**Project: Michigan Space Weather Modeling Framework**

**Solicitation RA-133W-14-RP-0132**

**Contents by Milestones**

[Milestone 4.1.1](#_Installing_and_Using) (January 30, 2015) Documentation of Algorithms

[Milestone 4.1.5](#_Sensitivity_to_Input) (May 15, 2015) Sensitivity to Input Parameters

[Milestone 4.2.1](#_Model_Products_and) (March 31, 2016) Model Products and Displays

[Milestone 4.2.2](#_Solar_Wind_Propagation) (April 30, 2016) MHD Propagation of Solar Wind from L1

[Milestone 4.2.5](#_Automated_Validation_Tools_1) (March 31, 2016) Automated Validation Tools and Sensitivity Analyses

[Milestone 4.3.1](#_Radiation_Belt_Environment) (March 31, 2017) Model Products and Displays

Table of Contents

1 Installing and Using the Geospace Model 3

1.1 Code version 3

1.2 Installation, Configuration, Compilation 4

1.2.1 Configuring out unused components 4

1.2.2 Installation, Configuration and Compilation 4

1.2.3 Maintenance 6

1.3 Setting up the run directory 6

1.4 Running the Geospace model 9

1.5 Regular Restart of the Geospace model 10

1.6 Restarting the Model due to Sudden Increase of Solar Wind Speed 11

2 Solar Wind Propagation 12

2.1 Ballistic propagation 12

2.2 MHD propagation 13

2.3 Comparison of ballistic and MHD propagation 14

2.4 Comparison of ballistic and MHD propagation with WIND satellite data 16

3 Sensitivity to Input Parameters 18

3.1 Explicit versus Implicit Time Stepping Schemes 18

3.2 Physical Parameters 19

3.2.1 BATS-R-US Parameters 19

3.2.2 Ridley Ionosphere Model Parameters 21

3.2.3 Rice Convection Model Parameters 22

3.3 Results from our work prior to the SWPC challenge 22

4 Model Products and Displays 25

4.1 Magnetometer Stations and Grids 25

4.2 Ring of Geosynchronous Values 27

4.3 Virtual Geomagnetic Indices 30

4.3.1 Calculation and Calibration of K Indices 31

4.3.2 Calculation of AE Indices 32

5 Automated Validation Tools and Sensitivity Analyses 33

5.1 Automated Validation Tools 33

5.2 Using Ballistic vs. MHD Propagated Solar Wind for the Geospace Model 34

**Documentation of Algorithms**

Milestone 4.1.1

# Installing and Using the Geospace Model

## Code version

The Geospace model is part of the Space Weather Modeling Framework (SWMF). The SWMF Git repository is stored on GitLab server of the University of Michigan. We have provided read-only access to SWPC personnel. See the

SWMF/doc/GitLab\_instructions.docx

for detailed instructions on how to access the GitLab repository. To clone the SWMF (with no history and only the four models used at SWPC) do the following:

gitlabclone SWMF --depth=1

cd SWMF

./Config.pl -clone=BATSRUS,RCM2,Ridley\_serial,RBE

The models listed after -clone= will be cloned from the GitLab repositories hosted on the UM GitLab server. Cloning a subset of models reduces the size of the source code substantially. A further reduction can be achieved by using the Configure.pl script to configure out SWMF components that are not used in the Geospace model running at SWPC and by removing the .git directories:

Scripts/Configure.pl # Create Build directory with GM,IM,IE,RB components only

cd Build

In addition to the source code, the RBE model also needs some data files. These can be obtained by executing the following commands inside the SWMF (or Build) directory:

gitlabclone SWMF\_data --depth=1 # this may take a while

cd SWMF\_data

rm -rf .git GM IM PT PW SC SP UA # remove extra directories

cd ..

This will create a directory tree under

SWMF\_data/RB/RBE/data

that contains the RBE data files. At this point the SWMF (or Build) directory tree can be renamed or moved to another machine. The Build/ uses only 122MB disk space.

## Installation, Configuration, Compilation

### Installation, Configuration and Compilation

The SWMF needs to be installed first with the appropriate Fortran compiler selected. Since currently SWPC uses the Intel Fortran compiler on a Cray machine, switch to the Intel programming environment with

module swap PrgEnv-cray PrgEnv-intel

and install the code with

./Config.pl –install=BATSRUS,RCM2,Ridley\_serial,RBE -compiler=ifortftn,intelcc

Note that the models cloned should be listed after -install=, otherwise the installation will attempt to clone all the missing models from herot.

If a new version of the SWMF and/or compiler is used, it is best to do a short test. Assuming that the code can run with the “mpiexec” command on the machine,

make -j test\_swpc NP=4

will execute a short test on 4 cores and compilation will be done in parallel due to the -j flag. The expected output is three empty difference files for the GM, IE and RB components:

-rw-r--r-- 1 gtoth hpcc 0 Jul 30 13:53 test\_swpc\_gm.diff

-rw-r--r-- 1 gtoth hpcc 0 Jul 30 13:53 test\_swpc\_ie.diff

-rw-r--r-- 1 gtoth hpcc 0 Jul 30 13:53 test\_swpc\_rb.diff

If the test fails then one or both of these files will contain information about the failure. The reference solution files are stored in the Param/SWPC/TestOutput/ directory.

Running test\_swpc takes care of configuring the SWMF the way it is used for the Geospace model. In fact

make -j test\_swpc\_compile

can be used to configure and compile the SWMF for the Geospace model. But it is probably better to describe what is happening under the hoods, so that the configuration of the SWMF for the Geospace model is clear.

The SWMF has to be configured with four components for the Geospace Model: the Global Magnetosphere model (BATS-R-US) in the GM/BATSRUS directory, the Inner Magnetosphere model (RCM) in the IM/RCM2 directory, the Inner Electrodynamics model (RIM) in the IE/Ridley\_serial directory and the Radiation Belt Environment (RBE) model in the RB/RBE directory:

./Config.pl –v=Empty,GM/BATSRUS,IM/RCM2,IE/Ridley\_serial,RB/RBE

The first “Empty” value means that all other components are switched off first (in case the full SWMF is used and the other components were configured already) and then the three specified components are switched on. BATS-R-US has many options that can be configured. The most important are the grid and the user and equation modules. These can be configured as

./Config.pl -g=GM:8,8,8,400,1 -o=GM:u=Default,e=Mhd

The –g flag sets the size of the grid blocks to 8x8x8 cells, the maximum number of grid blocks per processor to 400, and 1 of these per processor can be advanced with the part implicit time stepping scheme (setting the last value to 1 signals that the part implicit time stepping will not be used, which speeds up the compilation by switching off optimization for this complicated module). The u= flag sets the user module to Default (nothing special) and the e= flag sets the equation module to Mhd for GM/BATSRUS. There are many other options that can be configured for each model, as can be seen by typing e.g.

GM/BATSRUS/Config.pl –h

but the default settings are appropriate for the Geospace model. The current configuration is shown if Config.pl is run without arguments.

The optimization level of the compiler affects execution speed, but also accuracy and reliability. We have been using the –O2 optimization level for many years with earlier versions of the ifort compiler, because the –O3 did not work properly.

We recommend performing comprehensive testing whenever the compiler version and/or the optimization level are changed. As a minimum, test\_swpc should be rerun.

After the configuration is complete, the SWMF.exe code can be compiled with

make –j SWMF

where the optional –j speeds up the compilation by using multiple cores in parallel. Some of the output files may require post-processing, so it is best to compile the post processing code PostIDL.exe with

make PIDL

After successful compilation the executables will be in the bin/ directory

bin/SWMF.exe

bin/PostIDL.exe

If the source files are modified (e.g. a new feature is installed) on herot, the code can be updated from the main SWMF directory (but not from Build!) with

gitall pull

After an update, the code can be recompiled without cleaning the existing object files. The Build can be recreated with Configure.pl.

If necessary (e.g. to change the optimization level or to test a different compiler version), the compiled executable and object files can be removed with

make clean

prior to a recompilation. The code can be uninstalled (e.g. to move it to a different directory) with

./Config.pl –uninstall

Note that uninstall preserves the cloned repositories (share, util, GM/BATSRUS… etc).

### Maintenance

The SWMF is maintained at the University of Michigan.

## Setting up the run directory

The SWMF in general (and the Geospace model in particular) runs in a directory that is dedicated for the run and separate from the Git distribution. This directory contains several files and subdirectories, so it has to be created with the SWMF (after the code is configured) with

make rundir

The resulting run/ directory can now be renamed or moved into a different location if desired. Note, however, that the directory contains symbolic links to the executables. One can replace the links with copies of the executables if desired, e.g.

rm –f run/SWMF.exe; cp bin/SWMF.exe run/

rm –f run/GM/PostIDL.exe; cp bin/PostIDL.exe run/GM/

There are a number of other items in the run directory that are potentially useful in other applications, but are currently not used by the SWPC Geospace model. These can be removed:

rm –f run/Param run/GM/Param

If the run directory is moved and the links are removed, it may be a good idea to add some file into it that indicates where the corresponding source code can be found. For sake of simplicity we keep using “run/” as the name of the run directory.

There are a number of input files that need to be put into the run directory. One of these has a fixed name and it is required for any run with the SWMF:

run/PARAM.in

The PARAM.in file is different for initial startup and continuation of the run using restart files. The initial startup should be done only once and the recommended file is in

Param/SWPC/PARAM.in\_SWPC\_v2\_init

that should be copied into run/PARAM.in for initial startup. For restarted runs, which will be all the runs except for the very first one, the

Param/SWPC/PARAM.in\_SWPC\_v2\_restart

file should be used.

