A PHYSICS-BASED SOFTWARE FRAMEWORK FOR SUN-EARTH CONNECTION MODELING

G. Toth[1,2], O. Volberg[1], Aaron J. Ridley[1], Tamas I. Gombosi[1], Darren DeZeeuw[1], Kenneth C. Hansen[1], David R. Chesney[1], Quentin F. Stout[1], Kenneth G. Powell[1], Kevin Kane[1], Robert Oehmke[1]

[1]*Center for Space Environment Modeling, The University of Michigan, Ann Arbor, MI 48109-2143, USA*
[2]*Department of Atomic Physics, Eotvos University, Budapest 1117, Hungary*

Abstract. The Space Weather Modeling Framework (SWMF) has been developed to provide NASA and the modeling community with a high-performance computational tool with "plug-and-play" capabilities to model the physics from the surface of the Sun to the upper atmosphere of the Earth. Its recently released working prototype includes five components for the following physics domains: Inner Heliosphere, Global Magnetosphere, Inner Magnetosphere, Ionosphere Electrodynamics and Upper Atmosphere. The SWMF is a structured collection of software building blocks to develop components for Sun-Earth system modeling, to couple them, and to assemble them into applications. A component is created from the user-supplied physics module by adding a wrapper, which provides the control functions, and coupling interface to perform the data exchange with other components. In its fully developed form, the SWMF will incorporate several more components (for example Solar Energetic Particles and Radiation Belt). It can also incorporate different versions − developed by the Sun-Earth modeling community − for each of the components. The SWMF Control Module is responsible for component registration, processor layout for each component, execution, and coupling schedules. We discuss the SWMF architecture, physics and implementation of component coupling, and results of some preliminary simulations that involve all five components.

Keywords. Space weather, software framework, numerical modeling, model coupling, component based software.

1. Introduction

The Sun-Earth system is a complex natural system of many different interconnecting elements. The solar wind transfers significant mass, momentum and energy to the magnetosphere, ionosphere, and upper atmosphere, and dramatically affects the physical processes in each of these physical domains. One example of such interaction effects is the population of highly energetic particles, which are produced in the inner magnetosphere. These particles precipitate into the upper atmosphere causing the aurora. A variety of complex electric currents in the magnetosphere, ionosphere, and upper atmosphere is another example of these interaction effects.

The various domains of the Sun-Earth system can be simulated with stand-alone models if simplifying assumptions are made about the interaction of the particular domain with the rest of the system. Sometimes the effects of the other domains can be taken into account by the use of satellite and ground based measurements. In other cases statistical and/or phenomenological models can be used. For the prediction of extreme space weather events, however, one must use first principles based physics models for all the involved domains and these models must be run and coupled in an efficient manner, so that the simulation can run faster than real-time. The ability to simulate and predict space weather phenomena is important for many applications, for instance the success of spacecraft missions and the reliability of satellite communication equipment. In extreme cases, the magnetic storms may have significant effects on the power grids used by millions of households.

As an illustrative example of modeling multiple domains of the Sun-Earth system with a highly integrated numerical code, we describe the evolution of the space plasma simulation program BATS-R-US developed at the University of Michigan. Originally BATSRUS was designed as a very efficient, massively parallel MHD code for space physics applications [*Powell et al*, 1999; *Gombosi et al*, 2001]. It is based on a block adaptive Cartesian grid with block based domain decomposition, and it employs the Message Passing Interface (MPI) standard for parallel execution. Later the code was coupled to an ionosphere model [*Ridley et al*, 2001], various upper atmosphere models [*Ridley et al*, 2003] and the inner magnetosphere model [*De Zeeuw et al*, 2003]. These couplings were done in a highly integrated manner resulting in a monolithic code, which makes it rather difficult to select an arbitrary subset of the various models, to replace one model with another one, to change the coupling schedules of the interacting models, and to run these models concurrently on parallel computers. Thus, although BATS-R-US is successfully used for the global MHD simulation of space weather [*Gombosi et al*, 2004], monolithic programs like it have their limitations.

The Center for Space Environment Modeling (CSEM) at the University of Michigan and its collaborators are building a Space Weather Modeling Framework (SWMF). The SWMF is designed to couple the models of the various physics domains in a flexible yet efficient manner, which makes the prediction of space weather feasible on massively parallel computers. Each model has its own dependent variables, a mathematical model in the form of equations of evolution, and a numerical scheme with an appropriate grid structure and temporal discretization. The physics domains may overlap each other or they can interact with each other through a boundary surface. The SWMF should be able to incorporate models from the community and couple them with minimal changes in the software of an individual model. In this paper we present the design and implementation of the first working prototype of the SWMF.

2. Architecture of the SWMF

The SWMF aims at providing a flexible and extensible software architecture for multi-component physics-based space weather simulations, as well as for various space physics applications [*G. Toth et al*, 2003*; Volberg et al*, 2003]. The main SWMF design goals were defined in *Volberg et al* [2002] as:

- Incorporate computational physics modules with only modest modification.
- Achieve good parallel performance in the coupling of the physics components.
- Allow physics components to interact with the SWMF as efficiently as possible.

One of the most important features of the SWMF is that it can incorporate different computational physics modules to model different domains of Sun-Earth system. Each module for a particular domain can be replaced with alternatives, and one can use only a subset of the modules if desired.

