A99-33571

AIAA 99-3273
A Parallel Solution-Adaptive Scheme for Ideal Magnetohydrodynamics

C. P. T. Groth, D. L. De Zeeuw, K. G. Powell,
T. I. Gombosi, and Q. F. Stout

University of Michigan
Ann Arbor, MI

14th Computational Fluid Dynamics Conference

June 28 – July 1, 1999/Norfolk, VA

AIAA-99-3273

A PARALLEL SOLUTION-ADAPTIVE SCHEME FOR IDEAL MAGNETOHYDRODYNAMICS

C. P. T. Groth[*] D. L. De Zeeuw[†] K. G. Powell[‡] T. I. Gombosi[§] and Q. F. Stout[¶]
University of Michigan, Ann Arbor, Michigan 48109, U.S.A.

Abstract

A parallel adaptive mesh refinement (AMR) scheme is described for solving the hyperbolic system of partial-differential equations governing ideal magnetohydrodynamic (MHD) flows in three space dimensions. This highly parallelized algorithm adopts a cell-centered upwind finite-volume discretization procedure and uses limited solution reconstruction, approximate Riemann solvers, and explicit multi-stage time stepping to solve the MHD equations in divergence form, providing a combination of high solution accuracy and computational robustness across a large range in the plasma β (β is the ratio of thermal and magnetic pressures). A flexible block-based hierarchical data structure is used to facilitate automatic solution adaption on Cartesian mesh using physics-based refinement criteria. In addition, the data structure naturally lends itself to domain decomposition, thereby enabling efficient and scalable implementations of the method on MIMD (multiple-instruction multiple-data) distributed-memory multi-processor architectures. The model has been developed on several massively parallel computer platforms and high parallel performance has been achieved (342 GFlops has been attained on a 1,490-processor Cray T3E-1200 with near-perfect scalability). Numerical results for MHD simulations of magnetospheric and heliospheric plasma flows are described to demonstrate the validity and capabilities of the approach for space physics applications.

I. Introduction

Gaining a better understanding of the often complicated physical processes associated with the flows of electrically conducting fluids and plasmas is of great interest and current importance both for fundamental scientific research and new engineering technology. Numerical modeling is playing an ever increasing role in the study of these flows; however, the solution of the governing nonlinear partial differential equations (PDEs) for such flows, even for the somewhat rudimentary mathematical representation given by the hyperbolic system of equations of ideal magnetohydrodynamics (MHD), is challenging and necessitates special consideration. Solutions of the compressible MHD equations are characterized by complicated nonlinear wave structure and admit strong shocks and contact discontinuities. For many space and astrophysical plasma applications, the solutions may also exhibit rapid transitions between a wide range of highly different solution regimes including transitions from low speed subsonic to hypersonic and hyper-Alfvènic regimes, from high-density high-temperature to low-density low-temperature regimes, and from strongly magnetized (low-β) to weakly magnetized (high-β) regimes, where β is the ratio of thermal and magnetic pressures. In addition, the ideal MHD equations exhibit solution degeneracies of a type that do not arise in conventional gas dynamics and, as they are normally written, have the added constraint of zero divergence of the magnetic field imposed by the condition that there are no observed magnetic monopoles. Numerical methods are needed that are capable of reliably and accurately resolving complex nonlinear wave structure, can cope with the discontinuous and degenerate nature of the solutions, and at the same time remain valid for a wide range of solution parameters.

Numerical simulations of conducting fluids and plasmas in three space dimensions also place heavy demands on available computational resources. The solution of problems with between ten and one hun-

[*] Assistant Research Scientist, Atmospheric, Oceanic and Space Sciences, Senior Member AIAA

[†] Assistant Research Scientist, Atmospheric, Oceanic and Space Sciences, Member AIAA

[‡] Associate Professor, Aerospace Engineering, Senior Member AIAA

[§] Professor, Atmospheric, Oceanic and Space Sciences

[¶] Professor, Electrical Engineering and Computer Science dred million degrees of freedom or more is typically required, and for time-dependent calculations, the solutions may require updating at between ten and one hundred thousand instances in time. For this reason, the ability to routinely perform such simulations virtually necessitates efficient and flexible algorithms that can harness the potential of current and future generations of massively parallel computer architectures.

In the present work, a parallel adaptive mesh refinement (AMR) scheme is described for solving the equations governing ideal MHD flows in three space dimensions. This highly parallelized algorithm adopts a cell-centered upwind finite-volume discretization procedure and uses limited solution reconstruction, approximate Riemann solvers, and explicit multi-stage time stepping to solve the MHD equations in divergence form, providing a combination of high solution accuracy and computational robustness across a large range in the plasma β. A flexible block-based hierarchical data structure is used to facilitate automatic solution adaption on Cartesian mesh using physics-based refinement criteria. In addition, the data structure naturally lends itself to domain decomposition, thereby enabling efficient and scalable implementations of the method on MIMD (multiple-instruction multiple-data) distributed-memory multi-processor architectures. This combination of a robust finite-volume discretization of the governing equations, block-based AMR, domain decomposition, and parallel implementation yields an efficient, reliable, and powerful numerical method for large-scale simulations of MHD flows on high-performance parallel machines.

A description of the parallel solution-adaptive scheme is given in the following sections. Included are a presentation of the particular form of the MHD equations which are solved, details of the spatial and temporal discretization procedures used, and descriptions of the AMR strategy and hierarchical block-based data structure for Cartesian mesh. The parallel implementation of the algorithm MIMD distributed-memory architectures is also discussed along with the resulting parallel performance. In particular, aspects of the algorithm and implementation that are important to achieving high-speed scalable performance are discussed. Finally, numerical results for MHD simulations of magnetospheric and heliospheric plasma flows are described to demonstrate the validity and capabilities of the approach for space physics applications.

II. Governing Equations

A. Symmetrizable Form of MHD Equations

The hyperbolic system of PDEs governing compressible ideal MHD flows can be written in non-dimensional weak conservation (divergence) form as

$$\frac{\partial \tilde{\mathbf{U}}}{\partial \tilde{t}} + \left(\tilde{\nabla} \cdot \tilde{\mathbf{F}}\right)^{\mathrm{T}} = \tilde{\mathbf{S}} + \tilde{\mathbf{Q}}, \qquad (1)$$

where the solution and source-like vectors $\tilde{\mathbf{U}}$ and $\tilde{\mathbf{S}}$ are given by

$$\tilde{\mathbf{U}} = \begin{bmatrix} \tilde{\rho} \\ \tilde{\rho}\tilde{\mathbf{u}} \\ \tilde{\mathbf{B}} \\ \tilde{E} \end{bmatrix} \qquad \tilde{\mathbf{S}} = -\begin{bmatrix} 0 \\ \tilde{\mathbf{B}} \\ \tilde{\mathbf{u}} \\ \tilde{\mathbf{u}} \cdot \tilde{\mathbf{B}} \end{bmatrix} \tilde{\nabla} \cdot \tilde{\mathbf{B}}, \qquad (2)$$

$\tilde{\mathbf{F}}$ is a flux tensor having the form

$$\tilde{\mathbf{F}} = \begin{bmatrix} \tilde{\rho}\tilde{\mathbf{u}} \\ \tilde{\rho}\tilde{\mathbf{u}}\tilde{\mathbf{u}} + \left(\tilde{p} + \frac{1}{2}\tilde{B}^2\right)\mathbf{I} - \tilde{\mathbf{B}}\tilde{\mathbf{B}} \\ \tilde{\mathbf{u}}\tilde{\mathbf{B}} - \tilde{\mathbf{B}}\tilde{\mathbf{u}} \\ \tilde{\mathbf{u}}\left(\tilde{E} + \tilde{p} + \frac{1}{2}\tilde{B}^2\right) - \left(\tilde{\mathbf{B}} \cdot \tilde{\mathbf{u}}\right)\tilde{\mathbf{B}} \end{bmatrix}^{\mathrm{T}}, \qquad (3)$$

the total plasma energy, $\tilde{E}$, is given by

$$\tilde{E} = \frac{1}{2}\tilde{\rho}\tilde{u}^2 + \frac{\tilde{p}}{\gamma - 1} + \frac{1}{2}\tilde{B}^2. \qquad (4)$$