The #COMPONENTMAP command in the PARAM.in file specifies which component runs on which processor. This may be customized for best performance.

There are additional files that are needed for running the Geospace model in operational mode. The names of these files are defined in the PARAM.in file, so their names can be modified. For sake of simplicity here we use the customary names to refer to the various files. The

IMF.dat

file contains the solar wind and interplanetary magnetic field data propagated to the inflow boundary of the global magnetosphere domain (typically at x=32Re in the GSM coordinate system). The IMF file is a simple ASCII file that should have the following format:

#START

2006 12 14 7 0 0 0 5.00 -3.01 -0.56 -585.89 -11.35 1.73 1.39 171087.25 comments

2006 12 14 7 1 0 0 -2.30 -2.82 -0.82 -583.52 -12.21 1.25 1.43 172988.88 or other

2006 12 14 7 2 0 0 -2.01 -2.77 -0.77 -580.37 -13.88 -1.11 1.56 172606.37 columns

…

where the “#START” indicates the start of the solar wind data containing 15 space separated columns per line. The first 7 columns are integers containing the date and time from year down to milliseconds in coordinated universal time, UTC. The next three columns contain the 3 components of the interplanetary magnetic field BSW in nT, the next 3 columns are the 3 components of the velocity vSW in km/s, the next column is the number density nSW in cm-3, and the last column is the ion temperature TSW in K. The vector quantities are in the GSM coordinate system by default (this can be modified as described in the GM/BATSRUS/PARAM.XML file for the #SOLARWINDFILE command). Since BATS-R-US runs in single fluid MHD mode in the Geospace model using protons and negligible electron temperature, the mass density will be the number density times the proton mass, and the thermal pressure will be pSW = nSWkTSW. **Important notes:**

1. The X component of the magnetic field should not be set to zero (CCMC tends to do that).
2. For proper linear interpolation in time the file should contain at least two data rows.
3. The time series should be monotonically increasing, but it does not have to be uniformly spaced.
4. There can be additional characters at the end of the lines as long as the 7 integers and 8 real numbers are present and properly separated with spaces.

The IMF file can be continuously updated with new data lines if the PARAM.in file contains the newly added command

#REFRESHSOLARWINDFILE

T DoReadAgain

The effect of this command is that the code stops running when the simulation reaches the time of the final data point and waits until the solar wind file gets updated. The update can be a simple append to the end, or it can be a complete replacement of the file too. The code ignores incomplete lines in the solar wind file, so this is a robust feature.

Another input file contains the coordinates of the magnetometer stations, an example can be found in

Param/SWPC/magin\_GEM.dat

There can be several input files containing satellite trajectories so that synthetic in-situ data can be extracted from the simulation. The satellite trajectory files should be produced in advance (unlike the solar wind file) and read at the beginning of the simulation. An example for defining several trajectory files is provided in

Param/SWPC/SATELLITES.in

that can be copied into the run directory, modified as needed, and included into the PARAM.in file with the

#INCLUDE

SATELLITES.in

command. See the description of the #SATELLITEFILE command in GM/BATSRUS/PARAM.XML for more detail.

## Running the Geospace model

There are some important differences between running the Geospace model in *research mode* (which includes testing and validation studies as well) and in *real-time or operational mode*. In research mode the input is prepared, the code is typically run only once and the output is post-processed and analyzed after the run finished. In real-time (operations) mode the inputs are updated while the code is running continuously and the output is also post-processed and used at the same time. We have already developed most of the tools needed for real-time mode, but some of these can and will be further optimized and customized as part of the collaboration between SWPC and UofM.

In operational mode the first step is to create an initial state. The run/PARAM.in file should be based on the Param/SWPC/PARAM.in\_SWPC\_init file with the following modifications:

* The #STARTTIME should be set to the time (in UTC) when the time dependent simulation will be started.
* The #ENDTIME should be set to the time when the full restart is planned.
* The #CPUTIMEMAX command should be removed (or the maximum time should be set to a value that is consistent with the planned periodic daily restarts, in seconds).
* The F10.7 flux should be set to the current (or very recent) value in the #IONOSPHERE command.

In addition, the solar wind data file (IMF.dat) should contain at least 2 data lines with the current (or very recent) solar wind conditions. Once the input is all set, the run can be started as

mpiexec –n 256 SWMF.exe > runlog \_`date +%y%m%d%H%M`

or a similar command in the run directory. This may be done from a job script.

In addition to running the SWMF.exe, there are a few scripts that need to run. The output generated for the 2D cut files (in run/GM/IO2/) should be continuously post-processed with

PostProc.pl –r=10 >& PostProc.log &

The –r=10 flag instructs the script to check for new output every 10 seconds. The PostProc.log file contains information about the host machine that the process is running on, the start time and the expected end time. This script runs the PostIDL.exe code parallel on 4 cores by default. The number of cores can be changed (see PostProc.pl –h for various options). Note that the script stops after 2 days if not killed sooner.

In operational mode the Geospace model will be restarted at least once a day (regular restart) and whenever the solar wind speed increases suddenly. This means that the SWMF has to save restart files quite frequently (we plan to do that once a minute of simulation time) and the restart files cannot be overwritten (overwriting is the default behavior). We added a new feature controlled by the #RESTARTOUTDIR command that creates a new restart directory every time with a name containing the date and time in the YYYYMMDD\_HHMMSS format.

## Regular Restart of the Geospace model

For the daily restart, the code can stop simply because it reaches the final simulation time that is set in the #ENDTIME command. Alternatively a clean stop can be achieved by creating the SWMF.STOP file by

touch SWMF.STOP

in the run directory. The existence of this file is checked periodically if the #CHECKSTOP command is present in the PARAM.in file. Checking the file very frequently can slow down the progress of the code. Once the file is noticed, the code will generate a final restart file and stop.

For restart, the PARAM.in file should be replaced with the version that can do a restart and is based on the Param/SWPC/PARAM.in\_SWPC\_restart file. It should contain the most up-to-date F10.7 flux. The solar wind data file (IMF.dat) should be updated too.

If the Restart.pl script is running continuously in the background with the –r flag, the script will link to the last restart directory. However, when multiple restart trees are created with the #RESTARTOUTDIR command, then it is necessary to select the last restart directory with

Restart.pl –i SWMF\_RESTART.YYYYMMDD\_HHMMSS

If the Geospace model is run with repeated restarts indefinitely, the number of time steps and the simulation time could accumulate to values that would cause issues. To avoid this problem, the PARAM.in file uses the #ENDTIME command instead of the usual #STOP command in the last session. If the code stops because it reached the time specified with the #ENDTIME command, the SWMF restart file (SWMF\_RESTART.YYYYMMDD\_HHMMSS/RESTART.out) will be written with 0.0 simulation time in the #TIMESIMULATION command and 0 time step in the #NSTEP command, but the #STARTTIME command will have the date and time specified in the #ENDTIME in PARAM.in.

In addition, the time step counter nStep in the BATS-R-US model also accumulates indefinitely by default. To avoid this, the value of nStep has to be reset periodically. Since nStep=0 indicates the beginning of a simulation, a positive value should be used. A simple solution is to add the #NSTEP command with nStep set to 10 right after the command that includes the BATS-R-US restart header in the PARAM.in file used for restarts. This has been added to the Param/SWPC/PARAM.in\_SWPC\_restart template.

The Geospace model can be restarted with the usual command:

mpiexec –n 256 SWMF.exe >& runlog\_`date +%y%m%d%H%M`

## Restarting the Model due to Sudden Increase of Solar Wind Speed

Most restarts will be needed, because the solar wind speed increases, so that the propagation of the latest solar wind data from the L1 point to the upstream boundary will take less time than assumed previously. This means that the Geospace model has to go back to an earlier time and use the updated solar wind data, i.e. it has to restart from an earlier restart file.

First of all the running code should be stopped. This may be easiest to do with some Unix command (kill) or from the queue system software (e.g. qdel), or by creating an SWMF.KILL file in the run directory:

touch SWMF.KILL

Using SWMF.KILL in combination with the #CHECKKILL command in the PARAM.in file ensures that the BATSRUS model stops at a point that is not in the middle of writing output or restart files.

For restart, the PARAM.in file should be replaced with the version that can do a restart and is based on the Param/SWPC/PARAM.in\_SWPC\_restart file. It should contain the most up-to-date F10.7 flux. The solar wind data file (IMF.dat) should be updated too. The restart directory with the new propagated date-time value should be selected:

Restart.pl –i SWMF\_RESTART.YYYYMMDD\_HHMMSS

All the newer restart directories should be removed. The post processing script PostProc.pl should be stopped (killed) to avoid processing files that are being removed. Then the output that is newer than the new date-time should be removed (or archived into a different directory, if desired). This cleanup operation will require a new script or a new option for the Restart.pl script.

