International Journal of Computational Fluid Dynamics
Publication details, including instructions for authors and subscription information:
http://www.informaworld.com/smpp/title~content=t713455064

A parallel solution-adaptive scheme for multi-phase core flows in solid propellant rocket motors

J. S. Sachdev [a]; C. P. T. Groth [a]; J. J. Gottlieb [a]
[a] University of Toronto, Ontario Institute for Aerospace Studies, Toronto, Canada

Online Publication Date: 01 February 2005

To cite this Article: Sachdev, J. S., Groth, C. P. T. and Gottlieb, J. J. (2005) 'A parallel solution-adaptive scheme for multi-phase core flows in solid propellant rocket motors', International Journal of Computational Fluid Dynamics, 19:2, 159 — 177

To link to this article: DOI: 10.1080/10618560410001729135
URL: http://dx.doi.org/10.1080/10618560410001729135

International Journal of Computational Fluid Dynamics, Vol. 19, No. 2, February 2005, 159–177

Taylor & Francis
Taylor & Francis Group

A parallel solution-adaptive scheme for multi-phase core flows in solid propellant rocket motors

J.S. SACHDEV*, C.P.T. GROTH† and J.J. GOTTLIEB‡

University of Toronto, Ontario Institute for Aerospace Studies, 4925 Dufferin Street, Toronto, M3H 5T6, Canada

(Received 17 March 2004)

The development of a parallel adaptive mesh refinement (AMR) scheme is described for solving the governing equations for multi-phase (gas–particle) core flows in solid propellant rocket motors (SRMs). An Eulerian formulation is used to describe the coupled motion between the gas and particle phases. A cell-centred upwind finite-volume discretization and the use of limited linear reconstruction, Riemann solver based flux functions for the gas and particle phases, and explicit multi-stage time-stepping allows for high solution accuracy and computational robustness. A Riemann problem is formulated for prescribing boundary data at the burning surface and a mesh adjustment algorithm has been implemented to adjust the multi-block quadrilateral mesh to the combustion interface. A flexible block-based hierarchical data structure is used to facilitate automatic solution-directed mesh adaptation according to physics-based refinement criteria. Efficient and scalable parallel implementations are achieved with domain decomposition on distributed memory multi-processor architectures. Numerical results are described to demonstrate the capabilities of the approach for predicting SRM core flows.

Keywords: Adaptive mesh refinement; Parallel computing; Multi-phase flow; Solid propellant rocket motors

1. Introduction

The internal flow of a solid propellant rocket motor (SRM) is very complex and not yet completely understood. The flow of the propellant products from the combustion chamber through the nozzle and the plume of the rocket is a high speed, high temperature, multi-phase, chemically reactive, turbulent flow. It is inherently three-dimensional due to rocket rotation, turbulent flow, mass injection and propellant grain geometry. Moreover, the combustion of the solid propellant occurs in a thin, high temperature layer between the propellant grain and the main flow cavity, known as the combustion interface. This topologically complex surface evolves as the propellant burns. Propellant burning rates must be controlled to avoid catastrophic failure and solid particles are often added to the propellant to enhance combustion and burning stability. 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. The particles added in many SRMs such as the booster rockets on the Space Shuttle, are typically composed of aluminum. These reactive particles are used to stabilize the internal flow due to possible combustion instabilities and also act as a fuel, injecting hot gas into the combustion chamber as they burn. The aluminum particles can account for as much as 20% of the propellant by mass. Non-aluminized propellants are, however, common in some military rocket applications, such as the CRV-7 rocket system, resulting in reduced smoke and thermal signatures. Here, inert particles, such as aluminum oxide, Al_2O_3, are typically added to damp out combustion instabilities in the flow chamber and account for 2–3% of the solid propellant by mass.

Modern numerical methods are a potential tool for studying the characteristics of the core flows of SRMs, as well as aiding rocket motor design and optimization.

*Corresponding author Email: j.sachdev@utoronto.ca
†Email: groth@utias.utoronto.ca
‡Email: gottlieb@utias.utoronto.ca

International Journal of Computational Fluid Dynamics
ISSN 1061-8562 print/ISSN 1029-0257 online © 2005 Taylor & Francis Ltd
http://www.tandf.co.uk/journals
DOI: 10.1080/10618560410001729135

Recent examples of the application of numerical methods to the modeling of SRM flows include the studies by Ciucci *et al.* (1998), Orbekk (1998), Daniel *et al.* (1999) and Sabnis (2003). Computational tools for the simulation of SRMs under various operating conditions are also being developed at the Center for Simulation of Advanced Rockets, see Najjar *et al.* (2000, 2002, 2003) and Dick and Heath (2002). The development and description of a parallel solution-adaptive method for predicting axisymmetric, multi-phase SRM core flows is the focus of the present paper. This parallel algorithm has been devised with a view to enable the computation of complex rocket motor flows on a more routine basis. A preliminary discussion of the algorithm was presented by Sachdev *et al.* (2003).

In the proposed approach, an Eulerian formulation is used to describe the coupled motion of both the gas and particle phases. A cell-centred upwind finite-volume discretization procedure is used on multi-block quadrilateral meshes to solve the governing partial differential equations in weak conservation form. Limited linear solution reconstruction and Riemann-solver based flux functions are used to evaluate the numerical fluxes for the gas and particle phases and an explicit multi-stage time-stepping procedure is used to integrate the governing equations in time. The injection of gas and particle into the core flow from the burning propellant is formulated as a Riemann problem and boundary data is obtained by an iterative scheme. The combustion interface is held stationary in this study; however, future work will involve a regressing burning surface. To prepare for this capability, a mesh adjustment algorithm has been implemented to adjust the multi-block quadrilateral mesh locally to the combustion interface. Adaptive mesh refinement (AMR) is used to accurately resolve multiple solution scales associated with SRM flows. A flexible block-based hierarchical data structure is used to facilitate automatic solution-directed mesh adaptation according to physics-based refinement criteria. This block-based data structure also lends itself naturally to domain decomposition and thereby enables efficient and scalable implementations of the algorithm on distributed-memory multiprocessor architectures.

A review of the governing equations is given in the next section followed by a description of the algorithm outlined above. The paper concludes by describing numerical results for inviscid flows that illustrate the capabilities of the proposed approach for predicting SRM core flows. The reference rocket motor chosen is based on the CRV-7 rocket system, which is SRM developed for military applications. The characteristics of the reference SRM and the propellant are summarized in table 4. 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 aluminum oxide particles account for 2–3% of the solid propellant by mass. Therefore, the modeling of burning particles and the transport of smoke are not required. In addition, the propellant grain is assumed to be stationary and uninhibited, and erosive and transient burning effects are neglected. Furthermore, the propellant ignition is assumed to be instantaneous. The proposed scheme is currently being extended to three-dimensional turbulent flow with a propagating propellant interface. This numerical scheme will provide a basic numerical tool for future work involving the effects of spin rate, structural oscillations (e.g. Greatrix (1999)), ignition dynamics (e.g. Greatrix *et al.* (1988) and Alavilli *et al.* (2000)) and more sophisticated burning models (e.g. King (1979), Renie and Osborn (1983), Greatrix and Gottlieb (1987), Brewster (2000) and Surzhikov *et al.* (2000)).

2. Governing equations

2.1 Conservation equations

An Eulerian formulation is used for both the gas and particle phases (Marble 1970). The Euler equations of inviscid compressible gasdynamics are used to describe the behavior of the gas-phase and a similar set of equations are used to describe the behavior of the particle-phase. The particle-phase is assumed to be inert (non-reacting), dilute (negligible volumetric fraction) and disperse (no particle–particle interactions) (Rudinger 1980) and consist of mono-sized spherically-shaped particles with uniform physical properties. Since the particle-phase is considered to be disperse, there can be no particle-pressure 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). 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. The set of partial differential equations governing the coupled motion of the disperse gas–particle flow for an axisymmetric coordinate system is given by

$$\frac{\partial \mathbf{U}}{\partial t} + \frac{\partial \mathbf{F}_r}{\partial r} + \frac{\partial \mathbf{F}_z}{\partial z} + \frac{\mathbf{S}}{r} = \mathbf{P}, \tag{1}$$

where $\mathbf{U}$ represents the conserved variable solution vector,

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

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 $\mathbf{v}$, u_r and u_z are the radial and axial components of the particle velocity $\mathbf{u}$, and E and E_p are the total energy per unit volume of the gas and particle-phases. The vectors $\mathbf{F}_r$ and $\mathbf{F}_z$ are the flux vectors in the r and z directions respectively and the vector $\mathbf{S}$ represents the sources due to the axisymmetric flow geometry,

$$\mathbf{F}_r = \begin{bmatrix} \rho v_r \\ \rho v_r^2 + p \\ \rho v_r v_z \\ v_r(E+p) \\ \sigma_p u_r \\ \sigma_p u_r^2 \\ \sigma_p u_r u_z \\ u_r E_p \end{bmatrix}, \quad \mathbf{F}_z = \begin{bmatrix} \rho v_z \\ \rho v_r v_z \\ \rho v_z^2 + p \\ v_z(E+p) \\ \sigma_p u_z \\ \sigma_p u_r u_z \\ \sigma_p u_z^2 \\ u_z E_p \end{bmatrix}, \tag{3}$$

$$\mathbf{S} = \begin{bmatrix} \rho v_r \\ \rho v_r^2 \\ \rho v_r v_z \\ v_r(E+p) \\ \sigma_p u_r \\ \sigma_p u_r^2 \\ \sigma_p u_r u_z \\ u_r E_p \end{bmatrix}.$$

The vector $\mathbf{P}$ represents the sources due to the gas–particle interaction

$$\mathbf{P} = \begin{bmatrix} 0 \\ -\frac{\sigma_p}{\tau_v}(v_r - u_r) f(Re_p) \\ -\frac{\sigma_p}{\tau_v}(v_z - u_z) f(Re_p) \\ -\frac{\sigma_p}{\tau_v}(\mathbf{v} - \mathbf{u}) \cdot \mathbf{u} f(Re_p) - \frac{\sigma_p c_p}{\tau_T}(T - T_p) \\ 0 \\ \frac{\sigma_p}{\tau_v}(v_r - u_r) f(Re_p) \\ \frac{\sigma_p}{\tau_v}(v_z - u_z) f(Re_p) \\ \frac{\sigma_p}{\tau_v}(\mathbf{v} - \mathbf{u}) \cdot \mathbf{u} f(Re_p) + \frac{\sigma_p c_p}{\tau_T}(T - T_p) \end{bmatrix}. \tag{4}$$

The relaxation times associated with the momentum (drag) and heat transfer between the gas and particle-phases are

$$\tau_v = \frac{m_p}{3\pi d_p \mu}, \quad \tau_T = \frac{m_p c_p}{2\pi d_p \kappa}, \tag{5}$$

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, $\tau_T/\tau_v = \frac{3}{2}Pr$. The function *f*($Re_p$) is a correction to the Stokes drag law for spherical particles which has the form

$$C_\mathrm{D} = \frac{24}{Re_p} f(Re_p) = \frac{24}{Re_p}\left(1 + \frac{1}{6} Re_p^{\frac{2}{3}}\right). \tag{6}$$

This drag coefficient, C_D, is valid for flow situations satisfying $Re_p < 1000$, where $Re_p = (\rho d_p/\mu)|\mathbf{v} - \mathbf{u}|$. The ratio of the particle density to gas density is assumed to be large enough such that the Basset history, fluid acceleration and added mass forces can be neglected (Rudinger, 1980).

