

Parallel, Adaptive-Mesh-Refinement MHD for Global Space-Weather Simulations

Kenneth G. Powell*, Tamas I. Gombosi*, Darren L. De Zeeuw*, Aaron J. Ridley*,
Igor V. Sokolov*, Quentin F. Stout* and Gábor Tóth*[†]

**Center for Space Environment Modeling
University of Michigan*

[†]Department of Atomic Physics, Loránd Eötvös University, Budapest, Hungary

Abstract. The first part of this paper reviews some issues representing major computational challenges for global MHD models of the space environment. These issues include mathematical formulation and discretization of the governing equations that ensure the proper jump conditions and propagation speeds, regions of relativistic Alfvén speed, and controlling the divergence of the magnetic field. The second part of the paper concentrates on modern solution methods that have been developed by the aerodynamics, applied mathematics and DoE communities. Such methods have recently begun to be implemented in space-physics codes, which solve the governing equations for a compressible magnetized plasma. These techniques include high-resolution upwind schemes, block-based solution-adaptive grids and domain decomposition for parallelization. We describe the space physics MHD code developed at the University of Michigan, based on the developments listed above.

INTRODUCTION

Global computational models based on first principles represent a very important component of efforts to understand the intricate processes coupling the Sun to the geospace environment. The hope for such models is that they will eventually fill the gaps left by measurements, extending the spatially and temporarily limited observational database into a self-consistent global understanding of our space environment.

Presently, and in the foreseeable future, magnetohydrodynamic (MHD) models are the only models that can span the enormous distances present in the magnetosphere. However, it should not be forgotten that even generalized MHD equations are only a relatively low-order approximation to more complete physics; they provide only a simplified description of natural phenomena in space plasmas.

NON-RELATIVISTIC MAGNETOHYDRODYNAMICS

The governing equations for an ideal, non-relativistic, compressible plasma may be written in a number of different forms. While the different forms of the MHD equations describe the same physics at the differential equation level, there are important practical differences

when one solves discretized forms of the various formulations.

According to the Lax-Wendroff theorem [1] only conservative schemes can be expected to get the correct jump conditions and propagation speed for a discontinuous solution. This fact is much less emphasized in the global magnetosphere simulation literature than the more controversial divergence of B issue. In some test problems the non-conservative discretization of the MHD equations can lead to significant errors, which do not diminish with increased grid resolution.

Fully Conservative Form

The fully conservative form of the equations is

$$\frac{\partial \mathbf{U}}{\partial t} + (\nabla \cdot \mathbf{F})^T = 0, \quad (1)$$

where \mathbf{U} is the vector of conserved quantities and \mathbf{F} is a flux diad,

$$\mathbf{U} = \begin{pmatrix} \rho \\ \rho \mathbf{u} \\ B \\ E_{mhd} \end{pmatrix} \quad (2)$$

$$\mathbf{F} = \begin{pmatrix} \rho \mathbf{u} \\ \rho \mathbf{u} \mathbf{u} + \left(p + \frac{1}{2\mu_0} B^2 \right) \mathbf{I} - \frac{1}{\mu_0} \mathbf{B} \mathbf{B} \\ \mathbf{u} \mathbf{B} - \mathbf{B} \mathbf{u} \\ \mathbf{u} \left(E_{mhd} + p + \frac{1}{2\mu_0} B^2 \right) - \frac{1}{\mu_0} (\mathbf{u} \cdot \mathbf{B}) \mathbf{B} \end{pmatrix}^T \quad (3)$$

where E_{mhd} is the magnetohydrodynamic energy, given by

$$E_{mhd} = \frac{1}{2} \rho u^2 + \frac{1}{\gamma - 1} p + \frac{1}{2\mu_0} B^2 \quad (4)$$

Symmetrizable Formulation

Symmetrizable systems of conservation laws have been studied by Godunov [2] and Harten [3], among others. One property of the symmetrizable form of a system of conservation laws is that an added conservation law

$$\frac{\partial(\rho s)}{\partial t} + \frac{\partial(\rho s u_x)}{\partial x} + \frac{\partial(\rho s u_y)}{\partial y} + \frac{\partial(\rho s u_z)}{\partial z} = 0$$

for the entropy s can be derived by a linear combination of the system of equations. For the ideal MHD equations, as for the gasdynamic equations, the entropy is $s = \log(p/\rho^\gamma)$. Another property is that the system is Galilean invariant; all waves in the system propagate at speeds $u \pm c_w$ (for MHD, the possible values of c_w are the Alfvén, magnetofast and magnetoslow speeds). Neither of these properties holds for the fully conservative form of the MHD equations.