Now the Geospace model and the post processing code can be restarted with the usual commands:

mpiexec –n 256 SWMF.exe >& runlog\_`date +%y%m%d%H%M`

PostProc.pl –r=10 >& PostProc.log &

# Solar Wind Propagation

## Ballistic propagation

Solar wind data arrives near real time to SWPC. This information is measured at the L1 point that is approximately 0.01 AU from the Earth towards the Sun. Let us denote the distance between the satellite and the upstream boundary of the Geospace model with

D = xSAT – xBOUNDARY

where x is the GSM coordinate parallel with the Sun-Earth line. In the current model setup xBOUNDARY = 32 RE = 204,096 km, while xSAT should be obtained from the satellite trajectory. At any given time one can assume that the solar wind propagates with the measured velocity VX(t) so it will arrive to the boundary of the Geospace model domain at

Clearly t’ will be in the future which provides the predictive capability for the Geospace model. In practice the solar wind data is taken as a discrete time series with some uniform frequency Δt, so tn = n Δt and

The solar wind data file will contain the date and time corresponding to together with the measured solar wind parameters. This will work as long as is a monotonically increasing series, i.e.

.

As long as does not change rapidly, this will be true. However, if

then . Note that is the ratio of the measurement frequency and the time it takes for the solar wind to propagate from L1 to the boundary, which is much less than 1 if the measurements are taken frequently enough.

Whenever the above condition is met, the Geospace model will have to be restarted from a time that is less than and all simulation results (and predictions) beyond this time should be discarded.

## MHD propagation

An alternative to ballistic propagation is using a 1D MHD code to propagate the solar wind data from L1 to the inflow boundary of the Geospace model. To evaluate the feasibility and/or usefulness of this approach, we have implemented the necessary tools to propagate L1 data to the boundary with the BATS-R-US code configured to 1D. For the purposes of evaluation the complete L1 data is propagated without restarts. In true operation mode one would need to do restarts when the solar wind speed suddenly increases, as described in the previous section. This would make the propagation procedure more complicated, but it should not change the results.

From the operational perspective it is important to make sure that the propagation is fast enough so that not much time is spent on it. Fortunately this is indeed the case: running the 1D MHD code for 1 hour simulation time takes less than 1 second wall clock time on 4 Intel i7 cores. The code uses the 5th-order accurate finite difference scheme (Chen et. al, 2016) with 320 grid cells. Running on a grid with twice as many cells makes very small differences only.

In the following we describe how the propagation can be performed using the scripts in the SWPCTEST repository (see the section on [Validation Tools](#_Automated_Validation_Tools) to see how this repository can be obtained). Inside the SWMF/SWPCTEST directory typing

make propagate1d EVENTS=2,3,5

will do the 1D MHD propagation for events 2, 3 and 5. For sake of better understanding the mains steps are described below.

First, GM/BATSRUS is configured to 1D MHD and compiled to produce the executable

bin/BATSRUS.exe

Then the run directory

GM/BATSRUS/run\_L1toBC

is created and the input files from SWPCTEST/Inputs/event\*/L1.dat and wind.dat (if present) are copied into the run directory for the selected events. The PARAM.in file is copied from

GM/BATSRUS/Param/EARTH/PARAM.in.L1toBC

and modified by the

SWPCTEST/Scripts/change\_param.pl

script: the start and end times are set according to the first and last dates in the L1.dat file, and the xMax position of the grid is set to the satellite position in the L1.dat file. In addition, if the wind.dat file is present in the run directory, then the IMF is extracted at the location of the WIND spacecraft using GSE vector components. Otherwise the date is extracted x=32 Re (the boundary of the Geospace model) with GSM vector components. Then the code is run as

mpiexec –n 4 BATSRUS.exe > runlog

The main result is the

GM/IO2/log\_n000000.log

file, which contains the MHD quantities extracted at the 32 Re or WIND position. After the variable names and the header lines are slightly edited, the log file is copied back into the

SWPCTEST/Inputs/event\*/IMF\_mhd.dat

files for the selected events. The results are stored in Git, so this procedure does not have to be repeated as long as the code or some of the parameters are not modified.

## Comparison of ballistic and MHD propagation

The SWPCTEST repository contains input data for 10 events. Events 1 through 6 are those that were used in the SWPC challenge that was performed by CCMC. Events 7 to 10 contain L1 data and WIND satellite data for 4 events that were selected by Michele Cash who studied the effect of adjusting the normal direction of the propagation plane. For sake of brevity, we refer to the first 6 events as the “SWPC challenge” events, and the last 4 as the “WIND comparison” events.

For the very first event, the October 29, 2003 (Halloween) storm there is no complete L1 data, because ACE could not measure some of the plasma parameters. Therefore we have to limit the comparison to events 2 to 10.

The ballistic and MHD propagation methods were described in the previous section. We used the

make ballistic

command to run the SWPCTEST/Idl/ballistic\*.pro scripts that generate the

SWPCTEST/Inputs/event\*/IMF\_ballistic.dat

files for events 2 to 10. Note that in these files Bx is not set to zero (unlike in the IMF files provided by CCMC). Typing

make propagate1d\_plot

generates the comparison plots

SWPCTEST/Inputs/event\*/mhd\_vs\_ballistic.eps

SWPCTEST/Inputs/event\*/mhd\_vs\_ballistic.pdf

These plot files have been committed into the SWPCTEST repository. As and example, the plot for event 6 is shown in Figure 2.1. Overall there is a good agreement between the MHD and ballistic propagated solar wind data, but there are also some significant differences. In particular, the MHD solution contains larger density peaks. The MHD solution shows a shock at about 24 hours after the start time. This shock is not present in the ballistic propagation, which suggest that the shock is a result of some MHD dynamics that is not captured by the simple ballistic propagation. The other events typically show less difference between the results of the two propagation methods.



Figure 2.1. Comparison of solar wind data propagated by 1D MHD (black line) and ballistic method (blue line) for event 6 (August 5, 2011).

## Comparison of ballistic and MHD propagation with WIND satellite data

The previous section compared the results from ballistic and MHD propagation. While one would expect the MHD propagation to be more accurate as it captures the physics better, this expectation should be checked against data. Events 7 to 10 were selected by Michele Cash to allow comparison with WIND satellite data. WIND only measures the magnetic field, so the comparison is limited to the three components of B. Typing

make propagate1d\_wind\_plot

will generate the comparison plots

SWPCTEST/Inputs/event\*/mhd\_vs\_ballistic\_vs\_wind.eps

SWPCTEST/Inputs/event\*/mhd\_vs\_ballistic\_vs\_wind.pdf

among the WIND data, the MHD and the ballistic propagation results for events 7 to 10. These plots are saved into the SWPCTEST reporistory.

Figure 2.2 shows the comparison plot for event 9 as an example. Clearly, the MHD and ballistic propagation methods give very similar results (at least for the magnetic field), while the WIND data has significant deviations. This suggest that the differences between WIND data and the traditional ballistic propagation method (studied by Cash et al. 2016, doi:10.1002/2015SW001321) is not resolved by using MHD propagation. In fact, it seems that the differences between the two propagation methods are much smaller than the “error” relative to the WIND data.

****

Figure 2.2. Comparison of solar wind and IMF propagated by 1D MHD (black line), ballistic method (blue line) and the WIND observations for event 9 (August 10, 1998). ACE and WIND were about 245 Re and 70-77 Re, respectively, from the Earth towards the Sun.

# Sensitivity to Input Parameters

Milestone 4.1.5

Many input parameters control the numerical algorithm, such as the time stepping scheme the numerical flux function, the limiter, the grid resolution etc. All these parameters have an influence on the results as well as on the computational performance (speed). Below we discuss the time stepping scheme, since it was changed relative to the initial settings. Other numerical parameters are discussed in the SWMF user manual doc/SWMF.pdf as well as the customized doc/SWPC.pdf manual, which contains only the commands that are used by SWPC.

There are also many parameters that have some physical meaning. One can modify/tune these to improve the accuracy of the predictions. These physical parameters will be listed and discussed below in the second section.

## Explicit versus Implicit Time Stepping Schemes

The following discussion is about the time stepping schemes used in BATSRUS. Since BATSRUS dominates the execution time for the Geospace application, the choice of the time integration scheme in BATSRUS has a major impact on the overall performance of the SWMF.

Explicit time integration uses the current state to calculate fluxes and source terms, and then uses these to advance the solution to the next time step. This method is fast but the time step is limited by numerical stability conditions, which make the time step small (about 0.1s or less for the magnetospheric runs using the typical grid resolution).

Implicit time integration uses fluxes and sources based on the already advanced state. This requires a solution of a large linear system, which is expensive. On the other hand the implicit time integration is stable for large time steps. In the SWPC application we typically try to use 5-second time steps.

The part implicit scheme uses the explicit time stepping in the grid blocks where the time step is stable, and the implicit time stepping where the explicit method would be unstable. The stability condition is evaluated every time step.

The time step may be reduced if the density or pressure changes too much at any location in the grid. This can happen both for the explicit and implicit time stepping, but it is more likely to happen for the implicit scheme, because the larger time steps result in larger changes in general.

When the time step is reduced, the part implicit scheme switches more-and-more blocks from implicit to explicit. If the time step gets very small, all blocks may become explicit. If the changes in density and pressure become smaller, the code gradually tries to recover the original time step.

We have 1-stage and 2-stage explicit time stepping schemes. In the explicit 1-stage scheme the fluxes and source terms are calculated from the current state and then multiplied with the time step Dt and added to the current state to get the new state. In the 2-stage scheme, on the other hand, we first advance the solution by a half time step Dt/2 and then use this intermediate state to recalculate the fluxes and sources, and then use these to update the solution with a full time step Dt.