The gas-phase is taken to be calorically perfect and the total energies of the two phases are then given by

$$E = \frac{p}{(\gamma - 1)} + \frac{\rho}{2}(\mathbf{v} \cdot \mathbf{v}), \tag{7}$$

$$E_p = \sigma_p c_m T_p + \frac{\sigma_p}{2}(\mathbf{u} \cdot \mathbf{u}), \tag{8}$$

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 provides a relationship between the gas pressure p and temperature T

$$p = \rho RT = \frac{\rho a^2}{\gamma}, \tag{9}$$

where $a = \sqrt{\gamma RT}$ is the sound speed and R is the gas constant.

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 their 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.

2.2 Degenerate hyperbolic system

It has been shown by previous authors that the set of equations governing disperse gas–particle flows is both hyperbolic and degenerate (Sauerwein and Fendell 1965, Saurel *et al.* 1994, Slater and Young 2001). A brief summary of the cause and implications of the degeneracy is as follows.

In order to assess the hyperbolicity and degeneracy of the two-phase flow equations of interest, consider the one-dimensional, weak conservation, form of the equations given above:

$$\frac{\partial \mathbf{U}}{\partial t} + \frac{\partial}{\partial x}\mathbf{F}(\mathbf{U}) = \mathbf{P}(\mathbf{U}), \qquad (10)$$

where $\mathbf{U} = [\rho, \rho v, E, \sigma_p, \sigma_p u, E_p]^T, \mathbf{F}(\mathbf{U})$ is the one-dimensional flow state vector of conserved variables and $\mathbf{F}(\mathbf{U})$ and $\mathbf{P}(\mathbf{U})$ are the flux and phase-interaction source vectors, respectively. This is an inhomogeneous hyperbolic system. In the "frozen" limit, the characteristic times scales of the particle drag and heat transfer, τ_v and τ_T can be assumed to be large, relative to the differences in velocities and temperatures of the two phases such that the phase-interaction terms can be neglected. In this limit, the equations governing the gas and particle-phases decouple. An eigenanalysis of the system in the frozen limit provides the set of real eigenvalues, λ_k,

$$\lambda_1 = v - a, \quad \lambda_2 = v, \quad \lambda_3 = v + a, \quad \lambda_{4,5,6} = u, \qquad (11)$$

indicating the hyperbolic nature of the equations. Note the three repeated eigenvalues associated with the particle-phase. The right eigenvectors corresponding to the frozen-limit eigenvalues are

$$(\mathbf{r}_1, \mathbf{r}_2, \mathbf{r}_3, \mathbf{r}_4, \mathbf{r}_5) = \begin{bmatrix} 1 & 1 & 1 & 0 & 0 \\ -a/\rho & 0 & a/\rho & 0 & 0 \\ a^2 & 0 & a^2 & 0 & 0 \\ 0 & 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 1 \end{bmatrix}. \qquad (12)$$

This is an incomplete set of eigenvectors as only five linearly independent eigenvectors can be found for the six eigenvalues. An eigenvector associated with the repeated $\lambda = u$ eigenvalue is missing, indicating a degeneracy in the eigensystem. Therefore, the set of hyperbolic conservation equations governing disperse gas–particle flows is said to be degenerate.

The preceding degeneracy is a direct result of the assumptions that the particle volume fraction is negligible (dilute) and that the effects of inter-particle collisions are not important (disperse). The lack of particle collisions means that there are no normal surface forces produced by the random motion of the particles themselves and hence there is no pressure-like term in the particle momentum and energy equations. Physically, the main implications of the degeneracy are two-fold. First, particle vacuums can exist, as there are no direct pressure forces to drive the particles from regions of high concentrations to those with low concentrations. Moreover, as interactions between particles have been neglected, the paths of particles can cross. Faster moving particles can freely over-take and pass slower moving particles without being subject to any particle–particle interaction forces.

The degenerate nature of the two-phase flow equations must be accounted for when designing a Godunov-type numerical scheme for their solution. Although Lagrangian modeling methods (e.g. Hwang and Chang (1988), Carrier *et al.* (1991) and Najjar *et al.* (2000)) can readily deal with particle trajectories that cross, Eulerian finite-volume methods can produce inaccurate and physically incorrect numerical solutions. This is a result of insufficient characteristic fields from which to reconstruct the solution and therefore solution information and accuracy can be lost. The solution algorithm proposed here will make use of the Riemann solver of Saurel *et al.* (1994) that allows particle paths to cross, however, flows involving the compression of the particle-phase are still problematic.

3. Numerical method

3.1 Finite volume scheme

In this work, an explicit higher-order Godunov-type finite-volume scheme is used to solve the gas–particle equations. Upwind finite-volume schemes for the gasdynamic equations were originally introduced by Godunov (1959). Application and development of these schemes for the gasdynamic equations has been well documented in literature, see van Leer (1973, 1974, 1977a,b, 1979, 1982), Roe (1981, 1984), Harten (1983, 1984), Colella and Woodward (1984), Osher (1984), Harten *et al.* (1987) and Yee (1987).

In this finite-volume approach, the governing equations are integrated over quadrilateral cells of a structured multi-block quadrilateral mesh. The preceding finite volume procedure applied to cell (i,j) results in the following semi-discrete form of the equations,

$$\frac{\mathrm{d}\mathbf{U}_{ij}}{\mathrm{d}t} = -\frac{1}{A_{ij}}\sum_k \vec{\mathbf{F}}_{ij_k} \cdot \hat{\mathbf{n}}\Delta\ell - \frac{\mathbf{S}_{ij}}{r} + \mathbf{P}_{ij}, \qquad (13)$$

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 inviscid numerical fluxes at the faces of each cell are determined from the solution of a Riemann problem. Given the left and right solution states, $\mathbf{U}_\mathrm{l}$ and $\mathbf{U}_\mathrm{r}$, at the cell interfaces, the numerical flux is given by

$$\vec{\mathbf{F}} \cdot \hat{\mathbf{n}} = \mathcal{F}(\mathbf{U}_\mathrm{l}, \mathbf{U}_\mathrm{r}, \hat{\mathbf{n}}) \qquad (14)$$

where $\mathcal{F}$ is evaluated by solving a Riemann problem in a direction defined by the normal to the face, $\hat{\mathbf{n}}$, with initial data $\mathbf{U}_\mathrm{l}$ and $\mathbf{U}_\mathrm{r}$. The left and right solution states are determined using the least squares piece-wise limited linear solution reconstruction procedure of Barth (1993). The modified limiter of Venkatakrishnan (1993) has also been implemented.

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. Hence, separate Riemann problems and solutions can be formulated for the two phases.

For the gas-phase, the flux functions of Roe (1981), Einfeldt (1988) (HLLE), Linde (2002) (HLLL), and the exact Riemann solver of Gottlieb and Groth (1988) have all been implemented. For the particle-phase, the Riemann solver proposed by Saurel *et al.* (1994) has been implemented and is discussed in the next section.

For time-accurate calculations, predictor–corrector and fourth order Runge-Kutta time-marching methods are used to integrate the set of ordinary differential equations that result from the spatial discretization of the governing equations. The optimally-smoothing multi-stage schemes developed by van Leer *et al.* (1989) are adopted for steady-state calculations.

3.2 Particle-phase riemann solver

Determination of the flux values at cell interfaces for the particle-phase is complicated by the degeneracy of the governing flow equations described above. A particle-phase Riemann solver was proposed by Saurel *et al.* (1994). This flux function is now briefly reviewed and its limitations are discussed.

Given the left and right particle-phase solution states defined by $\mathbf{W}_\mathrm{l} = [\sigma_\mathrm{pl}, u_\mathrm{l}, T_\mathrm{pl}]^T$ and $\mathbf{W}_\mathrm{r} = [\sigma_\mathrm{pr}, u_\mathrm{r}, T_\mathrm{pr}]^T$, three particle expansion wave configurations ($u_\mathrm{l} < u_\mathrm{r}$) and three particle compression wave configurations ($u_\mathrm{l} > u_\mathrm{r}$) were identified by Saurel *et al.*, and used to construct solutions to the Riemann problem (refer to figure 1). For the expansion configuration, if u_l, $u_\mathrm{r} > 0$ then the interface solution provided by the solution of the Riemann problem is simply the left state. If u_l, $u_\mathrm{r} < 0$ then the interface solution state is given by the right state. A strong expansion occurs when $u_\mathrm{l} < 0$ and $u_\mathrm{r} > 0$. In this case the particles of the left and right states are moving apart in opposite directions, leaving a vacuum state at the interface.

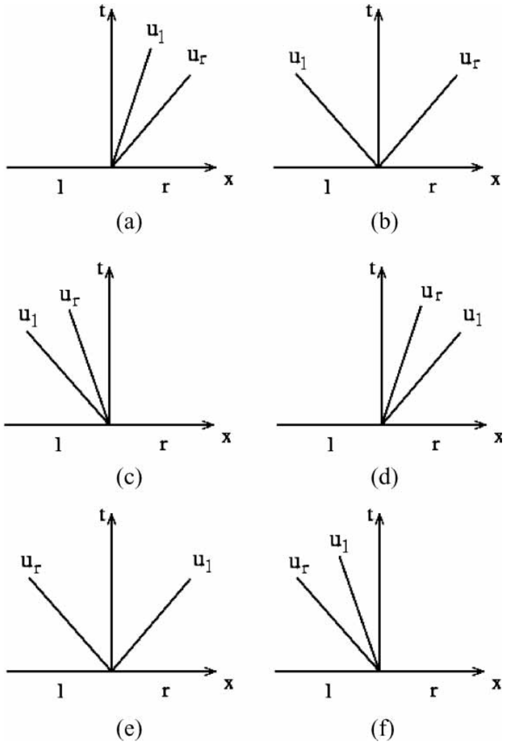


Figure 1. The six possible wave patterns for the particle-phase Riemann problem of Saurel *et al.* (1994).

For the compressive configurations ($u_\mathrm{l} > u_\mathrm{r}$), if u_l, $u_\mathrm{r} > 0$ then the interface solution for the Riemann problem is the left state and if $u_\mathrm{l}, u_\mathrm{r} < 0$ the solution state is the right state. A strong compression of the particles occurs when $u_\mathrm{l} > 0$ and $u_\mathrm{r} < 0$. In the proposed Riemann solver of Saurel *et al.*, the solution state is then given by $\sigma_p^* = \sigma_\mathrm{pl} + \sigma_\mathrm{pr}$, $u^* = (\sigma_\mathrm{pl} u_\mathrm{l} + \sigma_\mathrm{pr} u_\mathrm{r})/\sigma_p^*$ and $T_p^* = (\sigma_\mathrm{pl} T_\mathrm{pl} + \sigma_\mathrm{pr} T_\mathrm{pr})/\sigma_p^*$.

In all of the three compressive cases, the interface solution compromises the actual physics of the particle motions. For the mildly compressive cases where one solution state is catching up with the other, the solution information carried by the overtaken particles is lost. In reality, this solution information should be retained. For the more strongly compressive case, the two waves (populations of particles) should simply pass through each other. Instead, the particle concentrations are directly summed and density weighted averages are assigned to the particle velocity and temperature. Although this averaging procedure provides the correct average state for the combined left and right moving particle populations, it fails to recognize the presence of the two oppositely moving populations and this solution content is lost and not retained in the numerical approximation of the solution. This can lead to the unphysical results and difficulties near solid boundaries as described by Saurel *et al.* (1994) and Slater and Young (2001) and is a basic limitation of single-velocity Eulerian formulations for describing particle-phase motions. Lagrangian formulations, which track and solve for the individual particle motions, can account for multiple particle trajectories within a computational cell and thereby avoid the problems associated with the degeneracy of Eulerian formulations. However, Lagrangian methods generally require significantly more computing resources than an Eulerian approach, in terms of both memory and computing time. This is particularly true for two-phase particle-laden flows where the phases are strongly coupled and have a significant affect on each other. Efficient and scalable parallel implementations of Lagrangian particle tracking formulations are also difficult to achieve.