There are several a priori known problems, which need to be solved so that the heterogeneous computational models of the different domains of the Sun-Earth system can properly inter-operate. An incomplete list of these problems:

1. There are serial and parallel models.
2. An individual model is usually developed for stand-alone execution.
3. Input/output operations do not take into account potential conflicts with other models.
4. Models often do not have checkpoint and restart capabilities.
5. The majority of models are written in non-object oriented languages (e.g. Fortran 77 and Fortran 90), which means that data and procedure name conflicts can easily occur.
6. The efficient coupling of any arbitrary pair of parallel applications, each of them having its own grid structure and data decomposition, is not easily achieved.

There are several potential solutions that provide the necessary interoperability mechanism between parallel modules [*Edjlali et al*, 1997]. The most promising approach is to define a standard set of interface functions that every physics component must provide. In the SWMF a component is created from a physics module, for example BATSRUS, by making some minimal required changes in the module and by adding two relatively small units of code:

1. A wrapper, which provides the standard interface to control the physics module;
2. A coupling interface, to perform the data exchange with other components.

From a component software technology perspective, both the wrapper and coupling interface are component interfaces: the wrapper is an interface with the high-level Control Module (CON) of the framework, and the coupling interface is an interface with another component. As shown in Figure 1, the wrapper interface functions have standard names, which makes swapping between various versions of a component possible. Both the wrapper and the coupling interface are constructed from the building blocks provided by the framework.

The physics modules must comply with a minimum set of requirements before they are transformed into a component:

1. The parallelization mechanism (if any) should employ the MPI standard;
2. The module needs to be compiled as a library that could be linked to another executable;
3. The module should be portable to a specific combination of platforms and compilers, which include Linux workstations and NASA super-computers [*Volberg et al*, 2002];
4. The stand-alone module must successfully run a model test suite provided by the model developer on all the required platform/compiler combinations;
5. The module should read input from and write output to files which are in a subdirectory unique for the component;
6. A module should be implemented in Fortran 77 and/or Fortran 90;
7. A module should be supplied with appropriate documentation.

The SWMF requirements for a component are defined in terms of a set of methods to be implemented in the component wrapper. The methods enable the component to perform the following tasks:

1. being registered by the Control Module;
2. being initialized in parallel configuration;
3. accept and check input parameters obtained from the Control Module;
4. provide grid description to Control Module;
5. initialize for session execution;
6. execute one time step which cannot exceed a specified simulation time;
7. receive and provide data to another component via an appropriate coupler;
8. write its state into a restart file when requested;
9. finalize at the end of the execution.

The structure of a component and its interaction with the Control Module (CON) and another component are illustrated in Figure 1.

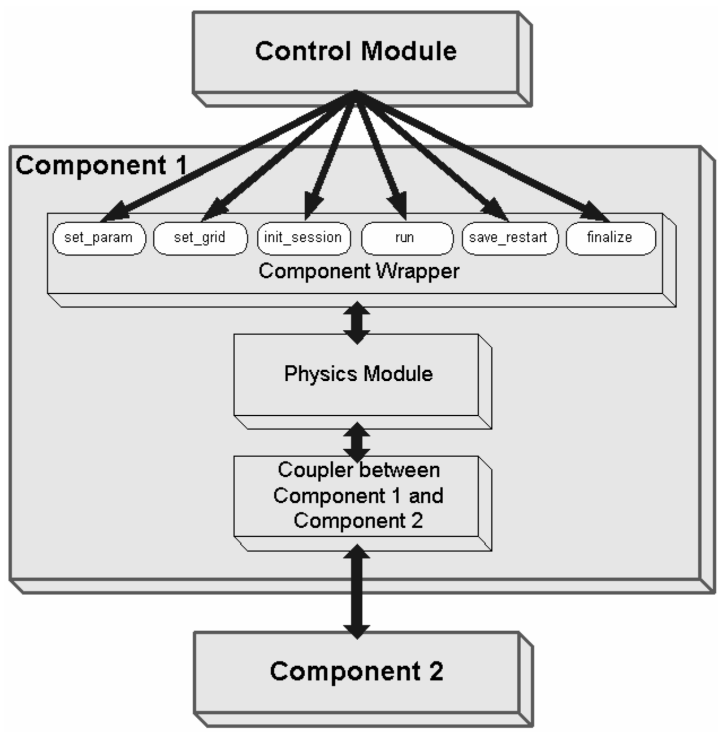

Figure 1. Integration of physics modules into the SWMF architecture

The framework's layered architecture is shown in Figure 2. The Framework Services consist of software units (classes) which implement component registration, session control, and input/output operations of initial parameters. The Infrastructure consists of utilities, which define physics constants and different coordinate systems, time and data conversion routines, time profiling routines and other lower level routines. The Superstructure Layer, Physics Module Layer, and Infrastructure Layer constitute the "sandwich-like" architecture similar to the Earth System Modeling Framework (ESMF) [*Hill et al,* 2004]. The SWMF will also contain a web-based Graphical User Interface, which is not part of the ESMF design.

