and $\epsilon = u_\tau^3/\kappa y_w$ are used to fully specify k and ϵ for $y^+ \leq$30-250.

Interaction between the phases occurs due to momentum (drag) and heat transfer caused by the collisions between the solid particles and the gas molecules. This collision process tends to relax the system into an equilibrium state in which the two phases have equal velocity and temperature. The relaxation times associated with the momentum and heat transfer between the gas and particle-phases are

$$\tau_v = \frac{m_p}{3\pi d_p \mu f(\mathrm{Re_p})}\,, \qquad \tau_T = \frac{m_p c_p}{2\pi d_p \kappa}\,, \tag{16}$$

where m_p is the particle mass, d_p is the particle diameter, and c_p, μ, and κ, are the gas specific heat at constant pressure, viscosity, and thermal conductivity, respectively. Note that the ratio of the relaxation times can be related to the Prandtl number of the gas, $\tau_T/\tau_v = \frac{3}{2}\mathrm{Pr}$. The function $f(\mathrm{Re_p})$ is a correction to the Stokes drag law, $f(\mathrm{Re_p}) = 1 + \frac{1}{6}\mathrm{Re_p}^{2/3}$, which is valid for slip Reynolds numbers, $\mathrm{Re_p} = (\rho d_p/\mu)\big|\vec{\mathbf{v}} - \vec{\mathbf{u}}\big|$, satisfying $\mathrm{Re_p} < 1000$.[38] The vector $\mathbf{P}$ contains the gas-particle interaction source terms and has the form

$$\mathbf{P} = \left[0,\, -\frac{\sigma_p}{\tau_v}(v_r - u_r), -\frac{\sigma_p}{\tau_v}(v_z - u_z), -\frac{\sigma_p}{\tau_v}(\vec{\mathbf{v}} - \vec{\mathbf{u}})\cdot\vec{\mathbf{u}} - \frac{\sigma_p c_p}{\tau_T}(T - T_p),\, 0,\, 0,\right.$$
$$\left. 0,\, \frac{\sigma_p}{\tau_v}(v_r - u_r),\, \frac{\sigma_p}{\tau_v}(v_z - u_z),\, \frac{\sigma_p}{\tau_v}(\vec{\mathbf{v}} - \vec{\mathbf{u}})\cdot\vec{\mathbf{u}} + \frac{\sigma_p c_p}{\tau_T}(T - T_p)\right]^T. \tag{17}$$

In simulations involving moving boundaries, such as the propellant grain of a rocket motor, the velocity of the boundary and the rate of change of cell areas must be accounted for. The transport of the conserved variables due to the motion of the boundaries is accounted for by including the cell interface velocity, $\vec{\mathbf{w}}$, as included in Equation (1). The final term on the right-hand side of Equation (1) is the rate of change of the cell area which can be approximated by the geometric conservation law, which is written in semi-discrete form as

$$\frac{\mathrm{d}A}{\mathrm{d}t} = \sum_{k} \vec{\mathbf{w}} \cdot \hat{\mathbf{n}} \, \Delta\ell \, , \tag{18}$$

which states that the change in cell area is equal to the area swept by the moving surfaces.[44]

B. Spatial Discretization

In this work, the hyperbolic flux, which includes the gas-phase inviscid flux, the particle-phase flux, and the transport due to the moving boundary, is determined using a high-order Godunov scheme. Upwind finite-volume schemes for the gas-dynamic equations were originally introduced by Godunov in 1959.[45] Application and development of these schemes for the gas-dynamic equations has been well documented in literature.[46,47,48,49,50,51,52,53,54,55,56,57,58,59] The hyperbolic numerical fluxes at the faces of each cell are determined from the solution of a Riemann problem. Given the left and right solution state vectors, $\mathbf{U}_l$ and $\mathbf{U}_r$, and velocity $\vec{\mathbf{w}}$ of the cell interface, the numerical flux is given by

$$(\vec{\mathbf{F}} - \vec{\mathbf{w}}\mathbf{U}) \cdot \hat{\mathbf{n}} = \mathcal{F}(\mathbf{U}_l, \mathbf{U}_r, \vec{\mathbf{w}}, \hat{\mathbf{n}}) \, , \tag{19}$$

where $\mathcal{F}$ is evaluated by solving the Riemann problem in a direction defined by the normal to the face, $\hat{\mathbf{n}}$. The left and right solution states are determined using limited piece-wise linear solution reconstruction.[60] The modified limiter of Venkatakrishnan has also been implemented.[61] The velocity of the cell interface can either be prescribed or dependent on the local flow variables.

Frozen flow conditions are assumed for the solution of the Riemann problem. In the frozen flow limit, the phase interactions terms vanish and the gas and particle-phases fully decouple. Therefore, separate Riemann problems and solutions can be formulated for the two phases. The gas-phase Riemann solutions must account for the flux contribution by the turbulent kinetic energy as well as moving boundary transport terms. Roe's approximate Riemann solver[52] has been reformulated to account for these additional terms. Here there are six eigenvalues, given for the x-component flux as

$$\lambda_1 = u - w - c, \quad \lambda_{2,3} = u - w, \quad \lambda_4 = u - w + c, \quad \lambda_{5,6} = u - w, \tag{20}$$

where c is a modified sound speed given by $c = \sqrt{a^2 + \frac{2}{3}\gamma k}$. The corresponding conserved variable right eigenvectors for the gas-phase are

$$\mathbf{R} = \begin{bmatrix} 1 & 1 & 0 & 1 & 0 & 0 \\ u-c & u & 0 & u+c & 0 & 0 \\ v & v & \rho & v & 0 & 0 \\ h-cu & \frac{u^2+v^2}{2} & \rho v & h+cu & \frac{3\gamma-5}{3(\gamma-1)} & 0 \\ k & 0 & 0 & k & 1 & 0 \\ \epsilon & 0 & 0 & \epsilon & 0 & 1 \end{bmatrix} , \tag{21}$$

and the characteristic variables are given by

$$\alpha = \left[\frac{1}{2c^2}\left(\frac{2}{3}k\mathrm{d}\rho + \rho c\mathrm{d}u + \mathrm{d}p + \frac{2}{3}\rho\mathrm{d}k\right), \; \left(1 - \frac{2}{3}\frac{k}{c^2}\right)\mathrm{d}\rho - \frac{\mathrm{d}p}{c^2} - \frac{2}{3}\frac{\rho}{c^2}\mathrm{d}k, \; \mathrm{d}v, \right.$$
$$\left. \frac{1}{2c^2}\left(\frac{2}{3}k\mathrm{d}\rho - \rho c\mathrm{d}u + \mathrm{d}p + \frac{2}{3}\rho\mathrm{d}k\right), \; \mathrm{d}k, \; \mathrm{d}\epsilon \right]. \tag{22}$$

For the particle-phase, the Riemann solver proposed by Saurel *et al.*[33] has been implemented. The limitations of this Riemann solver are discussed by Saurel *et al.*[33] and in our previous work.[16] To account for the transport due to the moving boundary, the intermediate solution state for the particle-phase Riemann problem is found in the reference frame of the moving boundary. The solution state is then transformed back into the inertial frame where the interface particle-phase flux can be calculated.

Computation of a Green-Gauss integration over the diamond-path defined at the cell interfaces to determine the primitive-variable gradients for evaluation of the viscous (elliptic) fluxes.[62] The primitive variable solution states at cell corners are determined using bilinear interpolation.[63,64] At solid wall boundaries (adiabatic, isothermal, or transpiring walls), the diamond path integration is limited to a single triangle since the primitive variable solution state on the boundary are known.

C. Temporal Discretization

The semi-discrete form of the conservation equations can be written in a dual-time formulation where t is the physical time and τ is the pseudo-time.[24] For steady-state problems, the derivative of the conserved state variables with respect to the physical time is ignored and the steady-state solution is achieved ($\tau \to \infty$) through the use of a parallel multigrid method coupled with an explicit optimally-smoothing multi-stage temporal discretization scheme as the smoother.[23] Both V and saw-tooth multigrid cycles have been implemented and a full-multigrid cycle can also be used for start-up purposes. The matrix Γ is a preconditioner which can be used to improve the rate of solution convergence or improve the conditioning of the compressible flow equations in the low Mach number limit.[65,66,67,68] Unsteady calculations are achieved through the use of the full dual time-stepping approach.[24,69,70,71,72,73] Here, the pseudo-time residual is converged to steady-state using the parallel multigrid method described above. The physical time derivative can be formed by standard time-marching methods and an implicit formulation is considered here.

The semi-discrete form of the equations, Equation (1), can be rewritten in a more compact form as

$$\frac{\mathrm{d}\mathbf{U}}{\mathrm{d}\tau} + \mathbf{R}^*(\mathbf{U}) = 0\,, \tag{23}$$

where the preconditioner, Γ, has been neglected and $\mathbf{R}^*(\mathbf{U})$ is the dual time residual defined by

$$\mathbf{R}^*(\mathbf{U}) = \frac{\mathrm{d}\mathbf{U}}{\mathrm{d}t} + \mathbf{R}(\mathbf{U})\,. \tag{24}$$

The physical time residual, $\mathbf{R}(\mathbf{U})$, contains both solution flux and source terms. The physical time derivative appearing in the pseudo-time residual can be represented discretely by an unconditionally stable implicit scheme to allow for large time-steps. A family of consistent unconditionally stable schemes are given by

$$\mathbf{R}^*(\mathbf{U}^{(m)}) = \frac{(1+\xi)\mathbf{U}^{(m)} - (1+2\xi)\mathbf{U}^{(n)} + \xi\mathbf{U}^{(n-1)}}{\triangle t} + \theta\mathbf{R}(\mathbf{U}^{(m)}) + (1-\theta+\phi)\mathbf{R}(\mathbf{U}^{(n)}) - \phi\mathbf{R}(\mathbf{U}^{(n-1)})\,, \tag{25}$$

where the coefficients for several methods can be found in Ref. 74. Second-order backward differencing with $(\theta, \xi, \phi) = (1, 1/2, 0)$ is typically used.