Godunov showed that the fully conservative form of the MHD equations (eq. 1) is not symmetrizable [2]. The symmetrizable form may be written as

$$\frac{\partial \mathbf{U}}{\partial t} + (\nabla \cdot \mathbf{F})^T = \mathbf{Q}, \quad (5)$$

where

$$\mathbf{Q} = -\nabla \cdot \mathbf{B} \begin{pmatrix} 0 \\ \frac{1}{\mu_0} \mathbf{B} \\ \mathbf{u} \\ \frac{1}{\mu_0} \mathbf{u} \cdot \mathbf{B} \end{pmatrix} \quad (6)$$

Vinokur separately showed that eq. (5) can be derived starting from the primitive form, if no stipulation is made about $\nabla \cdot \mathbf{B}$ in the derivation. Powell showed that this symmetrizable form can be used to derive a Roe-type approximate Riemann solver for solving the MHD equations in multiple dimensions [4].

The MHD eigensystem arising from eq. (1) or eq. (5) leads to eight eigenvalue/eigenvector pairs. The eigenvalues and associated eigenvectors correspond to an entropy wave, two Alfvén waves, two magnetofast waves, two magnetoslow waves, and an eighth eigenvalue/eigenvector pair that depends on which form of

the equations is being solved. This last wave (which describes the jump in the normal component of the magnetic field at discontinuities) has a zero eigenvalue in the fully conservative case, and an eigenvalue equal to the normal component of the velocity, u_n , in the symmetrizable case. The expressions for the eigenvectors, and the scaling of the eigenvectors, are more intricate than in gasdynamics [5].

We note that while eq.(1) is fully conservative, the symmetrizable formulation (given by eq. 5) is formally not fully conservative. Terms of order $\nabla \cdot \mathbf{B}$ are added to what would otherwise be a divergence form. The danger of this is that shock jump conditions may not be correctly met, unless the added terms are small, and/or they alternate in sign in such a way that the errors are local, and in a global sense cancel in some way with neighboring terms. This downside, however, has to be weighed against the alternative; a system (i.e., the one without the source term) that, while conservative, is not Galilean invariant, has a zero eigenvalue in the Jacobian matrix, and is not symmetrizable.

SEMI-RELATIVISTIC PLASMAS

While the solar-wind speed remains non-relativistic in the solar system, the intrinsic magnetic fields of several planets in the solar system are high enough, and the density of the solar wind low enough, that the Alfvén speed,

$$V_A = \sqrt{\frac{B^2}{\mu_0 \rho}} \quad (7)$$

can reach appreciable fractions of the speed of light. In the case of Jupiter, the Alfvén speed in the vicinity of the poles is of order ten! Even Earth has a strong enough intrinsic magnetic field that the Alfvén speed reaches twice the speed of light in Earth's near-auroral regions.

Limiting the Alfvén Speed

For these regions, solving the non-relativistic ideal MHD equations does not make sense. Having waves in the system propagating faster than the speed of light, besides being non-physical, causes a number of numerical difficulties. However, solving the fully relativistic MHD equations is overkill. What is called for is a semi-relativistic form of the equations, in which the flow speed and acoustic speed are non-relativistic, but the Alfvén speed can be relativistic. A derivation of these semi-relativistic equations from the fully relativistic equations is given in [6]; the final result is presented here.