The 1-stage scheme has a temporal discretization error proportional to Dt, while the two stage scheme makes the error term proportional to Dt2. The 2-stage scheme is about twice as expensive as the 1-stage scheme. For the explicit time stepping scheme the time step is small, so the temporal error term is small compared to the spatial discretization error that is typically proportional to Dx2 where Dx is the cell size.

Based on several experiments, we now recommend using the 1-stage explicit time stepping scheme for operational use, as it can continuously maintain faster than real time performance on 64 processor cores while the results remain essentially the same as with the original implicit time stepping scheme. The PARAM.in files in the Param/SWPC directory were modified accordingly.

## Physical Parameters

### BATS-R-US Parameters

#BODY command. The location of the inner boundary of the BATS-R-US domain is controlled by the rBody parameter of the #BODY command. The typical value is at 2.5 RE. Increasing this value is probably not a good idea, because it will impact the accuracy of the solution near the Earth. Lowering the value say to 2 RE is certainly possible, however given the 1/4 RE grid resolution near the Earth it is unlikely to improve results. For extreme storms (like the Halloween storm) the magnetopause may be pushed very close and in this case a smaller value of rBody may be useful. For typical storms this is probably not needed. The rCurrents parameter determines where the field-aligned currents (to be passed to the ionosphere electrodynamics model) is calculated. This should be at least a couple of grid cells away from the inner boundary, so that the currents are not influenced by the boundary conditions too much. We typically set rCurrents = rBody + 0.5 RE. One may move this slightly further, but not closer for the given grid resolution. For finer grids the distance between rCurrents and rBody could be reduced. The BodyNDim and BodyTDim parameters determine the density and temperature of the “body”. The temperature is not used at all (except in plots). The number density is used (except when overwritten by the #CPCPBOUNDARY command), and it has a major influence. The value 28/cc was found to be a good compromise for quiet times, but it is too small for storms. We have experimented with doubling it to 56/cc, and it improved the solution for storms, but it seemed to be too large for quiet times. The #CPCPBOUNDARY command allows making the density a function of activity, as discussed next.

#CPCPBOUNDARY command. If the UseCpcpBc parameter is true, the density at the inner boundary will depend on the cross polar cap potential (CPCP, calculated by the IE model) in a linear fashion:

RhoBc = Rho0Cpcp + RhoPerCpcp \* Cpcp

where the densities are given in amu/cc and the cross polar cap potential in keV. Both the constant Rho0Cpcp and the coefficient RhoPerCpcp can be adjusted by this command. The values of these coefficients have a major impact on the predictive capability of the Geospace code. The selected values Rho0Cpcp=28amu/cc and RhoPerCpcp=0.1 amu/cc/keV were found to be significantly better than the simple constant value given by nBodyDim parameter of the #BODY command. However, there is definitely room for optimization.

#IMCOUPLING command. The TauCoupleIm parameter determines how fast the MHD pressure (and density) is relaxed towards the pressure (and density) of the inner magnetosphere model (RCM in the current Geospace setup). The default value of TauCoupleIm = 20 seconds is a good choice, but not unique. Using a much shorter time may lead to numerical stability issues. Using a much longer time will reduce the effectiveness of the RCM-BATS-R-US coupling, which is a crucial part of the Geospace model. Doing some experiments in the 5 to 60 second range may be useful. The DoImSatTrace command only influences the satellite output files, so it is not discussed further. The DoCoupleImPressure logical should be true for the Geospace model, because it provides the crucial pressure coupling between the inner and global magnetosphere models. The DoCoupleImDensity may be both true or false. When true, the MHD density is relaxed towards the density determined by the RCM model. If it is false, the MHD density is determined by the MHD model and the inner boundary condition (see #CPCPBOUNDARY command). In fact, our first set of runs in preparation for the SWPC challenge used the DoCoupleImDensity false setting, and the results were good. We later changed to the true setting, because for some storms the MHD density in the inner magnetosphere seemed to become unrealistic. We found that the change to DoCoupleImDensity=T did not have a strong effect on the Dst (and presumably on the dB and dB/dt values), but this setting may be revisited. If the only goal is to predict local magnetic perturbations (dB), the optimal choice may well be the DoCoupleImDensity=F. The DoFixPolarRegion parameter determines what happens in the polar region that is outside the closed field line region controlled by the RCM model. The DoFixPolarRegion=F setting does not do anything special, which means that the enhanced pressure (and density) can diffuse (due to numerical diffusion) from the closed field line region into the polar region. This does not seem to be a major problem in the SWPC-type simulations, but we (and CCMC) have also experimented with an alternative setting using the DoFixPolarRegion=T. In this case the density and temperature are forced to fixed values of PolarNDim and PolarTDim out to the rFixPolarRegion radius in the polar (open field line) region. We have not attempted to optimize the PolarNDim, PolarTDim, and rFixPolarRegion parameters.

#IMCOUPLINGSMOOTH command. The dLatSmoothIm parameter determines how the nudging of the MHD pressure/density is smoothed out at the edges of the RCM domain. The default is to apply the same relaxation time (see the TauCoupleIm parameter in the #IMCOUPLING command) in the whole closed field line domain. When dLatSmoothIm is applied, the relaxation rate changes smoothly from the 1/TauCoupleIm value to zero within the dLatSmoothIm latitude distance from the open-closed latitude boundary. Some runs at CCMC observed oscillations at the open closed boundary in the MHD domain, and this smoothing seemed to help to suppress it. It is unclear if this algorithm has a significant effect on the products that SWPC aims at providing.

#BORIS command. The speed of light limits the maximum value of the Alfven speed in the semi-relativistic formulation of the MHD equations, while in classical MHD the Alfven speed can be arbitrarily large as it simply depends on the ratio of the magnetic field and the square root of density. While the classical Alfven speed in the magnetosphere is unlikely to exceed the speed of light (although it can be comparable), solving the semi-relativistic equations with an artificially *reduced speed of light* is still beneficial, so the UseBorisCorrection parameter should be set to true and the BorisClightFactor parameter set to a value less than 1 so that the Alfven speed is limited to c’=c\*BorisClightFactor. This “Boris correction” with a reduced speed of light is used in essentially all global MHD codes applied to the Earth magnetosphere (e.g. LFM and OpenGGCM). The benefit of limiting the Alfven speed is two-fold: 1) it allows larger time steps for the explicit time stepping scheme (see section 1) and thus improves the code speed; and 2) it reduces numerical diffusion. This has been extensively discussed by Toth et al. (2011, JGR, doi:10.1029/2010JA016370). The recommended value is BorisClightFactor=0.01 that corresponds to an artificial speed of light c’=3000km/s. Using a larger value will increase the numerical diffusion and make the code run slower. Making the value much smaller may lead to numerical instabilities. Using c’ that is comparable with the solar wind speed is unphysical, and should be avoided. This means that there is not much room to change this parameter and it is unlikely that a slightly different value would have a significant influence on the performance of the Geospace model.

### Ridley Ionosphere Model Parameters

#IONOSPHERE command. Detailed description of this command is given in the SWPC.pdf document, and not repeated here. The TypeConductanceModel defines the model version. The default model 5 is the most sophisticated and it has been found to produce very realistic CPCP values in a wide range of circumstances. But models 2, 3 and 4 are also possible choices. All these models have several adjustable parameters. The F107Flux parameter should be set to the actual observed value, so it is not intended to be a tunable parameter. The StarLightPedConductance and PolarCapPedConductance parameters can be experimented with, if desired, although the default parameters are likely to be close to the optimal choice. This does not mean that some improvement cannot be achieved for the specific skill scores that SWPC is optimizing for.

#BOUNDARY command. The LatBoundary defines where the lower latitude (in magnetic coordinates) boundary of the ionosphere electrodynamics domain. At this boundary the electric potential is set to zero, and this boundary condition is used by the Poisson solver. Since the field-aligned currents are usually located at high latitudes, the solution does not depend too much on the location of the boundary as long as it is at low magnetic latitude. The recommended setting LatBoundary = 10 degrees, is likely to be a good choice. Moving to lower value is unlikely to do much (except increasing computational cost for a given ionosphere grid resolution). Moving the boundary to higher latitude may have an improper influence on the electric potential (the solution of the Poisson equation) during strong geomagnetic events.

### Rice Convection Model Parameters

#COMPOSITION command. The FractionH and FractionO parameters determine the *assumed* ratio of H+ and O+ ions in the single fluid MHD plasma that provides the boundary conditions for the RCM. We found that this fraction has a major influence on the accuracy of the Geospace model in terms of predicting Dst and local magnetic perturbations. We have experimented with FractionH=0.7 and FractionO=0.3 (note that the sum has to be 1) as well as the recommended value FractionH=0.9 and FractionO=0.1. The smaller fraction of O+ seems to produce stronger Dst. This may not be a true physical effect, since in reality the O+ ratio is observed to increase during geomagnetic disturbances. It is best to think of these parameters in terms of the conversion of the MHD mass density and pressure into H+ and O+ number densities and temperatures for RCM. There is certainly room for experiments here. In the future, using the multi-fluid MHD code may provide a physics-based alternative (see the recent paper by Ilie et al. JGR, 2015, doi:10.1002/2015JA021157).