An explicit treatment of the pseudo-time residual results in the following general M stage time-marching scheme for integrating Equation (23) from time level n to time level $n+1$:

$$\begin{aligned} \mathbf{U}^{(0)} &= \mathbf{U}^{(n)} \\ \mathbf{U}^{(k)} &= \mathbf{U}^{(0)} + \beta_m \triangle\tau \mathbf{R}^*(\mathbf{U}^{(k-1)}) \quad \text{for } k = 1 \ldots M \\ \mathbf{U}^{n+1} &= \mathbf{U}^{M}\,. \end{aligned} \tag{26}$$

Melson *et al.*[75] showed that an implicit treatment of the physical time derivative can improve the stability characteristics of the discretization and has been adopted by many other researchers.[69,70,76,73] This modification results in a modified multi-stage scheme having the form

$$\begin{aligned} \mathbf{U}^{(0)} &= \mathbf{U}^{(n)} \\ \mathbf{U}^{(k)} &= \mathbf{U}^{(0)} + \beta_m \frac{\triangle\tau}{1 + (1+\xi)\triangle\tau/\triangle t} \mathbf{R}^*(\mathbf{U}^{(k-1)}) \quad \text{for } k = 1 \ldots M \\ \mathbf{U}^{n+1} &= \mathbf{U}^{M}\,. \end{aligned} \tag{27}$$

D. Adaptive Mesh Refinement

Adaptive mesh refinement techniques which automatically adapt the computational grid to the solution of the governing partial differential equations can be very effective in treating problems with disparate length scales. A description of the approach used here was provided in Refs. 16, 17 and follows the method developed by Groth *et al.* for computational magnetohydrodynamics.[77] A flexible block-based hierarchical data structure has been developed and is used in conjunction with the finite-volume scheme described above to facilitate automatic solution-directed mesh adaptation on multi-block quadrilateral mesh according to physics-based refinement criteria. The proposed AMR formulation borrows from previous work by Berger and co-workers,[78,79,80,81,82] Quirk,[83,84] and De Zeeuw and Powell[85] and has similarities with the block-based approaches described by Quirk and Hanebutte[84] and Berger and Saltzman.[82] A primary distinction of this work is the use of curvilinear (arbitrary quadrilateral) mesh as opposed to the Cartesian mesh that is used in most of the previous work. The use of quadrilateral mesh blocks makes the application of the block-based AMR more amenable to flows with thin boundary layers and permits anisotropic refinement as dictated by the initial mesh stretching. Note that cell-based AMR schemes on curvilinear mesh have been explored in previous work by Davis and Dannenhoffer,[86] Sun and Takayama,[87] and Bramkamp *et al.*[88]

In this work, the governing equations are integrated to obtain area-averaged solution quantities within quadrilateral computational cells and these cells are embedded in structured blocks consisting of $N_x \times N_y$ cells, where N_x and N_y are even, but not necessarily equal integers. Solution data associated with each block are stored in indexed array data structures and it is therefore straightforward to obtain solution information from neighbouring cells within blocks. Mesh adaptation is accomplished by the dividing and coarsening of appropriate solution blocks. In regions requiring increased cell resolution, a "parent" block is refined by dividing itself into four "children" or "offspring". Each of the four quadrants or sectors of a parent block becomes a new block having the same number of cells as the parent and thereby doubling the cell resolution in the region of interest. This process can be reversed in regions that are deemed over-resolved and four children are coarsened into a single parent block. The mesh refinement is constrained such that the grid resolution changes by only a factor of two between adjacent blocks and the minimum resolution is not less than that of the initial mesh. Standard multi-grid-type restriction and prolongation operators are used to evaluate the solution on all blocks created by the coarsening and division processes, respectively. Although several approaches are possible, for this study, the coarsening and division of blocks are directed using multiple physics-based refinement criteria.[89,90,91] Six flow quantities or refinement criteria, ϵ_k, are used herein:

$$\epsilon_1 \propto \left|\nabla\rho\right|, \quad \epsilon_2 \propto \left|\nabla\cdot\mathbf{v}\right|, \quad \epsilon_3 \propto \left|\nabla\times\mathbf{v}\right|, \tag{28}$$

$$\epsilon_4 \propto \left|\nabla\sigma_p\right|, \quad \epsilon_5 \propto \left|\nabla\cdot\mathbf{u}\right|, \quad \epsilon_6 \propto \left|\nabla\times\mathbf{u}\right|. \tag{29}$$

The first three quantities correspond to local measures of the density gradient, compressibility, and vorticity of the gas-phase and enable the detection of contact surfaces, shocks, and shear layers. Refinement criteria for the particle-phase are defined by the next three quantities which provide local measures of the particle concentration gradient and the divergence and vorticity of the particle velocity field. These quantities will enable the detection of high and low (vacuum) particle concentrations, particle-phase compression and expansion waves, and particle-phase shear layers.

In order that the solution algorithm for the multi-phase flow equations can be applied to all blocks in a more independent manner, some solution information is shared between adjacent blocks having common interfaces. This information is stored in an additional two layers of overlapping "ghost" cells associated with each block. At interfaces between blocks of equal resolution, these ghost cells are simply assigned the solution values associated with the appropriate interior cells of the adjacent blocks. At resolution changes, restriction and prolongation operators, similar to those used in block coarsening and division, are employed to evaluate the ghost cell solution values. Within the AMR approach, additional inter-block communication is also required at interfaces with resolution changes to strictly enforce the flux conservation properties of the finite-volume scheme.[79,80] In particular, the interface fluxes computed on more refined blocks are used to correct the interface fluxes computed on coarser neighbouring blocks and ensure that the fluxes are conserved at block interfaces.

A hierarchical tree-like data structure with multiple "roots", multiple "trees", and additional interconnects between the "leaves" of the trees is used to keep track of mesh refinement and the connectivity between solution blocks. This interconnected "forest" data structure is depicted in Figure 1. The blocks of the initial

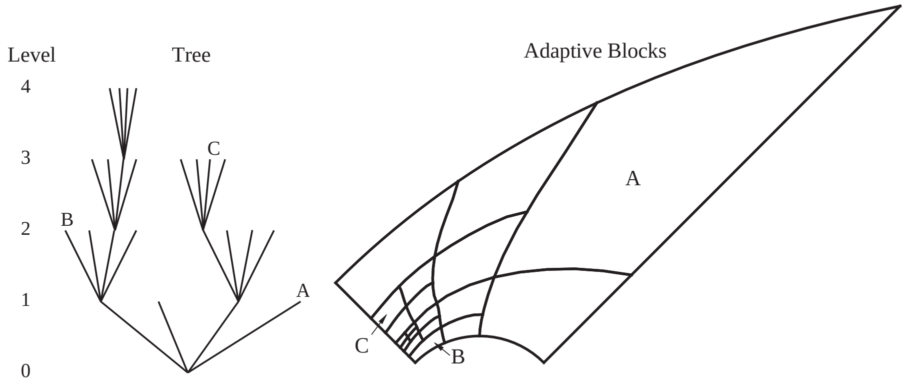


Figure 1. ***Solution blocks of a computational mesh with four refinement levels originating from one initial block and the associated hierarchical quadtree data structure. Interconnects to neighbours are not shown.***

mesh are the roots of the forest which are stored in an indexed array data structure. Associated with each root is a separate "quadtree" data structure that contains all of the blocks making up the leaves of the tree created from the original parent blocks during mesh refinement. One of the advantages of the hierarchical quadtree data structure is that it readily permits local mesh refinement at point in a calculation. Local modifications to the multi-block mesh can be performed without re-gridding the entire mesh and re-calculating solution block connectivity.

E. Parallel Implementation

Although the block-based AMR approach described above is somewhat less flexible and incurs some inefficiencies in solution resolution as compared to a cell-based approaches (i.e., for the same solution accuracy, generally more computational cells are introduced in the adapted grid), the block-based method offers many advantages over cell-based techniques when parallel implementation of the solution algorithm is considered and computational performance issues are taken into account. In particular, the multi-block quadrilateral mesh and quadtree data structure lends itself naturally to domain decomposition and thereby enables efficient and scalable implementations of the solution algorithm for the two-phase flow equations on distributed-memory multi-processor architectures.

A parallel implementation of the block-based AMR scheme has been developed using the C++ programming language and the MPI (message passing interface) library. Use of these standards greatly enhances the portability of the computer code and has enabled very good parallel performance. Domain decomposition is carried out by merely farming the solution blocks out to the separate processors, with more than one block permitted on each processor. A simple stack is used to keep track of available (open) processors. For homogeneous architectures with multiple processors all of equal speed, an effective load balancing is achieved by exploiting the self-similar nature of the solution blocks and simply distributing the blocks equally among the processors. In doing so, all blocks are treated equally and, currently, no use is made of the hierarchical data structure nor grid partitioning techniques to preferentially place neighbouring blocks on the same processors. With 10 blocks per processor, the maximum load imbalance attained by this simple block distribution procedure is less than 10% (near perfect load balancing is achieved if the number of block is a exact multiple of the number of available processors). For heterogeneous parallel machines, such as a network of workstations and computational grids, a weighted distribution of the blocks can be adopted to preferentially place more blocks on the faster processors and less blocks on the slower processors.

In order to carry out mesh refinement and inter-block communication, a complete copy of the hierarchical quadtree data structure is stored on each processor. This is possible because, unlike cell-based unstructured meshing techniques, the block-based tree data structure is not overly large. The structure need only retain the connectivity between the solution blocks as opposed to a complete map of the cell connectivity required by general unstructured mesh procedures. Inter-processor communication is mainly associated with block interfaces and involves the exchange of ghost-cell solution values and conservative flux corrections at every stage of the multi-stage time integration procedure. Message passing of the ghost-cell values and flux corrections is performed in an asynchronous fashion with gathered wait states and message consolidation, and as such, typically amounts to less than 5-8% of the total processor time.