3.3 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 used here to specify boundary data at the surface of the burning propellant. The treatment of the burning propellant boundary is very similar in spirit to the methods proposed by Gottlieb and Groth (1999) for imposing boundary data at a variety of flow boundaries based on the solution of Riemann problems.

Consider the combustion of a propellant surface. For the current analysis, the burning rate is computed by the application of the pressure dependent empirical St. Robert relation,

$$\rho_{\rm bs} v_{\rm bs} = (1-\alpha_s)\rho_s r_{\rm bs} = (1-\alpha_s)\rho_s \beta p_{\rm bs}^n \qquad (15)$$

where $\rho_{\rm bs}$, $v_{\rm bs}$ and $p_{\rm bs}$ are the density, normal velocity and pressure of the gas injected from the burning surface and $\rho_{\rm s}$ is the solid propellant density. The burning rate, $r_{\rm bs}$, is the same as the injected gas velocity and the burning rate constants β and n are functions of the chemical composition of the solid and the adiabatic flame temperature, $T_{\rm f}$. The mass fraction of solid particles in the propellant is given by $\alpha_{\rm s}$. Note that the speed of the propagating surface is $-r_{\rm bs}$. Although unsteady and erosive burning effects are not included here, it should be straightforward to include these influences in the present boundary scheme through the use of a modified burning rate.

The wave solution for the burning surface Riemann problem is shown in figure 2. There are two waves: a contact surface and a second wave, which can be a shock or a rarefaction wave. An exact Riemann solver has been constructed to solve this problem.

For the left-hand side boundary depicted in figure 2, the primitive variable states $\mathbf{V}_{\rm r}$, $\mathbf{V}_{\rm r}^*$ and $\mathbf{V}_{\rm l} = \mathbf{V}_{\rm l}^* = \mathbf{V}_{\rm bs}$ are given by

$$\mathbf{V}_{\rm r} = [\rho_{\rm r}, v_{\rm xr}, v_{\rm yr}, p_{\rm r}, \sigma_{\rm pr}, u_{\rm xr}, u_{\rm yr}, T_{\rm pr}]^T, \qquad (16)$$

$$\mathbf{V}_r^* = \left[\rho_r^*, v_{\rm bs}, v_{\rm yr}^*, p_{\rm bs}, \sigma_{\rm pr}^*, u_{\rm bs}, u_{\rm yr}^*, T_f\right]^T, \qquad (17)$$

$$\mathbf{V}_{\rm bs} = [\rho_{\rm bs}, v_{\rm bs}, 0, p_{\rm bs}, \sigma_{\rm pbs}, u_{\rm bs}, 0, T_f]^T \qquad (18)$$

where $u_{\rm bs} = v_{\rm bs}$. The resulting system of equations, the shock/rarefaction equations and the burning rate equation (15), is nonlinear and implicit in the unknown burning surface pressure or normal velocity, $p_{\rm bs}$ and $v_{\rm bs}$. A Newton iteration scheme is developed to solve this system with the burning surface pressure as the iterate. The propagation of the burning surface is accounted for through an application of a Galilean transformation to a frame in which the surface is stationary. In this frame, the right-state velocity is given by $v_{\rm xr}^s = v_{\rm xr} + r_{\rm bs}$. The burning surface velocity in the stationary frame, $v_{\rm bs}^s$, can now be determined by the iterative scheme.

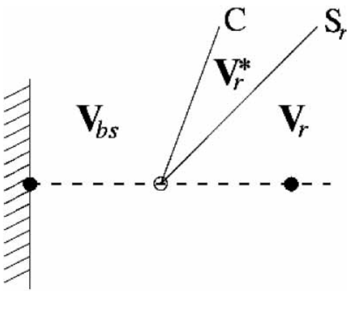


Figure 2. Burning surface wave pattern.

The right state pressure can be used as the initial guess. If the burning surface pressure is greater than or equal to the right state pressure, $p_{\rm bs} \geq p_{\rm r}$, then the right hand wave is a shock. The following equations are used to find the speed and density across the shock:

$$v_{\rm bs}^s = v_{\rm xr} + r_{\rm bs} + \frac{1}{\sqrt{C_1}}\frac{a_r}{\gamma}\left(\frac{p_{\rm bs}}{p_{\rm r}} - 1\right), \qquad (19)$$

$$v_{\rm bs}^{s\,\prime} = \frac{{\rm d}v_{\rm bs}}{{\rm d}p_{\rm bs}} = r_{\rm bs}^\prime + \frac{1}{\sqrt{C_1}}\frac{a_r C_2}{\gamma p_{\rm r}}, \qquad (20)$$

$$\rho_r^* = \rho_r\left[\frac{(\gamma+1)p_{\rm bs} + (\gamma-1)p_{\rm r}}{(\gamma-1)p_{\rm bs} + (\gamma+1)p_{\rm r}}\right],$$

$$C_1 = \frac{\gamma+1}{2\gamma}\frac{p_{\rm bs}}{p_{\rm r}} + \frac{\gamma-1}{2\gamma}, \qquad (21)$$

$$C_2 = 1 - \frac{1}{C_1}\frac{\gamma+1}{4\gamma}\left(\frac{p_{\rm bs}}{p_{\rm r}} - 1\right).$$

If $p_{\rm bs} < p_{\rm r}$, then the right-hand wave is a rarefaction wave. The following equations can be used to determine the speed and density across the rarefaction wave:

$$a_{\rm r}^* = a_{\rm r}\left(\frac{p_{\rm bs}}{p_{\rm r}}\right)^{\frac{\gamma-1}{2\gamma}}, \qquad (22)$$

$$v_{\rm bs}^s = v_{\rm xr} + r_{\rm bs} + \frac{2}{\gamma-1}(a_{\rm r}^* - a_{\rm r}), \qquad (23)$$

$$v_{\rm bs}^{s\,\prime} = r_{\rm bs}^\prime + \frac{a_{\rm r}^*}{\gamma p_{\rm bs}}, \qquad (24)$$

$$\rho_{\rm r}^* = \rho_{\rm r}\left(\frac{p_{\rm bs}}{p_{\rm r}}\right)^{\frac{1}{\gamma}}. \qquad (25)$$

The gas density and particle concentration injected into the core by the burning surface can be determined from $p_{\rm bs} = \rho_{\rm bs} {\rm R} T_f$ and $\sigma_{\rm bs} = \alpha_s \rho_s r_{\rm bs}/v_{\rm bs}^s$. The derivative of St. Robert's burning rate equation (15), with respect to the pressure at the burning surface is given by

$$r_{\rm bs}^\prime = n\frac{r_{\rm bs}}{p_{\rm bs}}. \qquad (26)$$

The pressure is then up-dated by

$$p_{\rm bs}^{(n+1)} = p_{\rm bs}^{(n)} - \frac{\rho_s r_{\rm bs} - \rho_{\rm bs} v_{\rm bs}^s}{\rho_s r_{\rm bs}^\prime - (\rho_{\rm bs} v_{\rm bs}^s)^\prime} \qquad (27)$$

where the derivative of the gas flux from the burning surface, $(\rho_{\rm bs} v_{\rm bs}^s)'$, with respect to $p_{\rm bs}$ is

$$\left(\rho_{\rm bs} v_{\rm bs}^s\right)' = \frac{\rho_{\rm bs} v_{\rm bs}^s}{p_{\rm bs}} + \rho_{\rm bs} v_{\rm bs}^{s\,\prime}. \tag{28}$$

The iteration process is complete when $|1 - p_{\rm bs}^{(n)}/p_{\rm bs}^{(n+1)}| < \varepsilon$. Finally, the burning surface velocity in the initial frame can be determined by $v_{\rm bs} = v_{\rm bs}^s - r_{\rm bs}$.

3.4 Mesh adjustment scheme for embedded boundaries

The combustion interface is considered to be stationary in the current study. Future work will include the application of the level set method (Sethian 1999) to allow for the regression of the propellant interface as it burns. To prepare for this capability, we have developed a mesh adjustment algorithm to adjust an underlying quadrilateral structured mesh to the location of the combustion interface. Alternative approaches considered in other studies for dealing with boundaries not aligned with the mesh include the cut-cell (e.g. Bayyuk *et al.* (1993), De Zeeuw and Powell (1993), Coirier and Powell (1995) and Aftosmis *et al.* (1998)) and immersed boundary (e.g. Peskin (2002)) methods.

Figure 3 demonstrates the grid adjustment algorithm for a circular interface. The initial mesh is shown in the left-side portion of the figure and the interface is indicated by a thick solid line. The mesh is adjusted by merely locating mesh nodes that are closest to the points where the spline defining the interface intersects mesh lines connecting the nodes of the mesh. The resulting grid is shown in the right-side portion of the figure. 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." These in-active nodes are retained to maintain the block-based data structure and may be reactivated in computations involving moving interfaces. Note that for a moving interface, the mesh will have to be readjusted accordingly. This is accomplished by returning the grid to the original unadjusted form, and then readjusting the mesh to the new interface location. The construction of interpolation operators for the solution is required to maintain conservation and an accurate representation of the solution.

The preceding mesh adjustment algorithm will generate a piecewise linear representation of the interface boundary which allows for the accurate calculation of cell areas and straight-forward application of the interface boundary condition. An advantage of this mesh adjustment algorithm is that the (i,j) data structure of a structured mesh is maintained. In addition, it is advantageous that this adjustment algorithm does not require communication between the multiple solution-blocks and is completely transparent to the block-based AMR scheme to be discussed in the next section. It can be seen from the adjusted mesh in Fig. 3 that triangular cells can be generated from this scheme which requires the flow solver to consider a number of special cases when determining the fluxes at cell interfaces.

Figure 4 demonstrates the grid adjustment algorithm for the combustion interface in a rocket motor. The initial structured multi-block grid is shown in the upper portion where the propellant interface is indicated by a thick solid line and the resulting adjusted grid is shown in the lower portion.

3.5 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. Following the approach developed by Groth *et al.* (1999, 2000) for computational magnetohydrodynamics, 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

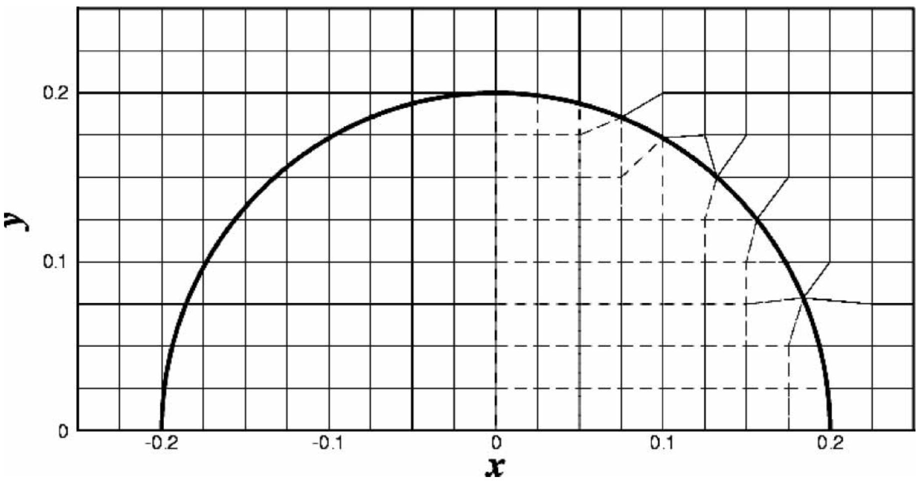


Figure 3. Initial (left panel of figure) and adjusted (right panel of figure) mesh for a circular cylinder.

