RAM-SCB User Manual

Vania Jordanova, Miles Engel, Steve Morley, Daniel Welling, Christopher Jeffery

Los Alamos National Laboratory

June 28, 2022

Contents

1 Introduction **5**
- 1.1 About This Manual . . . 5
- 1.2 TERMS OF USE & DISTRIBUTION POLICY . . . 5

2 Quick Start Guide **9**
- 2.1 Installation . . . 9
- 2.2 Execution . . . 9

3 Installation **11**
- 3.1 Installation of Required Libraries . . . 11
  - 3.1.1 Libraries From Modules . . . 11
  - 3.1.2 Installation From Source . . . 12
- 3.2 Installation, Configuration, and Compiling . . . 12
- 3.3 Testing the Installation . . . 13
- 3.4 Building Documentation . . . 14

4 Testing RAM-SCB **15**
- 4.1 Using and Interpreting Test Results . . . 15
- 4.2 Description of Available Regression Tests . . . 16
  - 4.2.1 Test 1 . . . 16
  - 4.2.2 Test 2 . . . 16
  - 4.2.3 Test 3 . . . 16
  - 4.2.4 Test 4 . . . 16
  - 4.2.5 EMIC Test . . . 16
- 4.3 Unit Testing . . . 16

5 Performing Simulations **17**
- 5.1 Making a Run Directory . . . 17
- 5.2 Inside a Run Directory . . . 17
- 5.3 Required Inputs . . . 19
- 5.4 Restarting Simulations . . . 20
- 5.5 Output . . . 20

6 Output Visualization **23**
- 6.1 SpacePy . . . 23
- 6.2 VTK: The Visualization Toolkit . . . 23
  - 6.2.1 VTK Data from RAM-SCB Restart Files . . . 23
  - 6.2.2 Visualization using VTK/ParaView . . . 23

7 RAM-SCB and SWMF **25**

8 Included Scripts **27**
8.1 Config.pl (SWMF Script) . . . 27
8.2 DiffNum.pl (SWMF Script) . . . 27
8.3 CatLog.py (RAM-SCB Script) . . . 27
8.4 Updating Input Files . . . 28
8.4.1 updateRamIndices.py . . . 28
8.4.2 provisionalIndices.py . . . 28
8.4.3 rtUpdateParam.py . . . 28

9 Complete List of Input Commands **29**
9.1 Input Commands for the RAM_SCB: IM Component . . . 29
9.1.1 Base . . . 29
9.1.2 Ram . . . 30
9.1.3 Scb . . . 32
9.1.4 Components . . . 34
9.1.5 Input . . . 36
9.1.6 Output . . . 37
9.1.7 Testing . . . 39

Index of Input Commands **40**

10 Bibliography **41**

Chapter 1

Introduction

The **R**ing current **A**tmosphere interactions **M**odel with **S**elf **C**onsistent magnetic field (**B**) is a unique code that combines a kinetic model of ring current plasma with a three dimensional force-balanced model of the terrestrial magnetic field. The kinetic portion, RAM, solves the kinetic equation to yield the bounce-averaged distribution function as a function of azimuth, radial distance, energy and pitch angle for four ion species (H^+, He^+, N^+, and O^+) and, optionally, electrons. The code is also set up to easily accept new ion species. The domain is a circle in the Solar-Magnetic (SM) equatorial plane with a configurable radial span (2 to 6.5 R_E by default). It has an energy range of approximately $100\,eV$ to $500\,KeV$. The 3-D force balanced magnetic field model, SCB, balances the $\mathbf{J} \times \mathbf{B}$ force with the divergence of the general pressure tensor to calculate the magnetic field configuration within its domain. The domain ranges from near the Earth's surface, where the field is assumed dipolar, to the shell created by field lines passing through the SM equatorial plane at radial distances which extend past the RAM domain to ensure proper overlap. The two codes work in tandem, with RAM providing anisotropic pressure to SCB and SCB returning the self-consistent magnetic field through which RAM plasma is advected.