F. Burning Surface Boundary Condition

The combustion of the solid propellant of the rocket motor occurs in a thin, high temperature layer between the solid propellant and the main flow cavity, known as the combustion interface. This layer is assumed to be small relative to the diameter of the rocket motor and large relative to the product of the propellant product velocities and chemical reaction relaxation times such that the finite-rate nature of the chemical reactions can be neglected and the injected gas-particle products can be assumed to be in chemical equilibrium. Boundary conditions for a regressing burning surface that injects gas-particle products into the flow cavity is formulated in terms of a Riemann problem and is used here to specify boundary data at the surface of the burning propellant. The burning rate is assumed to be dependent only on the local pressure as given by the empirical St. Robert relation.[92] An iterative solution of the burning surface Riemann problem was described in our previous work.[16,17] The treatment of the burning propellant boundary is very similar in spirit to the methods proposed by Gottlieb and Groth[93] for imposing boundary data at a variety of flow boundaries based on the solution of Riemann problems.

The eddy-viscosity length scale, Equation 14, included a parameter A^+ which can be adjusted to account for mass injection at the wall. The correction used by Tseng and Yang[1] and Cai *et al.*[2] is adopted here as is given by

$$A^+ = \frac{25}{5.15\left[v_w^+ + \dfrac{5.86P^+}{1+5v_w^+}\right] + 1.0}\,, \tag{30}$$

where v_w^+ is a non-dimensional injection rate, $v_w^+ = v_w/u_\tau$, and P^+ is a modification based on the pressure gradient. Note the effect of the pressure gradient is neglected in this study. The wall injection speed, v_w, is found from the solution of the burning surface Riemann problem and the friction velocity, u_τ, can be determined through an iterative solution of the law of wall.

G. Propellant Evolution

The evolution of the combustion interface is captured using the level set method[25,26] where the rate of regression of the burning surface is determined from the solution of the burning surface Riemann problem discussed previously. A mesh adjustment scheme has been developed in which the underlying body-fitted multi-block quadrilateral mesh is adjusted to the location of the combustion interface as it moves through the computational domain. This allows for the accurate calculation of cell areas and straight-forward application of the interface boundary condition. The mesh adjustment scheme is described in the next section. A brief description of the level set is included here.

Given the initial location of an arbitrary interface, $\Gamma(t\!=\!0)$, and a normal speed function defined on the front, F_0, the level set method computes the evolution of the interface by solving the *Hamilton-Jacobi*-type equation,

$$\frac{\partial\phi}{\partial t} + F\,|\nabla\phi| + \vec{\mathbf{U}}\cdot\nabla\phi = 0\,, \tag{31}$$

where the level set function $\phi = \phi(x,y)$ is initialized as a signed distance function computed from the interface location, $\vec{\mathbf{U}}$ is an underlying bulk flow-field, and F is a speed function computed from the normal speed function and must satisfy the condition

$$\nabla\phi\cdot\nabla F\ = 0\,, \tag{32}$$

which states that the normal front speed is extended in the normal direction from the front. The location of the front can be captured at future times by locating the zero contour of the level set function

$$\Gamma(t) = \big\{(x,y)|\phi(x,y,t)=0\big\}\,. \tag{33}$$

Due to the non-conservative nature of Equation (31), the level set function may not maintain a smooth signed-distance function. The level set function can by reinitialized as a signed distance function by iteratively solving the *Eikonal* equation

$$F|\nabla\phi| = 1\,, \tag{34}$$

on both the negative and positive sides of the zero contour with a uniform extension velocity of one at every time-step.

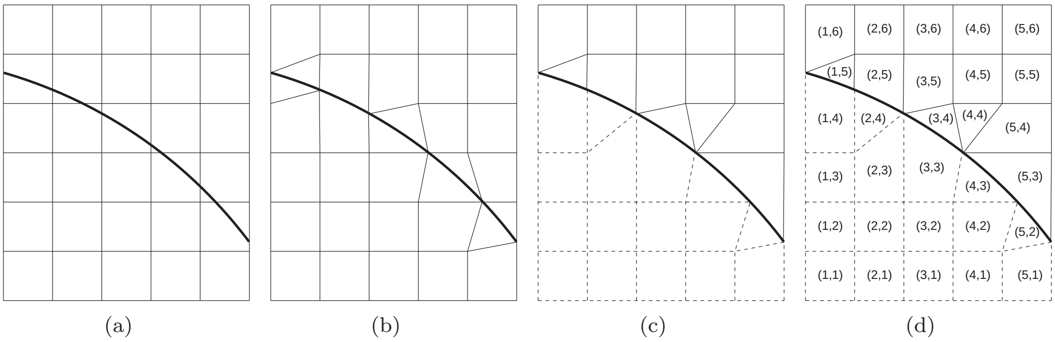
The evolution of the Hamilton-Jacobi-type Equation (31), and the associated solutions of Equations (32) and (34) are computed on a Cartesian mesh which overlaps the body-fitted mesh for the flow solution. The finite-difference solution procedure outlined by Sethian,[26] Sussman *et al.*,[94] and Peng *et al.*[95] is used to propagate the interface location.

H. Mesh Adjustment Scheme for Moving Embedded Boundaries

As related in the previous section, the location of the propellant interface is determined by the location of the zero-contour of the level set function, ψ. A mesh adjustment scheme was discussed in previous work in which a body-fitted multi-block mesh is locally adjusted to arbitrarily embedded boundaries that are not necessarily aligned with the mesh.[27] Not only does this scheme allow for rapid and robust mesh generation involving complex embedded boundaries, it also enables the solution of unsteady flow problems involving bodies and interfaces moving relative to the flow domain. The proposed mesh adjustment algorithm for stationary or moving embedded boundaries has similarities with the Cartesian cut-cell methods developed by De Zeeuw and Powell,[85] Bayyuk *et al.*,[96] and Murman *et al.*[73] Alternative approaches include overset mesh,[97,98,99] arbitrary Lagrangian-Eulerian (ALE) methods,[100] the interface tracking methods of Glimm *et al.*,[101] and immersed boundary methods.[102] Design criteria for the algorithm include the use of block-based adaptive mesh refinement on body-fitted multi-block quadrilateral mesh described above and that the scheme only produces cells of the same order of magnitude.

An example of the mesh adjustment algorithm is given in Figures 2(a)– 2(d). Here, the initial configuration shows a curved boundary embedded through the computational domain. The mesh adjustment is comprised of two main stages. The primary adjustment involves merely relocating the mesh nodes that are closest to the intersection point between the spline defining the embedded boundary and the mesh lines. The resulting adjusted grid is shown in Figure 2(b). A secondary adjustment is required to account for computational cells that are bisected diagonally by the embedded boundary. This is accomplished by choosing the closest not-aligned node to the embedded boundary and relocating it to that boundary point, as shown in Figure 2(c). The solid lines in the adjusted grid denote the "active" computational mesh and the dashed lines correspond to the portion of the mesh that is not needed for the computation and are deemed "in-active." Note that the in-active nodes are retained to maintain the block-based data structure and may be reactivated in computations involving moving boundaries. Mesh adjustment for moving embedded boundaries is accomplished by returning the grid to the original unadjusted form and then readjusting the mesh to the new location.

The preceding mesh adjustment scheme will generate a piece-wise linear representation of the embedded boundary while maintaining the (i, j) data structure of the original body-fitted mesh, as shown by Figure 2(d). The resulting mesh allows for accurate calculation of cell areas and straight-forward application of



(a) (b) (c) (d)

Figure 2. ***Mesh adjustment algorithm: (a) Initial mesh and embedded boundary (thick line), (b) result of primary adjustment, (c) result of secondary adjustment (dashed lines indicate inactive cells), and (d) example of*** (i, j)***-indexing on an adjusted mesh.***

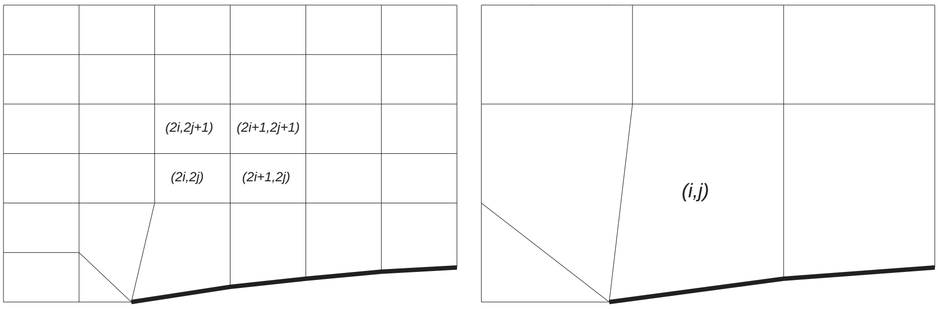


Figure 3. ***The fine and coarse grids for a two-stage multigrid method are shown in the left and right panels, respectively. The embedded boundary is indicated by the thick solid line and the corresponding indexing of the fine and coarse grids are indicated.***

boundary conditions. It can be seen from the adjusted mesh in Figure 2(c) that some triangular cells can be generated from this scheme. These cells are treated as quadrilateral cells with two coincident nodes. An advantage of this mesh adjustment algorithm is that very small cut-cells are not introduced and cell merging techniques are not required. It is found that typically the ratio of the smallest to largest cell areas produced by the proposed embedded boundary treatment is not less than about 0.2–0.25. Since only local alterations are made to the mesh, the need for inter-solution-block communication is not required and is, therefore, transparent to the parallel block-based adaptive mesh refinement scheme.