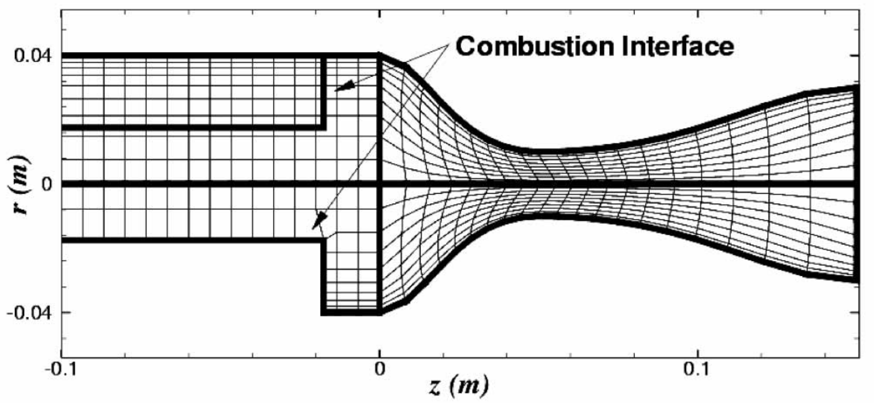


Figure 4. Initial (upper panel of figure) and adjusted (lower panel of figure) mesh for a cylindrical grain rocket motor.

solution-directed mesh adaptation on multiblock quadrilateral mesh according to physics-based refinement criteria. The proposed AMR formulation borrows from previous works by Berger (1982, 1984), Berger and Colella (1989), Berger and LeVeque (1989), Quirk (1991) and De Zeeuw and Powell (1993) and has similarities with the block-based approaches described by Quirk and Hanebutte (1993) and Berger and Saltzman (1994). 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 works. 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 works by Davis and Dannenhoffer (1993), Sun and Takayama (1999) and Bramkamp *et al.* (2000).

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 as done previously by Paillère *et al.* (1992) and Powell *et al.* (1993, 1999). Six flow quantities or refinement criteria, ϵ_k, are used herein,

$$\epsilon_1 \propto |\nabla\rho| \quad \epsilon_2 \propto |\nabla\cdot\mathbf{v}| \quad \epsilon_3 \propto |\nabla\times\mathbf{v}| \tag{29}$$

$$\epsilon_4 \propto |\nabla\sigma_p| \quad \epsilon_5 \propto |\nabla\cdot\mathbf{u}| \quad \epsilon_6 \propto |\nabla\times\mathbf{u}| \tag{30}$$

The first three quantities correspond to local measures of 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 multiphase 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 (Berger 1984, Berger and Colella 1989). In particular,

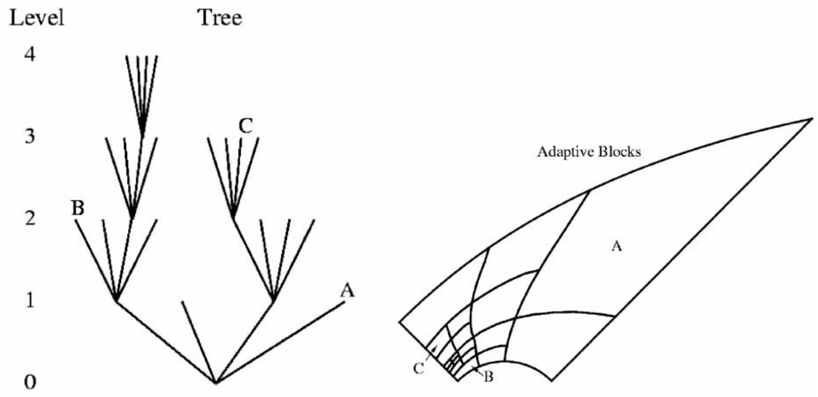


Figure 5. Solution blocks of a computational mesh with four refinement levels originating from one initial block and the associated hierarchical quadtree data structure. Interconnects to neighbours are not shown.

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.

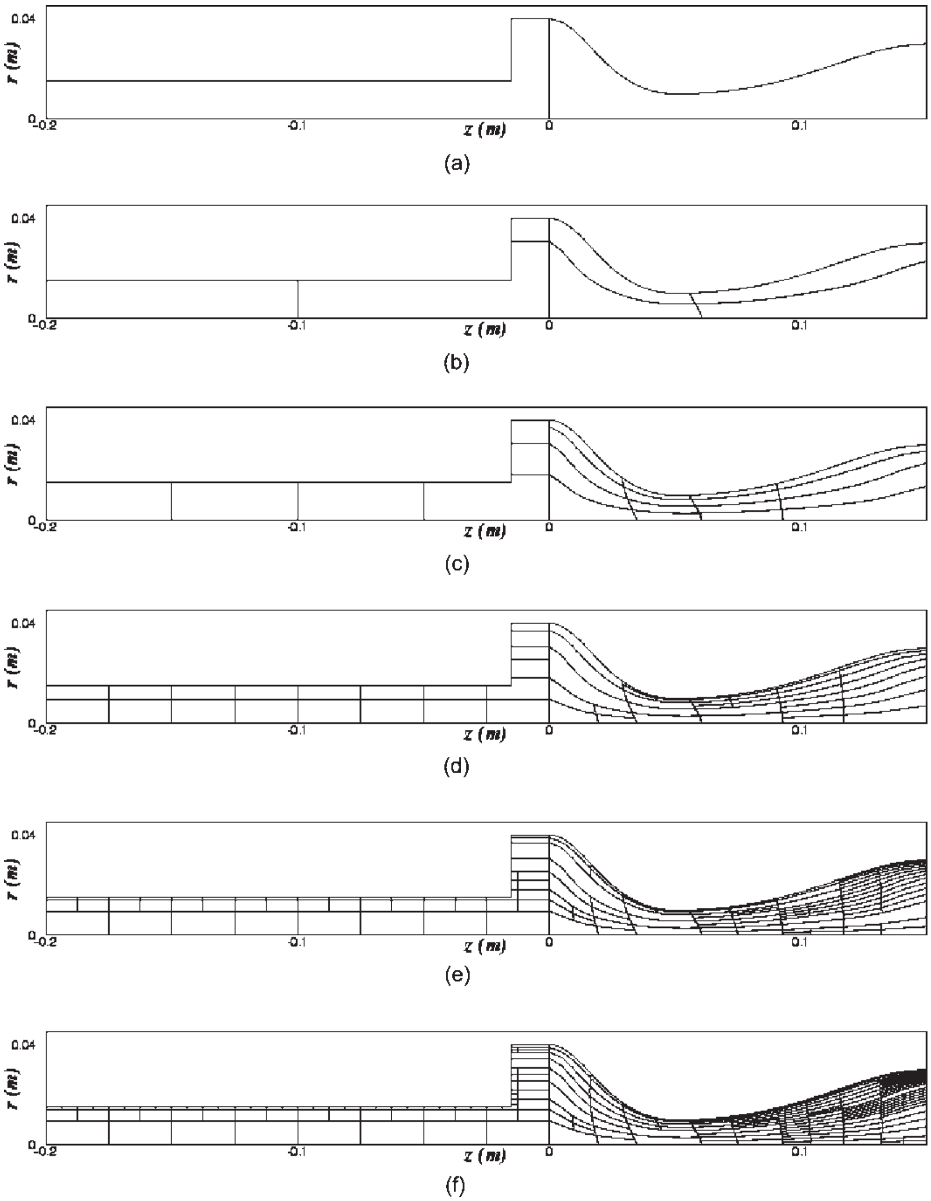
A hierarchical tree-like data structure with multiple “roots”, multiple “trees”, and additional interconnects between the “leaves” of the trees is used to keep track of mesh refinement and the connectivity between solution blocks. This interconnected “forest” data structure is depicted in figure 5. The blocks of the initial 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. Local modifications to the multi-block mesh can be performed without re-gridding the entire mesh and re-calculating solution block connectivity.

An example illustrating the adaptation of a two-dimensional multi-block quadrilateral mesh for a simplified rocket motor core flow geometry given in table 4 is shown in figure 6. The figure shows an initial grid (top pane) consisting of three blocks and 2304 cells (32×24 cell blocks were used) and five meshes derived by applying five levels of refinement. Note that the figures do not show the entire computational domain but only the nozzle section where the majority of the refinement occurs. The solution block boundaries are depicted in the figures. The statistics corresponding to the mesh refinement are summarized in table 1 where N_{blocks} is the total number of blocks, N_{cells} is the total number of cells, and η is a measure of the efficiency of the block-based AMR scheme defined by

$$\eta = 1 - N_{\mathrm{cells}}/N_{\mathrm{uniform}} \tag{31}$$

where N_{uniform} is the total number of cells that would have been used on a uniform mesh composed of cells of the finest size on the current mesh. The efficiency of the AMR scheme improves as the number of refinement levels increase. This indicates the ability of the block-based AMR approach to deal with flows having disparate spatial scales, providing reduced numbers of cells while maintaining cell resolution in areas of interest. Figures 14–17 indicate that the scheme is clustering solution blocks in areas of high acceleration and extreme particle concentrations (near all solid boundaries). The numerical results for this case are discussed in more detail in a later section. An enlargement of the throat-area of the nozzle after seven refinement levels is shown in figure 7. The mesh, in this case, has an refinement efficiency of $\eta = 0.956$ and consists of 2187 blocks and 1,679,616 cells. The application of the mesh adjustment algorithm to the corner of the combustion interface is shown in the inset. Note that each level of refinement in the grid introduces cells that are typically smaller by a factor of two in each spatial dimension. Practical calculations may have 10–15 levels of refinement. In the case of 15 levels of refinement, the finest cells in the mesh are more than 32,000 (2^{15}) times smaller than the coarsest cells. Also, note that the initial mesh stretching, in this case applied to the mesh in the radial direction, is retained by the mesh refinement procedure such that the refined blocks and the cells within them are clustered near the upper boundary of the mesh. Use of cell stretching and cell clustering in the initial mesh enables anisotropic refinement of the multi-block grid, which will be particularly important for resolving boundary and shear layers in subsequent studies of viscous core flows in rocket motors.

The mesh adjustment scheme discussed in the previous section works only on single solution blocks, however, the block-based AMR scheme requires the mesh to be completely unadjusted before refining. Therefore, the mesh must be readjusted after the mesh refinement occurs. Note that the constraint that limits grid resolution changes by a factor of two between adjacent blocks holds true even within the region inactivated by the mesh adjustment. In extreme cases such as those experienced by the rocket motor example, many solution blocks are completely



(a)

(b)

(c)

(d)

(e)

(f)

Figure 6. Illustration of AMR for a two-dimensional multi-block quadrilateral for a cylindrical grain rocket motor: (a) initial mesh (3 blocks, 2304 cells), (b) first refinement (12 blocks, 9216 cells), (c) second refinement (36 blocks, 27,648 cells), (d) third refinement (120 blocks, 214,272 cells), (e) fourth refinement (279 blocks, 214,272 cells) and (e) final refinement (570 blocks, 437,760 cells). Each block contains 32 × 24 cells. Computation cells are not shown.

ignored during the solution procedure if they fall within the region internal to the interface. This will have implications on the load balancing for the parallel implementation discussed in the next section.

Table 1. Statistics regarding the AMR for a two-dimensional multi-block quadrilateral for a cylindrical grain rocket motor.