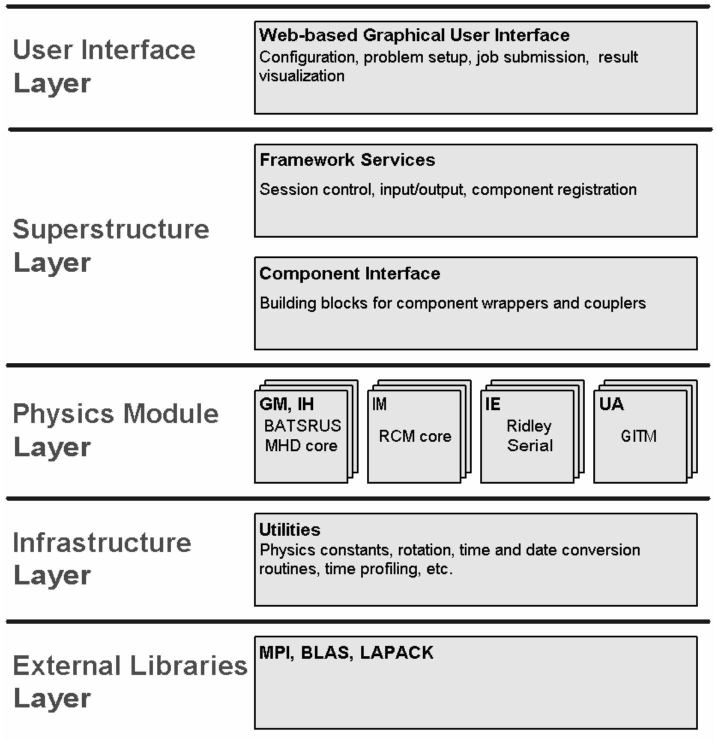

Figure 2. The layered hierarchy of the SWMF

```
ID    first  last  stride
#COMPONENTMAP
GM       0   999     2
IE       0     1     1
IH       1   999     2
IM      11    11     1
UA      12    15     1
#END
```

Figure 3. An example of the LAYOUT.in file

3. Execution of the SWMF

At the beginning of a run, the SWMF registers the components. The registration provides information about the component to CON, such as the name and the version of the component. In return, the control module assigns an MPI communication group to the component based on the processor layout defined in the LAYOUT.in file. An example of this file is shown in Figure 3. The first column identifies the components by their abbreviated names, while the rest of the columns define the ranks of the first and last processing elements (PE), and the PE stride. In the example shown in Figure 3, the Global Magnetosphere (GM) component runs on every even processor, the Ionospheric Electrodynamics (IE) component runs on the first two processors, the Inner Heliosphere (IH) component runs on every odd processor, the Inner Magnetosphere (IM) component runs on processor 11, and the Upper Atmpsphere (UA) component runs on 4 processors from ranks 12 to 15. Only registered components can be used in the run.

The execution is completed in sessions. In each session the parameters of the framework and the components can be changed. The parameters are read from the PARAM.in file, which may contain further included parameter files. These parameters are read and broadcast by CON and the component specific parameters are sent to the components for reading and checking. The CON related parameters define the initial time, coupling schedules, frequency of saving restart files, final simulation time of the session, and other information which is not restricted to a single component. At the beginning of each session the components are initialized and the interacting components are coupled together for the first time. The SWMF currently contains two different session execution models.

The sequential execution model of the session is based on the BATS-R-US module, and is backward compatible with it. In this model the components are synchronized at every time step, so typically only one component is executing (possibly on many processors) at any given time. The progress of the simulation is controlled by the GM module, and the coupling patterns and schedules are mostly predetermined.

In the concurrent execution model, the components communicate only when necessary. This is possible because the coupling times are known in advance. The components advance to the coupling time and only the processors involved in the coupling need to communicate with each other. In this execution model all components are 'equal', any physically meaningful subset can be selected for the run, and their coupling schedules are determined by the parameters given in the PARAM.in file. The concurrent execution model allows execution of more than one component on the same PE, in which case the components advance in a time-shared manner. The possibility of dead-locks is carefully avoided.

The coupling of the components is realized either with plain MPI calls, which are specifically designed for each pair of interacting components, or via the general SWMF coupling toolkit. The toolkit can couple components based on the following types of distributed grids:

- 2D or 3D block adaptive grid
- 2D or 3D structured grid

Structured grids include uniform and non-uniform spherical and Cartesian grids.

The toolkit obtains the grid descriptors from the components at the beginning of the run. The grid descriptor defines the geometry and parallel distribution of the grid. At the time of coupling the receiving component requests a number of data values at specified locations of the receiving grid (for example all grid points at one of the boundaries). The geometric locations are transformed, sometimes mapped, to the grid of the provider component. Based on the grid descriptor of the provider component, the data values are interpolated to the requested locations and sent back to the requesting component. The interpolation weights and the MPI communication patterns are calculated in advance and saved into a 'router' for sake of efficiency. The router is reused as much as possible in subsequent couplings. The routers are updated only if one of the grids has changed (e.g. due to grid adaptation) or when the mapping between the two components has changed (e.g. due to the rotation of one grid relative to the other). In certain cases the coupling is achieved via an intermediate grid, which is stored by the receiving component, but its structure is based on the providing component. The intermediate grid can be described to CON the same way as the base grid of the receiving component.

As specified in the PARAM.in file, CON instructs the components to save their current state into restart files periodically. This makes possible the seamless continuation of a run from a given point of the simulation. Check-point restart is an essential feature of a robust, user-friendly, and fault-tolerant software design.

At the end of the last session each component finalizes its state. This involves writing out final plot files, closing log files, and printing performance and error reports. After the components have finalized, CON also finalizes and stops the execution.

4. The SWMF Prototype

The recently released working prototype of the SWMF includes components for five physics domains: Global Magnetosphere (GM), Inner Heliosphere (IH), Ionosphere Electrodynamics (IE), Upper Atmosphere (UA) and Inner Magnetosphere (IM):

1. The GM and IH components are based on the University of Michigan's BATS-R-US MHD module. The highly parallel BATS-R-US code uses a 3D block-adaptive Cartesian grid.
2. The IM component is the Rice Convection Model (RCM) developed at Rice University. This serial module uses a 2D non-uniform spherical grid.
3. The IE component is a 2D spherical electric potential solver developed at the University of Michigan. It can run on 1 or 2 processors since the northern and southern hemispheres can be solved in parallel.
4. The UA component is the Global Ionosphere Thermosphere Model (GITM) recently developed at the University of Michigan as a fully parallel 3D spherical model.