The finite volume scheme described above can be directly applied to the adjusted mesh. The coarse grids required for the parallel multigrid method are generated by the standard approach on the unadjusted initial mesh. The mesh on each level is then adjusted to the embedded boundary using the mesh adjustment algorithm. This method of coarse grid generation is straight-forward, however, the standard restriction and prolongation operators near the embedded boundary are no longer valid since the $(2i, 2j)$-indexing on the fine mesh may not correspond to the (i, j)-indexing on the coarser mesh as indicated in Figure 3. The restriction operator used here for cells near the embedded boundary requires the calculation of the area of intersection between the coarse grid and fine grid cells allowing for an accurate area-weighted average, $U_{i,j}A_{i,j} = \sum_m \sum_n U_{2i+m,2j+n}(A_{2i+m,2j+n} \cap A_{i,j})$, where $A_{i,j}$ and $U_{i,j}$ are the area and solution state of the coarse grid cell and $A_{2i+m,2j+n}$ and $U_{2i+m,2j+n}$ are the area and solution state of the contributing fine grid cells. The area of intersection between the fine and coarse grid cells, $A_{2i+m,2j+n} \cap A_{i,j}$, is determined using a polygon clipping algorithm.[103] The search space for contributing fine grid cells can be limited to $(m, n) \in [-2, 3]$. The prolongation operator is reduced to injection for cells near the embedded boundary. Bilinear interpolation is used elsewhere.

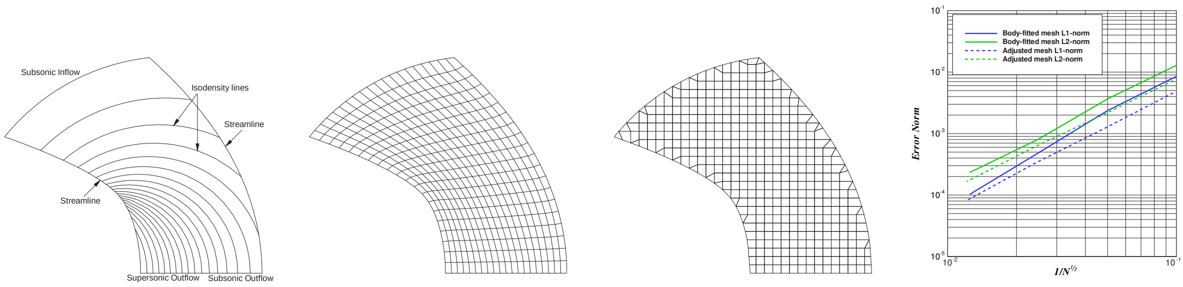
III. Numerical Verification and Validation

A. Accuracy Assessment: Ringleb's Flow

In order to first assess and demonstrate the accuracy of the inviscid spatial discretization scheme, the predictions of the proposed algorithm are considered for a test problem for which an exact analytic solution exists. Ringleb's flow is a hodograph solution to the Euler equations that is widely used in validation studies.[104,105] The flow pattern involves an isentropic, irrotational flow contained between two streamlines and is shown in Figure 4(a) for the streamlines corresponding to $k = 0.75$ and $k = 1.5$ (where $k = 1/\psi$). The inflow boundary is defined by the iso-velocity contour, $q = 0.5$, corresponding to a subsonic inflow. A mixed supersonic and subsonic outflow occurs at the lower boundary.

The accuracy of the spatial discretization was assessed by comparing the computed solution on a series of refined meshes to the analytic solution. The L_1- and L_2-norms of the difference in the solution densities were used as the measure of solution accuracy. The error-norms were computed for a uniformly refined body-fitted mesh involving 100, 400, 1600, and 6400 cells and for a uniformly refined Cartesian mesh with an embedded boundary representing the Ringleb's flow domain involving 107, 429, 1710, and 6832 active computational cells. Sample body-fitted and embedded boundary meshes with 400 and 429 cells are shown in Figures 4(b) and 4(c). The L_1- and L_2-norms of the solution error are plotted in Figure 4(d). The slopes of the L_1- and L_2-norms are 2.11 and 1.94 for the body-fitted mesh, respectively. For the Cartesian mesh with the embedded boundary, the slopes of the L_1- and The and L_2-norms are 1.93 and 1.81, respectively.

However, considering only the finest two meshes, the L_1- and L_2-norms are 2.03 and 1.89. This indicates that the scheme is indeed second order accurate for both the body-fitted and embedded boundary meshes.

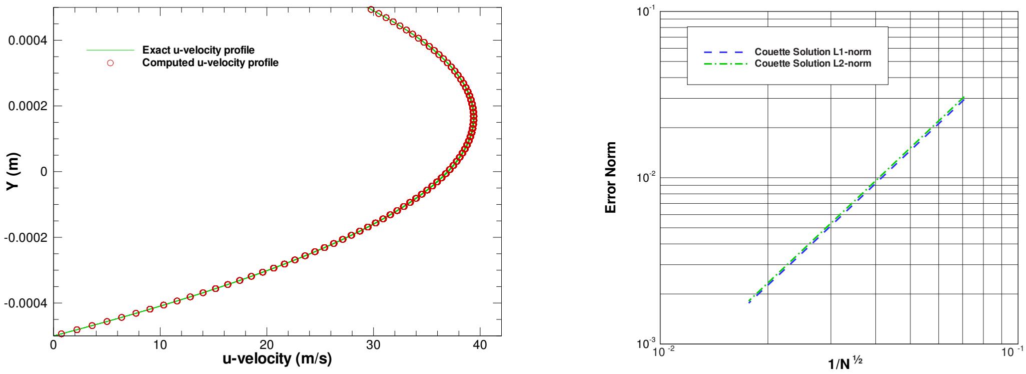


(a) (b) (c) (d)

Figure 4. ***Ringleb's flow: (a) Flow pattern, (b) Sample body-fitted mesh with 400 cells, (c) Sample embedded boundary Cartesian mesh with 429 cells, and (d) L_1- and L_2-norms of the solution error.***

B. Accuracy Assessment: Laminar Couette Flow

The accuracy of the viscous spatial discretization scheme is demonstrated by considering the computation of laminar flow in a channel with a moving wall. Classical planar Couette flow,[106] is considered with an upper wall velocity of 29.4 m/s and a favourable pressure gradient of -635.54 Pa. The predicted x-direction velocity component is plotted and compared to the exact analytic solution for this incompressible isothermal flow in Figure 5(a). The L_1- and L_2-norms of the solution error are plotted in Figure 5(b). The slopes of the L_1- and L_2-norms are 2.03 and 2.04 respectively, indicating that the viscous discretization scheme is indeed second-order accurate.

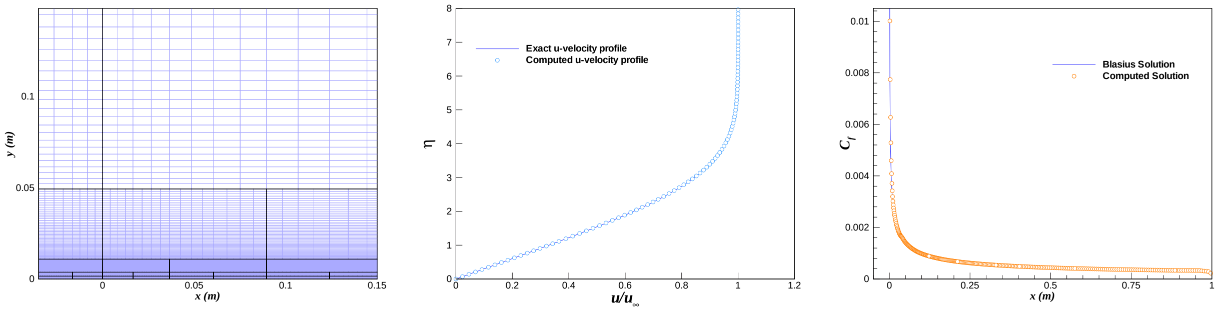


(a) (b)

Figure 5. ***Laminar Couette Flow: (a) u-velocity profile (1 block and 3200 cells), (b) L_1- and L_2-norms of the solution error.***

C. Validation: Laminar Flat-Plate Boundary Layer

The computation of laminar flow over a flat plate at zero incidence is considered to further demonstrate the accuracy of the viscous spatial discretization procedure. The free-stream Mach number and Reynolds number, based on the length of the plate, for the case considered are $\mathrm{M} = 0.2$ and of $\mathrm{Re} = 10,000$, respectively. The exact solution of the incompressible boundary layer equations first obtained by Blasius is given by Schlichting.[106] The calculated boundary layer solution is shown in Figure 6. The mesh consisting of 92 blocks and 70,656 cells is shown in Figure 6(a) for the leading edge region of the plate. Note the anisotropic non-uniform mesh spacing for this case. The first node normal to the plate is located at a distance of approximately 3×10^{-5}m. The prediction of the non-dimensional x-direction velocity component and the skin friction coefficient are shown in Figures 6(b) and 6(c). The x-direction velocity component is plotted at $\mathrm{Re}_\mathrm{x} = 8,000$. It can be seen that the x-direction velocity component and the skin friction coefficient are in excellent agreement with the Blasius solution, providing further validation of the schemes accuracy.

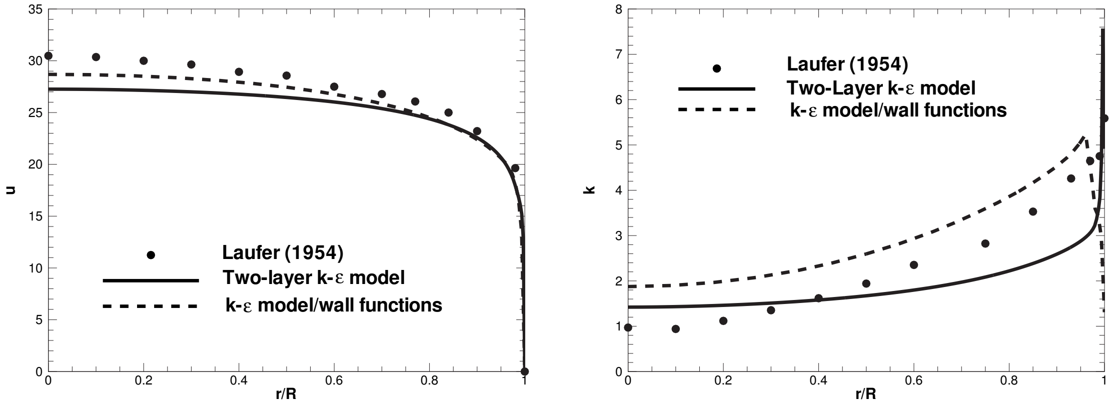


(a) (b) (c)

Figure 6. *Laminar flat plate boundary layer: (a) Body-fitted mesh at the leading edge (92 blocks and 70,656 cells), (b) Non-dimensional velocity components at* $\mathrm{Re}_x = 8,000$*, and (c) Estimation of the skin friction coefficient.*

D. Validation: Fully-Developed Turbulent Pipe Flow

The validation of the parallel AMR scheme for turbulent flows has also been considered by comparing numerical results to the experimental data of Laufer[107] for fully-developed turbulent flow in a pipe with $\mathrm{Re} = 500,000$. Solutions for both the two-layer two-equation $k-\epsilon$ turbulence model and k-ϵ turbulence model with wall functions are compared to measured mean axial velocity and turbulent kinetic energy profiles in Figure 7. The computations with the two-layer model were performed using 64 cells in the radial direction with some 3-5 of those cells lying within the laminar sublayer. The first cell off the wall was located at $y^+ \approx 2$. The calculations with the wall function formulation were performed using just 24 cells in the radial direction with the first cell at $y^+ \approx 45$. The agreement between the experimental data and numerical results for this case is generally quite good. As should be expected, the present implementations of the two-layer and standard k-ϵ model with wall functions are both capable of accurately reproducing the characteristic features of fully-developed pipe flow.