	N_{blocks}	N_{cells}	η
Initial mesh	3	2304	0.000
First refinement	12	9216	0.000
Second refinement	36	27,648	0.250
Third refinement	120	92,160	0.375
Fourth refinement	279	214,272	0.637
Final refinement	570	437,760	0.814

3.6 Parallel implementation

Although the block-based AMR approach described above is somewhat less flexible and incurs some inefficiencies in solution resolution as compared to 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

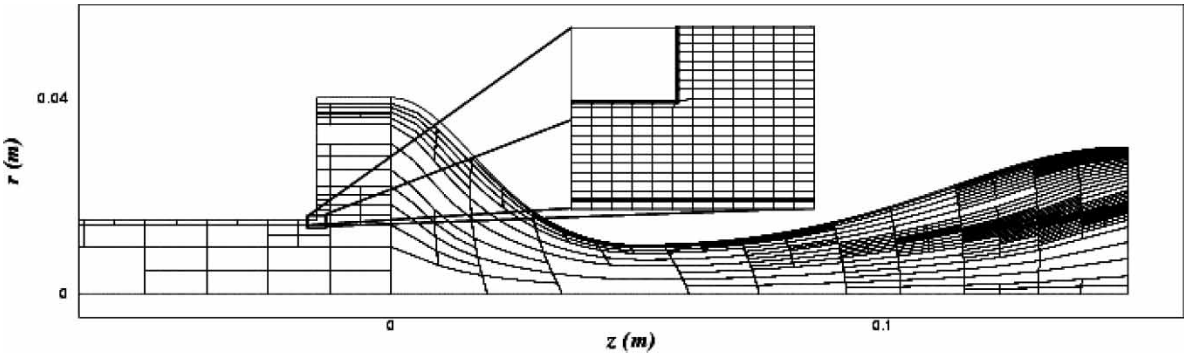


Figure 7. Illustration of the AMR for a two-dimensional multi-block quadrilateral for a cylindrical grain rocket motor after six refinements with 2187 blocks and 1,679,616 cells ($\eta = 0.956$). The inset depicts the adjusted mesh at the corner of the propellant interface. Computation cells are shown only in the inset.

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 message passing interface (MPI) library (Gropp *et al.* 1999). 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 amongst 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 blocks is an exact multiple of the number of processors available). 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 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 needs only to 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.

Implementation of the algorithm has been carried out on a Beowulf cluster of 4-way Hewlett-Packard ES40 and ES45 servers with a total of 104 Alpha processors. A low-latency Myrinet network and switch is used to interconnect the servers in the cluster. The parallel performance and scalability of the proposed method on this parallel architecture was assessed by conducting a series of numerical experiments using up to 48 processors. An explicit algorithm with nearly perfect load balancing should prove to be highly scalable. However, application of the mesh adjustment scheme and the block-based AMR scheme can lead to imperfect load balancing and therefore, a less scalable algorithm. Note that the percentage of inactivity of the solution blocks due to the adjustment of the mesh to an arbitrary interface is currently not considered when distributing the solution blocks to available processors. To investigate the effect of the mesh adjustment algorithm and the block-based AMR scheme on the parallel implementation, the parallel relative speed-up, S_p, and the parallel relative efficiency, E_p, are determined for fixed-size problems involving perfect and uneven load balancing. Both cases involve the two-phase flow rocket motor test case of figure 6. The geometry and characteristics of this rocket motor are given in table 4. Note that in a fixed-size problem, the total workload is constant and independent of the number of processors. The parallel relative speed-up, S_p, given by $S_p = t_1/t_p$, and the parallel relative efficiency, E_p, given by $E_p = S_p/p$, are determined as a function of the number of processors, p, where t_p is the time required to solve the problem using p processors and t_1 is the time required to solve the problem using a single processor. The results of these scale-up tests are presented in figures 8 and 9.

To achieve perfect load balancing the propellant interface was removed and combustion products were injected through the side-wall of the rocket motor.

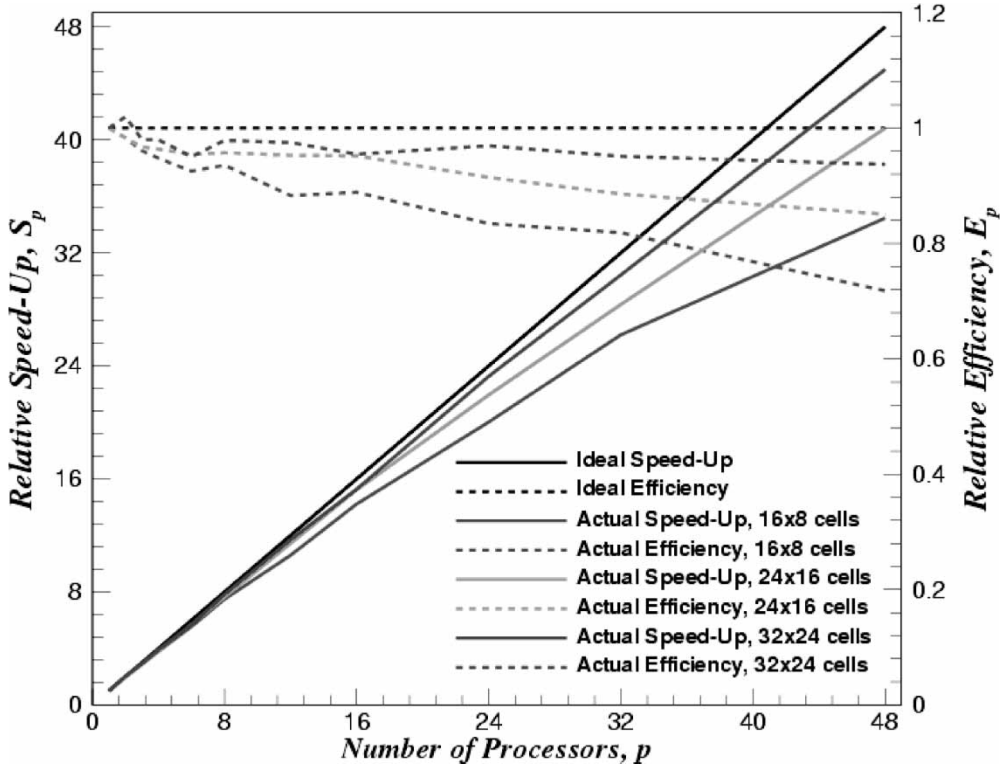


Figure 8. Parallel relative speed-up, S_p and parallel relative efficiency, E_p, for a fixed-size problem with perfect load-balancing using up to 48 processors.

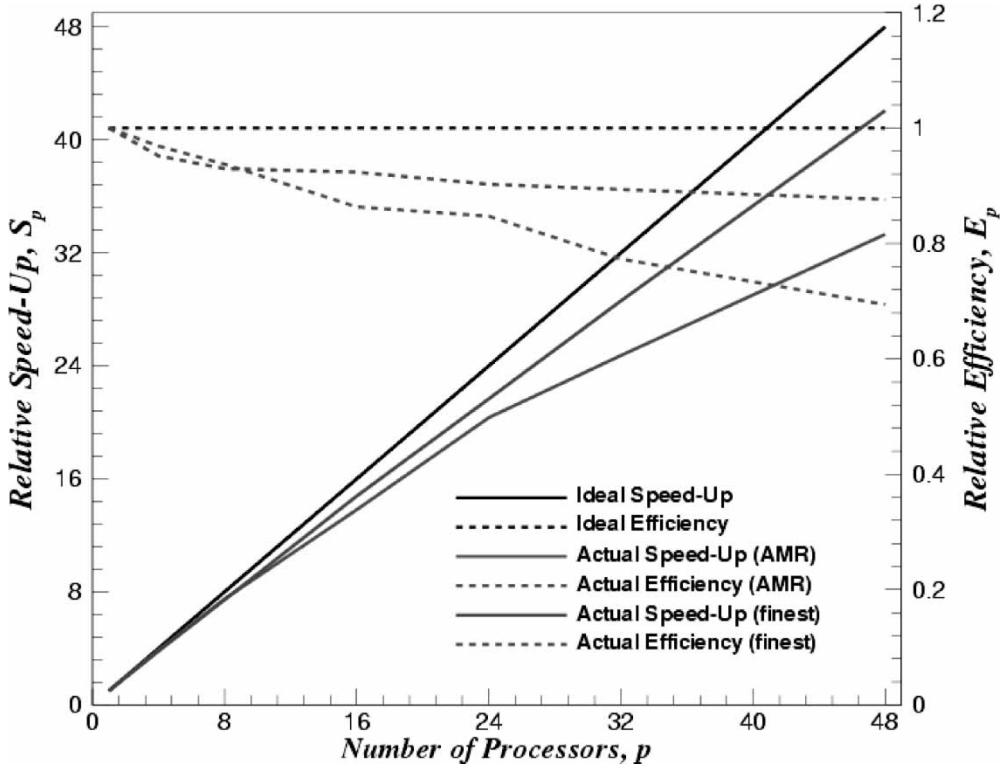


Figure 9. Parallel relative speed-up, S_p, and parallel relative efficiency, E_p, for a fixed-size problem involving the block-based AMR scheme using up to 48 processors and 32×24 cells per solution block.

Table 2. Statistics regarding the parallel relative speed-up, S_p and parallel relative efficiency, E_p, for a fixed-size problem with perfect load-balancing using up to 48 processors.

	Block size											
	16 × 8		16 × 16		24 × 16		24 × 24		32 × 24		32 × 32	
p	S_p	E_p	S_p	E_p	S_p	E_p	S_p	E_p	S_p	E_p	S_p	E_p
1	1.00	1.00	1.00	1.00	1.00	1.00	1.00	1.00	1.00	1.00	1.00	1.00
4	3.79	0.95	3.78	0.95	3.86	0.96	3.77	0.94	3.92	0.98	3.82	0.96
8	7.49	0.94	7.53	0.94	7.66	0.96	7.69	0.96	7.83	0.98	7.71	0.97
16	14.22	0.89	14.62	0.91	15.23	0.95	14.89	0.93	15.28	0.96	15.55	0.97
24	20.03	0.83	21.29	0.89	21.95	0.91	23.19	0.97	23.26	0.97	23.03	0.96
32	26.20	0.82	27.42	0.86	28.35	0.88	30.08	0.94	30.44	0.95	30.39	0.95
48	34.46	0.72	39.58	0.82	40.82	0.85	44.03	0.92	44.98	0.94	43.60	0.91

The results of this scale-up test are shown in figure 8 for three different block sizes. Table 2 presents the performance data for these and three additional block sizes. It can be seen that with perfect load balancing, block sizes of 32 × 24 cells per block provide the best scale-up with a nearly linear 44.98 parallel relative speed-up and 93.7% parallel relative efficiency for up to 48 processors. The results for the different block sizes also indicate that when solving a particular rocket motor core flow, the number of cells used in each solution block will ultimately affect the trade-off between parallel efficiency and efficiency of the block-based mesh refinement procedure, with large block sizes improving the former but reducing the latter.

The rocket configuration depicted in figure 6 has been used to study the effect of the mesh adjustment scheme on the parallel performance of the algorithm (imperfect load balancing). For this case, the size of each solution block was 32 × 24 cells. This imperfectly-load-balanced case also included block-based AMR scheme so as to investigate the scalability of the numerical algorithm with increased numbers of blocks. Improved load balancing is expected as the number of blocks increases and is obtained. A steady state solution was determined on the initial mesh of figure 6(a) and an additional 1000 iterations were performed on each successively refined mesh resulting from the AMR procedure. The refined mesh are shown in figure 6(b)–(f). The relative parallel speed-up and relative parallel efficiency for this case is presented in figure 9. The relative parallel speed-up and efficiency for the entire AMR procedure and for the finest mesh level are shown in this figure. The relative parallel speed-up and efficiencies for the entire AMR procedure and each refinement level are summarized in table 3. It should be noted that for a high number of processors, the load balancing is poor for the first few refinement levels due to the low number of available solution blocks. The poor load-balancing leads to the poor parallel scalings for the first two refinement levels which achieve only 19 and 53% efficiency, respectively. However, the overall AMR scheme is 75% efficient for up to 48 processors. Moreover, on the final mesh, the load-balancing is greatly improved and this is reflected in the performance parameters. For the fine mesh, the speed-up is much closer to linear and achieves 88% parallel efficiency for up to 48 processors.