A detailed description of the 5 components and their coupling with each other exceeds the limitations of this paper. Below we briefly outline the coupling mechanisms so that the complexity and variability of the couplings may be better appreciated.

4.1 Coupling the Inner Heliosphere to the Global Magnetosphere

The GM is driven by the solar wind flow. One method of accomplishing this is to use an upstream solar wind monitor to feed the upstream boundary conditions of the GM. Alternatively, the GM can be embedded into a full MHD IH model that provides all relevant flow conditions to the GM at the necessary time and location. The GM obtains a temporally and spatially varying set of data at its inflow boundary, which consists of the solar wind density, pressure, velocity, and magnetic field.

4.2 Coupling the Global Magnetosphere and the Inner Magnetosphere

This coupling is challenging, as the IM needs the magnetic field line flux tube volumes and the average density and pressure in the flux tubes connected to its 2D spherical grid points. This requires the integration along many magnetic field lines in GM. In turn GM needs to know where each of its 3D grid points are mapped onto the ionosphere along the magnetic field lines, so that the total pressure calculated by IM can be applied in GM. A parallel method has been developed to accomplish these steps in an efficient manner.

4.3 Coupling the Global Magnetosphere and the Ionosphere Electrodynamics

The IE requires the field aligned currents on its 2D spherical grid. The currents are calculated in GM at an appropriate radial distance (e.g. 3.5 Earth radii), and mapped along the dipole field lines of the planet to the ionosphere. In return, IE provides the electric potential on its grid. This potential is mapped back to the inner boundary of GM (typically at 2.5 Earth radii). The potential is used to calculate the electric field and the corresponding plasma velocities, which are used as the inner boundary condition for GM.

4.4 Coupling the Inner Magnetosphere and the Ionosphere Electrodynamics

This is a one-way coupling. IE provides the radial current and the electric potential for the 2D spherical grid of IM. The coupling is complicated by the non-uniformity of the IM grid.

4.5 Coupling the Ionosphere Electrodynamics and the Upper Atmosphere

The upper atmosphere model provides the Pederson and Hall conductivities and field aligned currents (in addition to the field aligned currents obtained from GM) to the IE. On the other hand, the electric potential calculated by IE is mapped to the upper atmosphere model in each grid cell along the magnetic field lines. The gradient of the potential provides the electric field which is used to drive the ion motion.

5. The First Simulation Results

We present some preliminary simulations which involve all five components of the SWMF prototype. This simulation is mainly used as a test bed for the development of the framework, thus the grid resolution is relatively coarse and the setup of the simulation is relatively simple. The first part of the simulation involves the IH, GM and IE components only. Both the IH and GM grids contain about a quarter million cells. For GM the cell sizes range from 0.5 to 4 Earth radii, while for IH they range from 0.125 to 16 Solar radii. The IE component has a 2D spherical grid with an approximately 1.5 degree resolution. All 3 components are run in a steady-state mode, and there is no coupling between IH and GM. The purpose of this start up is to obtain a physically reasonable steady-state solution in a relatively short time, but the steady state solutions do not match at the IH-GM boundary (for a real production run this would not be the case). The steady state solution obtained by the IH component is shown in Figure 4.

The simulation is continued in a time-accurate mode for an hour of physical time. All five components are involved and they are fully coupled in this part of the simulation. Since the steady state obtained with GM and IH do not match at the inflow boundary of GM, the magnetosphere undergoes a severe compression due to the incoming density jump in the solar wind. Figure 5 also shows how the orientation of the upwind magnetic field changes. Meanwhile the interaction of the Global Magnetosphere with the Inner Magnetosphere model causes a gradual increase of the plasma pressure on the night side of the Earth as shown in Figure 6. The compression of the magnetotail also increases the pressure, but the increase confined to the closed field lines is characteristic to the influence of the IM component. Another consequence of the IM-GM coupling is the strengthening of region 2 currents. These currents are used by the Ionosphere Electrodynamics model to solve for the electric potential (Figure 7). Finally we show the electron density and velocity field obtained by the Upper Atmosphere model in Figure 8. The UA component is strongly dependent on the conductance and electric potential calculated by the IE component. The IM component uses a non-uniform 78 by 48 2D spherical grid, while the UA component runs on a 3D spherical grid with 5 degrees angular resolution and 25 altitude levels.

In the future we plan to apply the SWMF to more realistic problems, which involve the initiation and propagation of Coronal Mass Ejection from the Sun to the Earth. Such a simulation will require finer grid resolution and much longer time evolution.