(a) (b)

Figure 7. *Turbulent pipe flow: (a) radial profiles of axial velocity,* u*, (b) radial profiles of turbulent kinetic energy,* k*.*

E. Validation: 1D Moving Piston Problem

A one-dimensional moving piston problem has been used to assess the conservation properties of the proposed scheme for moving boundaries and interfaces.[73] Although the moving piston problem is a rather simple inviscid problem, analytic expressions can be determined for the resulting shock wave that forms ahead of the piston and rarefaction wave that is generated behind the piston and it provides a good test of the scheme's conservation properties. If the method accurately conserves mass, momentum, and energy at the moving interface, the predicted shock strength and speed must match the analytic expression. Poor agreement would indicate solution content may be lost at the moving surface. The predicted non-dimensional pressure and density fields are shown in Figure 8 for a piston moving at Mach 2 into a quiescent gas. Comparison with the analytical results reveals that the proposed numerical scheme accurately predicts the shock position and strength on the compression-side of the piston, as well as the rarefaction wave solution behind the piston, indicating that the conservation properties of the finite-volume formulation are maintained when a moving interface is introduced.

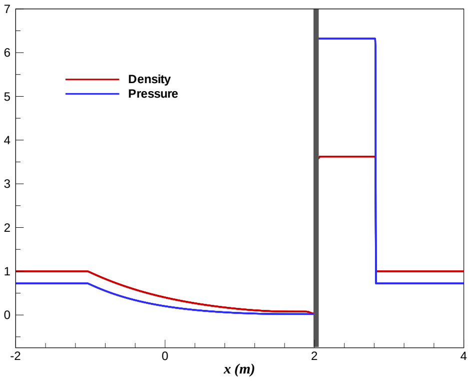


Figure 8. ***Non-dimensional pressure and density ratios for the 1D moving piston at time 3ms for a piston moving at Mach 2.***

F. Validation: Translating Ellipse

The flow produced by the motion of a rigid ellipse-shaped body translating at Mach 1.5 in a channel filled with initially stationary air at atmospheric conditions is used as a further example to illustrate the effectiveness of the proposed algorithm for more complex flows with moving boundary boundaries. The calculated pressure contours are shown for this case in Figures 9(a) and 9(b) at two different times during the computation as the ellipse moves along the channel. It can be seen that the mesh adjustment scheme is very effective in dealing with the moving boundary of the ellipse. The mesh is continuously adjusted to conform to the moving ellipse. It is also evident from the figures that the AMR scheme has successfully detected the bow shock which forms at the leading edge of the ellipse and the reflected and wake shock structures that form at solid boundaries and behind the ellipse.

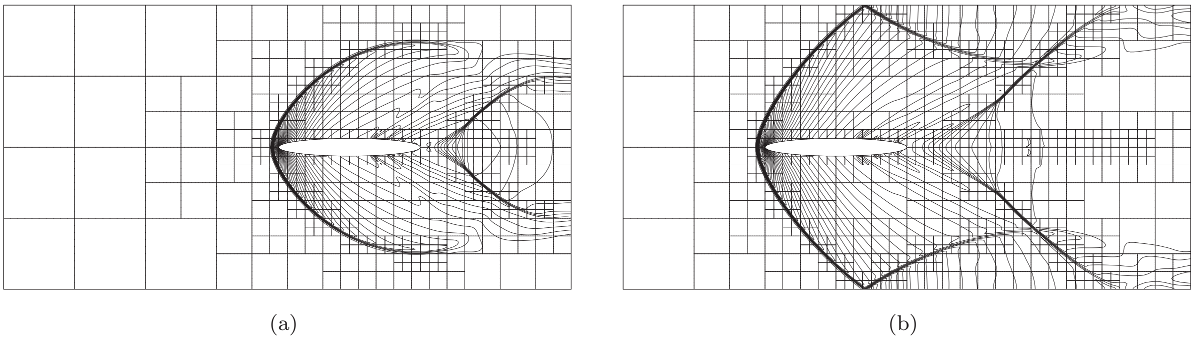
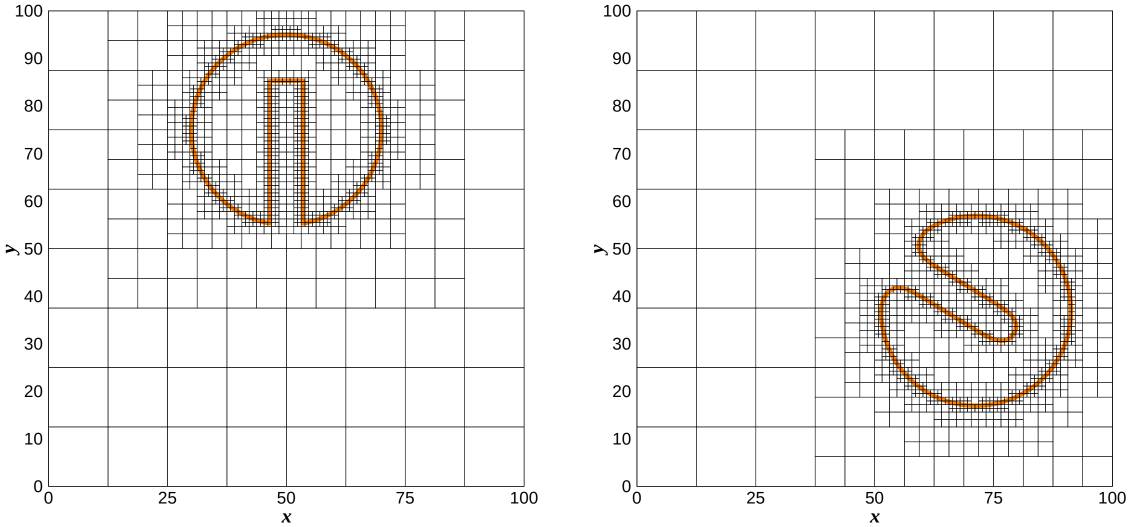


Figure 9. ***An ellipse translating at Mach 1.5 at (a) $t = 0.735ms$ (574 blocks, 40,600 cells), and (b) $t = 1.470ms$ (838 blocks, 83,800 cells). Pressure contours and solution block boundaries are shown.***

G. Validation: Level Set Evolution of Zalesak's Disk

Finally, the validity of the level set approach for treating evolving interfaces has also been investigated. A popular test case for assessing the performance of front tracking algorithms is the evolution of a rigid slotted disk in a rotating flow, also known as Zalesak's disk.[108, 109, 110] The problem configuration used here is the same as that used by Sussman *et al.*[108] The rotating two-dimension velocity field is taken to be $(u, v) = (\pi/314)(50 - y, x - 50)$. The initial configuration of the disk is shown in Figure 10(a) where the disk boundary is given by the zero contour shown in orange. Initially, the mesh includes four levels of refinement with 1,462 blocks and 146,200 cells. Figure 10(b) shows the computed position of the disk after a rotation of 120° about the origin. The final mesh contains 1,408 blocks and 140,800 cells. It can be seen that the level set method coupled with the block-based AMR scheme is capable of maintaining a rather accurate definition of the disk. Additional refinement could be employed to further reduce the diffusion of the disk as it rotates.



(a)

(b)

Figure 10. ***Zalesak's disk: (a) Initial configuration (1,462 blocks, 146,200 cells) and (b) After*** 120° ***rotation about the origin (1,408 blocks, 140,800 cells). The zero contour is shown by the solid orange line and the block boundaries are given by the solid black lines.***

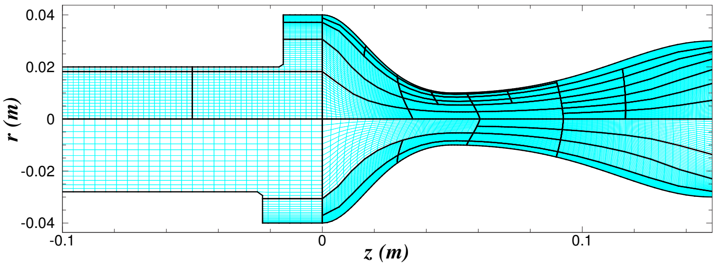
IV. CRV Solid Propellant Rocket System

Predicted SRM core flow results for a configuration typical of CRV-7 rocket systems are now described to further demonstrate the viability and capability of the proposed scheme for multi-phase flow with evolving combustion interfaces. The predicted results are presented in Figures 11 and 12 for a cylindrical grain rocket motor with a 40 mm internal radius, a nozzle throat radius of 10 mm, and an internal port radius of 20 mm. The propellant grain consists of 97% AP-HPTB and 3% inert particles by mass. The rocket motor, propellant, and particle characteristics are summarized in Table 1. The configuration of the rocket and propellant grain are shown in Figure 11 where the (adjusted) body-fitted mesh is shown at two different times after ignition, along with the corresponding level set solutions. The initial adjusted multi-block mesh with 4 levels of refinement, 72 blocks, and 27,648 cells and the subsequent re-adjusted mesh at the later time with 3 levels of refinement, 21 blocks, and 8,064 cells is shown in Figure 11(a) and the level set solutions and overlapping Cartesian mesh with 5 levels of refinement, 49 blocks, and 14,700 cells is depicted in Figure 11(b). The two curves representing the zero level set and the position of the combustion interface are shown in blue. Note that as the propellant grain regresses, a larger volume is available for the core flow of the propellant gases.