The semi-relativistic ideal MHD equations are of the form

$$\frac{\partial \mathbf{U}_{sr}}{\partial t} + (\nabla \cdot \mathbf{F}_{sr})^T = 0 \quad (8)$$

where the state vector, \mathbf{U}_{sr} , and the flux diad, \mathbf{F}_{sr} , are

$$\mathbf{U}_{sr} = \begin{pmatrix} \rho \\ \rho \mathbf{u} + \frac{1}{c^2} \mathbf{S}_A \\ \mathbf{B} \\ \frac{1}{2} \rho u^2 + \frac{1}{\gamma-1} p + e_A \end{pmatrix} \quad (9)$$

$$\mathbf{F}_{sr} = \begin{pmatrix} \rho \mathbf{u} \\ \rho \mathbf{u} \mathbf{u} + p \mathbf{I} + \mathbf{P}_A \\ \mathbf{u} \mathbf{B} - \mathbf{B} \mathbf{u} \\ \left(\frac{1}{2} \rho u^2 + \frac{\gamma}{\gamma-1} p \right) \mathbf{u} + \mathbf{S}_A \end{pmatrix}^T \quad (10)$$

In the above,

$$\mathbf{S}_A = \frac{1}{\mu_0} (\mathbf{E} \times \mathbf{B}) \quad (11)$$

$$e_A = \frac{1}{2\mu_0} \left(B^2 + \frac{1}{c^2} E^2 \right) \quad (12)$$

$$\mathbf{P}_A = e_A \mathbf{I} - \frac{1}{\mu_0} \mathbf{B} \mathbf{B} - \frac{1}{\mu_0 c^2} \mathbf{E} \mathbf{E} \quad (13)$$

are the Poynting vector, the electromagnetic energy density, and the electromagnetic pressure tensor, respectively. The electric field \mathbf{E} is related to the magnetic field \mathbf{B} by Ohm's law.

Lowering the Speed of Light

This new system of equations has wave speeds that are limited by the speed of light; for strong magnetic fields, the modified Alfvén speed (and the modified magnetofast speed) asymptote to c . The modified magnetoslow speed asymptotes to a , the acoustic speed. This property offers the possibility of a rather tricky convergence-acceleration technique for explicit time-stepping schemes, first suggested by Boris [7]; the wave speeds can be lowered, and the stable time-step thereby raised, by artificially lowering the value taken for the speed of light. This method is known as the ‘‘Boris correction.’’

The equations in section are valid in physical situations in which $V_A > c$. A slight modification yields a set of equations, the steady-state solutions of which are independent of the value taken for the speed of light. Defining the true value of the speed of light to be c_0 , to distinguish it from the artificially lowered speed of light, c , the equations are:

$$\frac{\partial \mathbf{U}_{sr}}{\partial t} + (\nabla \cdot \mathbf{F}_{sr})^T = \mathbf{Q}_{c_0} \quad (14)$$

where the state vector, \mathbf{U}_{sr} , and the flux diad, \mathbf{F}_{sr} , are as defined above, and the new source term in the momentum

equation is

$$\mathbf{Q}_{c_0} = \frac{1}{\mu_0} \left(\frac{1}{c_0^2} - \frac{1}{c^2} \right) \mathbf{E} \mathbf{V} \cdot \mathbf{E} \quad (15)$$

An implementation of the semi-relativistic equations has been made in the BATSUS code developed at the University of Michigan [8, 6].

SOLUTION TECHNIQUES

Finite-Volume Scheme

The MHD equations are well suited for finite volume methods when the governing equations are integrated over a computational cell i , yielding

$$\frac{d\mathbf{U}_i}{dt} = -\frac{1}{V_i} \sum_{\text{faces}} \mathbf{F} \cdot \hat{\mathbf{n}} A - \frac{\mathbf{Q}_i}{V_i} \sum_{\text{faces}} \mathbf{B} \cdot \hat{\mathbf{n}} A, \quad (16)$$

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

$$\mathbf{Q}_i = - \begin{bmatrix} 0 \\ \frac{1}{\mu_0} \mathbf{B}_i \\ \mathbf{u}_i \\ \frac{1}{\mu_0} \mathbf{u}_i \cdot \mathbf{B}_i \end{bmatrix}. \quad (17)$$

The numerical face fluxes, $\mathbf{F} \cdot \hat{\mathbf{n}}$, are defined in terms of the left and right interface solution states, \mathbf{U}_L and \mathbf{U}_R , as follows

$$\mathbf{F} \cdot \hat{\mathbf{n}} = \mathcal{F}(\mathbf{U}_L, \mathbf{U}_R, \hat{\mathbf{n}}), \quad (18)$$

where \mathbf{U}_L and \mathbf{U}_R are the state vectors at the left and right sides of the interface.