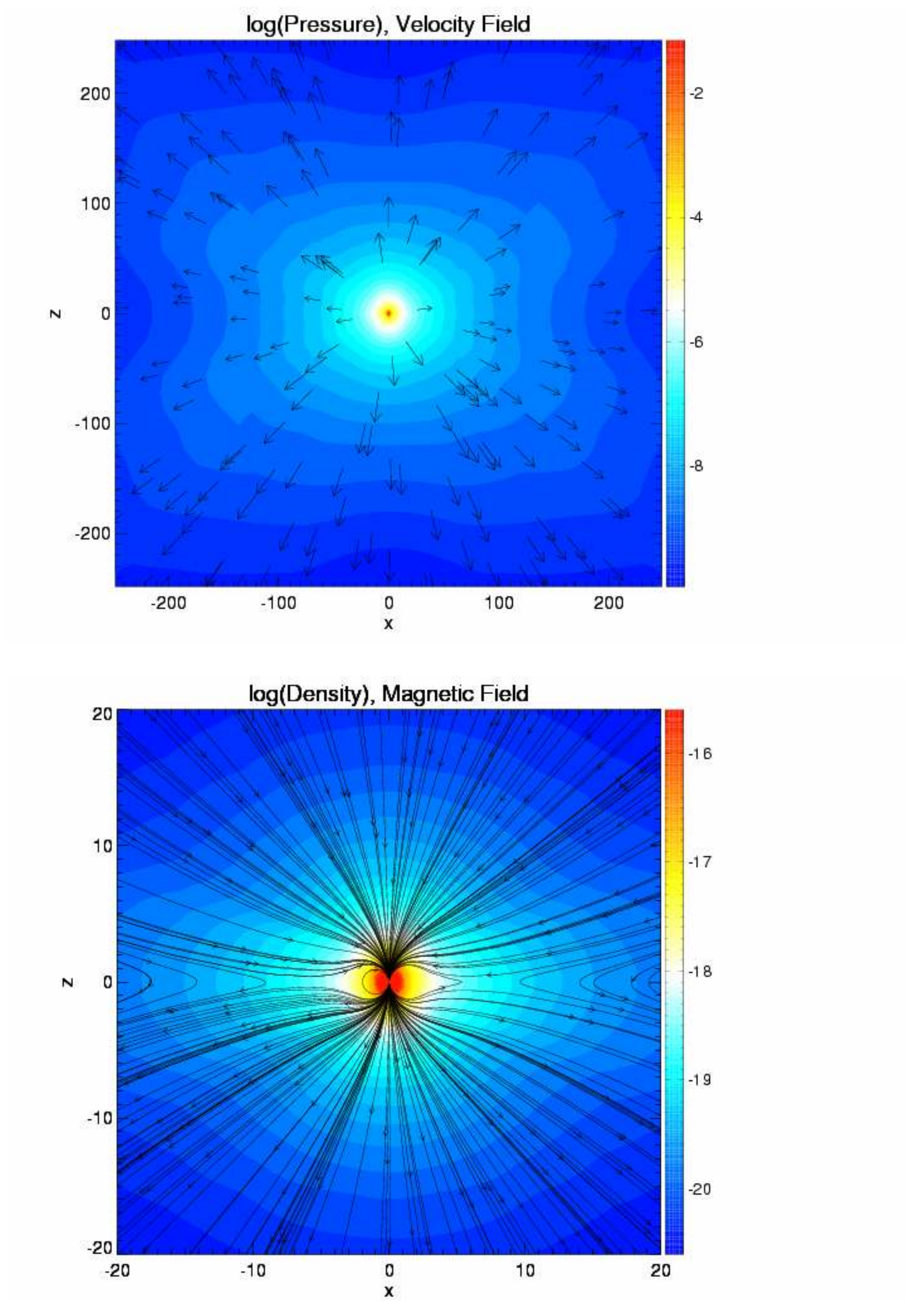

Figure 4. The steady state solution obtained by the Inner Heliosphere model. The top panel depicts the overall pressure and velocity fields out to 256 Solar radii, while the bottom panel shows the magnetic field and density in the vicinity of the Sun.