An indication of how a parallel numerical algorithm can incorporate a fixed workload per processor can be determined by considering a scaled-size problem, in which the workload scales with the number of processors. Here, the important performance parameters are the parallel scaled speed-up, S_p, given by $S_p = (t_1/t_p)p$ and the parallel scaled efficiency, E_p, given by $E_p = S_p/p$. The scaled parallel performance parameters are determined using the rocket motor test case with the propellant interface intact. It can be seen in figure 10 that when considering a scaled-size problem, the parallel scaled speed-up is nearly linear and maintains at least 80% efficiency for up to 48 processors, regardless of the poor load-balancing produced

Table 3. Statistics regarding the parallel relative speed-up, S_p and parallel relative efficiency, E_p, for the AMR procedure using up to 48 processors.

	AMR scheme		First Refinement		Second Refinement		Third Refinement		Fourth Refinement		Fifth Refinement	
p	S_p	E_p	S_p	E_p	S_p	E_p	S_p	E_p	S_p	E_p	S_p	E_p
1	1.00	1.00	1.00	1.00	1.00	1.00	1.00	1.00	1.00	1.00	1.00	1.00
4	3.87	0.96	3.76	0.92	3.87	0.97	3.98	1.00	3.82	0.96	3.81	0.95
8	7.50	0.94	5.33	0.67	7.24	0.91	8.01	1.00	7.40	0.93	7.44	0.93
16	13.81	0.86	9.34	0.58	11.56	0.72	14.46	0.90	14.15	0.88	14.78	0.92
24	20.33	0.84	9.28	0.39	15.54	0.64	21.88	0.91	22.30	0.93	21.66	0.90
32	24.73	0.77	9.28	0.29	16.06	0.50	28.56	0.89	27.51	0.86	28.60	0.89
48	33.31	0.69	9.23	0.19	25.56	0.53	36.61	0.76	38.13	0.79	42.06	0.88

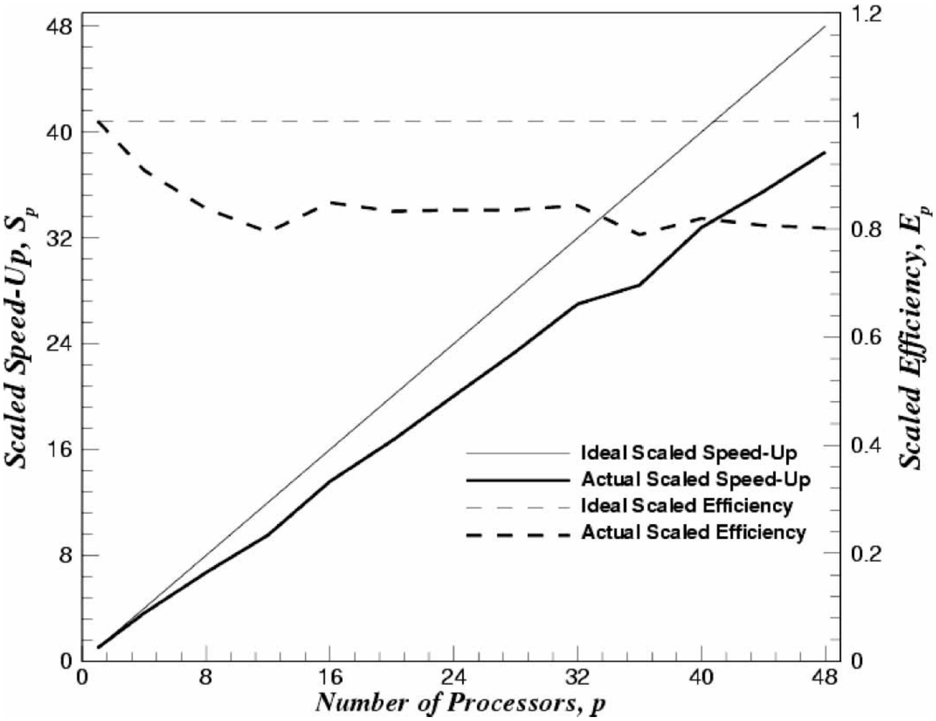


Figure 10. Parallel scaled speed-up, S_p, and parallel scaled efficiency, E_p, for a scaled-size problem using up to 48 processors.

by the mesh adjustment. The over-head time, t_o, given by $t_o = pt_p - t_1$ and normalized by $\max(t_o)$ and the normalized wall-time, t_w, given by $t_w = t_p/t_1$ are presented in figure 11 as a function of the number of processors. An algorithm is considered to be highly scalable if the iso-efficiency function, the relationship between the over-head time and the number of processors, is linear. It can be seen in figure 11 that the iso-efficiency function is indeed linear and the normalized wall-time remains fairly constant for up to 48 processors.

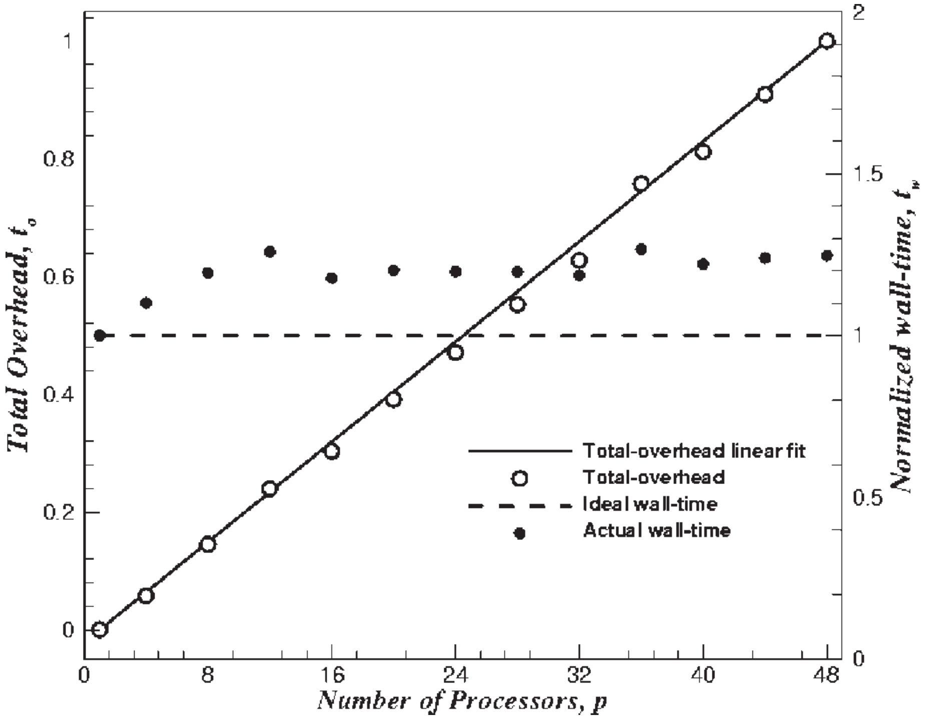


Figure 11. Parallel overhead time, t_o, and wall-time, t_w for a scaled-size problem using up to 48 processors.

4. Numerical results

4.1 Ringleb's flow

In order to first assess and demonstrate the accuracy of the spatial discretization scheme, the predictions of the proposed parallel 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 (e.g. Barth and Fredrickson (1990), Coirier and Powell (1995) and Hartmann and Houston (2002)). The flow pattern, shown in figure 12(a), involves an isentropic, irrotational flow contained between two streamlines. The analytic solution can be parameterized in terms of a non-dimensional velocity q, and the streamline number, $k = 1/\psi$:

$$x(q,k) = \frac{1}{2\rho}\left(\frac{2}{k^2} - \frac{1}{q^2}\right) - \frac{J}{2}, \tag{32}$$

$$y(q,k) = \pm\frac{1}{\rho k q}\sqrt{1-(q/k)^2}, \tag{33}$$

$$c = \sqrt{1 - \frac{\gamma - 1}{2}q^2} \tag{34}$$

$$\bar{\rho} = c^{2/(\gamma-1)}, \tag{35}$$

$$J = \frac{1}{c} + \frac{1}{3c^3} + \frac{1}{5c^5} - \frac{1}{2}\ln\left(\frac{1+c}{1-c}\right) \tag{36}$$

where the density, $\bar{\rho}$, is made non-dimensional by the stagnation density and the velocity q, and sound-speed c, are made non-dimensional by the stagnation sound-speed. The flow angle, θ, can be determined from

$$\theta = 2\pi - \sin^{-1}(q/k). \tag{37}$$

The flow pattern shown in figure 12(a) is defined by the streamlines corresponding to $k = 0.75$ and 1.5, and 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. Although Ringleb's flow involves only the gas-phase it does provide a good test of the gasdynamic portion of the solver and overall accuracy of the scheme.

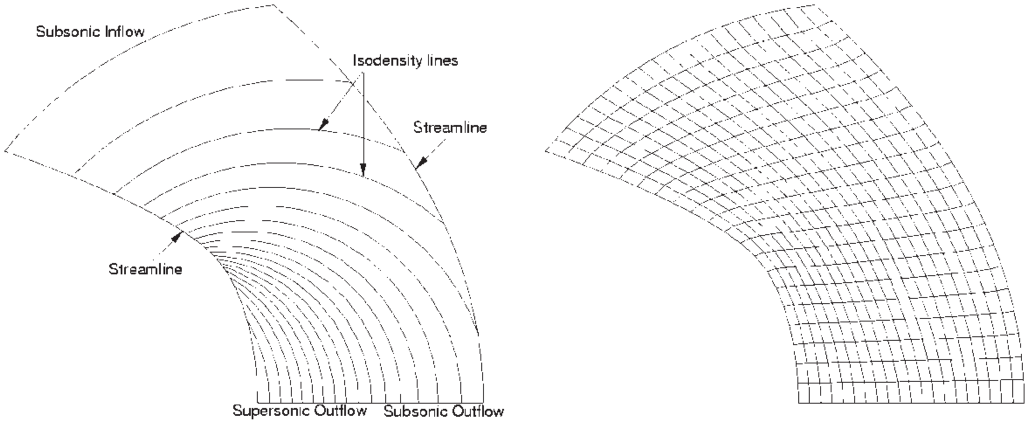
The accuracy of the spatial discretization was assessed by comparing the computed solution on a series of meshes involving 100, 400, 1600 and 6400 cells to the analytic solution. The L_1- and L_2-norms of the difference in the solution densities were used as the measure of solution accuracy and were taken to have the form

$$L_p = \left(\frac{\sum|\rho_{ij} - \rho_{\text{exact}}(x_{ij}, y_{ij})|}{N}\right)^{1/p} \tag{38}$$

where N is the total number of computational cells and ρ_{ij} is the computed density at cell (i,j). The exact density at a point (x,y) can be found by iteratively solving the equation for the constant velocity lines,

$$\left(x - \frac{J}{2}\right)^2 + y^2 = \frac{1}{4\rho^2 q^4} \tag{39}$$

for the non-dimensional sound-speed c, from which the non-dimensional density is computed by equation (35). A sample mesh with 400 cells is shown in figure 12(b). Reflection boundary conditions are applied at the lateral boundaries and the exact solution is used at the inflow and outflow boundaries. The left and right states for the evaluation of the Riemann problem on the inflow and outflow boundaries are determined from a characteristic boundary condition with a specified static pressure based on the exact solution at the Gauss point. The proposed algorithm performs well for this test case. The L_1- and L_2-norms of the solution error are plotted in figure 13. The slopes of the L_1- and L_2-norms are 2.11 and 1.94, respectively, indicating that the scheme is indeed second order accurate.