TVD-MUSCL

Because the MHD equations are a system of hyperbolic conservation laws, many of the techniques that have been developed for the Euler equations can be applied relatively straightforwardly. In particular, the high-resolution finite-volume approach of van Leer [9] (i.e. approximate Riemann solver + limited interpolation scheme + multi-stage time-stepping scheme) is perfectly valid. The Rusanov/Lax-Friedrichs approximate Riemann solver can be applied directly; no knowledge of the eigensystem of the MHD equations is required other than the fastest wave speed in the system. A Roe-type scheme [10] can be constructed for non-relativistic

MHD, but requires more work, because of the complexity of the eigensystem. In addition, an HLLC-type Riemann solver has been derived by Linde [11]; it is less dissipative than the Rusanov/Lax-Friedrichs scheme, but less computationally intensive than the Roe scheme. Whichever approximate Riemann solver is chosen to serve as the flux function, standard interpolation schemes and limiters can be used to construct a finite-volume scheme.

CONTROLLING $\nabla \cdot \mathbf{B}$

One way in which the numerical solution of the MHD equations differs from that of the gasdynamic equations is the constraint that $\nabla \cdot \mathbf{B} = 0$. Enforcing this constraint numerically, particularly in shock-capturing codes, can be done in a number of ways, but each way has its particular strengths and weaknesses. This issue is explained more fully in the references such as [12], [4, 8], [13], [14], [15]. Tóth has published a numerical comparison of many of the approaches for a suite of test cases [16].

BLOCK-BASED AMR ON CARTESIAN GRIDS

Adaptive mesh refinement techniques that 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 underresolving the solution in regions deemed of interest (e.g., high-gradient regions) and, conversely, avoid overresolving the solution in other less interesting regions (low-gradient regions), thereby saving orders of magnitude in computing resources for many problems. For typical solar wind 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 timescales 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.

Adaptive Blocks

Borrowing from previous work by Berger and coworkers [17, 18, 19, 20, 21] and Quirk [22, 23], 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

FIGURE 1. (left) Self-similar blocks used in parallel block-based AMR scheme. (right) Self-similar blocks illustrating the double layer of ghost cells for both coarse and fine blocks.

scheme described above. The method has some similarities with the block-based approaches described by Quirk and Hanebutte [23] and Berger and Saltzman [21]. Here the governing equations are integrated to obtain volume-averaged solution quantities within rectangular Cartesian computational cells. 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 nondimensional 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 $12 \times 12 \times 12 = 1728$ cells are used (see Figure 1). 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.

Computational grids are 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 of the volume of physical space that they occupy. Starting with an initial mesh consisting of blocks of equal size (i.e., equal resolution), 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 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. Conversely, in regions that are deemed overresolved, 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 2. 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 Figure 1. Any of the blocks shown in Figure 1 can in turn be refined, and so on, leading to successively finer blocks. In

the present method, mesh refinement is constrained such that the cell resolution changes by only a factor of 2 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 Figure 1. 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 multistage time-stepping algorithm, ghost cell values are reevaluated to reflect the updated solution values of neighboring blocks. With the AMR approach, additional interblock communication is also required at interfaces with resolution changes to strictly enforce the flux conservation properties of the finite-volume scheme [17, 18, 19]. 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 that the fluxes are conserved across block interfaces.

PARALLEL IMPLEMENTATION

The parallel block-based AMR solver was designed from the ground up with a view to achieving very high performance on massively parallel architectures. The underlying upwind finite-volume solution algorithm, with explicit time stepping, has a very compact stencil and is therefore highly local in nature. The hierarchical data structure and self-similar blocks make domain decomposition of the problem almost trivial and readily enable good load-balancing, a crucial element for truly scalable computing. A natural load balancing is accomplished by simply distributing the blocks equally amongst the processors. Additional optimization is achieved by ordering the blocks using the Peano-Hilbert space filling curve to minimize inter-processor communication. 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. The parallel implementation of the algorithm has been carried out to such an extent, that even the grid adaptation is performed in parallel.