In addition, the specific internal energy of the plasma is related to the pressure and density via the ideal-gas equation of state, $\tilde{e} = \tilde{p}/(\gamma - 1)\tilde{\rho}$ and, for ideal MHD, the current density, $\tilde{\mathbf{j}}$, follows from a reduced form of Ampère's law and is given by $\tilde{\mathbf{j}} = \tilde{\nabla} \times \tilde{\mathbf{B}}$. Represented in the equation set are five equations for the conservation of mass, momentum, and total plasma energy, as well as three equations describing the time evolution of the magnetic field as described by Faraday's Law. The non-dimensional variables $\tilde{\rho}$, $\tilde{\mathbf{u}}$, $\tilde{p}$, and $\tilde{\mathbf{B}}$ correspond to the non-dimensional density, velocity, pressure, and magnetic field, respectively, and are related to their dimensional counterparts by the following non-dimensionalization:

$$\begin{aligned} \tilde{\rho} &= \rho/\rho_\circ && (5) \\ \tilde{\mathbf{u}} &= \mathbf{u}/a_\circ && (6) \\ \tilde{p} &= p/\rho_\circ a_\circ^2 && (7) \\ \tilde{\mathbf{B}} &= \mathbf{B}/\sqrt{\mu_\circ \rho_\circ a_\circ^2} && (8) \end{aligned}$$

where $\rho_\circ$ and $a_\circ$ are the density and ion-acoustic wave speed of a suitable reference solution state, and

γ is the specific heat ratio of the plasma. In this formulation, the vector $\tilde{\mathbf{S}}$ contains terms arising in the MHD equations that cannot be expressed in divergence form and the column vector $\tilde{\mathbf{Q}}$ contains pure source terms associated with the modeling of any additional physical processes, such as gravitational acceleration.

The form of the ideal MHD equations given by Eqs. (1)–(4) is somewhat non-standard. The terms of column vector $\tilde{\mathbf{S}}$ are proportional to $\nabla \cdot \tilde{\mathbf{B}}$, and arise solely from expressing Faraday's Law in divergence form. It is more usual to make use of the solenoidal condition, $\nabla \cdot \tilde{\mathbf{B}} = 0$, to further simplify Eq. (1). The solenoidal condition amounts to an initial condition for the MHD equations and, physically, implies that there are no magnetic monopoles. However, imposing this constraint in finite-volume based numerical solutions of the MHD equations has proven to be very challenging.[1] Mathematically, dropping the source terms of $\tilde{\mathbf{S}}$ changes the character of the MHD equations. Godunov has shown that Eq. (1) with the additional constraint is not symmetrizable.[2,3] The equations possess a degenerate eigensystem, having only seven identifiable characteristic fields, and, furthermore, are not formally Galilean invariant. Godunov found that the MHD equations can only be rendered symmetrizable by adding the terms contained in $\tilde{\mathbf{S}}$. Equation (1) with the source terms of $\tilde{\mathbf{S}}$ has eight characteristic fields, satisfies an additional transport equation or balance law for the plasma entropy, and is Galilean invariant. Moreover, this form of the MHD equations allows the derivation of an eight-wave approximate Roe-type Riemann solver that can be used in the construction of an upwind finite-volume scheme.[4]

The approach taken here, as first advocated by Powell,[4] is to solve the governing PDEs in their symmetrizable form. This permits the construction of a finite-volume scheme, based on approximate Riemann solvers, that tightly couples the fluid-dynamics and magnetic field equations, is conservative, correctly represents the propagation speeds of solution disturbances, and thereby satisfies the appropriate Rankine-Hugoniot jump relations for discontinuities in the solution. Although the solenoidal condition is not enforced to machine accuracy, it can be shown that it is satisfied to the level of the truncation error of the solver without requiring the use of projection schemes and/or staggered mesh arrangements.[5,6] Furthermore, Tóth and Odstrčil[6] have found that the addition of the terms proportional to $\nabla \cdot \tilde{\mathbf{B}}$ improve results for multidimensional MHD calculations with several methods and, in fact, reduce errors in the computed parallel magnetic force.

B. Intrinsic and Deviatoric Magnetic Fields

For some problems in which a strong "intrinsic" magnetic field dominates much of the flow solution (e.g., the intrinsic dipolar field of earth strongly influences the magnetospheric magnetic field configuration arising from the interaction of the solar wind with the earth's magnetic field), improved solution accuracy can be gained by solving for the deviatoric magnetic field, $\tilde{\mathbf{B}}_1$, which is the measure of the deviation of the full magnetic field from the intrinsic field, $\tilde{\mathbf{B}}_\circ$ and is defined as follows:

$$\tilde{\mathbf{B}}_1 = \tilde{\mathbf{B}} - \tilde{\mathbf{B}}_\circ . \tag{9}$$

This approach was first employed by Tanaka[7] and can lead to improved numerical solutions by alleviating the necessity of resolving the often large spatial gradients associated with the intrinsic fields and by ensuring that the divergence of the intrinsic component of the magnetic field is by definition zero. Assuming that the intrinsic magnetic field is both current free and time invariant such that

$$\frac{\partial \tilde{\mathbf{B}}_\circ}{\partial \tilde{t}} = 0 \ , \quad \tilde{\nabla} \cdot \tilde{\mathbf{B}}_\circ = 0 \ , \quad \tilde{\nabla} \times \tilde{\mathbf{B}}_\circ = 0 \, , \tag{10}$$

then the symmetrizable form of the ideal MHD equations can be rewritten as

$$\frac{\partial \tilde{\mathbf{U}}_1}{\partial \tilde{t}} + \left(\tilde{\nabla} \cdot \tilde{\mathbf{F}}_1\right)^{\mathrm{T}} + \left(\tilde{\nabla} \cdot \tilde{\mathbf{G}}\right)^{\mathrm{T}} = \tilde{\mathbf{S}}_1 + \tilde{\mathbf{Q}}_1 \tag{11}$$

where

$$\tilde{\mathbf{U}}_1 = \begin{bmatrix} \tilde{\rho} \\ \tilde{\rho}\tilde{\mathbf{u}} \\ \tilde{\mathbf{B}}_1 \\ \tilde{E}_1 \end{bmatrix} \qquad \tilde{\mathbf{S}}_1 = -\begin{bmatrix} 0 \\ \tilde{\mathbf{B}} \\ \tilde{\mathbf{u}} \\ \tilde{\mathbf{u}} \cdot \tilde{\mathbf{B}}_1 \end{bmatrix} \tilde{\nabla} \cdot \tilde{\mathbf{B}}_1 \, , \tag{12}$$

$$\tilde{\mathbf{F}}_1 = \begin{bmatrix} \tilde{\rho}\tilde{\mathbf{u}} \\ \tilde{\rho}\tilde{\mathbf{u}}\tilde{\mathbf{u}} + \left(\tilde{p} + \frac{1}{2}\tilde{B}_1^2\right)\mathbf{I} - \tilde{\mathbf{B}}_1\tilde{\mathbf{B}}_1 \\ \tilde{\mathbf{u}}\tilde{\mathbf{B}}_1 - \tilde{\mathbf{B}}_1\tilde{\mathbf{u}} \\ \tilde{\mathbf{u}}\left(\tilde{E}_1 + \tilde{p} + \frac{1}{2}\tilde{B}_1^2\right) - \left(\tilde{\mathbf{B}}_1 \cdot \tilde{\mathbf{u}}\right)\tilde{\mathbf{B}}_1 \end{bmatrix}^{\mathrm{T}} , \tag{13}$$

$$\tilde{\mathbf{G}} = \begin{bmatrix} 0 \\ \tilde{\mathbf{B}}_\circ \cdot \tilde{\mathbf{B}}_1\mathbf{I} - \left(\tilde{\mathbf{B}}_\circ\tilde{\mathbf{B}}_1 + \tilde{\mathbf{B}}_1\tilde{\mathbf{B}}_\circ\right) \\ \tilde{\mathbf{u}}\tilde{\mathbf{B}}_o - \tilde{\mathbf{B}}_\circ\tilde{\mathbf{u}} \\ \left(\tilde{\mathbf{B}}_\circ \cdot \tilde{\mathbf{B}}_1\right)\tilde{\mathbf{u}} - \left(\tilde{\mathbf{u}} \cdot \tilde{\mathbf{B}}_1\right)\tilde{\mathbf{B}}_\circ \end{bmatrix}^{\mathrm{T}} , \tag{14}$$

and where the total plasma energy in terms of the deviatoric magnetic field, $\tilde{E}_1$, is given by

$$\tilde{E}_1 = \frac{1}{2}\tilde{\rho}\tilde{u}^2 + \frac{\tilde{p}}{\gamma - 1} + \frac{1}{2}\tilde{B}_1^2 . \tag{15}$$

Note that Eqs. (11)–(15) have been derived under the assumption that $\partial \mathbf{B}_\circ/\partial \tilde{t} = 0$ and $\tilde{\nabla} \times \tilde{\mathbf{B}}_\circ = 0$. It is possible to derive a more general formulation of the equations that accounts for time variations in the intrinsic magnetic field and current systems associated with the intrinsic field.