#DECAY command. If the UseDecay parameter is true, the phase density function of the RCM model decays at a rate 1/DecayTimescale, where the DecayTimeScale parameter is given in seconds. The logical should be true, because without this ad hoc decay the RCM model tends not to recover after a major storm. It is unclear what physics is missing from RCM, or why this happens, but we know that without the decay the Geospace model gives unrealistic solutions in the recovery phase, and eventually crashes. The recommended value for the DecayTimeScale is 36000 seconds (or 10 hours). This value gives fairly good agreement with the measured decay rate of the Dst index in a few events but one could experiment with different values in the range of 5 to 20 hours. One could potentially adjust this rate to follow observed decay rates.

## Results from our work prior to the SWPC challenge

The following figures were generated from the many runs we performed prior to the SWPC challenge. Both figures show the simulated vs. observed Dst indexes for the April 6th 2000 storm. Figure 3.1 shows how the code performs for different locations for the inner boundary (2.25 vs. 2.5 RE) and the fixed density (28 vs. 56/cc) (see the #BODY command for BATS-R-US); and different H+/O+ ratios (see the #COMPOSITION command for RCM). Figure 3.2 shows dependence on various dynamic settings for the density at the inner boundary (see the #CPCPBOUNDARY command for BATS-R-US) and various options for coupling between the global and inner magnetosphere models (see the #IMCOUPLING and #IMCOUPLINGSMOOTH commands). Note that this study is limited to a single storm and for a single observable (Dst), so it does not replace a more comprehensive optimization. Also, the code has changed since this study was done. Therefore, while indicative, the results shown by the figures are not meant to replace the need for a new more extensive (more parameters) and comprehensive (more events and more observables) study with the current code.

20000406dst_HOratio.pdf

Figure 3.1. Comparison of simulated and observed (black line) Dst for various parameter settings. The H/O ratio is used at the RCM boundary condition. The density (rho) is set at the inner boundary of BATS-R-US. The 2.5 and 2.25 RE indicate the location of the inner boundary of the BATS-R-US code.

20000406dst_density.pdf

Figure 2. Comparison of simulated and observed (black line) Dst for various parameter settings. The H/O ratio used at the RCM boundary condition is fixed to 9/1 in these runs. The density (rho) at the inner boundary is either set to a fixed value, or to a value dependent on the cross polar cap potential. Runs with “imcouplerho” have the density coupling switched on between RCM and BATS-R-US. Runs with “fixpolar” have the polar region fix switched on with the indicated density and temperature. The “smooth” means that the run uses the #IMCOUPLINGSMOOTH command to smooth the relaxation rate towards the RCM pressure and density at the edges of the closed field line region.

# Model Products and Displays

Milestones 4.2.1 and 4.3.1

## Magnetometer Stations and Grids

One of the main products of the Geospace model is the predicted magnetic perturbation at a given location due to the various current systems. For validation purposes the model can calculate the perturbations at the locations of magnetometer stations. This can be conveniently achieved with the the #MAGNETOMETER command:

#MAGNETOMETER

magin\_GEM.dat NameMagInputFile

-1 DnOutput

60.0 DtOutput

This example uses the list of stations provided in the magin\_GEM.dat file and the output is saved with a frequency of 60 seconds. The magnetogram input file has the following format:

#COORD

MAG

#START

YKC 68.93 299.36

MEA 61.57 306.20

…

In this example the coordinate system is magnetic (MAG) given by the #COORD command. In each line following the #START command, the 3-letter station name is followed by the longitude and latitude (in degrees) of the station in the selected coordinate system. The output produced by this command is a plain text file that contains a time series of magnetic perturbations for all the listed stations. For example the test\_swpc test produces the file

GM/IO2/magnetometers\_e20140410-000200.mag

with the content

12 magnetometers: YKC MEA NEW FRN IQA PBQ OTT FRD HRN ABK WNG FUR

nstep year mo dy hr mn sc msc station X Y Z dBn dBe dBd dBnMhd dBeMhd dBdMhd dBnFac dBeFac dBdFac dBnHal dBeHal dBdHal dBnPed dBePed dBdPed

539 2014 04 10 00 02 00 000 1 1.42600E+06 1.79558E+06 5.95158E+06 -3.15835E+01 -3.04657E+01 6.23890E+00 -1.43739E+01 -1.86302E+01 -1.39347E+00 -1.72097E+01 -1.18355E+01 7.63238E+00 0.00000E+00 0.00000E+00 0.00000E+00 0.00000E+00 0.00000E+00 0.00000E+00

539 2014 04 10 00 02 00 000 2 1.59177E+06 2.58581E+06 5.60881E+06 -2.59753E+01 -1.45323E+01 6.27083E+00 -1.11501E+01 -1.91161E+01 -1.98222E+00 -1.48252E+01 4.58378E+00 8.25305E+00 0.00000E+00 0.00000E+00 0.00000E+00 0.00000E+00 0.00000E+00 0.00000E+00

…

The first line contains the number and names of stations, the second line describes the columns, and the following lines contain the values for each station and for each time step saved. The nstep column contains the number of time steps taken by the MHD model, the ‘year … msc’ columns describe the time down to milliseconds, the next ‘station’ column contains the index of the station in the order listed in the #MAGNETOMETER command. The ‘X Y Z’ columns contain the position of the station in the coordinate system of the Geospace model (GSM), the

‘dBn dBe dBd’ columns are the north, east, down components of the magnetic perturbations. The remaining columns contain the contributions due to the currents in the MHD domain, the field aligned currents (FAC) in the gap region between the MHD domain and the ionosphere, and the ionospheric Hall and Pedersen currents.

For purposes of operational use it is better to place the virtual magnetometers on a grid around the globe or a region of interest. This can be achieved with the #MAGNETOMETERGRID command.

#MAGNETOMETERGRID  
ascii TypeFileMagGrid (ascii, tec, real4, real8)

GEO TypeCoordMagGrid (GEO, MAG, SMG)

72 nLonMagGrid

33 nLatMagGrid

0. LonMinMagGrid

360. LonMaxMagGrid

-80. LatMinMagGrid

80. LatMaxMagGrid

-1 DnSaveMagGrid

60.0 DtSaveMagGrid

This example will produce a 72 times 33 grid stretching from over 0 to 360 degrees in longitude and -80 to 80 degrees in latitude in the geographic (GEO) coordinate system. The ASCII file is saved once a minute and it has the same format as the IDL output files from BATSRUS. The output file will be in the GM/IO2 directory named, for example, as

GM/IO2/mag\_grid\_e20140410-000100.out

with the following content:

Magnetometer grid (GEO) [deg] dB (North-East-Down) [nT]

0 6.0000000000E+01 2 0 15

72 33

Lon Lat dBn dBe dBd dBnMhd dBeMhd dBdMhd dBnFac dBeFac dBdFac dBnHal dBeHal dBdHal dBnPed dBePed dBdPed

0.0000000000E+00 -8.0000000000E+01 3.7952601246E+01 5.1934660250E+01 2.7506730373E+00 1.5602241992E+01 1.4539702224E+01 -1.5082398669E+01 2.0765859544E+01 6.9429100923E+00 7.7712964235E+00 5.8020725472E+00 2.7806526731E+01 1.3351968044E+01 -4.2175728366E+00 2.6455212030E+00 -3.2901927615E+00

5.0000000000E+00 -8.0000000000E+01 -5.3659043423E+01 2.6712463200E+01 2.9420644645E+01 -1.6861565547E+01 2.0276776496E+01 -8.6573109728E+00 -1.5232033399E+01 2.0480065726E+01 -7.9383787154E-01 -2.6654740088E+01 -1.0975638255E+01 2.4758317813E+01 5.0892956109E+00 -3.0687407670E+00 1.4113475676E+01

…

The first line provides information about the units (degrees and nT) and the coordinate system (GEO). The second line contains an iteration number (for this file it is always 0), the simulation time (since the last full restart), the number of dimensions (2), the number of scalar parameters (0), and the number of variables saved (15). The third line gives the size of the magnetometer grid (72 times 33). The fourth line contains the names of the coordinates (Lon Lat) and the names of the variables saved (dBn … dBdPed). The variables have the same names and the same meaning as explained above for the ‘magnetometer\_\*.mag’ output. The following 72\*33 lines contain the coordinate and variable values for each grid point in the Fortran ordering: the first coordinate (longitude) varies fastest. This output file can be read into IDL, SpacePy, or into any other plotting package.

## Ring of Geosynchronous Values

A useful forecasting product requested by SWPC is the MHD model results at the geosynchronous orbit. Previously, this capability was available in only a limited manner via the “sph” plot area type for the #SAVEPLOT command in the GM portion of the PARAM input file. This plot area type would allow for the saving of the results about a sphere of given radius at a given time cadence. Parameters such as the latitude and longitude resolution and the coordinate system of the plot file were hard-coded, meaning that the user was required to post-process such files to reduce the data and convert the coordinate system. A simpler, more flexible output type is required for operational used.

