In *Proceedings SC'98*

Adaptive Parallel Computation of a Grand-Challenge Problem: Prediction of the Path of a Solar-Coronal Mass Ejection

Q.F. Stout, D.L. De Zeeuw, T.I. Gombosi,
C.P.T. Groth, H.G. Marshall, K.G. Powell
The University of Michigan, Ann Arbor, MI 48109 USA

Abstract

One of the ways that the Sun interacts with the Earth is through the solar wind, which is an ionized multi-component fluid that emanates from the Sun and travels radially outward at hundreds of kilometers per second. Solar-wind transients, such as Coronal Mass Ejections (CME's), can be particularly important. In rare cases, CME's have affected the lower atmosphere of the Earth, causing regional power-grid failures. More regularly, CME's pose threats to satellites and spacecraft. Due to the extreme range of temporal and spatial scales involved in solar-wind phenomena, it had previously been impossible to predict CME propagation to Earth with faster-than-real-time, well-resolved calculations. Our team has now developed a highly scalable solution-adaptive scheme for predicting CME propagation. The solution-adaptive technique is an adaptive mesh refinement (AMR) scheme for magneto-hydrodynamic (MHD) calculations. The physical domain is decomposed into three-dimensional blocks, where each block forms a regular grid. In regions of relatively high gradients, blocks are successively refined. Blocks are distributed to processors, with communication between neighboring blocks is handled by asynchronous message passing. The benchmark calculation achieved 212 Gflops on a 1024-processor Cray T3E-1200, with the grid adapting over the course of the calculation from 2048 blocks to 11,729 blocks, where each block was composed of 10x10x10 cells. On a 512-processor Cray T3E-600, our benchmark simulations were performed 16 times faster than real time.

Keywords: adaptive mesh refinement (AMR), adaptive blocks, magneto-hydrodynamics (MHD), coronal mass ejection (CME), space weather prediction, heliospheric plasma, hyperbolic PDEs, upwind methods, Riemann solver

1 Introduction

Just above the visible Sun is the Sun's corona, where the solar wind originates. The solar wind, composed of ionized gases with speeds of several hundred kilometers per second, flows out radially into the solar system, interacting with the planets. The structure of the solar wind is complicated, governed by hydrodynamic and electromagnetic physics, and exhibits temporal scales that range over at least five orders of magnitude and spatial scales that range over at least eight orders of magnitude. Transients in the solar wind, such as *Coronal Mass Ejections* (CME's), evolve and propagate to the Earth, directly affecting the conditions in our upper atmosphere. The interaction of a CME with the Earth's magnetosphere can cause a storm with significant magnetic turbulence and an abundance of highly energetic "killer electron". These storms can adversely affect satellites orbiting Earth, such as the $200 million AT&T satellite that suffered a failure on 10 January 1997 due to the effects of a CME. They can also have effects in the lower atmosphere of the earth, adversely affecting surface power grids because the transmission lines behave like giant antennas.

It is important to be able to predict the behavior of a CME so that appropriate measures can be taken to minimize its impact, much as hurricane and tornado predictions can save lives and money. Besides having appropriate observational capability for early detection of a CME, it is important that one be able to accurately simulate its evolution faster than real time. Due to the complicated physics of the solar wind, the range of temporal and spatial scales, and the speed at which a CME travels, such predictive capability has been out of reach until very recently. This paper describes a parallel, solution-adaptive code that can simulate CME propagation and evolution faster than real time, at a useful resolution.

Our code, which we call **BATS-R-US** (**B**lock **A**daptive-**T**ree **S**olar-wind **R**oe-type **U**pwind **S**cheme), has been developed by a team which has worked together for a few years. The code's performance is a combination of many factors, which include use of powerful highly parallel machines, new numerical approaches to magneto-hydrodynamics (MHD), and a parallel implementation which achieves high efficiency and scalability though at least a thousand processors. The parallel computers used were Cray T3Es. Most of our work was done a 512 processor Cray T3E-600, provided by NASA to support National Grand Challenge projects such as ours. In May 1998 SGI/Cray also ran our code on a 1024 processor Cray T3E-1200, to demonstrate the power of this new machine.