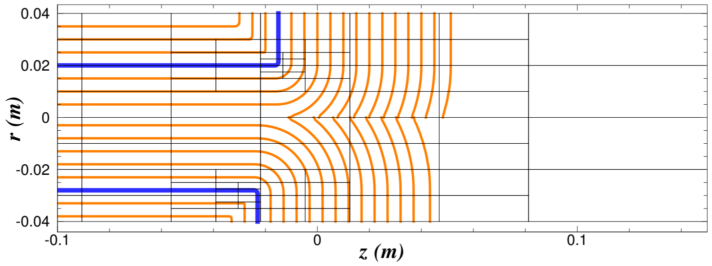
The predicted core flow solutions at the two times after the propellant ignition are given in Figure 12. The upper panel of each figure shows the solution at the early time and the lower shows the solution at the later time. The burning of the solid propellant leads to a head end pressure approaching 3 MPa and produces sonic flow conditions at the nozzle throat and supersonic outflows in the rocket nozzle with Mach numbers in excess of 3 as shown Figure 12(a). The particle-phase concentration contours through the converging-diverging nozzle are depicted in Figure 12(b). Due the the high mass of the inert particles, the particle-phase velocity lags relative to the gas-phase velocity after the rapid acceleration through the nozzle. The limiting particle-phase concentration contours also show that the particles are unable to expand in the nozzle due to their relative high mass. As expected, low speed recirculation zones are found in the area following the propellant grain and before the converging section of the nozzle. A consequence of this recirculation, is the formation of an area of low particle concentration since the inert particles are pushed away by the recirculation. A region of high particle concentration can also be found at the upper wall of the converging section of the nozzle. A similar result has been found by.[10] The high particle concentration zone found at the rocket centre-line and the low concentration zone at the walls of the diverging section of the nozzle are well predicted.

Table 1. Motor, propellant, and particle characteristics.

	Reference Motor
Combustion chamber length	40.0 cm
Inner casing diameter	8.00 cm
Internal port diameter	4.00 cm
Nozzle throat diameter	2.00 cm
Grain length	38.5 cm
Propellant density (ρ_s)	1740 kg/m^3
Propellant specific heat (c_s)	1510 J/kg K
Propellant flame temperature (T_f)	3060 K
Propellant surface temperature (T_s)	1130 K
Propellant burning rate ($r_{bs} = \beta p^n$)	0.5[p(kPa)]$^{0.33}$mm/s
Particulate mass fraction (α_s)	0.03
Gas specific heat (c_p)	1845 J/kg$K$
Specific gas constant (R)	318 J/kg$K$
Gas thermal conductivity (κ)	0.184 W/mK
Gas absolute viscosity (μ)	8.19 $\times 10^{-5}$kg/m s
Gas ratio of specific heats (γ)	1.21
Average particle diameter (d_p)	10 μm
Particulate solid density (ρ_p)	2700 kg/m^3
Particulate specific heat (c_m)	900 J/kg K

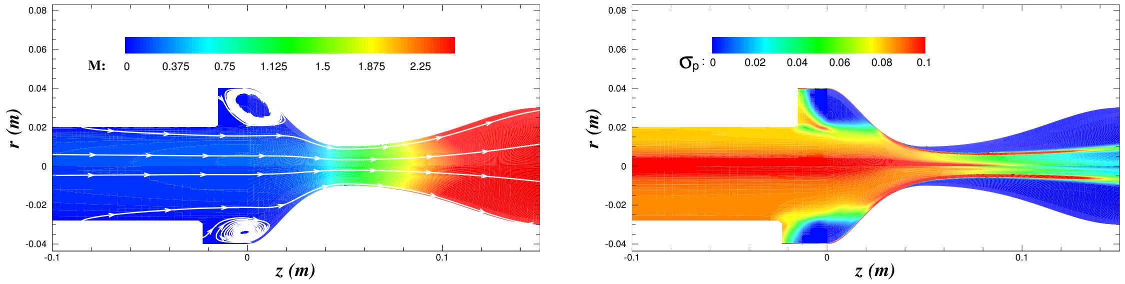


(a)



(b)

Figure 11. ***Adjusted mesh and level set solution for a cylindrical grain rocket motor with evolving combustion interface: (a) the (adjusted) body-fitted rocket motor mesh at two different times after ignition (initial adjusted mesh is shown above and has 4 levels of refinement, 72 blocks, 27,648 cells,*** $\eta = 0.625$***, re-adjusted mesh at the later time is shown below has 3 levels of refinement, 21 blocks, 8,064 cells,*** $\eta = 0.5625$***) and (b) the level set solutions and overlapping Cartesian mesh at two different times after ignition (5 levels of refinement, 49 blocks, 14,700 cells,*** $\eta = 0.952$***). The curves representing the zero level set and the position of the combustion interface are shown in blue.***



(a) (b)

Figure 12. *Predicted solution for a cylindrical grain rocket motor with evolving combustion interface: (a) Mach number distribution with gas-phase streamlines and (b) particle concentration contours for two different times after ignition of the solid propellant.*

V. Concluding Remarks

A parallel adaptive mesh refinement scheme has been described for solving the governing equations for turbulent multi-phase core flows in solid propellant rocket motors. The application of a finite-volume discretization procedure and a parallel block-based AMR strategy has provided a powerful tool for predicting SRM core flows. Coupling the flow solution with the level set method and the mesh adjustment scheme allows for the regression of the combustion interface. The validity of the scheme has been shown through various numerical assessments and the viability of the method has been demonstrated for a SRM core flow. Future work will involve extending the method to three dimensional flow.

Acknowledgements

This research was supported by a Premier's Research Excellence Award from the Ontario Ministry of Energy, Science, and Technology and by the Natural Sciences and Engineering Research Council of Canada. Funding for the parallel computing facility used to perform the computations described herein was obtained from the Canadian Foundation for Innovation and Ontario Innovation Trust (CFI Project No. 2169). The authors are very grateful to these funding agencies for this support. The first author is also grateful for the financial assistance provided by the Ontario Graduate Scholarship program.

References

[1] Tseng, I. S. and Yang, V., "Computational of a Double-Base Homogeneous Propellant in a Rocket Motor," *Combust. Flame*, Vol. 96, 1994, pp. 325–342.

[2] Cai, W., Ma, F., and Yang, V., "Two-Phase Vorticoacoustic Flow Interactions in Solid-Propellant Rocket Motors," *J. Propulsion Power*, Vol. 19, No. 3, 2003, pp. 385–396.

[3] Ciucci, A. and Iaccarino, G., "Numerical analysis of turbulent flow and alumina particle trajectories in solild rocket motors," Paper 97–2860, AIAA, July 1997.

[4] Ciucci, A., Iaccarino, G., and Amato, M., "Numerical investigation of 3D two-phase turbulent flows in solid rocket motors," Paper 98–3966, AIAA, July 1998.

[5] Orbekk, E., "Internal flow analysis of a technology demonstrator rocket motor with new CFD code," Paper 98–3967, AIAA, July 1998.

[6] Sabnis, J., Madabhushi, R., Gibeling, H., and McDonald, H., "On the Use of k-e Turbulence Model for Computation of Solid Rocket Internal Flows," Paper 89–2558, AIAA, 1989.

[7] Sabnis, J. S., "Numerical Simulations of Distributed Combustion in Solid Rocket Motors with Metalized Propellant," *J. Propulsion Power*, Vol. 19, No. 1, January-February 2003, pp. 48–55.

[8] Berglund, M. and Fureby, C., "Large Eddy Simulation of the FLow in a Solid Rocket Motor," Paper 2001–0895, AIAA, January 2001.

[9] Najjar, F. M., Hasselbacher, A., Ferry, J., Wasistho, B., Balachandar, S., and Moser, R. D., "Large-Scale Multiphase Large-Eddy Simulation of Flows Inside Solid-Rocket Motors," Paper 03–3700, AIAA, June 2003.

[10] Vuillot, F., Basset, T., Dupays, J., Daniel, E., and Lupoglazoff, N., "2D Navier-Stokes stability computations for solid rocket motors: rotational, combustion and two-phase flow effects," Paper 97–3326, AIAA, January 1997.

[11] Daniel, E., Basset, T., and Loraud, J. C., "Eulerian approach for unsteady two-phase reactive solid rocket motor flows loaded with aluminum particles," Paper 98–3697, AIAA, July 1999.

[12] Balachandar, S., Ferry, J. P., and Bagchi, P., "Fundamental two-phase flow modeling efforts at CSAR," Paper 00–3569, AIAA, July 2000.

[13] Dick, W. A. and Heath, M. T., "Whole system simulation of solid propellant rockets," Paper 02–4345, AIAA, July 2002.

[14] Blazek, J., "Flow Simulation in Solid Rocket Motors Using Advanced CFD," Paper 03–5111, AIAA, July 2003.

[15] Karimian, S. M. H. and Amoli, A., "Two-dimensional simulation of SRM internal ballistics on a moving grid," Paper 99–2801, AIAA, June 1999.

[16] Sachdev, J. S., Groth, C. P. T., and Gottlieb, J. J., "Parallel Solution-Adaptive Scheme for Multi-Phase Core Flows in Rocket Motors," Paper 03–4106, AIAA, 2003.

[17] Sachdev, J. S., Groth, C. P. T., and Gottlieb, J. J., "A Parallel Solution-Adaptive Scheme for Predicting Multi- Phase Core Flows in Solid Propellant Rocket Motors," *Int. J. Comput. Fluid Dyn.*, Vol. 19, No. 2, 2005, pp. 157–175.