RAM-SCB has grown from a research-grade code with limited options and static magnetic field (RAM) to a rich, highly configurable research and operations tool with a multitude of new physics and output products. This manual provides a guide to users who want to learn how to install, configure, and execute RAM-SCB simulations. While the code is designed to make these steps as straight-forward as possible, it is strongly recommended that users review the publications listed in the Bibliography to ensure a thorough understanding of the physics included in the model. Additionally, all users are asked to review the terms of use.

1.1 About This Manual

Users who want to install and begin quickly should start at Chapter 2, which outlines the path from installation to simulation with little detail. The installation process is discussed fully in Chapter 3. Instructions on performing simulations, as well as several example simulations, are given in Chapter 5. An outline of using this code in the Space Weather Modeling Framework is found in Chapter 7. Useful scripts included in the distribution are described, in brief, in Chapter 8. Finally, a complete list of all param file commands is found in Chapter 9.

1.2 TERMS OF USE & DISTRIBUTION POLICY

Use of the RAM-SCB software implies agreement with the terms herein. RAM-SCB is open source software that has been developed at Los Alamos National Laboratory (LANL).

The references below represent critical milestones for RAM-SCB. Please cite the appropriate work when using RAM-SCB to give the developers proper credit.

Citation	Information
Jordanova et al. [1996]	First description of ring current model (RAM) using dipolar magnetic field
Jordanova et al. [2006]	First extension of RAM for non-dipolar magnetic field and coupling with SCB
Zaharia et al. [2006]	Description of SCB model and coupling with RAM
Jordanova et al. [2010]	Full description of RAM extension for non-dipolar magnetic field
Welling et al. [2011]	First full description of one-way coupling with SWMF
Welling et al. [2015]	Description of two-way coupling of RAM-SCB with SWMF
Engel et al. [2019]	Improved implementation of SCB and robust numerics

Chapter 2

Quick Start Guide

This section provides a bare-bones approach for getting started with RAM-SCB. The instructions are aimed at users familiar with Unix-like environments. Tutorial-like guides for each step can be found in other chapters.

2.1 Installation

Installation and compilation of RAM-SCB requires a Fortran compiler (with OpenMP support), Perl interpreter, an MPI library compiled with the preferred Fortran compiler, and two external libraries which have additional requirements (most notably a C compiler, typically standard on Unix-like systems.) The required libraries are:

1. GNU Scientific Library
2. Unidata's NetCDF library

If these libraries are installed in non-standard locations, use environment variables to make them visible to RAM-SCB. The two variables to be set are `GSLDIR` and `NETCDFDIR`.

Installation of RAM-SCB is handled via the Config.pl script found in the installation directory. The `-h` option will print help, but the user will almost exclusively use Config.pl as follows:

```
Config.pl -install -compiler=gfortran -mpi=openmpi -openmp -ncdf=${NETCDFDIR} -gsl={GSLDIR}
```

Other compiler/MPI combinations are of course available, see Chapter 3.

Finally, use GNU Make to compile and, if desired, test:

```
make
make test
```

2.2 Execution

RAM-SCB is run from *run directories*, create one via `make`:

```
make rundir RUNDIR=~/desired_run_location
```

Run directories are soft-linked to the installation. This means you can move them freely and create multiple run directories for simultaneous simulations. A fresh run directory has everything you need to execute the code immediatetly, simply start the executable.

```
./ram_scb.exe
```

To customize the simulation, edit the PARAM.in file. Chapter 9 has a complete description of all parameters. Additional input files required must be placed in either the `input_ram` or `input_scb` subdirectories; output is located in similarly named locations.

Output data comes in two formats: simple ASCII and NetCDF files. While there are a plethora of tools available, we recommend users try the Python-based tools discussed in Chapter 6.

Chapter 3

Installation

Installation and compilation of RAM-SCB requires a Fortran compiler (with OpenMP support), MPI compiled against the chosen Fortran compiler, a Perl interpreter, and several external libraries which have additional requirements (most notably a C compiler, typically standard on Unix-like systems.) The configuration is done with the Config.pl script; compilation with GNU Make. The Config/Make system follows the tiered makefile standards of the Space Weather Modeling Framework (SWMF) project; users of the SWMF should feel right at home using RAM-SCB.