We first describe the solution-adaptive numerical analysis scheme used (Section 2), and then describe the parallel implementation (Section 3). Then we explain how the CME was simulated (Section 4), and show the achieved performance (Section 5). We finish with some concluding remarks (Section 6).

2 A Solution-Adaptive Scheme for Solving the Governing Equations

The BATS-R-US code solves the three-dimensional form of the ideal magneto-hydrodynamic (MHD) equations. For solar wind and heliospheric calculations the basic MHD equations are supplemented with additional source terms, which represent the effects of the solar gravitational force on the plasma and a volumetric heating term. The latter is a somewhat empirical term that has been incorporated in an attempt to model micro-physical processes not represented by an ideal MHD description of the plasma such as coronal heating processes and heat and radiation transfer effects. The volumetric heating term is required for a more realistic model of the solar wind based on the ideal MHD equations.

Before the advent of modern high-resolution upwind schemes, researchers solving hyperbolic systems of conservation laws had a choice between extremely dissipative first-order schemes, such as Lax-Friedrichs and Rusanov methods, or second-order centered-differenced based schemes, such as the Lax-Wendroff method, which were much less dissipative but could not treat even weakly discontinuous solutions (e.g., shock waves) without introducing non-physical and potentially destabilizing oscillations in the approximate solutions. In the last 10-15 years, research into upwind finite-volume schemes and approximate Riemman solvers have led to more robust and lower dissipation schemes. These algorithmic advances yielded first-order methods with the minimum dissipation necessary to provide numerical stability. The upwind schemes provided robustness nearly equal to that of the early excessively dissipative first-order schemes combined with solution accuracy nearing that of the early second-order schemes. At the same time, other research into higher-order limited reconstruction techniques provided a way to extend the accuracy of the new upwind schemes to second order or higher, while avoiding the deleterious oscillations associated with earlier higher-order methods.

The BATS-R-US code was designed to take advantage of these advances in upwind methods, approximate Riemann solvers, and limited solution reconstruction. In the MHD model, a cell-centered upwind finite-volume formulation is adopted to solve the governing equations of ideal MHD in divergence form. The limited solution reconstruction of van Leer [5], is used to ensure second-order accuracy away from discontinuities, while simultaneously providing the stability required for monotonic non-oscillatory solutions. In addition, two of the more popular approximate Riemann solvers: the linearized approximate Riemann solver of Roe and a modified version of the HLLE method of Harten, Lax, Van Leer, and Einfeldt recently proposed by Linde, are employed in the evaluation of the numerical flux function. These flux functions were originally designed for gasdynamics. They have been extended and re-derived for ideal MHD as part of the development of the BATS-R-US algorithm. In the process, a number of other very challenging algorithmic issues were solved which limited earlier upwind MHD schemes.

Finally, a multi-stage method of lines approach with implicit treatment of source terms is used to integrate the ordinary differential equations that result from this upwind spatial discretization of the MHD equations. The resulting finite-volume scheme solves for the hydrodynamic and electromagnetic effects in a tightly coupled manner, provides accurate resolution of discontinuous solutions and complicated wave structures, and works equally well across a range of several orders of magnitude in plasma beta.

Further details can be found in [3] and [7], and in the milestone release documentation for BATS-R-US, [1].