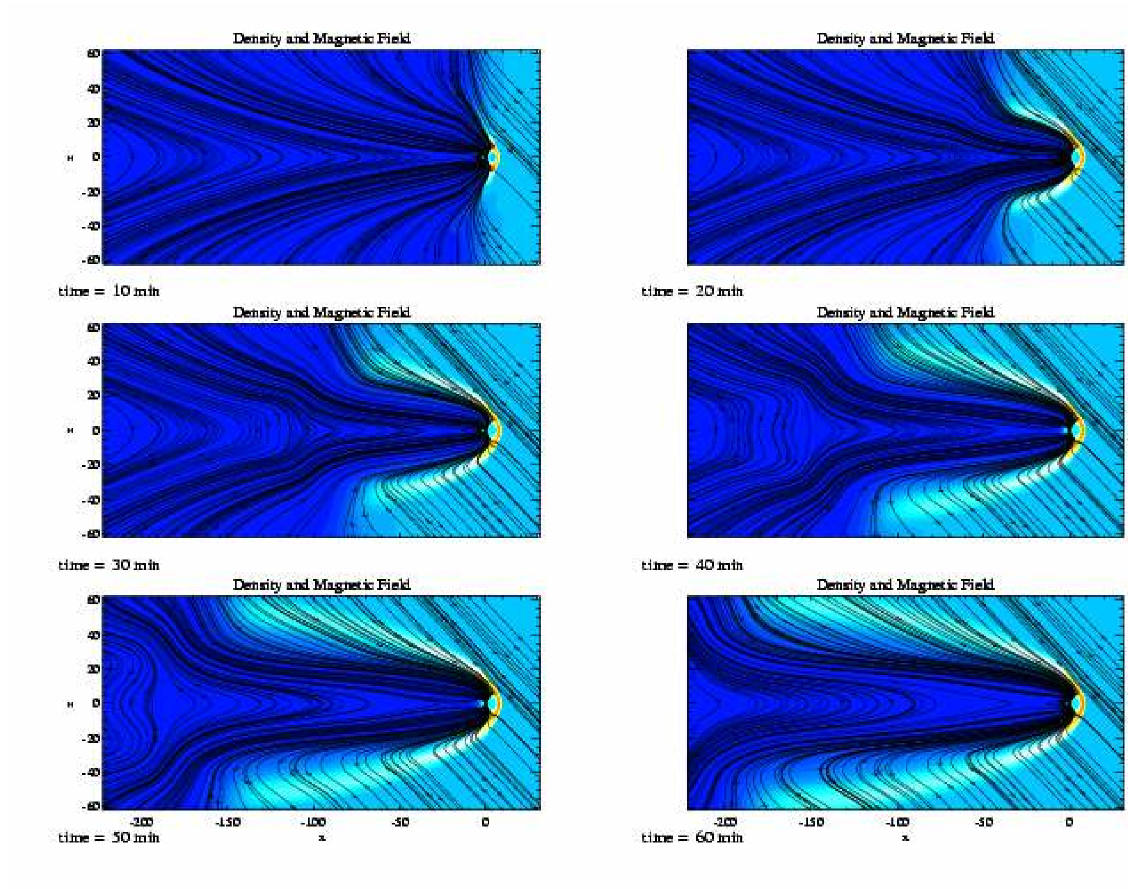

Figure 5. The evolution of the Global Magnetosphere due to the sudden changes in the solar wind density and magnetic field. The units on the axes are in Earth radii.

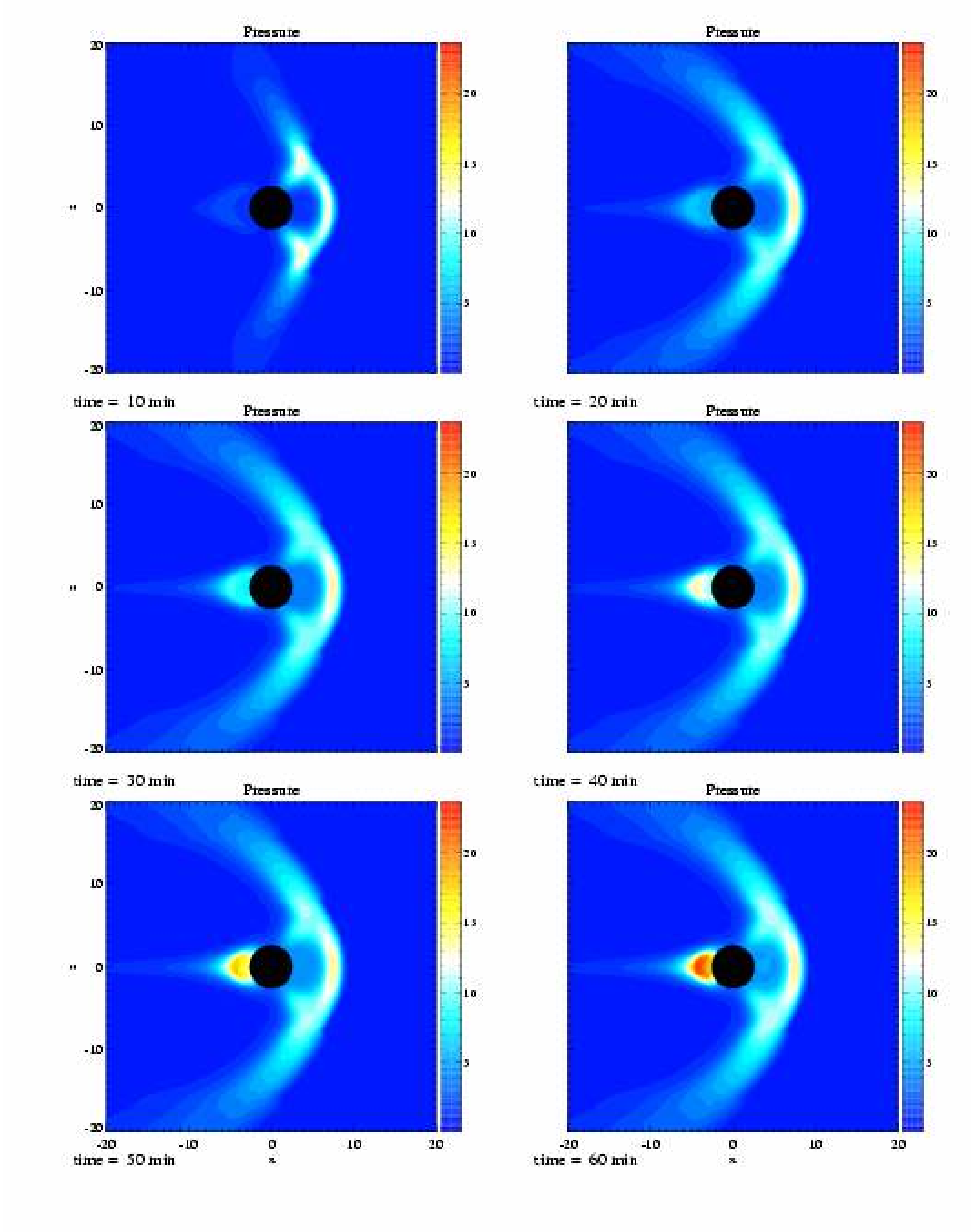

Figure 6. The interaction of GM with IM increases the pressure on the night side of the Earth.