Other features of the parallel implementation include the use of FORTRAN 90 as the programming language

FIGURE 2. Parallel speedup of BATSRUS on various architectures. Black dashed lines represent perfect scaling from single node performance.

and the message passing interface (MPI) library for performing the interprocessor communication. Use of these standards greatly enhances the portability of the code and leads to very good serial and parallel performance. The message passing is performed in an asynchronous fashion with gathered wait states and message consolidation.

Implementation of the algorithm has been carried out on Cray T3E supercomputers, SGI and Sun workstations, on Beowulf type PC clusters, on SGI shared-memory machines, on a Cray T3D, and on several IBM SP2s. BATSRUS nearly perfectly scales to 1,500 processors and a sustained speed of 342 GFlops has been attained on a Cray T3E-1200 using 1,490 PEs. For each target architecture, simple single-processor measurements are used to set the size of the adaptive blocks. The scaling of BATSRUS on various architectures is shown in Figure 2.

Implicit Time-Stepping

In BATSRUS we have a number of time stepping algorithms implemented. The simplest and least expensive scheme is a multistage explicit time stepping, for which the time step is limited by the CFL stability condition. We have also implemented an unconditionally stable fully implicit time stepping scheme [27, 28]. The second order implicit time discretization (BDF2) requires the solution of a non-linear system of equations for all the flow variables. This can be achieved by the Newton-

Krylov-Schwarz approach: a Newton iteration is applied to the non-linear equations; a parallel Krylov type iterative scheme is used to solve the linear systems; the convergence of the Krylov solver is accelerated with a Schwarz type preconditioning. We have implemented two Krylov solvers: BiCGSTAB and GMRES. A modified block incomplete LU (MBILU) preconditioner is applied on a block by block basis. Since every block has a simple Cartesian geometry, the preconditioner can be implemented very efficiently. The resulting implicit scheme requires about 20-30 times more CPU time per time step than the explicit method, but the physical time step can be 1,000 to 10,000 times larger. This implicit algorithm has a very good parallel scaling due to the Krylov scheme and the block by block application of the preconditioner.

In BATSRUS, we can combine explicit and implicit time stepping. Magnetosphere simulations include large volumes where the Alfvén speed is quite low (tens of km/s) and the local CFL number allows large explicit time steps (tens of seconds to several minutes). In these regions implicit time stepping is a waste of computational resources. Since the parallel implicit technique we use is fundamentally block based we only treat those blocks implicitly where the CFL condition would limit the explicit time step to less than the selected time step (typically ~ 10 s). Needless to say, this combined explicit-implicit time stepping represents more computational challenges (such as separate load balancing of explicit and implicit blocks). Overall, this solution seems to be a very promising option, but other potential avenues need to be explored before one makes a final decision about the most efficient time-stepping algorithm for space MHD simulations. These questions will be discussed in an upcoming paper [29].

APPLICATIONS

BATSRUS has been extensively applied to global numerical simulations of the inner heliosphere including CME propagation [30, 31], the coupled terrestrial magnetosphere-ionosphere [32, 33, 34], and the interaction of the heliosphere with the interstellar medium [35]. In addition, it has also been successfully applied to a host of planetary problems ranging from comets [36, 37], to Mercury [38], Venus [39], Mars [40], Saturn [41], to planetary satellites [42, 43].

In this section we briefly summarize our most ambitious space weather simulation so far, in which we used BATSRUS to simulate an entire space weather event, from its generation at the Sun through the formation and evolution of a CME, to its interaction with the magnetosphere-ionosphere system [30, 31]. In this simulation we resolved multiple spatial and temporal

FIGURE 3. 3D representation of the steady-state solar wind solution. The shading represents $\log |\mathbf{B}|$ in the (x, z) - and (x, y) -planes. The thin black lines are the computational mesh and the thick solid lines are magnetic field lines: grey denotes the last closed field lines, black is open field lines expanding to the interplanetary medium just above the heliospheric current sheet, and finally, white lines show open magnetic field lines in the (y, z) -plane.

scales and took advantage of frequent grid refinements and coarsening to follow the CME through interplanetary space. The total number of cells varied between 800,000 and 2 million as the solution evolved. The simulation used 13 levels of grid refinement. The simulation ran faster than real time on a 512 node Cray T3E-600 supercomputer. This simulation demonstrates that we have the necessary experience to undertake the research outlined in this proposal.