3.1 Installation of Required Libraries

There are two required libraries that must be installed to use RAM-SCB: GNU Scientific Library and Unidata's NetCDF library. It is possible to find pre-compiled binaries for each library, allowing the user to skip the configure, make, and install steps. However, if a binary cannot be found that matches your system and compilers, you will be forced to install from source. Some systems may already have these libraries installed; be sure to check before going through unneccessary installation work.

3.1.1 Libraries From Modules

Well maintained clusters often use the `module` interface for loading and unloading software packages. Be sure to explore the available packages on a new system before going through the arduous task of installing from source!

```
module avail
```

will list all available software libraries on the system. If a needed software is not available it is a good idea to contact your computing support to see if they can build the module for you before installing from source. You can load a library, unload a library, and list loaded software using the module commands:

```
module load package_name
module unload package_name
module list
```

As an example, here is what you would type on LANL's Grizzly HPC cluster to get started with RAM-SCB:

```
module purge

module load gcc/10.3.0
module load hdf5-serial/1.10.7 netcdf-serial/4.4.0 openmpi/4.1.1
```

The first command purges all currently loaded modules to make sure there is no interference from previously loaded modules. The next command loads gcc/gfortran and GSL. Finally we load the modules for HDF5 and NetCDF4 (NetCDF4 depends on HDF5), and MPI. The Perl installation on this system includes core CPAN libraries, including the *Version* package required by RAM-SCB's configuration script. All dependencies are now loaded and ready to use.
It is often helpful to place commands to configure the environment in your shell configuration script (e.g., `~/.bashrc` or `~/.cshrc` depending on your shell) to load modules upon login. Note that if you are running using a job scheduler (e.g. slurm, moab, etc...), placing the module commands in your shell configuration script may not be adequate and they may need to be loaded in your job script.

Common issues

Most installations of NetCDF4 and GSL should come with command line utilities to help determine library locations and flags. Specifying `-gsl` and `-ncdf` flags when running `Config.pl`should auto-detect the install locations. If the command line utilities were not built, the paths must be provided, e.g. `-gsl=[...]`.

On some systems the user may find that running `Config.pl` shows the warning "Can't locate share/Scripts/Config.pl in @INC". In this case, add the RAM-SCB directory to the Perl path; in `bash` this is done with `export PERL5LIB=‘pwd‘`.

Finally, not all Perl installations provide CPAN modules. In case `Config.pl` reports an error about `CPAN::Version`, please work with your system administrator to install the module, or see
the CPAN module installation instructions.

3.1.2 Installation From Source

If installing from source, follow the instructions provided with the software, making sure to use the same compiler that you are using with RAM-SCB to install the dependencies.

If you change the library locations by any method, you should reinstall the code. These library paths are resolved at runtime, and changes in path may result in the code not finding dependencies or finding incompatible versions.

Library	Environment Variable	Config.pl Switch
GNU Scientific Library	`GSLDIR=[...]`	`-gsl=[...]`
NetCDF	`NETCDFDIR=[...]`	`-netcdf=[...]`

Table 3.1: List of required libraries and methods for expressing their location to RAM-SCB. Note that the Config.pl switches override environment variables. If none are given, the Fortran compiler will search in the default library location.

3.2 Installation, Configuration, and Compiling

`Config.pl` handles the installation and configuration of RAM-SCB. To view the installation and configuration status, or to view help, use the following commands:

```
Config.pl
Config.pl -h
```

To install the code, use the `-install` flag:

```
Config.pl -install
```

Although RAM-SCB will try to use reasonable defaults based on your system, there are a number of flags that allow you to customize your installation:

- Use `-compiler` to select the Fortran 90 compiler. Common choices include gfortran, ifort, and pgf90.
- The `-mpi` flag allows the user to pick which version of MPI to use. Choose from openmpi, mpich, mpich2, and Altix. Alternatively, the `-nompi` flag may be set with no value. This option is fragile.
- Set `-ncdf` to the path of the NetCDF library installation. If used, this flag overrides the environment variable NETCDFDIR.
- Set `-gsl` to the path of the GSL library installation. If used, this flag overrides the environment variable GSLDIR.

To exemplify a typical installation, imagine a machine with several different Fortran compilers available. For each compiler available, the user has a corresponding installation of GSL. The user will use this command to properly install RAM-SCB:

```
Config.pl -install -compiler=pgf90 -mpi=mpich2 -gsl=~/libs/gsl_portland/
```

There are other Config.pl options that set up real precision, debug flags, and optimization level. Use `Config.pl -h` to learn about the available options.