III. Elements of the Parallel Solution-Adaptive Scheme

As discussed in the introduction, the numerical solution of hyperbolic systems of PDEs of the type given by Eqs. (11)–(15) is challenging. The algorithm described herein has been formulated to capitalize on recent developments in three areas of scientific computing. They are: 1) advances in upwind finite-volume methods for hyperbolic conservation laws; 2) significant advances in solution-adaptive techniques; and 3) advances in parallel computer design. A reliable, relatively efficient, and scalable parallel AMR finite-volume scheme for solving the ideal MHD equations has resulted.

A. Finite-Volume Scheme

Higher-order variants of the upwind finite-volume scheme originally proposed by Godunov[8] have proven effective in the solution of hyperbolic systems of equations. Robust and low dissipation shock-capturing schemes, providing reliable high-resolution of complicated wave structure and non-oscillatory solutions near discontinuities, have been devised for conventional gas dynamics.[9–22] The studies of Brio and Wu[23] and Zachary and Collela[24] represent the first application of Godunov-type techniques to the ideal MHD equations. Schemes based on Roe-type approximate Riemann solvers[14] were developed for the one-dimensional form of the ideal MHD equations in both studies. In subsequent work, Roe and Balsara[25] proposed a refinement to the eigenvector normalizations of the approximate Riemann solvers developed in this previous work and Tanaka[7] developed a Roe-type scheme for the three-dimensional MHD equations. Dai and Woodward[26,27] developed a nonlinear approximate Riemann solver for MHD and used it to formulate a higher-order Godunov-type scheme for multidimensional problems. Other flux function formulations have also been proposed. Croisille et al.[28] have developed a kinetic-based scheme and Linde[29] has developed a modified version of the HLLE method of Harten, Lax, Van Leer, and Einfeldt[30,31] for the MHD equations.

In the present work, an explicit higher-order Godunov-type method is used to solve Eq. (11). In this finite-volume approach, the governing equations are integrated over computational cells, i, yielding

$$\begin{aligned} \frac{d\tilde{\mathbf{U}}_{1_i}}{d\tilde{t}} &= -\frac{1}{V_i}\sum_{faces}\left(\tilde{\mathbf{F}}_1+\tilde{\mathbf{G}}\right)\cdot\mathbf{n}A \\ &\quad -\frac{\tilde{\mathbf{s}}_i}{V_i}\sum_{faces}\tilde{\mathbf{B}}_1\cdot\mathbf{n}A+\tilde{\mathbf{Q}}_{1_i}\,, \qquad (16)\end{aligned}$$

where V_i is the volume of cell i, A is the surface area of the faces forming the computational cell, n is the unit vector normal to the cell faces, $\tilde{\mathbf{U}}_{1_i}$ is the cell-averaged conserved solution vector, and $\tilde{\mathbf{s}}_i$ is given by

$$\tilde{\mathbf{s}}_i = -\begin{bmatrix} 0 \\ \tilde{\mathbf{B}}_{\circ_i}+\tilde{\mathbf{B}}_{1_i} \\ \tilde{\mathbf{u}}_i \\ \tilde{\mathbf{u}}_i\cdot\tilde{\mathbf{B}}_{1_i}\end{bmatrix}. \qquad (17)$$

The numerical face fluxes, $\left(\tilde{\mathbf{F}}_1+\tilde{\mathbf{G}}\right)\cdot\mathbf{n}$, are defined in terms of the left and right interface solution states, $\mathbf{U}_{1_\mathrm{L}}$ and $\mathbf{U}_{1_\mathrm{R}}$, as follows

$$\left(\tilde{\mathbf{F}}_1+\tilde{\mathbf{G}}\right)\cdot\mathbf{n} = \mathcal{F}\left(\mathbf{U}_{1_\mathrm{L}},\mathbf{U}_{1_\mathrm{R}},\mathbf{n}\right)\,, \qquad (18)$$

where $\mathbf{U}_{1_\mathrm{L}}$ and $\mathbf{U}_{1_\mathrm{R}}$ are determined using piece-wise linear solution reconstruction applied to each computational cell. The least-squares limited linear reconstruction procedure developed by Barth[32] is used here. Given the left and right interface states, the flux function, $\mathcal{F}$, is then evaluated by using an approximate Riemann solver. In the present work, the Roe-type linearized Riemann solver for MHD put forward by Powell[4] and the HLLE-type flux function due to Linde[29] have both been implemented and may be used. Note that the use of limited linear reconstruction and approximate Riemann solvers leads to a scheme that is robust, minimizes numerical discretization errors, and provides accurate resolution of discontinuities and shocks. Further details of this spatial discretization procedure and the approximate Riemann solvers used here are given elsewhere.[4,5,29,33]

For steady-state (time-invariant) calculations, a time-marching method-of-lines approach is adopted and the system of ordinary differential equations resulting from the application of the spatial discretization procedure defined above are integrated in time using the optimally-smoothing multi-stage schemes developed by van Leer et al.[34] The general M-stage scheme for integrating Eq. (16) from time level n to time level $n+1$ can be written as

$$\begin{aligned} \tilde{\mathbf{U}}^{(0)}_{1_i} &= \tilde{\mathbf{U}}^{n}_{1_i} \\ \tilde{\mathbf{U}}^{(m)}_{1_i} &= \tilde{\mathbf{U}}^{(0)}_{1_i} + \beta_m \Delta\tilde{t}^n \mathbf{R}\left(\tilde{\mathbf{U}}^{(m-1)}_{1_i}\right) \quad m = 1 \ldots M \\ \tilde{\mathbf{U}}^{(n+1)}_{1_i} &= \tilde{\mathbf{U}}^{(M)}_{1_i} \end{aligned}$$

where the residual $\mathbf{R}$ is equal to the right-hand-side of Eq. (16) and $\Delta\tilde{t}^n = \tilde{t}^{n+1} - \tilde{t}^n$ is the size of the time step. The multi-stage coefficients β_m and associated time-step constraints are given by van Leer et al.[34] One advantage of this integration procedure is its low storage requirements. Local time stepping is used to enhance the convergence of the scheme to the steady state solution. For time-accurate calculations, a simple two-stage second-order Runge-Kutta time integration procedure is used to solve Eq. (16).

B. Block-Based AMR on Cartesian Grids

Adaptive mesh refinement techniques which automatically adapt the computational grid to the solution of the governing PDEs can be very effective in treating problems with disparate length scales. Methods of this type avoid under-resolving the solution in regions deemed of interest (e.g., high-gradient regions) and, conversely, avoid over-resolving the solution in other less interesting regions (low-gradient regions) and thereby saving orders of magnitude in computing resources for many problems. For typical space plasma flows, length scales can range from tens of kilometers in the near Earth region to the Earth-Sun distance (1.5×10^{11} m) and time scales can range from a few seconds near the Sun to the expansion time of the solar wind from the Sun to the Earth ($\sim 10^5$ s). The use of AMR is extremely beneficial and almost a virtual necessity for solving problems with such disparate spatial and temporal scales.

Borrowing from previous work by Berger and coworkers,[35-39] Quirk,[40,41] and De Zeeuw and Powell[42] and keeping in mind the desire for high performance on massively parallel computer architectures, a relatively simple yet effective block-based AMR technique has been developed and is used in conjunction with the finite-volume scheme described above. The method has some similarities with the block-based approaches described by Quirk and Hanebutte[41] and Berger and Saltzman.[39] Here, the governing equations are integrated to obtain volume-averaged solution quantities within rectangular Cartesian computational cells. A representative cell is depicted in the schematic diagram of Fig. 1. The computational cells are embedded in regular structured blocks of equal sized cells. The blocks are geometrically self-similar with dimensions $\tilde{\ell}_x \times \tilde{\ell}_y \times \tilde{\ell}_z$ and consist of $N_x \times N_y \times N_z$ cells, where $\tilde{\ell}_x$, $\tilde{\ell}_y$, and $\tilde{\ell}_z$ are the non-dimensional lengths of the sides of the rectangular blocks and N_x, N_y, and N_z are even, but not necessarily all equal, integers. Typically, blocks consisting of anywhere between $4 \times 4 \times 4 = 64$ and $12 \times 12 \times 12 = 1,728$ cells are used. Refer to Fig. 2. Solution data associated with each block are stored in standard indexed array data structures. It is therefore straightforward to obtain solution information from neighboring cells within a block.