[18] Chen, H. C. and Patel, V. C., "Near-Wall Turbulence Models for Complex Flows Including Separation," *AIAA J.*, Vol. 26, No. 6, 1984, pp. 641–648.

[19] Rodi, W., "Experience with two-layer models combining the k-e model with a one-equation model near the wall," Paper 1991–0216, AIAA, 1991.

[20] Northrup, S. A. and Groth, C. P. T., "Solution of Laminar Diffusion Flames Using a Parallel Adaptive Mesh Refinement Algorithm," Paper 2005-0547, AIAA, 2005.

[21] Gao, X. and Groth, C. P. T., "Parallel Adaptive Mesh Refinement Algorithm for Turbulent Non-Premixed Combusting Flows," *Proceedings of the 2005 Combustion Institute Canadian Section Technical Meeting, Halifax, Canada, May 15-18, 2005*, Combustion Institute Canadian Section, 2005.

[22] M^{c}Donald, J. G. and Groth, C. P. T., "Numerical Modeling of Micron-Scale Flows Using the Gaussian Moment Closure," Paper 2005-5035, AIAA, June 2005.

[23] van Leer, B., Tai, C. H., and Powell, K. G., "Design of Optimally-Smoothing Multi-Stage Schemes for the Euler Equations," Paper 89-1933-CP, AIAA, June 1989.

[24] Jameson, A., "Time Dependent Calculations Using Multigrid, with Applications to Unsteady Flows Past Airfoils and Wings," Paper 91–1596, AIAA, June 1991.

[25] Osher, S. and Sethian, J. A., "Fronts propagating with curvature dependent speed: algorithms based on Hamilton-Jacobi formulations," *J. Comput. Phys.*, Vol. 79, 1988, pp. 12–49.

[26] Sethian, J. A., *Level Set Methods and Fast Marching Methods*, Cambridge Monographs on Applied and Computational Mathematics, Cambridge University Press, 2nd ed., 1999.

[27] Sachdev, J. S. and Groth, C. P. T., "A Mesh Adjustment Scheme for Embedded Boundaries," *Proceedings of the 3rd International Conference on Computational Fluid Dynamics, Toronto, Canada, July 12–16, 2004*, 2004.

[28] King, M. K., "Erosive burning of composite solid propellants: experimental and modeling studies," *J. Spacecraft Rockets*, Vol. 16, No. 7, June 1979, pp. 154–162.

[29] Renie, J. P. and Osborn, J. R., "Erosive Burning," *AIAA J.*, Vol. 21, No. 12, December 1983, pp. 1681–1689.

[30] Greatrix, D. R. and Gottlieb, J. J., "Erosive burning model for composite-propellant rocket motors with large length-to-diameter ratios," *Can. Aeronaut. Space J.*, Vol. 33, No. 3, September 1987, pp. 133–142.

[31] Brewster, M. Q., "Solid Propellant Combustion Response: Quasi-Steady (QSHOD) Theory Development and Validation," *Solid Propellant Chemistry, Combustion, and Motor Interior Ballistics*, edited by V. Yang, T. B. Brill, and W.-Z. Ren, Vol. 185 of *Progress in Astronautics and Aeronautics*, AIAA, Reston, VA, 2000, pp. 607–638.

[32] Surzhikov, S. T., Murphy, J. J., and Krier, H., "2D model for unsteady burning heterogenous AP/binder solid propellants," Paper 00–3573, AIAA, July 2000.

[33] Saurel, R., Daniel, E., and Loraud, J. C., "Two-Phase Flows: Second-Order Schemes and Boundary Conditions," *AIAA J.*, Vol. 32, No. 6, June 1994, pp. 1214–1221.

[34] Ben-Dor, G., Igra, O., and Wang, L., "Shock Wave Reflections in Dust-Gas Suspensions," *J. Fluids Eng.*, Vol. 123, March 2001, pp. 145–153.

[35] Slater, S. A. and Young, J. B., "The calculation of inertial particle transport in dilute gas-particle flows," *Int. J. Multiphase Flow*, Vol. 27, 2001, pp. 61–87.

[36] Saito, T., Marumoto, M., and Takayama, K., "Numerical investigations of shock waves in gas-particle mixtures," *Shock Waves*, Vol. 13, 2001, pp. 299–322.

[37] Igra, O., Hu, G., Falcovitz, J., and Wang, B. Y., "Shock wave reflection from a wedge in a dusty gas," *Int. J. Multiphase Flow*, Vol. 30, 2004, pp. 1139–1169.

[38] Rudinger, G., *Fundamentals of Gas-Particle Flow*, Elsevier Scientific Publishing Company, Amsterdam, 1980.

[39] Wolfshtein, M., "The Velocity and Temperature Distribution in One-Dimensional Flow With Turbulence Augmentation and Pressure Gradient," *Int. J. Heat Mass Transfer*, Vol. 12, 1969, pp. 301–318.

[40] Peirano, E. and Leckner, B., "Fundamentals of turbulent gas-solid flows applied to circulating fluidized bed combustion," *Prog. Energy Combust. Sci.*, Vol. 24, 1998, pp. 259–296.

[41] Vandromme, D. and Minh, H. H., "About the Coupling of Turbulence Closure Models with Averaged Navier-Stokes Equations," *J. Comput. Phys.*, Vol. 65, 1986, pp. 386–409.

[42] Siikonen, T., "An Application of Roe's Flux-Difference Splitting for $k - \epsilon$ Turbulence Model," *Int. J. Numer. Meth. Fluids*, Vol. 21, 1995, pp. 1017–1039.

[43] Geuzaine, P., *An Implicit Upwind Finite-Volume Method for Compressible Turbulent Flows on Unstructured Meshes*, Ph.D. thesis, Université de Liège, 1999.

[44] Thomas, P. D. and Lombard, C. K., "Geometric Conservation Laws and Its Application to Flow Computations on Moving Grids," *AIAA J.*, Vol. 17, No. 10, October 1979, pp. 1030–1037.

[45] Godunov, S. K., "Finite-Difference Method for Numerical Computations of Discontinuous Solutions of the Equations of Fluid Dynamics," *Mat. Sb.*, Vol. 47, 1959, pp. 271–306.

[46] van Leer, B., "Towards the Ultimate Conservative Difference Scheme. I. The Quest of Monotonicity," *Lecture Notes in Physics*, Vol. 18, Springer-Verlag, New York, 1973, pp. 163–168.

[47] van Leer, B., "Towards the Ultimate Conservative Difference Scheme. II. Monotonicity and Conservation Combined in a Second Order Scheme," *J. Comput. Phys.*, Vol. 14, 1974, pp. 361–370.

[48] van Leer, B., "Towards the Ultimate Conservative Difference Scheme. III. Upstream-Centered Finite-Difference Schemes for Ideal Compressible Flow," *J. Comput. Phys.*, Vol. 23, 1977, pp. 263–275.

[49] van Leer, B., "Towards the Ultimate Conservative Difference Scheme. IV. A New Approach to Numerical Convection," *J. Comput. Phys.*, Vol. 23, 1977, pp. 276–299.

[50] van Leer, B., "Towards the Ultimate Conservative Difference Scheme. V. A Second-Order Sequel to Godunov's Method," *J. Comput. Phys.*, Vol. 32, 1979, pp. 101–136.

[51] van Leer, B., "Flux-Vector Splitting for the Euler Equations," *Lecture Notes in Physics*, Vol. 170, Springer-Verlag, New York, 1982, p. 507.

[52] Roe, P. L., "Approximate Riemann Solvers, Parameter Vectors, and Difference Schemes," *J. Comput. Phys.*, Vol. 43, 1981, pp. 357–372.

[53] Roe, P. L., "Generalized Formulation of TVD Lax-Wendroff Schemes," Report 84-53, ICASE, January 1984.

[54] Colella, P. and Woodward, P. R., "The Piecewise Parabolic Method (PPM) for Gas-Dynamical Simulations," *J. Comput. Phys.*, Vol. 54, 1984, pp. 174–210.

[55] Harten, A., "High Resolution Schemes for Hyperbolic Conservation Laws," *J. Comput. Phys.*, Vol. 49, 1983, pp. 357–393.

[56] Harten, A., "On a Class of High Resolution Total-Variation-Stable Finite-Difference Schemes," *SIAM J. Numer. Anal.*, Vol. 21, 1984, pp. 1–23.

[57] Osher, S., "Riemann Solvers, the Entropy Condition, and Difference Approximations," *SIAM J. Numer. Anal.*, Vol. 21, No. 2, 1984, pp. 217–235.

[58] Yee, H. C., "Construction of Explicit and Implicit Symmetric TVD Schemes and Their Applications," *J. Comput. Phys.*, Vol. 68, 1987, pp. 151–179.

[59] Harten, A., Enquist, B., Osher, S., and Chakravarthy, S. R., "Uniformly High Order Accurate Essentially Non-Oscillatory Schemes, III," *J. Comput. Phys.*, Vol. 71, 1987, pp. 231–303.

[60] Barth, T. J., "Recent Developments in High Order K-Exact Reconstruction on Unstructured Meshes," Paper 93-0668, AIAA, January 1993.

[61] Venkatakrishnan, V., "On the Accuracy of Limiters and Convergence to Steady State Solutions," Paper 93-0880, AIAA, January 1993.

[62] Coirier, W. J., *An Adaptively-Refined, Cartesian, Cell-Based Scheme for the Euler and Navier-Stokes Equations*, Ph.D. thesis, University of Michigan, 1994.

[63] Holmes, D. and Connell, S. D., "Solution of the 2D Navier-Stokes equations on unstructured adaptive grids," Paper 89–1932, AIAA, June 1989.

[64] Zingg, D. W. and Yarrow, M., "A Direct Procedure for Interpolation on a Structured Curvilinear Two-Dimensional Grid," Technical Memorandum 102213, NASA, July 1989.

[65] Allmaras, S. R., "Analysis of a Local Matrix Preconditioner for the 2-D Navier-Stokes Equations," Paper 1993-3330, AIAA, 1993.