(a) Flow pattern.

(b) Sample mesh with 400 cells.

Figure 12. Ringleb's flow.

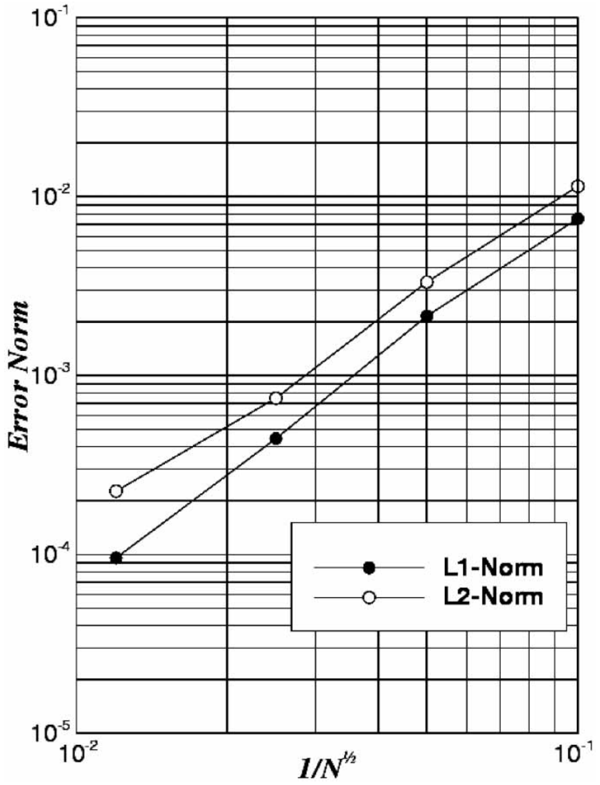


Figure 13. L_1- and L_2-norms of the solution error for Ringleb's flow as a function of the number of computational cells, N.

4.2 Rocket motor results

Predicted SRM core flow results for a configuration typical of the CRV-7 rocket system are now described to further demonstrate the viability and capability of the proposed scheme for multi-phase flows. The predicted results are shown in figures 14–17 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 14.5 mm. Predicted results of the type shown in the figures have also been validated quantitatively by comparing the two-dimensional predictions with the experimentally validated one-dimensional results obtained by Greatrix *et al.* (1987). The overall agreement for the centre-line pressure and Mach number is found to be very good. 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 4.

The solution procedure involved integrating the conservation equations to steady-state then refining the mesh and computing again. This is repeated for five mesh refinements. The initial mesh configuration is shown in figure 6(a) and the subsequent refinements are given by figure 6(b)–(f). All figures present the predictions for the finest mesh unless otherwise noted.

The burning of the solid propellant leads to a head end pressure in excess of 2.25 MPa, see figure 14, and produces sonic flow conditions at the nozzle throat and supersonic outflows in the rocket nozzle with Mach numbers approaching 3.25 as shown in figure 15. The Mach numbers and the gas-phase streamlines are depicted in figure 15 for the initial and final meshes in the upper and lower portions of the figures, respectively. Similarly, the particle-phase concentration contours through the converging–diverging nozzle are depicted in figure 16 for the initial and final meshes in the upper and lower portions of the figures, respectively. A comparison of the propellant gas and inert particle axial velocity components is shown in figure 17. This figure also depicts the gas and particle streamlines in the upper and lower portions of the figures, respectively.

It is clear from figure 17 that the particle-phase velocity lag relative to the gas-phase velocity after the rapid acceleration through the nozzle is represented in this figure. The particle streamlines show that the particles are unable to expand in the nozzle due to their relative high mass. As expected, a low speed recirculation zone is 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 high particle concentration near the junction between the propellant and the rocket casing since the inert particles are pushed away by the recirculation. This may be related to slag production. 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 Vuillot *et al.* (1997). However, this accumulation of particles may be somewhat unphysical due to the equation degeneracies and limitations of the particle-phase Riemann solver discussed previously. Reflective boundary conditions have been implemented at the upper boundary. Due to the degeneracy of the Eulerian formulation, the boundary condition instead behaves more like a slip boundary, where the particles will just slide along the wall until become entrained with the gas at the nozzle throat. 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. The improvement in the accuracy of the representation of the low-speed recirculation zone and high particle concentration regions due to the AMR scheme is evident from figures 15 and 16.

Finally it should be noted that, even with the regions of high particle concentration described above, the maximum value for the core flow particle concentration, σ_p, in this calculation was found to be about $0.35\,\mathrm{kg/m^3}$. For $\rho_p = 2700\,\mathrm{kg/m^3}$, this gives a maximum volume fraction for the particles of about 0.01% only, providing strong support for the assumption that particle volume effects are indeed negligible for this class of rocket motor flows.

5. Concluding remarks

A parallel AMR scheme has been described for solving the governing equations for multi-phase core flows in solid propellant rocket motors (SRMs). The two-phase flow

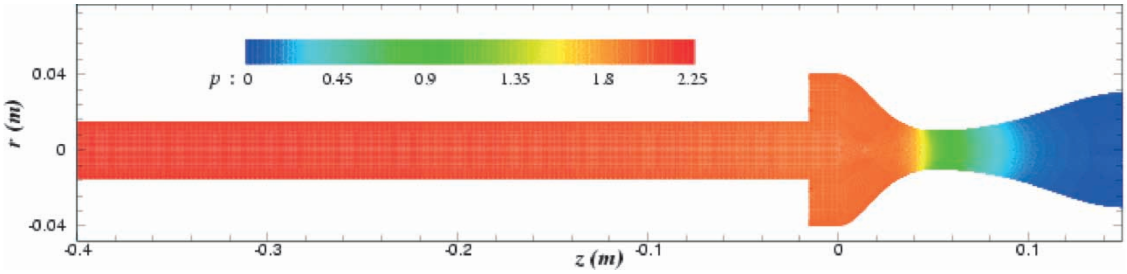


Figure 14. Predicted propellant gas pressure distribution for a cylindrical grain rocket motor.

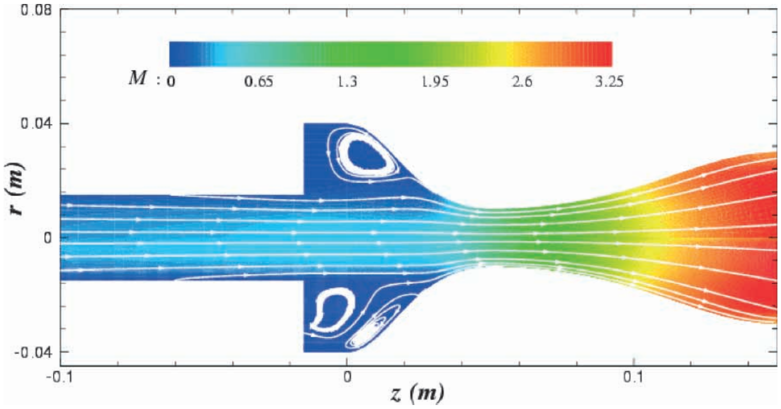


Figure 15. Predicted propellant gas Mach number distribution and streamlines for a cylindrical grain rocket motor calculated on the initial grid (upper panel of figure) and the grid after five mesh refinements (lower panel of figure).

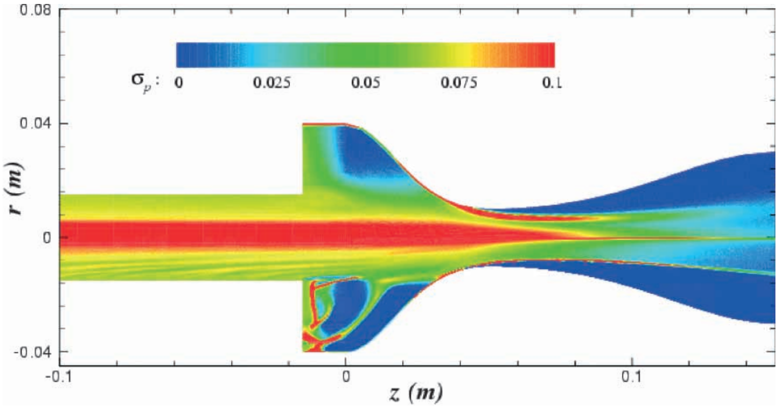


Figure 16. Predicted particle-phase concentration contours for a cylindrical grain rocket motor calculated on the initial grid (upper panel of figure) and the grid after five mesh refinements (lower panel of figure).

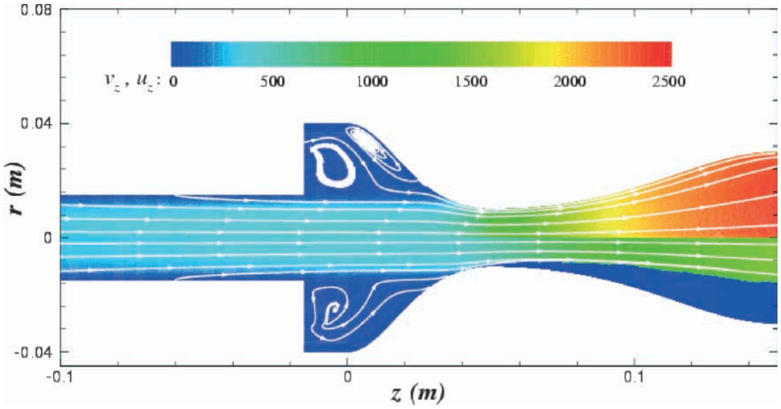


Figure 17. Axial velocity component and streamlines for the propellant gas-phase (upper panel of the figure) and inert particle-phase (lower panel of the figure) for a cylindrical grain rocket motor.

Table IV. Motor, propellant and particle characteristics.

	Reference motor
Combustion chamber length	40.0 cm
Inner casing diameter	8.00 cm
Internal port diameter	2.90 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 ($r_{\mathrm{bs}} = \beta p^n$)	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/(m K)
Gas absolute viscosity (μ)	8.19×10^{-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)

equations have been summarized and the degeneracy of the Eulerian formulation has been identified. The application of an upwind finite-volume discretization procedure and a parallel block-based AMR strategy has been shown to be a powerful tool for predicting SRM core flows. The proposed mesh adjustment algorithm effectively treats the combustion interface within the context of the block-based AMR scheme without resorting to cut and/or merged-cell approaches. The parallel implementation via domain decomposition can provide near-perfect parallel performance with both the mesh adjustment and AMR algorithms when good load balancing is achieved. Good load balancing can be readily achieved in the proposed method provided there are a sufficient number of blocks assigned to each processor (i.e. for large scale problems for which the algorithm has been designed).

Future work will include application of the level set method to allow for a regressing burning surface as well as extension to a viscous, turbulent core flow and the implementation of a parallel multi-grid method to improve the efficiency of the time integration procedure. Particle-phase treatments involving multiple particle families for coping with the degeneracy of the Eulerian formulation as suggested by Saurel *et al.* (1994) will also be explored. This numerical scheme will facilitate future investigations involving the effects of spin rate, structural oscillations (e.g. Greatrix (1999)), ignition dynamics (e.g. Greatrix *et al.* (1988)) and more sophisticated burning models (e.g. King (1979), Renie and Osborn (1983), Greatrix and Gottlieb (1987), Brewster (2000) and Surzhikov *et al.* (2000)).