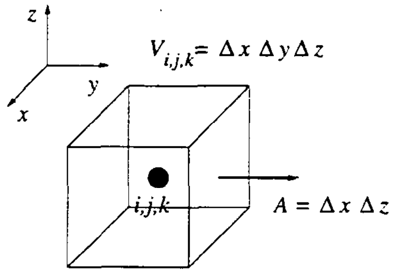
z
x
y
$V_{i,j,k} = \Delta x \, \Delta y \Delta z$
i,j,k
$A = \Delta x \, \Delta z$

Figure 1: Cartesian computational cell used in parallel block-based AMR scheme.

Computational grids are then composed of many self-similar blocks. Although each block within a grid has the same data storage requirements, blocks may be of different sizes in terms volume of the physical space that they occupy. Starting with an initial mesh consisting of blocks of equal size (i.e., equal resolution), adaption 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 eight "children" or "offspring". Each of the eight octants 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 ($\Delta\tilde{x} = \tilde{\ell}_x/N_x$, $\Delta\tilde{y} = \tilde{\ell}_y/N_y$ and $\Delta\tilde{z} = \tilde{\ell}_z/N_z$ for the newly created block are each halved from their value on the parent block). Conversely, in regions that are deemed over-resolved, the refinement process is reversed and eight children are coarsened and coalesced into a single parent block. In this way, the cell resolution is reduced by a factor of two. Standard multigrid-type restriction and prolongation operators are used to evaluate the solution on all blocks created by the coarsening and division processes, respectively.

Two neighboring blocks, one of which has been refined and one of which has not, are shown in Fig. 2. Any of the blocks shown in the figure can in turn be refined, and so on, leading to successively finer

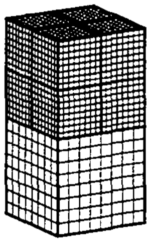

Figure 2: Self-similar blocks used in parallel block-based AMR scheme.

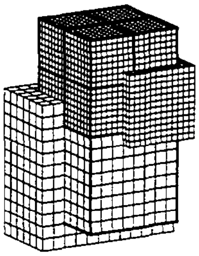

Figure 3: Self-similar blocks illustrating the double layer of ghost cells for both coarse and fine blocks.

blocks. In the present method, mesh refinement is constrained such that the cell resolution changes by only a factor of two between adjacent blocks and such that the minimum resolution is not less than that of the initial mesh.

In order that the update scheme for a given iteration or time step can be applied directly to all blocks in an independent manner, some additional 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 as shown in Fig. 3. 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. After each stage of the multi-stage time-stepping algorithm, ghost cell values are re-evaluated to reflect the updated solution values of neighboring blocks. With 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.[35-37] In particular, the interface fluxes computed on more refined blocks are used to correct the interface fluxes computed on coarser neighboring blocks so as to ensure the fluxes are conserved across block interfaces.

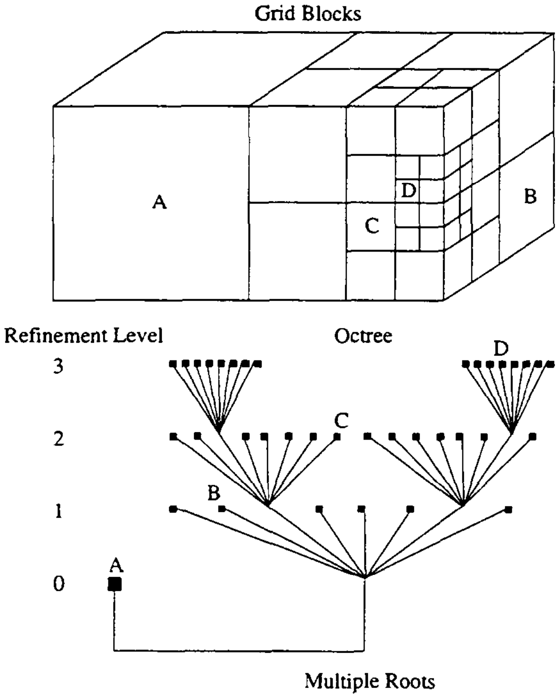

Figure 4: Solution blocks of a computational mesh with three refinement levels originating from two initial blocks and the associated hierarchical multi-root octree data structure. Interconnects to neighbors are not shown.

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 Fig. 4. 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 "octree" data structure that contains all of the blocks making up the leaves of the tree which were created from the original parent blocks during mesh refinement. Each grid block corresponds to a node of the tree. Traversal of the multi-tree structure by recursively visiting the parents and children of solution blocks can be used to determine block connectively. However, in order to reduce overhead associated with accessing solution information from adjacent blocks, the neighbors of each block are computed and stored directly, providing interconnects between blocks in the hierarchical data structure that are neighbors in physical space.

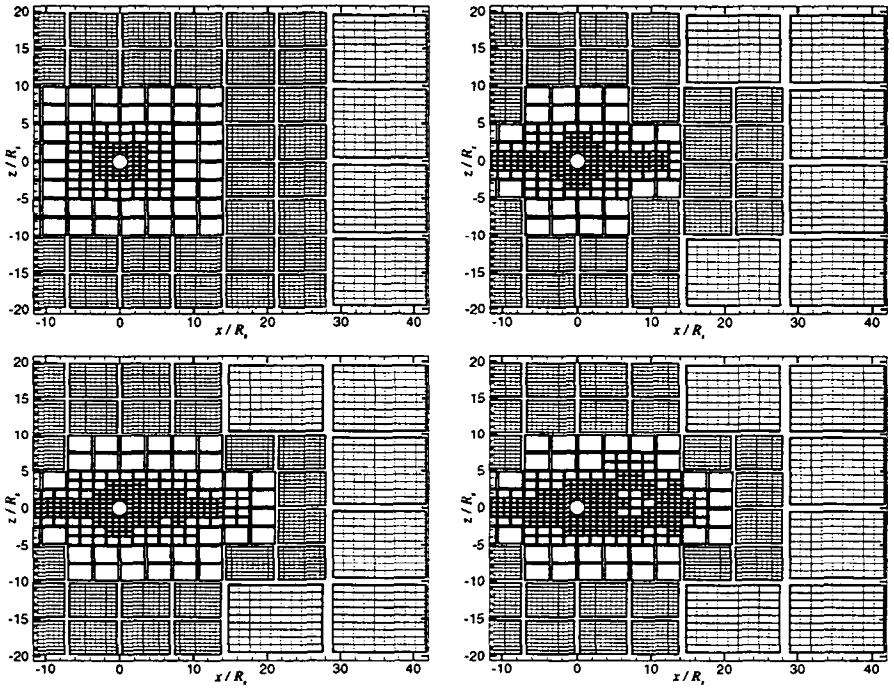

Figure 5: Evolution of a computational mesh illustrating grid adaptation in response to changes in the numerical solution. Cross sectional cuts through a 3D grid are shown for a solar wind calculation at four different instances in time. The computational cells are not shown for the smaller blocks.

One of the advantages of the preceding hierarchical data structure is that it is relatively easy to carry out local mesh refinement at anytime during a calculation. If, at some point in a computation, a particular region of the flow is deemed to be sufficiently interesting, better resolution of that region can be attained by refining the solution blocks in that region, without affecting the grid structure in other regions of the flow. Reducing the grid resolution in a region is equally easy. There is no need for completely re-meshing the entire grid and recalculating block connectivity every time a mesh refinement is performed. Although other approaches are possible, for this study, the coarsening and division of blocks are directed using multiple physics-based refinement criteria.[5,43,44] In particular, decisions as to when to refine or coarsen blocks are made based on comparisons of the maximum values of various local flow quantities determined in each block to specified refinement threshold values. Three flow quantities or refinement criteria, ϵ_k, are used herein. They have the forms

$$\epsilon_1 \propto \left|\bar{\nabla}\cdot\tilde{\mathbf{u}}\right| , \quad \epsilon_2 \propto \left|\bar{\nabla}\times\tilde{\mathbf{u}}\right| , \quad \epsilon_3 \propto \left|\bar{\nabla}\times\tilde{\mathbf{B}}\right| . \quad (19)$$

These quantities represent local measures of the compressibility and rotationality of the plasma as well as the stretching of the magnetic field. They have proven to be quite effective in detecting solution features such as shocks, velocity shears, and current systems in the plasma flow and directing the mesh adaption to more accurately resolve such features. Note that the refinement thresholds are dynamically adjusted so as to exercise some control over the total numbers of blocks, and hence cells, used in a calculation.