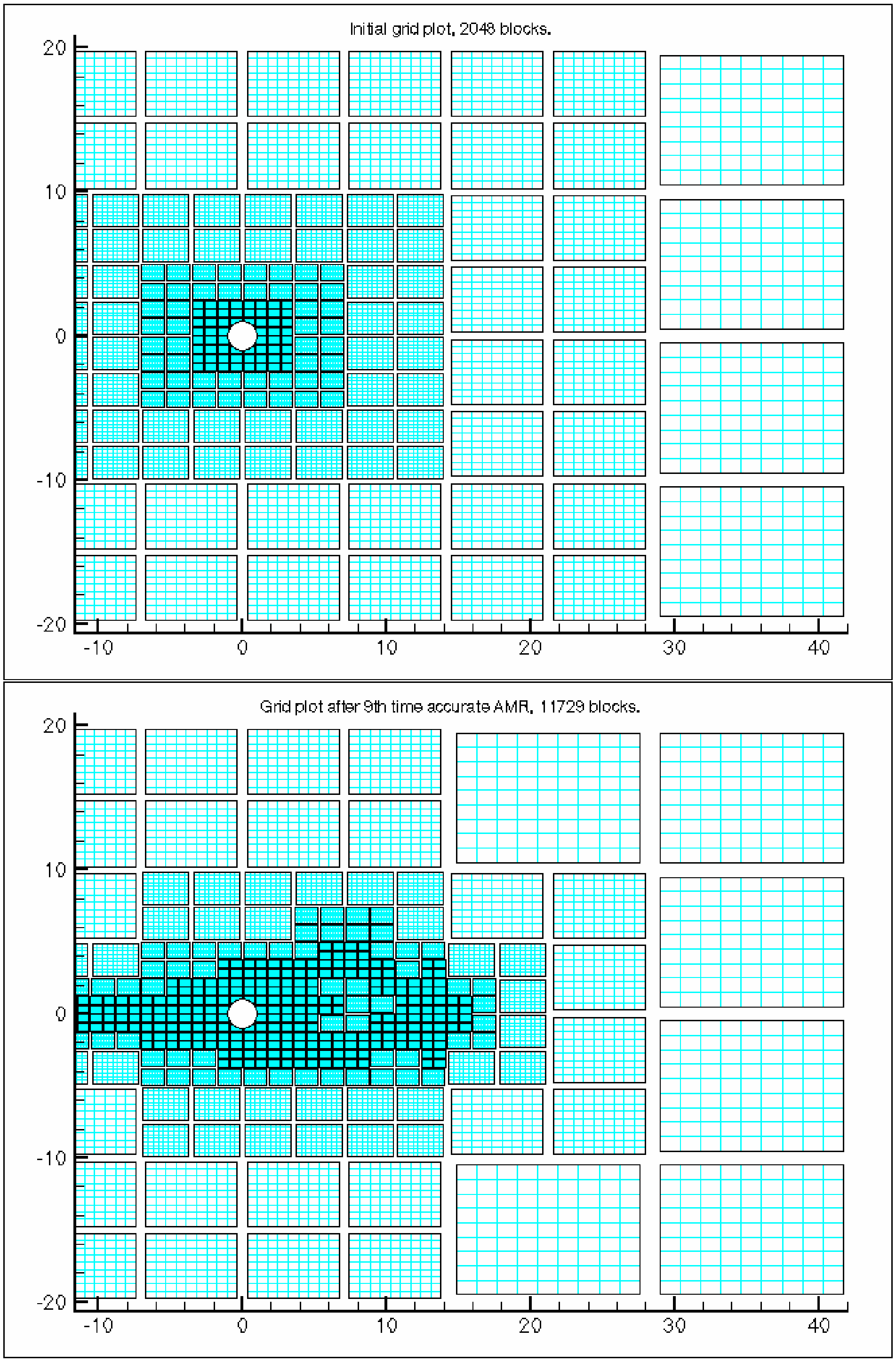
2.1 Solution Adaptation

Computational grids that automatically adapt to the solution of the governing partial differential equations are very effective in treating problems with disparate length scales, saving orders of magnitude in computing resources for many problems. *Adaptive mesh refinement* (AMR) techniques avoid under-resolving the solution in high-gradient regions and, conversely, avoid over-resolving the solution in low-gradient regions at the expense of the more critical regions. For space plasma flow problems, length scales can range from a few kilometers in the near Earth region to the Earth-Sun distance 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. For problems with disparate spatial and temporal scales such as these, the use of AMR is extremely beneficial and is almost a virtual necessity – simple structured grids would grossly under-resolve much of the problem, while over-resolving relatively uninteresting regions.

3 Algorithms and Parallel Implementation

The approach to adaptation taken in the BATS-R-US code is one of self-similar adaptive blocks. This approach was designed, from the ground up, with performance on massively parallel machines in mind. The basic unit of data is a three-dimensional *adaptive block* of grid cells. For a given run, all blocks have the same size; a typical run might use blocks of size $10 \times 10 \times 10$ cells. The size of the blocks is machine- and problem-dependent, based on factors such as the computational efficiency of each block, the communication/calculation ratios for cells, the number of blocks needed, etc.