After the code has been properly configured, compilation is simple:

```
make
```

Compilation is most likely to fail for two reasons. The first is RAM-SCB not finding MPI or another key library. The second is MPI or an external library that is installed using a mix of different Fortran compilers. Be vigilant when installing each library!

To remove object files before a fresh compilation, use

```
make clean
```

To uninstall RAM-SCB, simply use

```
Config.pl -uninstall
```

3.3 Testing the Installation

Running the RAM-SCB tests is an excellent way to evaluate the success and stability of your installation. To run the tests, simply type

```
make test
```

This will compile RAM-SCB, create a run directory, perform a short simulation and compare the results to a reference solution. If there is a significant difference between the test and reference solution, the test will fail. Details can be found in `*.test` files in the RAM-SCB directory.

For development there is also a unit test suite, which can be run by

```
make unittest
```

As new features are added, or as bugs are fixed, new unit tests to be added to the test suite to ensure correct behavior.

For a full description of running and interpreting test results, see Chapter 4.

3.4 Building Documentation

To generate a PDF of the latest User Manual, type

```
make PDF
```

The document will be located in the `doc/` directory.

Chapter 4

Testing RAM-SCB

RAM-SCB comes packaged with a test suite to determine if the current installation operates correctly and produces the expected results. Different tests evaluate different code capabilities and options. Testing is an especially powerful development tool for evaluating the impact, intentional or not, of changes to the source code. A summary of the tests and associated commands can be generated by using `make test_help`. Current test status can be seen at the github repository.

4.1 Using and Interpreting Test Results

Tests are called through GNU `make`. To run all available regression and functional tests, simply use the command

```
make test
```

in the RAM-SCB installation directory. To run a specific test (listed below), call that test by name:

```
make test1
```

A test simulation performs the following actions:

1. Compile the code with any options required by the particular test.
2. Create a run directory entitled `run_test`. If one exists, it will be deleted, so use caution when using this name for any other purpose besides automated testing.
3. Copy the required input and parameter files into the new test run directory.
4. Run the code. Test simulations are only long enough to perform the features being tested.
5. Perform a specialized version of `diff`, included in the distribution, to compare the results produced by the test simulation against reference solutions stored in the output directory of the installation location. If the two files have values that differ by a certain amount (default is an absolute difference of 10^{-9}), the test will fail.

Each of these steps can be called individually by using `make` *nametest_stepname*, where the step names are `compile`, `rundir`, `run`, and `check`.

If, at any point, the test fails, the details will be recorded in *nametest*.diff in the installation directory. If the test is successful, this file will be created but be empty. Tests that fail at the comparitive stage will list all of the differences. When this happens, additional tools to examine the file differences are recommended. An excellent visualization of file differences can be rendered with `gvimdiff`, an open source tool that is already installed on many Linux machines.

4.2 Description of Available Regression Tests

4.2.1 Test 1

Command: `make test1`

This is the most basic test for RAM-SCB. It runs RAM-SCB for 900 seconds using Volland-Stern electric field, constant dipole magnetic field (no SCB calculation), and LANL fluxes. **SHIELDS-RC** users should rely on this test.

4.2.2 Test 2

Command: `make test2`

Test 2 repeats the previous test, but stops half way through the simulation, writes restart files, and restarts the run. This exercises the code's ability to seamlessly restart a simulation at any arbitrary point.

4.2.3 Test 3

Command: `make test3`

Test 3 activates SCB, using a dipole outer boundary condition and the Weimer 2000 empirical electric field. This is the base test for SCB functionality and ionosphere-to-equator electric potential mapping.

4.2.4 Test 4

Command: `make test4`

Test 4 repeats the previous test, but stops half way through the simulation, writes restart files, and restarts the run. Like test2, this exercises the code's ability to seamlessly restart a simulation at any arbitrary point.

4.2.5 EMIC Test

Command: `make testEMIC`

This test is a repeat of Test 1 with FLC and EMIC scattering turned on

4.3 Unit Testing

In addition to the regression and functional testing, which is performed using short simulations that exercise a range of functionality to reproduce known results, RAM-SCB also includes a unit test suite.