This capability is now provided via the “shl” (short for “shell”) plot area type. This new plot type extracts a 1, 2, or 3D shell subsection of the solution and saves it to file using the typical file formats available. An example of the command syntax is as follows:

#SAVEPLOT  
1 nPlotFiles  
shl MHD idl\_ascii StringPlot  
1000 DnSavePlot  
-1. DtSavePlot  
GEO TypeCoordPlot  
6.6 rMin  
6.6 rMax  
0.0 LonMin  
360.0 LonMax  
5.0 dLon  
0.0 LatMin  
0.0 LatMax

This example command produces a single plot file that extracts values along a circle at a radius of 6.6RE at zero degrees latitude and 0-360 degrees longitude with a 5-degree resolution. The location is given in the coordinate system provided in the TypeCoordPlot parameter, in this case GEO. Note that the output grid resolution is only required when the plot range in a particular direction is nonzero. If we wanted a disk of results, the radius range would be set to 0.0 to 6.6 (or whatever range the user wishes), and the dR argument would be required immediately after rMax. Similarly, a full sphere volume could be specified by specifying a full latitude range (and associated dLat). Any combination of ranges can be used to create different lines, surfaces, or volumes. The variables extracted are determined by the StringPlot parameter. In this example the MHD values (density, pressure, velocity, magnetic field, current) are extracted. Note that the vector components are in the coordinate system used by BATS-R-US, which is GSM for the Geospace model. The output file will be in the GM/IO2 directory named, for example, as

GM/IO2/shl\_mhd\_3\_e20140410-000300-000.out

with the content

R deg deg Mp/cc km/s km/s km/s nT nT nT nPa uA/m2 uA/m2 uA/m2

710 6.0000000000E+01 3 0 11

1 37 1

r lon lat rho ux uy uz bx by bz p jx jy jz

6.6000000000E+00 0.0000000000E+00 0.0000000000E+00 2.4583163039E+01 5.8854344132E+01 -5.8063514183E+00 -3.6150663993E+01 5.2987720073E+01 -1.3123389781E-01 6.9710520456E+01 3.3653786500E+00 3.3112098982E-05 2.3393965787E-03 9.2568197086E-06

6.6000000000E+00 1.0000000000E+01 0.0000000000E+00 2.4036690524E+01 5.1661284323E+01 -4.6694979807E+00 -2.5127912605E+01 3.3168483422E+01 8.8319597920E-01 7.0031192027E+01 3.2844720732E+00 3.5934401442E-05 2.9073362345E-03 8.4130422557E-06

…

The first line contains the units, the second line the number of steps and the simulation time since the last full restart, the number of coordinates (always 3 in the current implementation), number of scalar variables (0) and the number of variables (11). The third line gives the grid size (1 x 37 x 1). The fourth line lists the names of the coordinates (r lon lat) and variables (rho … jz) for density, three velocity components, three magnetic field components, pressure and three current density components. The vectors are given in the coordinate system of the MHD model (GSM). The remaining lines contain the coordinate and variable values for each grid point on the spherical shell in the Fortran ordering (first coordinate varies fastest). This file can be read with the IDL macros, SpacePy or some other plotting package.

To verify that this new plotting output works correctly, a simple test was set up. The BATS-R-US domain was initialized with a density sine wave spanning the width of the domain. This is shown in Figure 4.1. Three 1-D extractions were made using the shell plot type: one for a range of longitudes, one for a range of latitudes, and one for a range of radial distances (Figure 4.1, blue, green, and red lines, respectively). Because the result can be obtained analytically, we can verify that the newly implemented algorithm is properly extracting and interpolating the MHD values. The result of this test is shown in Figure 4.2. The solid lines are the analytic reference solutions. The red dots show the MHD values as obtained via the three shell plots. The largest difference in any of the frames is 7.8x10-4 amu/cm-3, a small fraction of the actual plot values of 0.5 to 1.5 amu/cm-3. This test demonstrates that the new code has been implemented correctly.

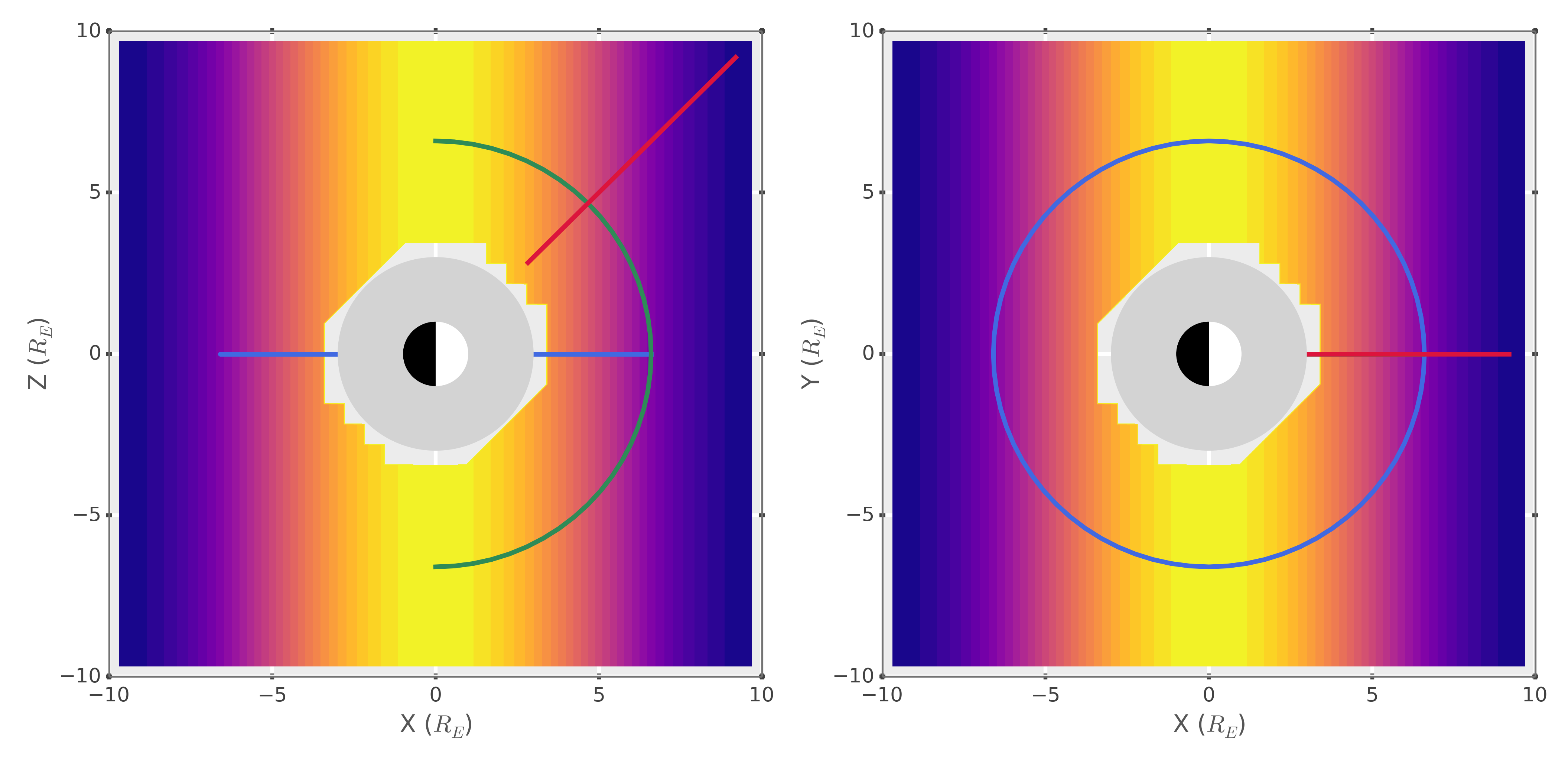


Figure 4.1: 2D cuts of the MHD domain when a simple density sine wave is imposed. The left plot is the X-Z plane; the right is the X-Y plane. Lines show extractions via the shell plot type.