An initial grid is composed of a number of blocks, all at the same level of refinement. Then, in regions that appear under-resolved (as defined by a physical or numerical error adaptation criteria), a block is refined with each of its eight octants becoming a block with the same number of cells as the original block, i.e., self-similar blocks. In regions that appear over-resolved eight blocks can be coalesced (coarsened) back to one. Typical simulations have 10-15 levels of refinement; some calculations have more than 20 levels of



Initial grid plot, 2048 blocks.

Grid plot after 9th time accurate AMR, 11729 blocks.

Figure 1: Initial Grid and Final Adapted Grid

refinement In the case of 20 levels of refinement, the finest cells on the mesh are approximately one million times smaller in linear dimension, and 10^18 times smaller in volume, than the coarsest cells on the mesh. Further discussion of adaptive blocks appears in [8].

Figure 1 shows a comparison of the initial grid and the final adapted grid. The black lines show the blocks while the cyan lines show individual cells. Strong gradients in magnetic field strength were used to determine which cells should be refined.

BATS-R-US is written in Fortran 90, with MPI used to handle all communication. Thus it is portable to a wide range of machines, and it has been run on Cray T3D and T3E, IBM SP2 and SGI Powerchallenge parallel systems, as well as Sun and SGI workstations.

4 Simulating a CME

In our initial studies of the formation and evolution of CMEs in the heliosphere, we simulated the initiation of a CME by local pressure and density enhancements. This was primarily for the sake of numerical convenience and not because it is thought that pressure enhancement is the most likely mechanism for triggering the onset of CMEs. The goal of these first studies was more to test the ability of the MHD model to simulate solar wind disturbances than to better understand CMEs.

The pressure enhancement was allowed to build up gradually to a maximum of 40:1 pressure increase and then to gradually decay over a period of 16 hrs. The width of the region of enhancement was about 0.06 R_s. The pressure pulse first “fills” the closed magnetic field line region with additional plasma. After a period of time the closed field magnetic configuration is unable to contain this additional plasma and the disturbance “pierces” the closed field lines. The resulting CME moves rapidly through the inner corona and propagates outward into interplanetary space, disrupting the heliospheric current sheet as it moves. A magnetic cavity propagates with the disturbance, which moves at velocities ranging from about 300-450 km/s. At 17-19 hrs into the simulation the pressure enhancements disappear and the CME field lines disconnect from the solar surface and the current sheet begins to reform.

In Figures 2 and 3, colors represent the log of the magnetic field magnitude. One can see that the grid adapts to the flow, and evolves over time.

5 Performance

The BATS-R-US code was designed from the ground up with parallelism in mind. The underlying basic algorithm that was chosen is highly local in nature, leading to minimal communication overhead. The data structures on which the code was built allow a natural partitioning of the data, and greatly facilitate load-balancing, a crucial element of truly scalable computing. Basically, balancing the number of blocks per processor balances the load. Because blocks have a high ratio of calculation to communication, the specifics of which blocks are mapped to which processors have negligible effects on the run time, which greatly simplifies block distribution. This is true even though each block must communicate with all of its neighbors in every time step.

Many performance gains were a consequence of the self-similarity of the blocks. Users of MPI Cartesian communicator (equal) grids and other parallel methods where each processor has an equal slice of a single regular grid enjoy the benefits of speed-up due to optimization of a single set of code duplicated on all grids on all processors. We achieve similar speed-ups for grids that are of unequal physical size, because of the