Here we only show a few highlights of this simulation. The detailed results have been published in JGR-Space Physics [31].

A steady state solar wind was obtained in the corotating frame for a tilted rotating Sun. The intrinsic magnetic field was approximated by the superposition of a tilted (with respect to the rotation axis) octupole and dipole. Figure 3 depicts a three-dimensional representation of the predicted pre-event steady-state solar wind solution in the vicinity of the Sun. The narrow dark region shown in Figure 3, which also coincides with regions of higher mesh refinement, corresponds to the beginning of the heliospheric current sheet. Due to the combined effects of magnetic tilt and solar rotation, the current sheet is tilted with respect to the rotation axis, and deformed, and resembles a “ballerina skirt.”

Figure 4 shows a 3D representation of the magnetic field configuration 9 hours after the initiation of the CME. 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. One can see that the closed field lines become greatly stretched by the outward moving plasma. This is due to the fact that the plasma β (the ratio of the kinetic and magnetic pressures) is quite large and the magnetic field

FIGURE 4. 3D representation of magnetic field lines 9 hours after the initiation of a CME. Grayscale represents $\log(B)$, white lines are open magnetic field lines, grey lines represent magnetic field lines with both ends connected to the Sun.

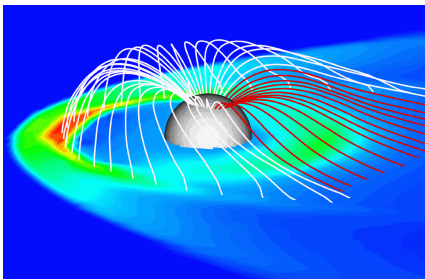


FIGURE 5. 3D representation of the last closed terrestrial field lines for southward IMF conditions. White field lines form the dayside magnetopause, while black ones map to the magnetotail. The grayscale represents normalized thermal pressure.

is “carried” by the outward moving plasma. We also note the decrease of magnetic field strength behind the leading edge of the outward moving disturbance.

The dynamic response of the global magnetosphere to the changing solar wind conditions produced by the density-driven CME was also computed as part of this simulation. The global magnetospheric configuration for quiet-time southward IMF conditions is shown in Figure 5. During the event the solar wind velocity remained nearly radial with the speed gradually decreasing from about 550 km/s to about 450 km/s. The solar wind dynamic pressure increased from its pre-CME value of 2.25 nP (at $t = 72$ hrs) to 4.6 nP at the peak of the event.

The ionospheric potential and convection patterns also change during the CME event. The ionospheric convection shows the two-cell pattern of ionospheric convection typical for southward-type IMF conditions. The convection pattern is also “twisted” due to the presence of a non-zero IMF B_y component. The most important change in the ionosphere is the doubling of the cross-cap potential drop from 30 kV at 70.5h to 60 kV some 27 hours later.

Overall, this simulated space weather event was not

very geoeffective. It is expected that we will be able to generate more geoeffective CMEs with the help of more realistic explosive event generation modules. This simulation, however, demonstrates the present capabilities of BATSUS.

CONCLUDING REMARKS

With the combination of adaptive mesh refinement, domain-decomposition parallelization, and robust finite-volume solvers, methods for solving the ideal MHD equations are developing into powerful tools for a number of applications. With attention to some issues particular to solar-wind modeling (high Alfvén speeds, strong embedded magnetic fields, pressure positivity and divergence control), these tools are becoming quite sophisticated. Much of the work to be done in improving these tools is in coupling them to solvers for regions in which semi-relativistic ideal MHD is not a sufficient model. The results presented in this paper, while preliminary, hint at the new abilities and insights that can be gained from this approach.