[66] Pierce, N. A. and Giles, M. B., "Preconditioned Multigrid Methods for Compressible flow Calculations on Stretched Meshes," *J. Comput. Phys.*, Vol. 136, 1997, pp. 425–445.

[67] Weiss, J. M. and Smith, W. A., "Preconditioning Applied to Variable and Constant Density Flows," *AIAA J.*, Vol. 33, No. 11, 1995, pp. 2050–2057.

[68] Turkel, E., "Preconditioning Techniques om Computational Fluid Dynamics," *Ann. Rev. Fluid Mech.*, Vol. 32, 1999, pp. 385–416.

[69] Alonso, J. J., Martinelli, L., and Jameson, A., "Multigrid Unsteady Navier-Stokes Calculations with Aeroelastic Applications," Paper 95-0048, AIAA, January 1995.

[70] Rumsey, C. L., Sanetrik, M. D., Biedron, R. T., Melson, N. D., and Parlette, E. B., "Efficiency and Accuracy of Time-Accurate Turbulent Navier-Stokes Computations," *Comput. & Fluids*, Vol. 25, No. 2, 1996, pp. 217–236.

[71] Dubuc, L., Cantariti, F., Woodgate, M., Gribben, B., Badcock, K. J., and Richards, B. E., "Solution of the Unsteady Euler Equations Ising an Implicit Dual-Time Method," *AIAA J.*, Vol. 36, No. 8, August 1998, pp. 1417–1424.

[72] Chassaing, J. C., Gerolymos, G. A., and Vallet, I., "Reynolds-stress model dual-time-stepping computation of unsteady three-dimensional flows," *AIAA J.*, Vol. 41, No. 10, October 2003, pp. 1882–1894.

[73] Murman, S. M., Aftosmis, M. J., and Berger, M. J., "Implicit Approaches for Moving Boundaries in a 3-D Cartesian Method," Paper 03–1119, AIAA, January 2003.

[74] Lomax, H., Pulliam, T. H., and Zingg, D. W., *Fundamentals of Computational Fluid Dynamics*, Scientific Computation, Springer, 2001.

[75] Melson, N. D., Sanetrik, M. D., and Atkins, H. L., "Time-Accurate Navier-Stokes Calculations with Multigrid Acceleration," *Proceedings of the Sixth Copper Mountain Conference on Multigrid Methods*, NASA Conf. Publication 3224, Copper Mountain, CO, 1993.

[76] Pierce, N. A. and Alonso, J. J., "Efficient Computation of Unsteady Viscous Flows by an Implicit Preconditioned Multigrid Method," *AIAA J.*, Vol. 36, No. 3, 1998, pp. 401–408.

[77] Groth, C. P. T., Zeeuw, D. L. D., Powell, K. G., Gombosi, T. I., and Stout, Q. F., "A Parallel Solution-Adaptive Scheme for Ideal Magnetohydrodynamics," Paper 99-3273, AIAA, June 1999.

[78] Berger, M. J., *Adaptive Mesh Refinement for Hyperbolic Partial Differential Equations*, Ph.D. thesis, Stanford University, January 1982.

[79] Berger, M. J., "Adaptive Mesh Refinement for Hyperbolic Partial Differential Equations," *J. Comput. Phys.*, Vol. 53, 1984, pp. 484–512.

[80] Berger, M. J. and Colella, P., "Local Adaptive Mesh Refinement for Shock Hydrodynamics," *J. Comput. Phys.*, Vol. 82, 1989, pp. 67–84.

[81] Berger, M. J. and LeVeque, R. J., "An Adaptive Cartesian Mesh Algorithm for the Euler Equations in Arbitrary Geometries," Paper 89-1930, AIAA, June 1989.

[82] Berger, M. J. and Saltzman, J. S., "AMR on the CM-2," *Appl. Numer. Math.*, Vol. 14, 1994, pp. 239–253.

[83] Quirk, J. J., *An Adaptive Grid Algorithm for Computational Shock Hydrodynamics*, Ph.D. thesis, Cranfield Institute of Technology, January 1991.

[84] Quirk, J. J. and Hanebutte, U. R., "A Parallel Adaptive Mesh Refinement Algorithm," Report 93-63, ICASE, August 1993.

[85] De Zeeuw, D. and Powell, K. G., "An Adaptively Refined Cartesian Mesh Solver for the Euler Equations," *J. Comput. Phys.*, Vol. 104, 1993, pp. 56–68.

[86] Davis, R. L. and Dannenhoffer, J. F., "Decomposition and Parallelization Strategies for Adaptive Grid-Embedding Techniques," *Int. J. Comput. Fluid Dyn.*, Vol. 1, 1993, pp. 79–93.

[87] Sun, M. and Takayama, K., "Conservative Smoothing on an Adaptive Quadrilateral Grid," *J. Comput. Phys.*, Vol. 150, 1999, pp. 143–180.

[88] Bramkamp, F., Ballmann, J., and Müller, S., "Development of a Flow Solver Employing Local Adaptation Based on Multiscale Analysis on B-Spline Grids," *Proceedings of the Eighth Annual Conference of the CFD Society of Canada, Montreal, Canada, June 11–13, 2000*, Vol. 1, CFD Society of Canada, 2000, pp. 111–118.

[89] Paillère, H., Powell, K. G., and De Zeeuw, D. L., "A Wave-Model-Based Refinement Criterion for Adaptive-Grid Computation of Compressible Flows," Paper 92-0322, AIAA, January 1992.

[90] Powell, K. G., Roe, P. L., and Quirk, J., "Adaptive-Mesh Algorithms for Computational Fluid Dynamics," *Algorithmic Trends in Computational Fluid Dynmaics*, edited by M. Y. Hussaini, A. Kumar, and M. D. Salas, Springer-Verlag, New York, 1993, pp. 303–337.

[91] Powell, K. G., Roe, P. L., Linde, T. J., Gombosi, T. I., and De Zeeuw, D. L., "A Solution-Adaptive Upwind Scheme for Ideal Magnetohydrodynamics," *J. Comput. Phys.*, Vol. 154, 1999, pp. 284–309.

[92] Kuo, K. K. and Summerfield, M., editors, *Fundamentals of Solid Propellant Combustion*, Vol. 90 of *Progress in Astronautics and Aeronautics*. AIAA, October 1986.

[93] Gottlieb, J. J. and Groth, C. P. T., "Collection of Boundary Conditions for One- and Some Multi-Dimensional Unsteady Flows of Polytropic Gases," *CASI J.*, Vol. 45, No. 2, June 1999, pp. 161–182.

[94] Sussman, M., Smereka, P., and Osher, S., "A Level Set Approach for Computing Solutions to Incompressible Two-Phase Flow," *J. Comput. Phys.*, Vol. 114, 1994, pp. 146–159.

[95] Peng, D., Merriman, B., Osher, S., Zhao, H., and Kang, M., "A PDE-based fast local level set method," *J. Comput. Phys.*, Vol. 155, 1999, pp. 410–438.

[96] Bayyuk, S. A., Powell, K. G., and van Leer, B., "A Simulation Technique for 2-D Unsteady Inviscid Flows Around Arbitrarily Moving and Deforming Bodies of Arbitrary Geometry," Paper 93–3391, AIAA, January 1993.

[97] Henshaw, W. D., "A Fourth-Order Accurate Method for the Incompessible Navier-Stokes Equations on Overlapping Grids," *J. Comput. Phys.*, Vol. 113, 1994, pp. 13–25.

[98] Fujii, K., "Unified Zonal Method Based on the Fortified Solution Algorithm," *J. Comput. Phys.*, Vol. 118, 1995, pp. 92–108.

[99] Henshaw, W. D. and Schwendeman, D. W., "An adaptive numerical scheme for high-speed reacting flow on overlapping grids," *J. Comput. Phys.*, Vol. 191, 2003, pp. 420–447.

[100] Hirt, C. W., Amsden, A. A., and Cook, J. L., "An Arbitrary Lagrangian–Eulerian Computing Method for All Flow Speeds," *J. Comput. Phys.*, Vol. 14, No. 3, March 1974, pp. 227–253, Reprinted in Vol. 135, 1997, pages 227–253.

[101] Glimm, J., Graham, M. J., Grove, J., Li, X. L., Smith, T. M., Tan, D., Tangerman, F., and Zhang, Q., "Front tracking in two and three dimensions," *Computers Math. Applic.*, Vol. 35, No. 7, 1998, pp. 1–11.

[102] Peskin, C. S., "The immersed boundary method," *Acta Numer.*, 2002, pp. 1–39.

[103] Sutherland, I. E. and Hodgman, G. W., "Reentrant polygon clipping," *Commun. ACM*, Vol. 17, No. 1, 1974, pp. 32–42.

[104] Barth, T. J. and Fredrickson, P. O., "Higher Order Solution of the Euler Equations on Unstructured Grids Using Quadratic Reconstruction," Paper 90-0013, AIAA, January 1990.

[105] Coirier, W. J. and Powell, K. G., "An Accuracy Assessment of Cartesian-Mesh Approaches for the Euler Equations," *J. Comput. Phys.*, Vol. 117, 1995, pp. 121–131.

[106] Schlichting, H., *Boundary-Layer Theory*, McGraw-Hill, Toronto, 7th ed., 1979.

[107] J.Laufer, "The Structure of Turbulence in Fully Developed Pipe Flow," Report 1174, NACA, 1954.

[108] Sussman, M., Fatemi, E., Smereka, P., and Osher, S., "An Improved Level Set Method for Incompressible Two-Phase Flow," *Comput. & Fluids*, Vol. 27, No. 5–6, 1998, pp. 663–680.

[109] Enright, D., Fedkiw, R., Ferziger, J., and Mitchell, I., "A Hybrid Particle Level Set Method for Improved Interface Capturing," *J. Comput. Phys.*, Vol. 183, 2002, pp. 83–116.

[110] Greaves, D., "A quadtree adaptive method for simulating fluid flows with moving interfaces," *J. Comput. Phys.*, Vol. 194, 2004, pp. 35–56.