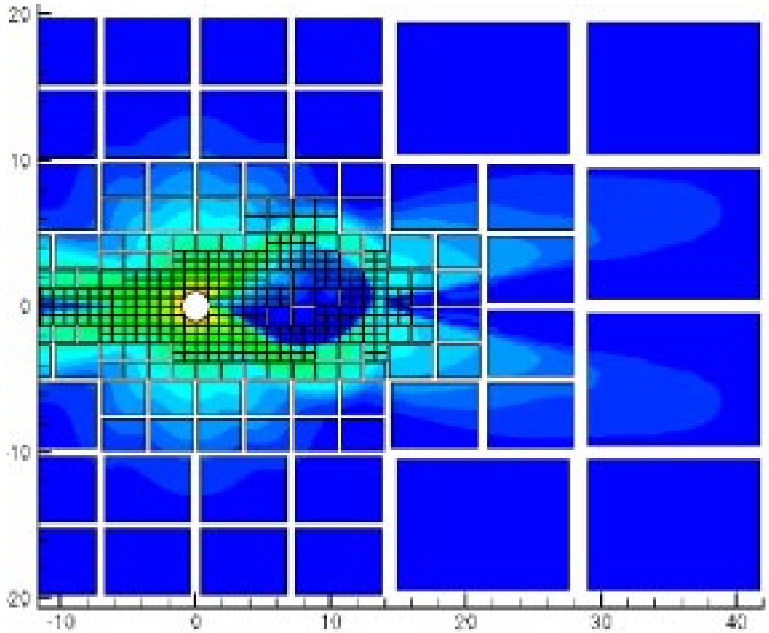


Figure 2: A 2D slice of the 3D solution for a CME at 8 hours after initialization.

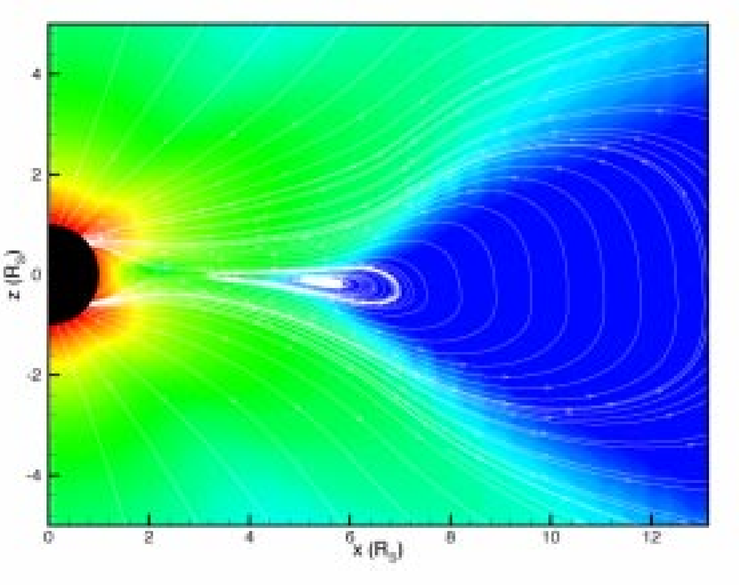


Figure 3: A 2D slice of the 3D solution for a CME at 20 hours after initialization.