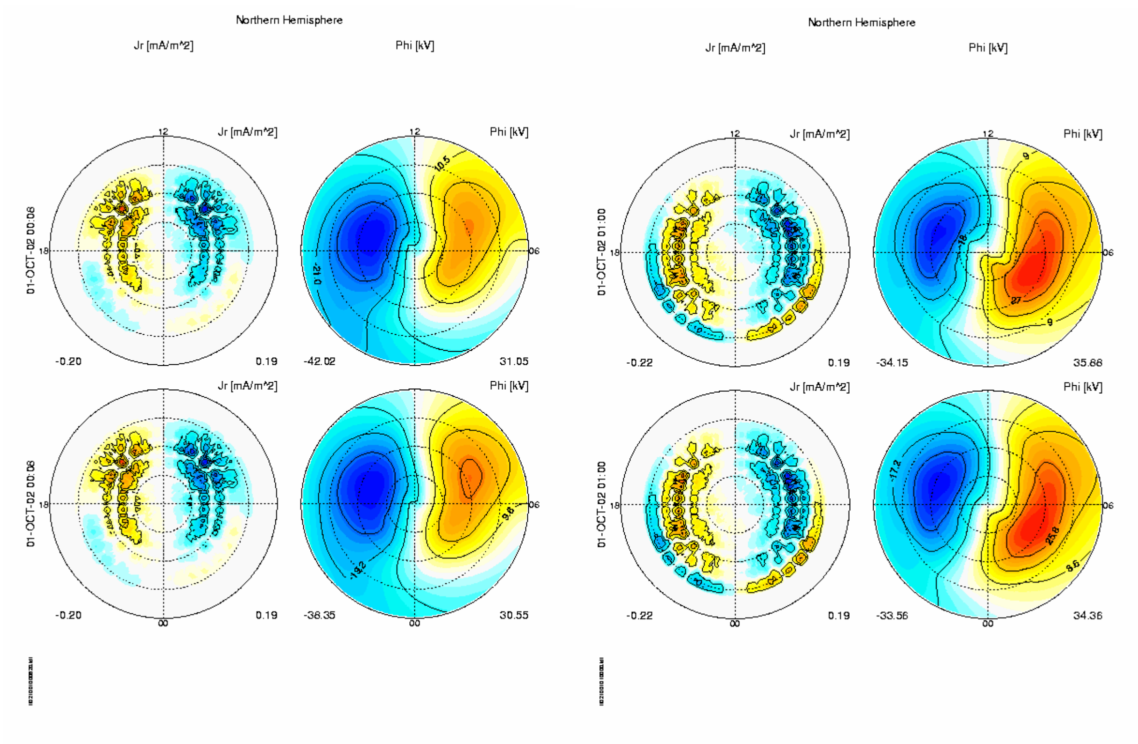

Figure 7. The field aligned current and the electric potential at the beginning (left panel) and at the end (right panel) of the one hour simulation. Note the strengthening of region 2 currents.

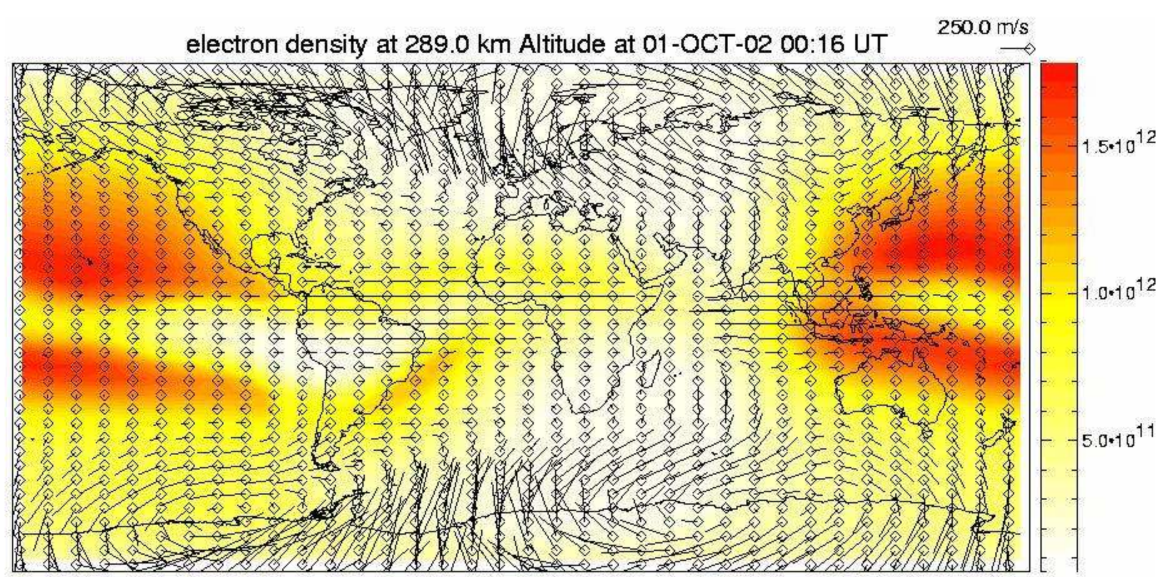

Figure 8. The electron density and velocity field obtained by the Upper Atmosphere model.

6. Conclusions and Future Work

The first results of our work are summarized as follows:

1. In its fully developed form, the SWMF comprises a series of interoperating models of physics domains, ranging from the surface of the Sun to the upper atmosphere of the Earth. In its current form the SWMF links together five physics models: Inner Heliosphere, Global Magnetosphere, Inner Magnetosphere, Ionosphere Electrodynamics and Upper Atmosphere.
2. The SWMF contains utilities and data structures for creating model components and coupling them.
3. A component is created from the user-supplied physics code by adding a wrapper, which provides the control functions, and a coupling interface which performs the data exchange with other components.
4. The SWMF contains a Control Module, which controls initialization and execution of the components and is responsible for component registration, processor layout for each component, and coupling schedules.
5. The framework allows a subset of the physics components to execute and can incorporate several different models for the same physics domain.
6. The SWMF parallel communications are based on the MPI standard. In its current implementation the SWMF builds a single executable.