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 authors would also like to express their appreciation to Professor David Greatrix of Ryerson University for the many insightful conversations regarding the internal ballistics of solid rocket motors.

References

Aftosmis, M.J., Berger, M.J. and Melton, J.E. Robust and efficient Cartesian mesh generation for component-base geometry. *AIAA J.*, 1998, **36**(6), 952–960.

Alavilli, P., Buckmaster, J., Jackson, T.L. and Short, M. Ignition-transient modeling for solid propellant rocket motors. *AIAA Paper*, 2000, 00–3567.

Barth, T.J. Recent developments in high order k-exact reconstruction on unstructured meshes. *AIAA Paper*, 1993, 93–0668.

Barth, T.J. and Fredrickson, P.O. Higher order solution of the Euler equations on unstructured grids using quadratic reconstruction. *AIAA Paper*, 1990, 90–0013.

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. *AIAA Paper*, 1993, 93–3391.

Berger, M.J. Adaptive mesh refinement for hyperbolic partial differential equations. PhD thesis, 1982, Stanford University.

Berger, M.J. Adaptive mesh refinement for hyperbolic partial differential equations. *J. Comput. Phys.*, 1984, **53**, 484–512.

Berger, M.J. and Colella, P. Local adaptive mesh refinement for shock hydrodynamics. *J. Comput. Phys.*, 1989a, **82**, 67–84.

Berger, M.J. and LeVeque, R.J. An adaptive Cartesian mesh algorithm for the Euler equations in arbitrary geometries. *AIAA Paper*, 1989b, 89–1930.

Berger, M.J. and Saltzman, J.S. AMR on the CM-2. *Appl. Numer. Math.*, 1994, **14**, 239–253.

Bramkamp, F., Ballmann, J. and Múller, 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, **1**, pp. 111–118, 2000 (CFD Society of Canada).

Brewster, M.Q. Solid propellant combustion response: Quasi-steady (QSHOD) theory development and validation. In Solid Propellant Chemistry, Combustion, and Motor Interior Ballistics, edited by V. Yang, T.B. Brill and W.-Z. Ren, pp. 607–638, 2000 (AIAA: Reston, VA), Volume 185 of *Progress in Astronautics and Aeronautics*.

Carrier, G., Fendell, F., Brent, D., Kimdrough, C., Loucks, S., Hess, E. and Acosta, P. Simple modeling of particle trajectories in solid rocket motors. *J. Propulsion Power*, 1991, **7**(2), 185–195.

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

Coirier, W.J. and Powell, K.G. An accuracy assessment of Cartesian-mesh approaches for the Euler equations. *J. Comput. Phys.*, 1995, **117**, 121–131.

Colella, P. and Woodward, P.R. The piecewise parabolic method (PPM) for gas-dynamical simulations. *J. Comput. Phys.*, 1984, **54**, 174–210.

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

Davis, R.L. and Dannenhoffer, J.F. Decomposition and parallelization strategies for adaptive gridembedding techniques. *Int. J. Comput. Fluid Dyn.*, 1993, **1**, 79–93.

De Zeeuw, D. and Powell, K.G. An adaptively refined Cartesian mesh solver for the Euler equations. *J. Comput. Phys.*, 1993, **104**, 56–68.

Dick, W.A. and Heath, M.T. Whole system simulation of solid propellant rockets. *AIAA Paper*, 2002, 02–4345.

Einfeldt, B. On Godunov-type methods for gas dynamics. *SIAM J. Numer. Anal.*, 1988, **25**, 294–318.

Godunov, S.K. Finite-difference method for numerical computations of discontinuous solutions of the equations of fluid dynamics. *Mat. Sb.*, 1959, **47**, 271–306.

Gottlieb, J.J. and Groth, C.P.T. Assessment of Riemann solvers for unsteady one-dimensional inviscid flows of perfect gases. *J. Comput. Phys.*, 1988, **78**, 437–458.

Gottlieb, J.J. and Groth, C.P.T. Collection of boundary conditions for one- and some multidimensional unsteady flows of polytropic gases. *CASI J.*, 1999, **45**(2), 161–182.

Greatrix, D.R. Combined structural oscillation effects on solid rocket internal ballistics. *AIAA Paper*, 1999, 99–2509.

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.*, 1987, **33**(3), 133–142.

Greatrix, D.R., Gottlieb, J.J. and Constantinou, T. Quasi-steady analysis of the internal ballistics of solid-propellant rocket motors. *Can. Aeronaut. Space J.*, 1987, **33**(2), 61–70.

Greatrix, D.R., Gottlieb, J.J. and Constantinou, T. Numerical model for pellet-dispersion igniter systems. *J. Propulsion Power*, 1988, **4**(5), 412–420.

Gropp, W., Lusk, E. and Skjellum, A., *Using MPI*, 1999 (MIT Press: Cambridge, MA).

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. *AIAA Paper*, 1999, 99–3273.

Groth, C.P.T., De Zeeuw, D.L., Gombosi, T.I. and Powell, K.G. Global three-dimensional MHD simulation of a space weather event: CME formation, interplanetary propagation, and and interaction with the magnetosphere. *J. Geophys. Res.*, 2000, **105**(A11), 25,053–25,078.

Harten, A. High resolution schemes for hyperbolic conservation laws. *J. Comput. Phys.*, 1983, **49**, 357–393.

Harten, A. On a class of high resolution total-variation-stable finite-difference schemes. *SIAM J. Numer. Anal.*, 1984, **21**, 1–23.

Harten, A., Enquist, B., Osher, S. and Chakravarthy, S.R. Uniformly high order accurate essentially non-oscillatory schemes, III. *J. Comput. Phys.*, 1987, **71**, 231–303.

Hartmann, R. and Houston, P. Adaptive discontinuous Galerkin finite element methods for nonlinear hyperbolic conservation laws. *SIAM J. Sci. Comput.*, 2002, **24**(3), 979–1004.

Hwang, C.J. and Chang, G.C. Numerical study of gas–particle flow in a solid rocket nozzle. *AIAA J.*, 1988, **26**(6), 682–689.

King, M.K. Erosive burning of composite solid propellants: experimental and modeling studies. *J. Spacecraft Rockets*, 1979, **16**(7), 154–162.

van Leer, B. Towards the ultimate conservative difference scheme. I. The quest of monotonicity. Lecture Notes in Physics, **18**, pp. 163–168, 1973 (Springer-Verlag: New York).

van Leer, B. Towards the ultimate conservative difference scheme. II. Monotonicity and conservation combined in a second order scheme. *J. Comput. Phys.*, 1974, **14**, 361–370.

van Leer, B. Towards the ultimate conservative difference scheme. III. Upstream-centered finite difference schemes for ideal compressible flow. *J. Comput. Phys.*, 1977a, **23**, 263–275.

van Leer, B. Towards the ultimate conservative difference scheme. IV. A new approach to numerical convection. *J. Comput. Phys.*, 1977b, **23**, 276–299.

van Leer, B. Towards the ultimate conservative difference scheme. V. A second-order sequel to Godunov's method. *J. Comput. Phys.*, 1979, **32**, 101–136.

van Leer, B. Flux-vector splitting for the Euler equations. Lecture Notes in Physics, **170**, p. 507, 1982 (Springer-Verlag: New York).

van Leer, B., Tai, C.H. and Powell, K.G. Design of optimally-smoothing multi-stage schemes for the Euler equations. *AIAA Paper*, 1989, 89–1933–CP.

Linde, T. A practical, general-purpose, two-state HLL Riemann solver for hyperbolic conservation laws. *Int. J. Numer. Meth. Fluids*, 2002, **40**, 391–402.

Marble, F.E. Dynamics of dusty gases. *Ann. Rev. Fluid Mech.*, 1970, **2**, 397–446.

Najjar, F.M., Balachandar, S., Alavilli, P.V.S. and Ferry, J. Computations of two-phase flow in aluminized solid propellant rockets. *AIAA Paper*, 2000, 00–3568.

Najjar, F.M., Ferry, J., Wasistho, B. and Balachandar, S. Full-physics large-scale multiphase large eddy simulations of flow inside solid rocket motors. *AIAA Paper*, 2002, 02–4343.

Najjar, F.M., Haselbacher, A., Ferry, J., Wasistho, B., Balachandar, S. and Moser, R.D. Large-scale multiphase large-eddy simulation of flows inside solid-rocket motors. *AIAA Paper*, 2003, 03–3700.

Orbekk, E. Internal flow analysis of a technology demonstrator rocket motor with new CFD code. *AIAA Paper*, 1998, 98–3967.

Osher, S. Riemann solvers, the entropy condition, and difference approximations. *SIAM J. Numer. Anal.*, 1984, **21**(2), 217–235.

Paillère, H., Powell, K.G. and De Zeeuw, D.L. A wave-model-based refinement criterion for adaptive-grid computation of compressible flows. *AIAA Paper*, 1992, 92–0322.

Peskin, C.S. The immersed boundary method. *Acta Numer.*, 2002, 1–39.

Powell, K.G., Roe, P.L. and Quirk, J. Adaptive-mesh algorithms for computational fluid dynamics. In Algorithmic Trends in Computational Fluid Dynmaics, edited by M.Y. Hussaini, A. Kumar and M.D. Salas, pp. 303–337, 1993 (Springer-Verlag: New York).

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.*, 1999, **154**, 284–309.

Quirk, J.J. An adaptive grid algorithm for computational shock hydrodynamics. PhD thesis, 1991, Cranfield Institute of Technology.

Quirk, J.J. and Hanebutte, U.R. A parallel adaptive mesh refinement algorithm, 1993, Report 93-63, ICASE.

Renie, J.P. and Osborn, J.R. Erosive burning. *AIAA J.*, 1983, **21**(12), 1681–1689.

Roe, P.L. Approximate Riemann solvers, parameter vectors, and difference schemes. *J. Comput. Phys.*, 1981, **43**, 357–372.

Roe, P.L. (1984) Generalized formulation of TVD Lax-Wendroff schemes, Report 84-53, ICASE.

Rudinger, G. Fundamentals of gas–particle flow, 1980 (Elsevier Scientific Publishing Company: Amsterdam).

Sabnis, J.S. Numerical simulations of distributed combustion in solid rocket motors with metalized propellant. *J. Propulsion Power*, 2003, **19**(1), 48–55.

Sachdev, J.S., Groth, C.P.T. and Gottlieb, J.J. Parallel solution-adaptive scheme for multi-phase core flows in rocket motors. *AIAA Paper*, 2003, 03–4106.

Sauerwein, H. and Fendell, F.E. Method of characteristics in two-phase flow. *Phys. Fluids*, 1965, **8**, 1564–1565 Research Note.

Saurel, R., Daniel, E. and Loraud, J.C. Two-phase flows: second-order schemes and boundary conditions. *AIAA J.*, 1994, **32**(6), 1214–1221.

Sethian, J.A. Level Set Methods and Fast Marching Methods, 2nd Ed., 1999 (Cambridge University Press: Cambridge Monographs on Applied and Computational Mathematics).

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

Sun, M. and Takayama, K. Conservative smoothing on an adaptive quadrilateral grid. *J. Comput. Phys.*, 1999, **150**, 143–180.

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

Venkatakrishnan, V. On the accuracy of limiters and convergence to steady state solutions. *AIAA Paper*, 1993, 93–0880.

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. *AIAA Paper*, 1997, 97–3326.

Yee, H.C. Construction of explicit and implicit symmetric TVD schemes and their applications. *J. Comput. Phys.*, 1987, **68**, 151–179.