ACKNOWLEDGEMENTS

This work was supported by DoD MURI grant F49620-01-1-0359, NSF KDI grant NSF ATM-9980078, NSF CISE grant ACI-9876943, and NASA AISRP grant NAG5-9406. G. Tóth is partially supported by the Education Ministry of Hungary (grant No. FKFP-0242-2000). We also acknowledge the contributions of the developers of TIEGCM and RCM to the coupled magnetosphere model. In particular, the contributions of Ray Roble, Stanislav Sazykin and Richard Wolf are acknowledged and appreciated.

REFERENCES

1. Lax, P. D., and Wendroff, B., *Communications on Pure and Applied Mathematics*, **13**, 217–237 (1960).
2. Godunov, S. K., “Symmetric form of the equations of magnetohydrodynamics (in Russian),” in *Numerical Methods for Mechanics of Continuum Medium*, Siberian Branch of USSR Acad. of Sci., Novosibirsk, 1972, vol. 1, pp. 26–34.
3. Harten, A., *J. Comput. Phys.*, **49**, 357–393 (1983).
4. Powell, K. G., An approximate Riemann solver for magnetohydrodynamics (that works in more than one dimension), Tech. Rep. 94-24, Inst. for Comput. Appl. in Sci. and Eng., NASA Langley Space Flight Center, Hampton, Va. (1994).
5. Roe, P. L., and Balsara, D. S., *SIAM J. Appl. Math.*, **56**, 57–67 (1996).