The SWMF is not only more flexible, but in some cases more efficient, than a highly integrated code. For example, when only the GM and IE components are used, the performance depends to a large extent on the execution model. The SWMF allows the concurrent execution of the two components. While IE solves for the electric potential on 2 processors, the GM code can proceed with the MHD simulation on the rest of the processors. As a result, the IE solution is applied to the inner boundary of GM with a small time shift, but this is an acceptable approximation. In this concurrent execution model, the SWMF can run almost twice as fast as the serial model with less flexible coupling schedules.

We will add two additional components for new physics domains in the next scheduled release: the Radiation Belt (RB) model developed at Rice University by A. Chan, D. Wolf and Bin Yu) and the Solar Energetic Particle (SP) model developed at the University of Arizona by J. Kota. The coupled GM and IM components will provide the time variations of the electric and magnetic fields in the region occupied by the outer radiation belt. The RB component will use those fields to estimate the time evolution of the outer belts, with particular emphasis on the relativistic electrons that are crucial for space-weather modeling.

The SP component will calculate both the acceleration of charged particles to high energies and the subsequent transport of these particles along the interplanetary magnetic field lines connecting the Earth to the site of acceleration, i.e. the strong shock driven by the CME. The code is an implicit scheme that provides numerical solutions to the Fokker-Planck equation including acceleration, focusing, convection, and random scattering in pitch angle. The code is designed so that the geometry of the magnetic field and the position and strength of the shock are constantly updated as the shock evolves. Multiple shocks are also allowed. The code will compute the predicted time profiles, energy spectra, and pitch-angle distribution of solar energetic particles events at the Earth.

Acknowledgments

We would like to express our gratitude to the governors of the Conference on Sun Earth Connection, A. T. Y. Lui and Y. Kamide. We also wish to thank the National Center for Atmospheric Research (NCAR), the NASA Goddard Space Flight Center (GSFC), Rice University and the University of Arizona for collaboration on this project. Our special thanks to J. Kota from the University of Arizona, A. Chan, D. Wolf, S. Sazykin and Bin Yu from Rice University, Cecelia Deluca from NCAR, J. Fisher from NASA GSFC, J. W. Larson and R. Jacob from Argonne National Laboratory. The SWMF project is funded by the NASA Earth Science Technology Office (the NASA CAN NCC5-614 grant). G. Toth is partially supported by the Hungarian Science Foundation (OTKA grant T047042).

References

De Zeeuw, D. L., S. Sazykin, A. Ridley, G. Toth, T. I. Gombosi, K. G. Powell, D. Wolf, Inner Magnetosphere Simulations - Coupling the Michigan MHD Model with the Rice Convection Model, *Fall AGU Meeting*, San Francisco, 2003.

Edjlali, G., A. Sussman, and J. Saltz, Interoperability of Data Parallel Runtime Libraries. In *Proceedings of the Eleventh International Parallel Processing Symposium*, IEEE Computer Society Press, 1997.

Gombosi, T. I., D. L. De Zeeuw, C. P. Groth, K. G. Powell, C. R. Clauer, and P. Song, From Sun to Earth: Multiscale MHD Simulations of Space Weather, in *Space Weather*, edited by P. Song, H. J. Singer, and G. L. Siscoe, vol. 125, pp. 169–176, AGU, 2001.

Gombosi T. I., K. G. Powell, D. L. De Zeeuw, R. Clauer, K. C. Hansen, W. B. Manchester, A. Ridley, I. I. Roussev, I. V. Sokolov, Q. F. Stout, G. Toth, Solution-Adaptive Magnetohydrodynamics for Space Plasmas: Sun-to-Earth Simulations, *Computing in Science and Engineering, Frontiers of Simulation*, March/April, p. 14-35, 2004.

Hill, C., C. DeLuca, V. Balaji, M. Suarez, A. da Silva, and the ESMF Joint Specification Team, The Architecture of the Earth System Modeling Framework, *Computing in Science and Engineering,* Volume 6, Number 1, 2004.

Powell, K.G., P.L. Roe, T.J. Linde, T.I. Gombosi, and D.L. De Zeeuw, A solution-adaptive upwind scheme for ideal magnetohydrodynamics, *J. Comp. Phys.,* 154, 284, 1999.

Ridley, A. J., D.L. De Zeeuw, T. I. Gombosi, and K.G. Powell, Using steady-state MHD results to predict the global state of the magnetosphere-ionosphere system, *J. Geophys. Res.,* 106, 30,067, 2001.

Ridley, A. J., T. I. Gombosi, D. L. De Zeeuw, C. R. Clauer, and A. D. Richmond, Ionospheric control of the magnetospheric configuration: Neutral winds, *J. Geophys. Res.*, 108(A8), 1328, 2003.

Toth, G., O. Volberg, A. Ridley, Space Weather Modeling Framework Manual, Code version 1.0, *Center for Space Environment Modeling, the University of Michigan,* Ann Arbor, Michigan, 2003.

Volberg, O., D. R. Chesney, D. L. De Zeeuw, K. C. Hansen, K. Kane, R. Oehmke, A.J. Ridley, I.V. Sokolov, G. Toth, T. Weymouth, Space Weather Modeling Framework: Design Policy for Interoperability, http://csem.engin.umich.edu, *Center for Space Environment Modeling, The University of Michigan,* Ann Arbor, Michigan, 2002.

Volberg, O., G. Toth, I. V. Sokolov, A. J. Ridley, T. I. Gombosi, D. L. De Zeeuw, K. C. Hansen, D. R. Chesney, Q. F. Stout, K. G. Powell, K. Kane, R. C. Oehmke, Doing it the SWMF Way: From Separate Space Physics Simulation Programs to The Framework for Space Weather Simulation, *Fall AGU Meeting*, San Francisco, 2003.