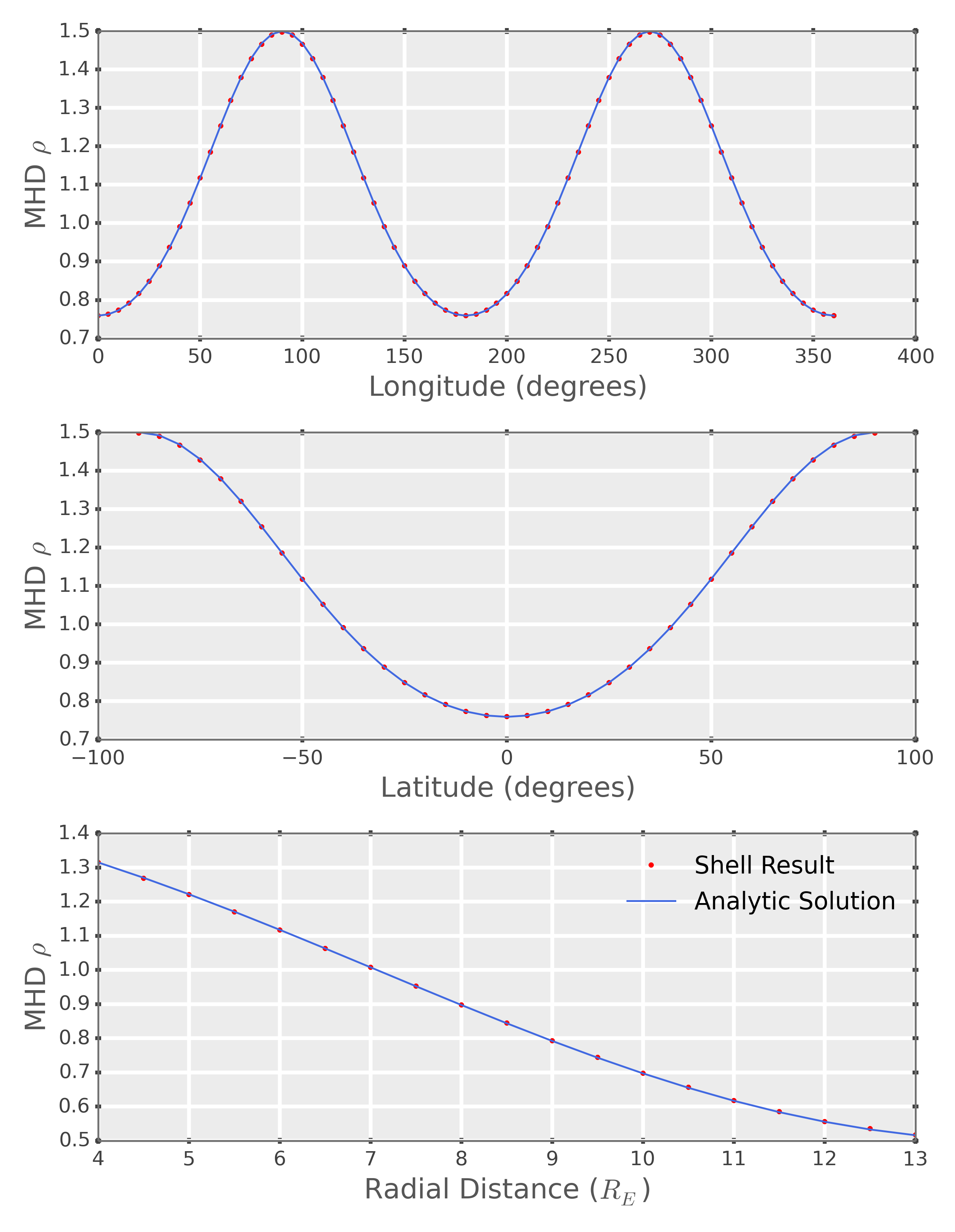


Figure 4.2: Analytic (blue) and extracted (red dots) density values for three different shell-plot slices. The difference between the two is negligible, demonstrating that the implementation of the shell plot area is correct.

## Virtual Geomagnetic Indices

Geomagnetic indices provide quick-look indicators of both global and local activity. As such, they can be a valuable output product from the SWMF. At current, three groups of indices are produced by the SWMF: K-type indices, including local K and the global K-index, KP; DST, an indicator of ring current strength; and the Auroral Electrojet indices (AL, AU, AE, and AO), which are indicators of high-latitude current system strength.

The three groups have been developed at different times. DST has been a long-standing output of BATS-R-US, long preceding the SWPC-CCMC validation challenge. The K-type indices were developed specifically for the challenge. This capability has been refined as part of this collaboration. The AE-type indices are a recent addition made specifically as part of the operational development of the SWMF.

### Calculation and Calibration of K Indices

The calculation of K indices closely follows the calculation of the real indices with several key differences. Similar to the real indices, virtual K indices begin with the N (north) and E (east) components of the magnetic field disturbance from background. No background field subtraction is required for the SWMF virtual magnetometers as they only report the disturbance. Next, the maximum and minimum of both N and E components over the past three hours are determined. In real K, this is done over 3-hour windows. However, because the virtual indices must be reported in real time, a sliding 3-hour window is used in the SWMF. Between the N and E components, the largest difference between maximum and minimum is selected. This is then converted to a K number using a semi-logarithmic scaling. The scaling factors are station-specific in order to obtain a desired occurrence distribution of K 1 through 9. Selection of the SWMF scaling factors is described below. Averaging the virtual K values creates planetary K, called KP.

The real time regional K indices produced by the SWMF are not placed at the real world standard K stations. Rather, they are placed evenly around the globe, one station at every hour local time. Additionally, they are placed all at the same magnetic latitude (60o). This overcomes two natural limitations of the real world observatories. First is the poor distribution of official K stations about the Asian continent. It also removes the need to adjust the results to compensate for seasonal effects that can affect the results.

Experiments were performed to choose the best combination of scaling factor and station latitude. A validation suite consisting of ten real-world events was used to test a variety of combinations (*Welling and Ridley,* Space Weather, 2010) It was found that a combination of 60o magnetic latitude and a K9 value at 600nT produced the best agreement with observations. The details of this validation are to be submitted to the Space Weather Journal.

To enable virtual K indices, use the following command in the PARAM.in file:

#GEOMAGINDICES  
180 nSizeKpWindow [min]  
60.0 DtOutput [sec]

The two parameters adjust the size of the rolling calculation window and the frequency that the output is written to file. It is not recommended to change the size of the calculation window, as this will affect the calibration. The results are written to GM/IO2/geomagindices\_<time>.log, where the time portion of the file name follows the usual SWMF formatting rules.

### Calculation of AE Indices

The calculation of the AE-type indices follows the calculation of the real world indices with a few important differences. The calculation is illustrated in Figure 4.3. A ring of magnetometers at constant SM latitude of 75o is spread such that there is one at every hour of magnetic longitude. Similar to the case of virtual K, using ideally placed stations overcomes issues with the real world observatories and simplifies the calculation. The horizontal components (north and east) of the magnetic disturbance at each virtual magnetometer is compared to each other (Figure 4.3, top frame, blue lines). The AU and AL indices at any point in time are defined as the maximum and minimum, respectively, of all stations. AE and AO are secondary products, defined and illustrated in Figure 4.3, bottom frame. These calculations mirror the real-world steps to obtain these indices.

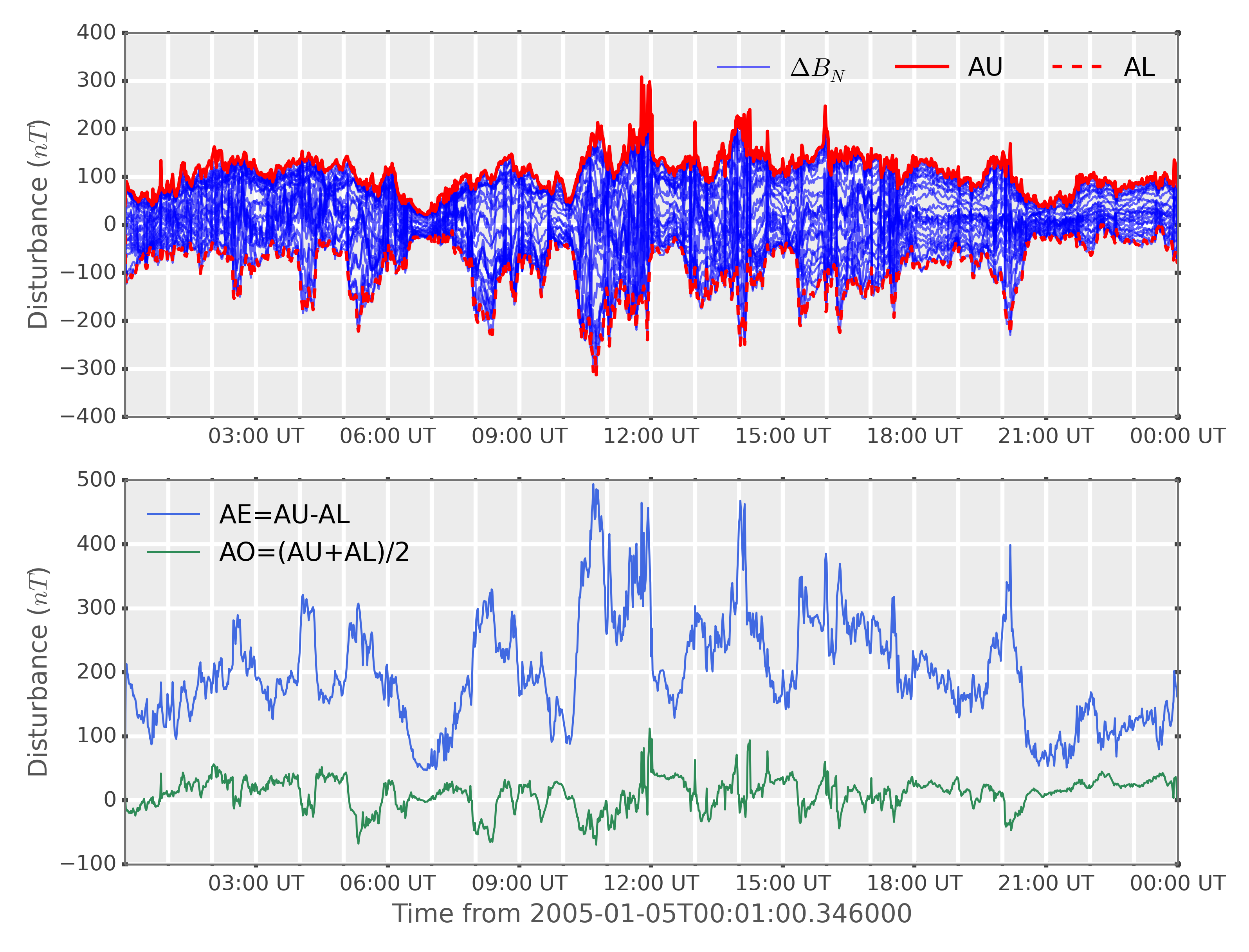


Figure 4.3: Illustration of AU and AL from individual high latitude stations (top frame) and the resulting AE and AO indices (bottom frame).

The process of calibrating these values is currently underway. The six SWPC/CCMC events will be the validation suite for this effort. Additionally, a month-long simulation will be used for extended-term validation. The results of this effort will determine the best latitude for the virtual AE stations.