An example illustrating the adaptation of the block-based Cartesian mesh to an evolving solution is shown in Fig. 5. The figure shows the grid at four different instances in time for an unsteady calculation and depicts both the solution blocks (thick lines) and computational cells (thin lines) of the evolving grid. As noted above, each level of refinement in the grid introduces cells that are smaller by a factor two in each dimension from those one level higher in the grid. Typically, calculations may have 10-15 levels of refinement; some calculations may have more than 20 levels of refinement. In the case of 20 levels of refinement, the finest cells on the mesh are more than one million times (2^{20}) smaller in each dimension than the coarsest cells. The block-based AMR approach described above has many similarities to the cell-based method proposed by De Zeeuw and Powell.[42] Although the block-based approach is somewhat less flexible and incurs some inefficiencies in solution resolution as compared to a cell-based approach (i.e., for the same solution accuracy, generally more computational cells are introduced into the adapted grid), as will now be shown, the block-based method offers many advantages over a cell-based technique when parallel implementations of the algorithms are considered and performance issues are taken into account.

C. Parallel Implementation

The current and future generations of massively parallel distributed-memory computers offer the potential of large increases in processing power and memory resources beyond those of single-processor machines. Capitalizing on the promise of these resources is, however, not always easily achieved. In many instances, solution algorithm speedup is achieved for small numbers of processors (1–32 processors); however, with added processors, not only does the method fail to scale as expected, but the performance of the algorithm may actually diminish with an increase in the number of processors.

The block-based AMR finite-volume scheme for MHD described in the preceding subsections has been designed with a view to achieving very high performance on massively parallel architectures. In particular, there are several important design features of the method that have enabled high parallel performance. They are as follows:

- For problems involving the numerical solution of PDEs, domain decomposition (i.e., the partitioning of the problem by dividing the computational domain into subdomains, and farming the subdomains off onto separate processors) is a natural and practical approach to parallelization. The hierarchical block-based data structure and self-similar nature of the solution blocks make domain decomposition of the problem almost trivial and readily enable good load balancing, a crucial element for truly scalable computing. Furthermore, the local nature of the mesh refinement process means that mesh adaptation can be performed routinely without re-mapping all of the subdomains to the processors, which would significantly increase interprocessor communication and reduce computational performance.

- The underlying upwind finite-volume solution algorithm, with explicit time stepping, has a very compact stencil and is therefore highly local in nature. This results in lower interprocessor communication requirements. For the block-based grid structure, update of the solution within the subdomains on each processor can proceed almost independently and communication is limited to block interfaces and mainly involves the exchange of ghost-cell solution values and conservative flux corrections. The compact stencil and block data structure also result in high data locality and therefore permits the more efficient use of processor memory and cache. Poor usage of processor cache can often severely limit the serial (single-processor) performance, and hence parallel performance, that may ultimately be achieved.

- The self-similar nature of the solution blocks also means that serial performance enhancements apply to all blocks and that fine grain parallelization of the algorithm is possible.

A parallel implementation of the block-based AMR scheme has been developed using the FORTRAN 90 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 serial and parallel performance. Domain decomposition is accomplished 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 neighboring blocks on the same processors. With 10 blocks per processor, the load imbalance attained by this simple block distribution procedure is less than 10% and, with 100 blocks per node, the load imbalance becomes less than 1%. For heterogeneous parallel machines, such as a network of workstations, 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 interblock communication, the complete hierarchical octree data structure is stored on every processor. This is possible because, unlike cell-based unstructured meshing techniques, the block-based tree data struc-

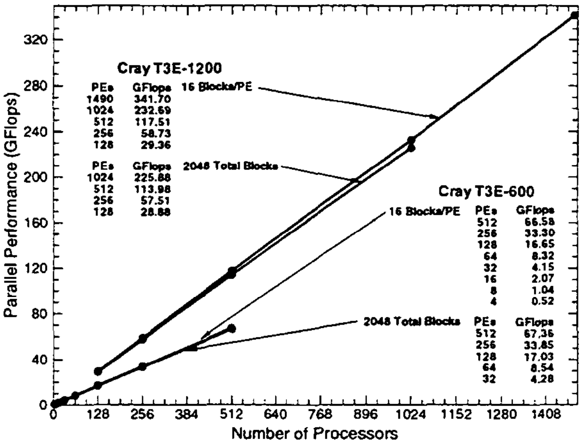

Figure 6: Parallel performance of the block-based AMR scheme for MHD on the Cray T3E-600 and T3E-1200 computers for both scale-up and speed-up problems.

ture is not unreasonably large. The octree data structure must only keep track of connectivity between the blocks and the number of solution blocks is clearly much less than the total number of cells in any given mesh. As mentioned, inter-processor communication is mainly restricted to block interfaces and primarily involves the exchange of ghost-cell solution values and conservative flux corrections. This communication of interface solution information is required at every stage of the multi-stage solution update 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 only amounts to less than 3-5% of the processor time in most cases.

Considerable fine grain parallelization of the algorithm has been carried out. In fact, the parallel implementation is such that even much of the grid adaptation is performed in parallel. Due to the self-similar nature of the solution blocks, enhancements to the serial performance of the algorithm also proved very effective in increasing the overall parallel performance of the method. The following strategies were particularly useful and when combined provided a more than two-fold increase in the single-processor performance:

- The use of indirect addressing and allocatable arrays was avoided when defining memory for primary solution variables. In addition, padding of arrays was used to align the primary solution variables in memory.

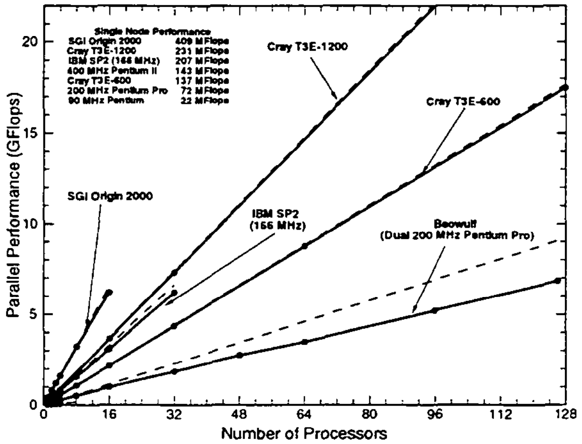

Figure 7: Parallel performance of the block-based AMR scheme for a variety of parallel architectures. The dashed line indicates ideal scale-up performance based on single node performance and solid lines indicate actual performance achieved on each of the machines for a scale-up problem with 8 blocks per processor.

- Inner loop optimization of the computationally intensive routines associated with the limited reconstruction procedure, flux function evaluation, and solution update was carried out. Standard loop unrolling and code inlining techniques were used. All decisions related to the evaluation of loop counters were removed. Furthermore, strip mining of computationally intensive loops using additional local array variables led to more efficient use of cache (i.e., high cache reuse is desirable) and considerable increases in performance.

Implementation of the algorithm has been carried out on SGI and Linux workstations, SGI shared-memory Origin 200 and Origin 2000 machines, a Cray T3D, both Cray T3E-600 and T3E-1200 parallel computers, several IBM SP2 machines, and a few Beowulf clusters. The parallel performance and scalability of the method for several of these architectures are shown in Figs. 6 and 7. Performance results for both Cray T3E-600 and T3E-1200 parallel computers are depicted in Fig. 6. The figure shows two curves for each of these machines: in the first, the performance measured in terms of the number of floating point operations performed per second is shown for a problem that has a fixed number of blocks per processor (the scale-up problem with 16 blocks per processor); in the second, the performance is shown for a problem that has a fixed size of 2,024 blocks (the speed-up problem). Of the two, the second case is more challenging. This is because as the speed-up problem is distributed across more and more processors, the ratio of communication overhead to computing cost increases. As can be seen from the figure, for both scale-up and speed-up problems, the parallel performance of the block-based AMR scheme is linear and nearly 100% efficient for up to 1,024 processors on both the Cray T3E-600 and T3E-1200. Furthermore, 342 GFlops has been attained on the Cray T3E-1200 for the scale-up problem using 1,490 processors. In Fig. 7, the parallel performance obtained on several other architectures is shown and compared to the Cray performance for another scale-up problem with 8 blocks per processor. Although there is generally a higher latency associated with message passing for these other machines, this second set of performance results demonstrate that the block-based algorithm method is portable to a wide range of machines and that reasonable scalability can be achieved, even for high-latency architectures such as a Beowulf cluster.