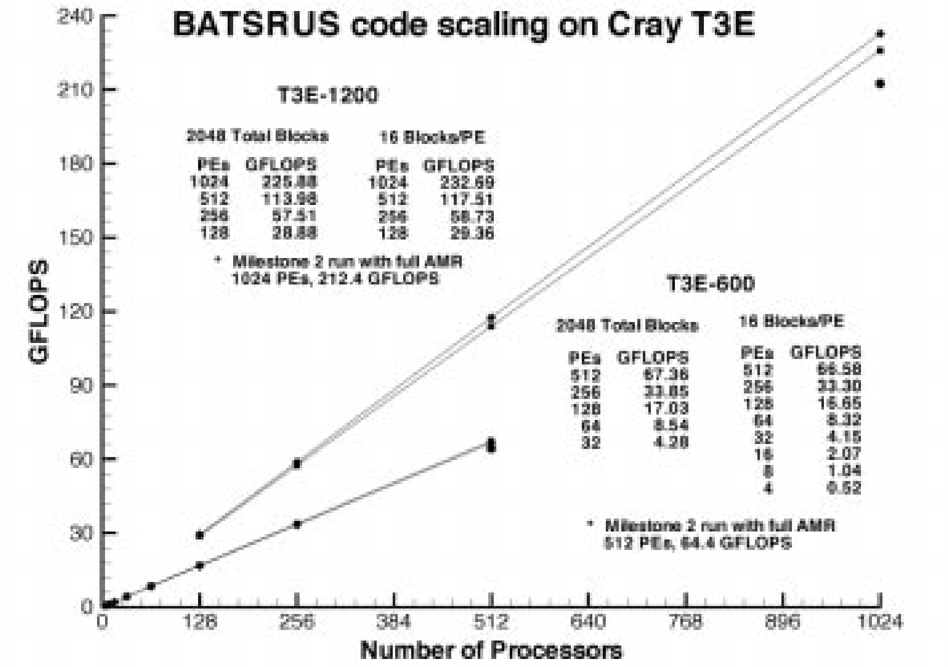


Figure 4: Performance results for benchmark calculations

self-similar properties of the blocks. Indeed each processor sees all blocks as exactly the same in the BATS-R-US model, where only the data distinguishes blocks as having different physical size and resolution. Stripping for cache in the most expensive part of the Roe solver benefited all blocks, as did improvements in the asynchronous message passing harness. The size of the optimal cache-line stripping varies with the CPU, as expected, but we found additional speed-ups by letting it vary with the direction of the Roe solver. The stripping size was also shortened for use with Cray streams.

We have developed a code that partially automates the determination of the best cache strip size for a machine/model pair. The script also helps us make choices of a few limited settings for the message passing harness as well (in particular the data size and setting of wait states). Such a script is useful because the BATS-R-US code is being used for a wide range of simulations (see [2,4,6]), and optimizations for one may not be appropriate for others.

5.1 Scaling

Figure 4 shows performance scaling results for two types of runs which simulate CME evolution and propagation. In the first, timings are shown for a problem that has a fixed size per processor; in the second, timings are shown for a problem that has a fixed total size. Both of these runs were carried out on a T3E-600 and a T3E-1200. As expected, the fixed size problem is somewhat less scalable, since as it is distributed across more and more processors the ratio of communications overhead to computing cost rises. However, unlike most other codes, this effect is minimal here. As can be seen from Figure 4, for both problems, the scaling to 1024 processors is nearly perfect.

We note that this calculation of the pressure-driven CME covered a period of 24 hours of simulated time. The calculation took about 1.5 hours to perform on a 512-node Cray T3E-600, indicating that the BATS-R-US code was running faster than real time by a factor of 16 for this particular simulation. This is important to note because a predictive model of space weather must be capable of running substantially faster than real time to be useful for forecasting purposes.

6 Concluding Remarks

Useful forecasts of Coronal Mass Ejections can be obtained by using a high-performance MPP to run the solution-adaptive parallel scheme developed by the authors for solving MHD problems. The solution-adaptive scheme helps resolve the vastly disparate scales in the problem, and the excellent parallel efficiency of the code allows runs with millions of degrees of freedom to be run faster than real time.

One area of future interest is the ability to couple the current code to other codes that model portions of the Sun-Earth connection, for example a model of the physics in the Earth's ionosphere, or sub-coronal solar physics. BATS-R-US is designed to be a *good neighbor* in running with other models. The self-similar block has naturally simplified load-balancing, but it is also adaptive in that it can fill whatever portion of the machine's memory and processors are allotted to it. For example, carrying out a BATS-R-US simulation of the Heliosphere simultaneously with a BATS-R-US simulation of the Earth's magnetosphere, the two could coexist (and in the future could coordinate resources). Depending on relative importance, the slice of the machine for each could be chosen with corresponding choices in potential resolution. The initial data for CME's, which comes from observations of the sun, have relatively large error bars, so multiple BATS-R-US simulations could be run in parallel to explore possible outcomes of initial errors/uncertainties. The results would be a "percentage" forecast that brackets the error from initial conditions at the Sun.

Finally, we note that versions of BATS-R-US are publicly available: consult [1] for relevant information.

Acknowledgements

This research is partially funded by NASA ESS Cooperative Agreement NCCS5-146, part of the National Grand Challenge program, and by NSF grant 9318181.

References

1. “BATS-R-US: Milestone Release Notes”, hpcc.engin.umich.edu/HPCC/codes/index.html

2. D.L. DeZeeuw, T.I. Gombosi, A.F. Nagy, K.G. Powell, and J.G. Luhmann, “A new axisymmetric MHD model of the interaction of the solar wind with Venus”, *J. Geophys. Res.*, 101, 4, 547–4,556, 1996.