## Radiation Belt Environment Model Output

The RBE model produces output files into the

RB/plots/

directory. The filenames contain the usual time stamp, the name of the species (e for electrons) and the .fls extension, for example test\_swpc creates a file named

20140410\_000300\_e.fls

The file contains one snapshot of the RBE results. The data file consists of the following parts:

The 1st record contains rc=1.0157 (radius of loss cone in Re), ir=51 (number of grid points in radial direction), ip=48 (number of grid points in the azimuthal direction), je=12 (number of grid points in the energy coordinate), ig=12 (number of grid points in the pitch angle coordinate), and ntime (number of snapshots, but it seems to be 2 when it is really 1, so it should be ignored).

The 2nd record (array of je elements) contains the energy grid in keV from 10keV to 4MeV:   
10, 17.241, 29.724 51.245, 88.349, 152.32, 262.61, 452.75, 780.56, 1345.7, 2320.1, 4000.

The 3rd record (array of ig elements) contains the sine of the pitch angles from 0.01 to 0.988.

The 4th record (array of ir elements) contains the magnetic latitude in degrees from 11.809 to 70.162 degrees.

The 5th record contains the time of the snapshot in hours followed by some parameters that can be ignored.

The next ir\*ip records start with the scalars lat1 (magnetic latitude), mlt (magnetic local time at the ionosphere), ro (radial distance at the minimum B in Re), mlto (magnetic local time at the minimum B), bo (magnetic field strength at minimum B in Tesla), irm1 (index of last closed field line in the latitude/radial grid), and the plasmasphere density, which is zero as it cannot be used in this version of RBE. These values are followed by je=12 (energies) lines each with ig=12 (pitch angles) elements of the directional differential flux function fy in units of (cm2 s sr keV)-1. The meaning of the directional differential flux is the number of electrons detected by a perfect detector per unit area (perpendicular to the motion of particles), per solid angle (direction), per energy (energy range) and per time. In effect the file contains the full 4D distribution function on the ir\*ip\*je\*ig grid.

Multi-snapshot files repeat the 5th record (with the time information) and the distribution function multiple times.

These plot files can be read by the IDL procedures found in the

RB/RBE/Idl/

directory:

plot\_fls.pro

plot\_fls\_L\_Time.pro

The first procedure can create a plot of the distribution of various quantities in the equatorial plane. The second procedure can be used to create a time evolution of the fluxes, however, that only works if the input file contains multiple snapshots, which is not the case with the settings used in the PARAM.in file used by SWPC. Therefore, we only describe how the first IDL procedure works.

Copy or link the plot\_fls.pro file into the RB/plots/ directory, start IDL there and then type

IDL> .r plot\_fls.pro

IDL> plot\_fls, fhead=’20140410\_000300\_e’

This will create a plot like shown in Figure 4.4. The plot can be customized by passing various parameters to it. These are described at the top of the IDL procedure. Here only the most useful and usable arguments are shown:

fhead - filename without extension. No default.

ilog - 1=logarithmic scale, 0=linear scale. Default is 1.

dir - 1=Sun on the left, 2=Sun on the right. Default is 1.

halfl - Size of the plot from origin in RE. Default is 20.

ie1 - index of lowest energy bin. Default is 0.

ie2 - index of highest energy bin. Default is 11.

iopt - plot 1=flux, 2=pressure, 3=density 4=temperature. Default is 1.

iunit - flux units: 1= /keV cm2 s sr, 2= /cm2 s sr. Default is 1. fmin - min value of (log)flux. Default is 0 (linear) or fmax-3 (log).

fmax - max value of (log)flux. Default is maximum found in output.

interactive- interactive mode requiring user input. Default is 0 (false).

The procedure performs a lot of calculations before producing the plots.



Figure 4.4: Example output from the plot\_fls IDL macro. The flux and the pitch angle distribution are shown in the equatorial plane. The results are produced by the RBE model in the short test\_swpc test.

# Automated Validation Tools and Sensitivity Analyses

Milestone 4.2.5

## Automated Validation Tools

In order to test the code and track performance versus time, an automated validation suite has been developed. This suite reproduces the entire SWPC selection analysis for the SWMF. It is expandable such that new events can be added easily. It also has the ability to propagate the solar wind drivers for each event from the L1 observation point to the upstream boundary of the SWMF computational domain via the newly developed MHD propagation code.

The automated validation suite can be obtained from the same Git server as the SWMF. The SWPCTEST repository should be checked out in the main SWMF directory:

cd SWMF

gitlabclone SWPCTEST

cd SWPCTEST

The repository contains the input files, solar wind drivers, run characteristics (run times, F10.7 fluxes, etc.) for every event as well as a set of scripts implemented in Make, Python, Perl, and IDL for performing the simulations and calculating metrics. All of this software is available freely except for IDL.

To execute the test, install the SWMF in the usual manner. Then, enter the SWPCTEST directory and type

make help

to see the available options. To run the code for all 6 events type

make test

To run a subset of events, type

make test EVENTS=2,3,5

This will configure and compile the SWMF, create separate run directories and copy input files for each event, and submit the simulations to the run queue. Note that as of now, the suite is set up to execute on the NASA Pleiades super computer. At this point, the user must wait for the simulations to finish.

Upon completion, the user should return to the SWPCTEST directory and type

make check

This will then collect the simulation results into the deltaB/Results directory for each event. Metrics are then calculated and saved. In each results directory (deltaB/Results/Event\*), a text file is generated with probability of detection (POD), probability of false detection (POF), and Heidke Skill Score (HSS) for high and mid latitude magnetometer groups for a variety of dB/dt thresholds. These same numbers are calculated for the results obtained for the SWPC selection study led by CCMC. This allows for a comparison of the current version of the code with the version originally submitted to SWPC. A summary table that contains the metrics for all events and stations together is written to the results directory. Finally, the results and metrics are archived in compressed tar file.

There are many options and commands that can be used through make to configure and control the test suite. The table below lists some make targets and variables of interest.

|  |  |
| --- | --- |
| **Target/Variable** | **Description** |
| make help | Print validation suite help information to screen |
| make test | Configure, compile, and execute code for each event. |
| make check | Collect results from successful runs, calculate and save validation metrics. |
| make propagate1d | Configure the code to perform the 1-D solar wind propagation for each event; save the results in the Inputs directory. |
| make … EVENTS=1,2,3 | Set the events to execute when make test is used. Defaults to all SWPC/CCMC study events. |
| make … QUEDIR=<directory> | Set the directory where the run directories will be placed. Important for parallel machines where execution is limited. |
| make … NP=64 | Set the number of processors to use during execution. |

## Using Ballistic vs. MHD Propagated Solar Wind for the Geospace Model

To test if the MHD propagation of solar wind parameters from the observations point at L1 to the upstream boundary of the SWMF (as described in Section 2) has an impact on SWMF dB/dt predictions, the automated validation suite was employed. First, the validation suite was run using the ballistically propagated solar drivers employed in the original CCMC/SWPC validation study. Then, the solar wind observations were propagated to 32 RE upstream using the 1D MHD propagation. The automated validation suite was then executed again, now using the results of the MHD propagation. The dB/dt prediction metrics from the two simulation sets are compared to see how the different solar wind propagation algorithms impact the results.

An example metrics comparison is shown in the table below. This table shows the Heidke Skill Score (HSS) for each event grouped by high- and mid-latitude magnetometers. The score is calculated per-event and for all events together. For this table, a cutoff threshold of 0.3 nT/s was used; metrics using the other thresholds from *Pulkkinen et al.* [2013] (0.7, 1.1, and 1.5 nT/s) were also calculated. Results are presented for both propagation methods. The difference between the two simulation sets metrics (MHD-propagated minus ballistically propagated) is shown; the relative difference (normalized to the original ballistically propagated run metrics) is also given. Results for other metrics (probability of detection and probability of false detection) for all threshold values can be found in the Excel worksheet accompanying this primer.



Notably absent from these calculations is Event 1, the famous Halloween storm of October 29, 2003. While the other five events all have both IMF and solar wind moment data from the ACE spacecraft, the SWEPAM instrument did not reliably measure the solar wind parameters during Event 1. Density and temperature are instead taken from the Geotail mission, which was in the near-Earth solar wind during storm onset. As such, propagation is not necessary for this event. It is excluded from all calculations for this comparison.

The above example is representative of the entire comparison: employing MHD propagation of the observed values to the upstream SWMF boundary does not improve the dB/dt predictions. Rather, it degrades the metrics considerably. The lone exception is Event 4, which is slightly improved when the MHD propagation is employed. The reduction in skill is especially noticeable at mid-latitude magnetometers; however, the number of threshold crossings is lower than at higher latitudes, so the metrics are typically less robust in general. At higher threshold values, the number of events drops significantly. These results strongly suggest that the MHD propagation will not add predictive value without further development.

There are several caveats to this comparison. First, there is a slight difference in the final position of the ballistic propagation (ending at 33 RE upstream to accommodate the different models run at CCMC) and the final position for the MHD propagation (32 RE, matching the true upstream boundary of the BATS-R-US model). Additionally, there is a discrepancy between the true x-coordinate of ACE for and the starting point used by the MHD propagation (235 RE). This results in a few minutes error that can be improved. Finally, it is noteworthy that the ballistically propagated input files set all IMF BX values to zero, again to accommodate less capable MHD models. These issues are currently being explored to determine their impact on the simulation results.