IV. Numerical Results

Simulation results are now described for several challenging problems in space plasma physics in order to demonstrate the capabilities of the present algorithm. Results are given for the terrestrial magnetosphere, the background solar wind, and a transient solar wind disturbance.

A. Simulation of the Magnetosphere

The intrinsic magnetic field of Earth presents an obstacle to the on-coming solar wind. The interaction of the solar wind with the Earth's magnetic field leads to the formation of a bow shock wave just upstream of the Earth. The solar wind is slowed, heated and deflected around the magnetic obstacle by hydrodynamic and electromagnetic forces. In addition, the terrestrial magnetic field is rather flexible and is distorted by the impinging solar wind. It is compressed on the sunward side and is stretched into a long tail behind the Earth on the night side. The Earth's magnetic field forms an enclosed cavity in which solar wind particles are prevented, for the most part, from entering. This cavity is called the magnetosphere.

Figures 8 and 9 show the results of a simulation of the interaction of the magnetosphere with the solar wind using the block-based AMR algorithm. In this calculation Earth is represented by a non-tilted non-rotating magnetic dipole field with a strength of 0.3 G at the equator. Inner boundary conditions for the magnetosphere are imposed at 3 R_e and a height-integrated ionosphere model is solved at 1.017 R_e which provides coupling between the ionosphere and magnetosphere[†]. The ionospheric conductances were taken to be constant with values of 5, 0, 5000 S for the Petersen, Hall, and field-aligned conductances, respectively. The solar wind is represented by a uniform solution state with a speed of 400 km/s (the on-coming solar wind flow is taken to be aligned with the x-axis of the Cartesian coordinate system such that $u_x = -400$ km/s and $u_y = u_z = 0$), an ion density of 5 cm^{-3}, and an ion-acoustic Mach number of 8. The interplanetary magnetic field (IMF) is assumed to have a strength of 10 nT and a southward orientation such that $B_z = -10$ nT and $B_x = B_y = 0$. For this calculation, the computational domain is a rectangular box that extends from $x = 192$ R_e to $x = -384$ R_e along the Sun-Earth direction and from -192 R_e to 192 R_e in the y, and z directions and 15,276 blocks, 977,664 cells, and 11 levels of refinement were used.

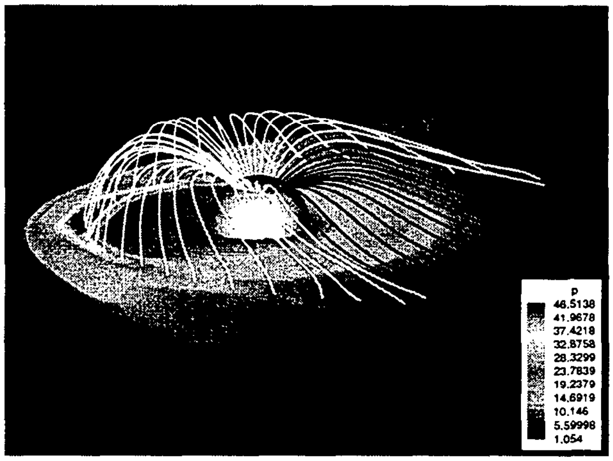

Figure 8: Simulation of the Earth's magnetosphere for southward IMF showing the computed plasma pressure distribution and last closed field lines.

Figure 8 depicts the three-dimensional configura-

[†] For magnetosphere simulations, the coupling between the magnetosphere and ionosphere must be taken into account. In the present model, a height-integrated electrostatic model of the ionosphere is used, in which closure of the field-aligned currents arising from the MHD solution is taken into account at the ionospheric boundary by applying the principle of current conservation and making use of Ohm's Law applied to a thin spherical shell.[45,46] The solution of an elliptic equation for the ionospheric electric potential on this spherical shell provides boundary conditions for plasma velocity of MHD solution at the magnetosphere-ionosphere interface.

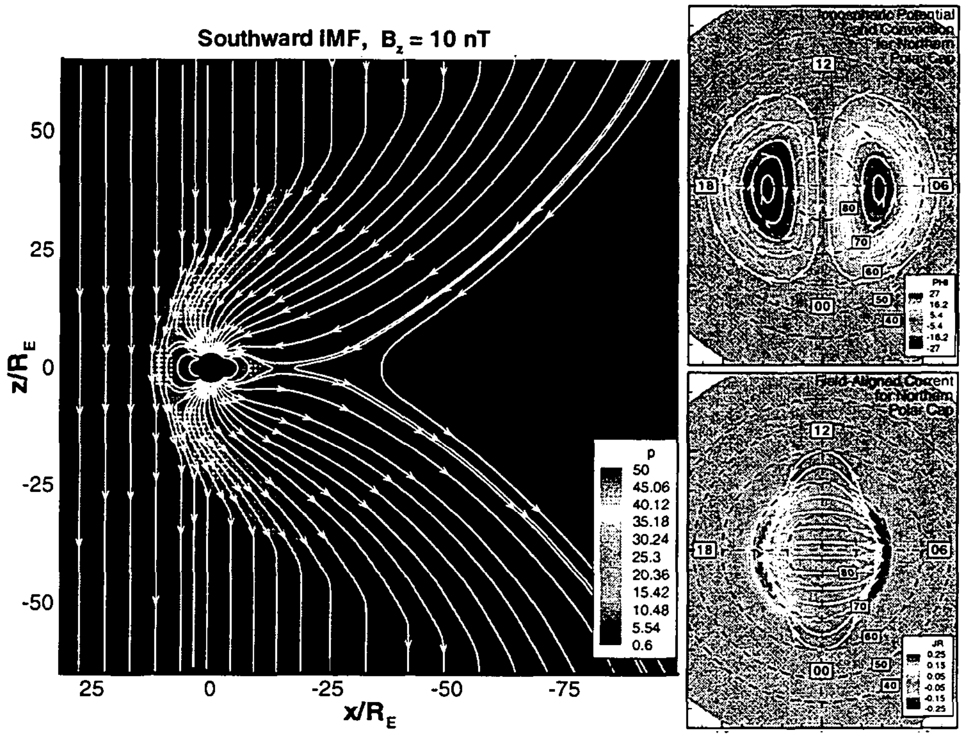

Figure 9: Simulation of the Earth's magnetosphere for southward IMF showing the plasma density distribution and open and closed field lines in the noon-midnight meridional plane and the ionospheric solution

tion of the computed magnetosphere looking from the side and down on the equatorial plane. The shading indicates variations in the plasma pressure (the region of higher pressure on the dayside defines the edge of the bow shock and magnetosheath) and the last closed magnetic field lines are also shown. The sphere represents the inner boundary of the magnetosphere calculation at 3 R$_e$. The last closed field lines effectively define the size of the magnetosphere. The white field lines close at the magnetopause and define the inner edge of the magnetosheath. The darker shaded field lines close through the tail and define the location of neutral line located at about 15-17 R$_e$ on the night side. A meridional cut through the solution in the $y = 0$ (noon-midnight) plane showing the plasma density distribution (color code) is given in Figure 9. Closed and open field lines are shown in white and the black lines define actual computational cells used in the computation. The locations of the bow shock, magnetopause, and neutral line are all quite evident. Notice the adaptation of the grid to the discontinuous features of the solution. The computed ionospheric solution for the northern hemisphere is also given in the figure, showing the field-aligned and ionospheric current and ionospheric potential and convection. A classic two cell convection pattern is obtained in this case.

The preceding magnetosphere simulation is actually a very challenging computation as the plasma β varies by several orders of magnitudes within the solution domain. Near the inner boundary, the terrestrial dipole field dominates and β is very small, while in the free streaming solar wind β is near unity. In addition, the Alfvén speed is very large in the vicinity of Earth making the equations very stiff. The robustness of the solver makes it possible to handle plasma flow problems of this type with relative ease.