3. T.I. Gombosi, K.G. Powell, Q.F. Stout, E.S. Davidson, D.L. De Zeeuw, L.A. Fisk, C.P.T. Groth, T.J. Linde, H.G. Marshall, P.L. Roe, B. van Leer, “Multiscale modeling of heliospheric plasmas”, *High Performance Computing 1997*. Available at www.eecs.umich.edu/˜qstout/abs/HPC97.html

4. R.M. Hberli, T.I. Gombosi, M.R. Combi, D.L. DeZeeuw, and K.G. Powell, “Modeling of cometary X-rays caused by solar wind minor ions”, *Science*, 276, 939–942, 1997.

5. B. van Leer, “Towards the ultimate conservative difference scheme. V. A second order sequel to Godunov’s method”, *J. Computational Physics* 32, 1979.

6. T.J. Linde, T.I. Gombosi, P.L. Roe, K.G. Powell, and D.L. DeZeeuw, “The heliosphere in the magnetized local interstellar medium: Results of a 3D MHD simulation”, *J. Geophys. Res.*, 103, 1889–1904, 1998.

7. K.G. Powell, P.L. Roe, T.J. Linde, T.I. Gombosi and D.L. De Zeeuw, “A solution-adaptive upwind scheme for ideal MHD”, submitted to *J. Computational Physics*, 1998.

8. Q.F. Stout, D.L. DeZeeuw, T.I. Gombosi, C.P.T. Groth, H.G. Marshall, K.G. Powell, “Adaptive blocks: A high-performance data structure”, *Proceedings SC’97*, 1997. Available at www.eecs.umich.edu/˜qstout/abs/SC97.html.

Author Biographies

Quentin F. Stout is Professor of Computer Science. His research interests are in parallel computing, algorithms, adaptive statistical designs, discrete mathematics, and scientific programming. He is Co-Principal Investigator of the NASA HPCC ESS Computational Grand Challenge Investigator Team. He is also the director of the Center for Parallel Computing at the University of Michigan.

Darren L. De Zeeuw is Assistant Research Scientist at the Space Physics Research Laboratory. His interests involve the development and implementation of algorithms to solve the multidimensional Euler and MHD equations using octree-based unstructured Cartesian grids with adaptive refinement and multigrid convergence acceleration. He is one of the primary developers of the code described in this paper.

Tamas I. Gombosi is Professor of Space Science and Professor of Aerospace Engineering. He has extensive experience in solar system astrophysics. His research areas include generalized transport theory, modeling of planetary environments, heliospheric physics, and more recently, multiscale MHD modeling of solar system plasmas. He is Interdisciplinary Scientist of the Cassini/Huygens mission to Saturn and its moon, Titan. Professor Gombosi is Senior Editor of the Journal of Geophysical Research Space Physics. He is Program Director and Co-Principal Investigator of the NASA HPCC ESS Computational Grand Challlenge Investigator Team at the University of Michigan.

Clinton P.T. Groth is Assistant Research Scientist at the Space Physics Research Laboratory. His research interests involve generalized transport theory and the development of higher-order moment closures for the solution of the Boltzmann equation via advanced numerical methods with applications to both rarefied gaseous and anisotropic plasma flows. This research has led to a new hierarchy of moment closures with many desirable mathematical features that appears to offer improved modeling of transitional rarefied flows.

Hal G. Marshall recently became Manager of User Support for HPC at the Supercomputing Institute of the University of Minnesota. He has interests in scientific computing using massively parallel computers. He is one of the primary developers of the code described in this paper.

Kenneth G. Powell is Associate Professor of Aerospace Engineering. His research areas include: solution-adaptive schemes, genuinely multidimensional schemes for compressible flows, and, most recently, numerical methods for magnetohydrodynamics. He was a National Science Foundation Presidential Young Investigator from 1988-1994. He is Co-Principal Investigator of the NASA HPCC ESS Computational Grand Challlenge Investigator Team at the University of Michigan.