6. Gombosi, T. I., Tóth, G., De Zeeuw, D. L., Hansen, K. C., Kabin, K., and Powell, K. G., *J. Comput. Phys.*, **177**, 176–205 (2002).
7. Boris, J. P., A physically motivated solution of the Alfvén problem, Tech. Rep. NRL Memorandum Report 2167, Naval Research Laboratory, Washington, D.C. (1970).
8. Powell, K. G., Roe, P. L., Linde, T. J., Gombosi, T. I., and Zeeuw, D. L. D., *J. Comput. Phys.*, **154**, 284–309 (1999).
9. van Leer, B., *J. Comput. Phys.*, **32**, 101–136 (1979).
10. Roe, P. L., *J. Comput. Phys.*, **43**, 357–372 (1981).
11. Linde, T. J., *A Three-Dimensional Adaptive Multifluid MHD Model of the Heliosphere*, Ph.D. thesis, Univ. of Mich., Ann Arbor (1998).
12. Brackbill, J., and Barnes, D., *J. Comput. Phys.*, **35**, 426–430 (1980).
13. Dai, W., and Woodward, P. R., *J. Comput. Phys.*, **142**, 331 (1998).
14. Londrillo, P., and Anna, L. D., *Astrophys. J.*, **530**, 508–524 (2000).
15. Dedner, A., Kemm, F., Kröner, D., Munz, C., Schnitzer, T., and Wesenberg, M., *J. Comput. Phys.*, **00**, 00–00 (2001), submitted.
16. Tóth, G., *J. Comput. Phys.*, **161**, 605–652 (2000).
17. Berger, M. J., *Adaptive Mesh Refinement for Hyperbolic Partial Differential Equations*, Ph.D. thesis, Stanford Univ., Stanford, Calif. (1982).
18. Berger, M. J., *J. Comput. Phys.*, **53**, 484–512 (1984).
19. Berger, M. J., and Colella, P., *J. Comput. Phys.*, **82**, 67–84 (1989).
20. Berger, M. J., and LeVeque, R. J., “An adaptive Cartesian mesh algorithm for the Euler equations in arbitrary geometries,” in *Proc. 9th AIAA Computational Fluid Dynamics Conference*, AIAA Paper No. 89-1930, Buffalo, NY, 1989.
21. Berger, M. J., and Saltzman, S., *Appl. Numer. Math.*, **14**, 239–253 (1994).
22. Quirk, J. J., *An Adaptive Grid Algorithm for Computational Shock Hydrodynamics*, Ph.D. thesis, Cranfield Inst. of Technol., Cranfield, England (1991).
23. Quirk, J. J., and Hanebutte, U. R., A parallel adaptive mesh refinement algorithm, Tech. Rep. 93-63, ICASE (1993).
24. Paillère, H., Powell, K. G., and De Zeeuw, D. L., “A wave-model based refinement criterion for adaptive-grid computation of compressible flows,” in *30th AIAA Aerospace Sciences Meeting*, AIAA-92-0322, Reno, Nevada, 1992.
25. Powell, K. G., Roe, P. L., and Quirk, J., “Adaptive-mesh algorithms for computational fluid dynamics,” in *Algorithmic Trends in Computational Fluid Dynamics*, edited by M. Y. Hussaini, A. Kumar, and M. D. Salas, Springer-Verlag, New York, 1993, pp. 303–337.
26. De Zeeuw, D. L., and Powell, K. G., *J. Comput. Phys.*, **104**, 55–68 (1993).
27. Tóth, G., Keppens, R., and Botchev, M. A., *Astron. Astrophys.*, **332**, 1159–1170 (1998).
28. Keppens, R., Tóth, G., Botchev, M. A., and van der Ploeg, A., *Int. J. for Num. Meth. in Fluids*, **30**, 335–352 (1999).
29. Tóth, G., De Zeeuw, D. L., Gombosi, T. I., Hansen, K. C., Powell, K. G., Ridley, A. J., Roe, P. L., and Sokolov, I. V., *J. Geophys. Res.*, **00**, 00–00 (2002), in preparation.
30. Gombosi, T. I., De Zeeuw, D. L., Groth, C. P. T., Powell, K. G., and Stout, Q. F., *J. Atmos. Solar-Terr. Phys.*, **62**, 1515–1525 (2000).
31. Groth, C. P. T., De Zeeuw, D. L., Gombosi, T. I., and Powell, K. G., *J. Geophys. Res.*, **105**, 25,053 – 25,078 (2000).
32. Gombosi, T. I., De Zeeuw, D. L., Groth, C. P. T., Powell, K. G., and Song, P., “The length of the magnetotail for northward IMF: Results of 3D MHD simulations,” in *Physics of Space Plasmas*, edited by T. Chang and J. R. Jasperse, MIT Press, Cambridge, Mass., 1998, vol. 15, pp. 121–128.
33. Song, P., De Zeeuw, D. L., Gombosi, T. I., Groth, C. P. T., and Powell, K. G., *J. Geophys. Res.*, **104**, 28,361–28,378 (1999).
34. Song, P., Gombosi, T., De Zeeuw, D., Powell, K., and Groth, C. P. T., *Planet. Space Sci.*, **48**, 29–39 (2000).
35. Linde, T. J., Gombosi, T. I., Roe, P. L., Powell, K. G., and De Zeeuw, D. L., *J. Geophys. Res.*, **103**, 1889–1904 (1998).
36. Gombosi, T. I., De Zeeuw, D. L., Häberli, R. M., and Powell, K. G., *J. Geophys. Res.*, **101**, 15233–15253 (1996).
37. Häberli, R. M., Gombosi, T. I., DeZeeuw, D. L., Combi, M. R., and Powell, K. G., *Science*, **276**, 939–942 (1997).
38. Kabin, K., Gombosi, T. I., De Zeeuw, D. L., and Powell, K. G., *Icarus*, **143**, 397–406 (2000).
39. Bauske, R., Nagy, A. F., Gombosi, T. I., De Zeeuw, D. L., Powell, K. G., and Luhmann, J. G., *J. Geophys. Res.*, **103**, 23,625–23,638 (1998).
40. Liu, Y., Nagy, A. F., Groth, C. P. T., De Zeeuw, D. L., Gombosi, T. I., and Powell, K. G., *Geophys. Res. Lett.*, **26**, 2689–2692 (1999).
41. Hansen, K. C., Gombosi, T. I., DeZeeuw, D. L., Groth, C. P. T., and Powell, K. G., *Adv. Space Res.*, **26**, 1681–1690 (2000).
42. Kabin, K., Gombosi, T. I., DeZeeuw, D. L., Powell, K. G., and Israelevich, P. L., *J. Geophys. Res.*, **104**, 2451–2458 (1999).
43. Kabin, K., Combi, M. R., Gombosi, T. I., De Zeeuw, D. L., Hansen, K. C., and Powell, K. G., *PSS*, **49**, 337–344 (2001).