B. Simulation of the Background Solar Wind

The corona is the outermost region of the solar atmosphere and is composed almost entirely of high temperature, low density, quasi-neutral, proton-electron plasma. The extension of the corona plasma to large heliocentric distances is what forms the solar wind and carries the charged coronal particles and magnetic field emanating from the Sun out into interplanetary space at supersonic speeds. Observations have shown that the solar wind is in many respects a two state phenomenon.[47,48] High speed wind originates from coronal holes generally located

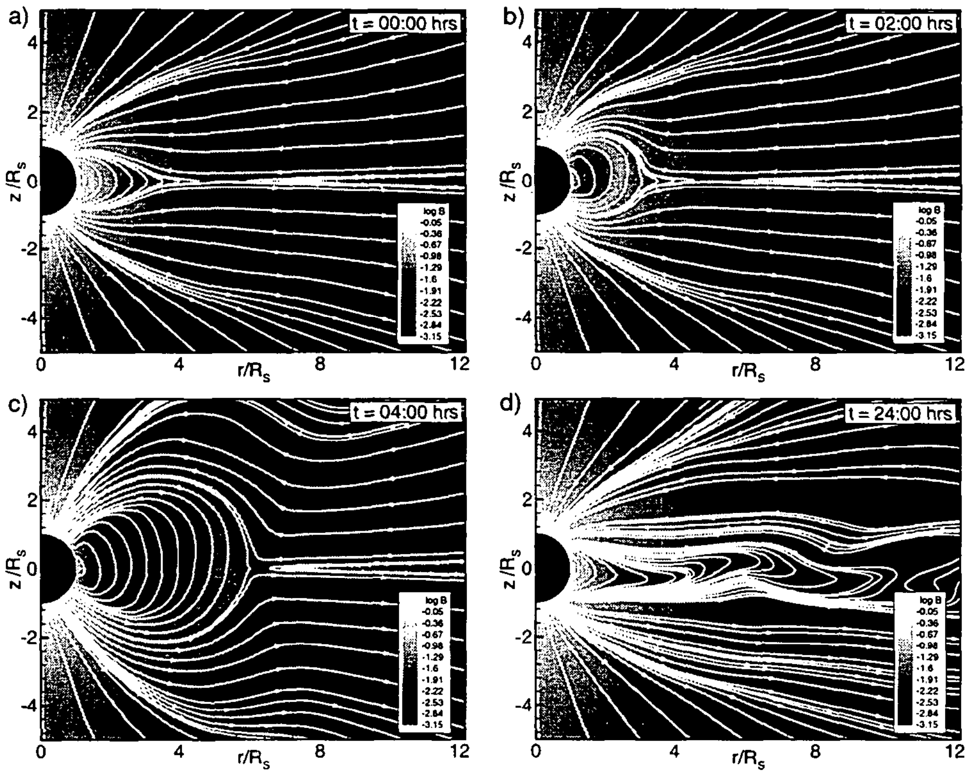

Figure 10: Four meridional snapshots of the computed CME solution depicting the initiation and evolution of the disturbance from $t = 0$ hrs to $t = 24$ hrs.

at high latitudes in the polar regions of the solar surface and the so-called slow wind originates from lower latitudes. A numerical solution of the nominal or background "steady" solar wind representative of conditions during solar minimum (i.e., for the quiet Sun) has also been developed using the parallel AMR MHD solution algorithm described herein. While the solution represents an oversimplified magnetic field configuration and is only relevant for solar minimum conditions, it represents an initial step towards developing more sophisticated models for the solar wind and is useful for the study of transient solar wind disturbances.

The steady-state solar wind from 1 R$_s$ to nearly 1/2 AU is modeled here by assuming that, at 1 R$_s$, the inner solar corona is a large rotating reservoir of hot plasma with an embedded magnetic dipole field. The plasma temperature (the sum of the ion and electron temperatures) of the reservoir is taken to be 3.1 MK, the plasma density is assumed to be 10^8 cm^{-3}, and the strength of the magnetic field at the poles is taken to be 1 G. In an attempt to incorporate the effects of coronal heating and heat transfer and enable the use of a realistic value for the specific heat ratio ($\gamma = 5/3$) such that the effects of adiabatic cooling at large heliocentric distances are correctly modeled, the coronal plasma is heated by an empirically formulated volumetric heating function in the vicinity of the Sun. Outflows from high-latitude coronal holes at the poles are then produced through the specification of boundary conditions and this heating function. The heat source is taken to be latitude dependent and decrease exponentially with radial distance from the Sun with a scale height of 4.5 R$_s$. The latitude dependence of the heating function is chosen such that outflows are produced from coronal holes for $0° \leq \theta \leq 30°$ where θ is the helio-colatitude, no outflow and a closed field-line structure are produced for $30° \leq \theta \leq 90°$, and a realistic solution of the solar wind is obtained at large heliocentric distances, with fast and slow streams. A rectangular computational domain is used for the calculation with $-35\,\mathrm{R}_s \leq x \leq 105\,\mathrm{R}_s$, $-35\,\mathrm{R}_s \leq y \leq 105\,\mathrm{R}_s$, $-50\,\mathrm{R}_s \leq z \leq 50\,\mathrm{R}_s$ and the adapted mesh consists of 9,444 blocks and 604,416 cells with 9 refinement levels and a minimum cell size of 1/16 R$_s$ at the solar surface.

Figure 10a shows a meridional cut through the numerical solution of the background solar wind obtained using the parallel adaptive MHD model. The

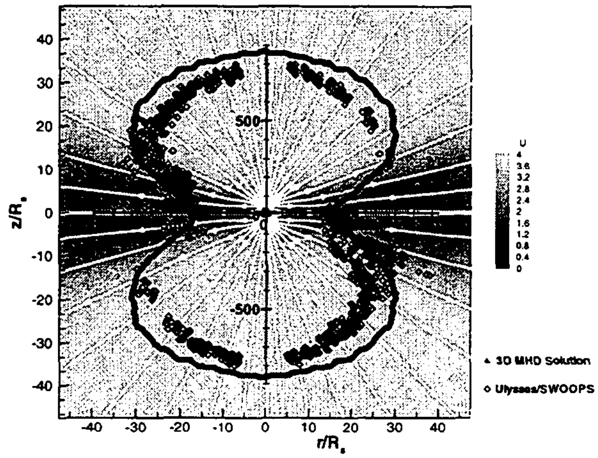

Figure 11: Predicted solar wind velocity and stream lines and polar plot of the asymptotic value of the computed solar wind speed compared to the observations of the Ulysses/SWOOPS instrument.[48]

shading represents the logarithm of the magnitude of the magnetic field and the white lines correspond to predicted field lines. It is evident that the solution, which is dictated by a complex balance of pressure, magnetic, gravitational, and inertial forces, has regions of open and closed magnetic field lines that lead to the formation of a "helmet" streamer configuration with associated neutral point and equatorial current sheet similar to the now classical solution obtained by Pneuman and Kopp.[49] However, unlike the Pneuman-Kopp model, because of the additional volumetric heating included in the present calculations the solution more correctly mimics the dual state features of the solar wind. A fast solar wind ($\sim$ 800 km/s) is produced above $\sim$ 30° heliolatitude, a slow solar wind ($\sim$ 400 km/s) is produced near the solar equator, and reasonable values for the solar wind temperature and and density and interplanetary magnetic field are obtained at 1/2 AU. The two-state nature of the computed solution is depicted in Figure 11. Shown is a polar diagram of the asymptotic values of the computed solar wind speed as a function of heliolatitude. These predicted results are consistent with Ulysses observations[48] as shown by the comparison to the Ulysses/SWOOPS data also shown in the figure. Note that the over-expanded plasma associated with streamers near the edges of coronal holes is the only source of the slow solar wind in this simulation as no mass flux is emerging from the closed field line regions.

C. Simulation of a Coronal Mass Ejection

Coronal mass ejections (CMEs) are highly transient solar events involving the expulsion of mass and magnetic field from the solar surface. On the order of 10^{12} kg of plasma may be expelled from the solar surface during a typical event. These dynamic events originate in closed magnetic field regions of the corona. They produce large-scale reconfiguration of the coronal magnetic field and generate large solar wind disturbances that appear to be the primary cause of major geomagnetic storms at Earth.

Some initial numerical studies of the formation and evolution of CMEs in the inner heliosphere have also been carried out using the block-based AMR scheme. In these studies, the background solar wind solution described above was used as the initial solution and CMEs were initiated solely by localized isothermal density and pressure enhancements and studied as an initial value problem. An example of such a CME simulation is shown in Figures 10a–d. The figure depicts four meridional snapshots of a computed CME solution. In this calculation, density and pressure enhancements were introduced at the solar surface at 11° in heliolatitude above the equatorial plane. The density was allowed to gradually increase and reach a maximum of a 40:1 increase in a three-dimensional region on the solar surface about 0.06 R_s wide ($\approx$ 3.5° in longitude and latitude) and then to gradually return to its original values over a period of 16 hrs. Figure 10 shows the initial solar wind solution and the density-driven CME at times of 2, 4, and 24 hrs after onset, respectively. It can be seen that the density enhancement first leads to the "filling" of the closed magnetic field lines with additional plasma and subsequent expansion of the closed field line region. After a period of time the closed field lines are unable to contain the additional plasma and the density disturbance disrupts the streamers. The resulting CME then moves more rapidly through the inner corona and propagates outward into interplanetary space, dragging out closed field lines with it and disrupting the heliospheric current sheet as it moves. As the CME is purely driven by a density enhancement, the emerging CME magnetic field is largely poloidal (having only radial and latitudinal components) without any significant helicity (twist), a clear limitation of the present simplified calculations as the helicity of the emerging magnetic flux is thought to be an important aspect of CMEs. A magnetic cavity propagates behind the front of the disturbance, which moves at velocities nearing 450 km/s. At 17-19 hrs into the simulation, the density and pressure enhancements at the solar surface have completely diminished and the CME field lines begin disconnecting from the solar surface resulting in the reformation of the current sheet. The solution at $t = 24$ hrs in the lower right panel of Figure 10 depicts this reformation process.

The preceding calculation of the CME covered a period of 24 hours of actual simulated time. The calculation took about 1.5 hours to perform on a 512-node Cray T3E-600, indicating that the parallel solver was running faster than real time by a factor of 16 for this particular simulation.

V. Summary

A new parallel solution-adaptive numerical scheme has been described for solving the governing equations of ideal MHD. The combination of a robust upwind finite-volume discretization procedure and a parallel block-based AMR strategy has yielded a reliable and powerful numerical method for performing large-scale simulations of MHD flows on high-performance parallel machines that is capable of resolving multiple solution scales. The viability of the method has been demonstrated for several problems of current interest in space plasma physics. Important details of the block-based AMR strategy, hierarchical data structure, and parallel implementation leading to very high parallel performance on MIMD distributed-memory architectures have been described. Linear scalability of the solution-adaptive approach has been demonstrated. The algorithm has achieved 342 GFlops on a 1,490-processor Cray T3E-1200 with near-perfect scalability. The method represents a significant improvement over the current generation of space plasma simulation tools, in terms of both computational performance and solution resolution capabilities, and thereby permits the study of a much wider class of problems.

Acknowledgments

This work was supported by the NSF-NASA-AFOSR interagency grant NSF ATM-9318181 and by NASA HPCC CAN NCCS5-146.

References

[1] Brackbill, J. U., and Barnes, D. C., "The Effect of Nonzero $\nabla \cdot \mathbf{B}$ on the Numerical Solution of the Magnetohydrodynamic Equations", *J. Comput. Phys.*, Vol. 35, pp. 426–430, 1980.

[2] Godunov, S. K., "Symmetric Form of the Equations for Magnetohydrodynamics", Report, Computer Center, Siberian Branch of USSR Academy of Sciences, 1972.

[3] Godunov, S. K., "An Interesting Class of Quasilinear Systems", *Sov. Math. Dokl.*, Vol. 2, pp. 947–949, 1961.

[4] Powell, K. G., "An Approximate Riemann Solver for Magnetohydrodynamics (That Works in More than One Dimension)", Report 94-24, ICASE, July 1994.

[5] 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", submitted to the Journal of Computational Physics, July 1998.

[6] Tóth, G., and Odstrčil, D., "Comparison of Some Flux Corrected Transport and Total Variation Diminishing Numerical Schemes for Hydrodynamic and Magnetohydrodynamic Problems", *J. Comput. Phys.*, Vol. 128, pp. 82–100, 1996.

[7] Tanaka, T., "Generation Mechanisms for Magnetosphere-Ionosphere Current Systems Deduced from a Three-Dimensional MHD Simulation of the Solar Wind-Magnetosphere-Ionosphere Coupling Processes", *J. Geophys. Res.*, Vol. 100, No. A7, pp. 12,057–12,074, 1995.

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

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

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

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

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

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

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

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

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

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

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

[19] Osher, S., and Chakravarthy, S. R., "High Resolution Schemes and the Entropy Condition", *SIAM J. Numer. Anal.*, Vol. 21, No. 5, pp. 955–984, 1984.

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

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

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

[23] Brio, M., and Wu, C. C., "An Upwind Differencing Scheme for the Equations of Ideal Magnetohydrodynamics", *J. Comput. Phys.*, Vol. 75, pp. 400–422, 1988.

[24] Zachary, A. L., and Colella, P., "A Higher-Order Godunov Method for the Equations of Ideal Magnetohydrodynamics", *J. Comput. Phys.*, Vol. 99, pp. 341–347, 1992.

[25] Roe, P. L., and Balsara, D. S., "Notes on the Eigensystem of Magnetohydrodynamics", *SIAM J. Appl. Math.*, Vol. 56, No. 1, pp. 57–67, 1996.

[26] Dai, W., and Woodward, P. R., "An Approximate Riemann Solver for Ideal Magnetohydrodynamics", *J. Comput. Phys.*, Vol. 111, pp. 354–372, 1994.

[27] Dai, W., and Woodward, P. R., "Extension of the Piecewise Parabolic Method to Multidimensional Ideal Magnetohydrodynamics", *J. Comput. Phys.*, Vol. 115, pp. 485–514, 1994.

[28] Croisille, J.-P., Khanfir, R., and Chanteur, G., "Numerical Simulation of the MHD Equations by a Kinetic-Type Method", *SIAM J. Sci. Comput.*, Vol. 10, No. 1, pp. 81–92, 1995.

[29] Linde, T. J., *A Three-Dimensional Adaptive Multifluid MHD Model of the Heliosphere*, PhD thesis, University of Michigan, May 1998.

[30] Harten, A., Lax, P. D., and van Leer, B., "On Upstream Differencing and Godunov-Type Schemes for Hyperbolic Conservation Laws", *SIAM Rev.*, Vol. 25, No. 1, pp. 35–61, 1983.

[31] Einfeldt, B., Munz, C. D., Roe, P. L., and Sjögreen, B., "On Godunov-Type Methods near Low Densities", *J. Comput. Phys.*, Vol. 92, pp. 273–295, 1991.

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

[33] Powell, K. G., Roe, P. L., Myong, R. S., Gombosi, T. I., and De Zeeuw, D. L., "An Upwind Scheme for Magnetohydrodynamics", Paper 95-1704-CP, AIAA, June 1995.

[34] 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.

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

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

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

[38] 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.

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

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

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

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

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

[44] Paillère, H., Powell, K. G., and De Zeeuw, D. L., "A Wave-Model-Based Refinement Criterion for Adaptive-Grid Computation of Compressible Flows", Paper 93-0322, AIAA, January 1993.

[45] Goodman, M. L., "A Three-Dimensional, Iterative, Mapping Procedure for the Implemenation of an Ionosphere-Magnetosphere Anisotropic Ohm's Law Boundary Condition in Global magnetohydrodynamic Simulations", *Ann. Geophys.*, Vol. 13, pp. 843–853, 1995.

[46] Amm, O., "Comment on "A Three-dimensional, Iterative, Mapping Procedure for the Implemenation of an Ionosphere-Magnetosphere Anisotropic Ohm's Law Boundary Condition in Global Magnetohydrodynamic Simulations" by Michael L. Goodman", *Ann. Geophys.*, Vol. 14, pp. 773–774, 1996.

[47] Schwenn, R., and Marsch, E., editors, *Physics of the Inner Heliosphere*, Springer-Verlag, New York, 1990.

[48] McComas, D. J., Bame, S. J., Barraclough, B. L., Feldman, W. C., Funsten, H. O., Gosling, J. T., Riley, P., Skoug, R., Balogh, A., Forsyth, R., Goldstein, B. E., and Neugebauer, M., "Ulysses' Return to the Slow Solar Wind", *Geophys. Res. Lett.*, Vol. 25, No. 1, pp. 1–4, 1998.

[49] Pneuman, G. W., and Kopp, R. A., "Gas-Magnetic Field Interactions in the Solar Corona", *Sol. Phys.*, Vol. 18, pp. 258–270, 1971.