The unit test suite is called using GNU `make`. To run, use

```
make unittest
```

Chapter 5

Performing Simulations

5.1 Making a Run Directory

Running RAM-SCB occurs not in the installation directory, but through *run directories* – special directories that keep your simulations separate from the installation and other simulations. To create a run directory, simply use the `make` interface:

```
make rundir
```

`make` will unpack some default inputs and organize a new directory aptly named `run`. Note that if a run directory exists that is called "run", RAM-SCB will not over write it and `make` will complain.

Run directories soft link to the installation directory, so as long as you don't move, uninstall, or otherwise break your installation, you can put the run directory where ever you want. Furthermore, you are not limited to a single run directory. Adding more allows you to run more simulations simultaneously from a single installation. Using the `PARAM` interface, described below, you can perform many very different simulations at once without re-compiling the code. The name and location of the run directory can be set using the RUNDIR variable as follows:

```
make rundir RUNDIR=/location/to/new/rundir
```

If running on a HPC cluster (or some other system with designated run spaces) it is important to place the run directories in the correct location. This can be done using the RUNDIR variable as above, or by moving the directory after to has been created:

```
make rundir
mv run /location/to/new/rundir
```

Both of the above methods produce the same result.

5.2 Inside a Run Directory

Let's look inside a typical run directory:

Figure 5.1 shows the default run directory organization. The IM directory is where most input and output occurs; subdirectories shaded green are soft-linked to the top run level for convenience. These directories are self explanatory outside of `output_swmf`, which is for runs that use SWMF output in stand-alone mode. Enough inputs are given in the input directories for a sample simulation right away. The red box in Figure 5.1 denotes the linked executable. The final files, denoted as yellow boxes, are the final input files. `w2k.dat` is the Weimer 2000 empirical model data file, `omni.txt` is solar wind input file from the OMNI database, and `RamIndices.txt` is a list of geomagnetic indices. The necessity of these three files is discussed below.

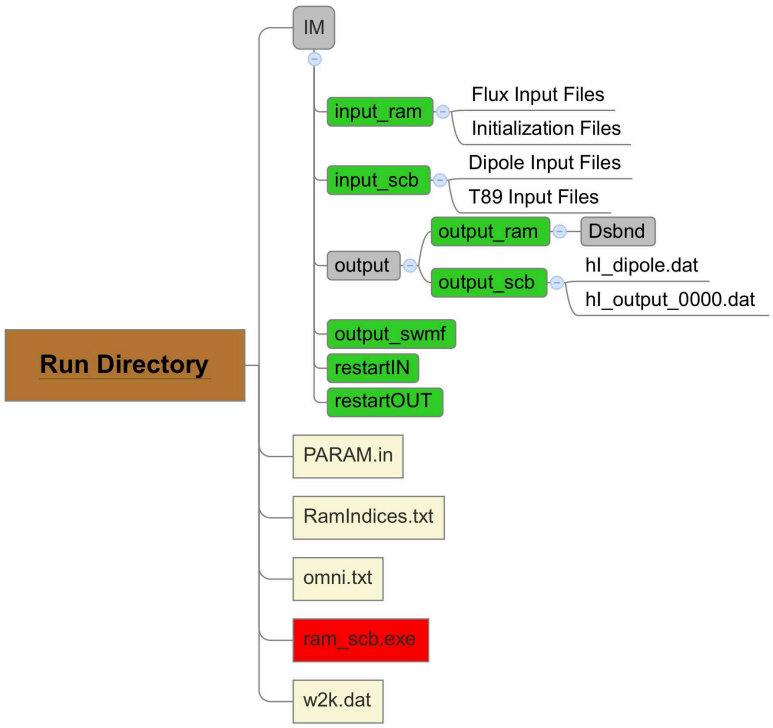

Figure 5.1: Run directory layout. Gray and green rounded boxes indicate subdirectories; green boxes are directories that are soft linked one level up for convenience. Yellow boxes are key input files, the red box is the linked RAM-SCB executable, and other data input files are listed without boxes.

Finally, there is `PARAM.in`, the run-time configuration file that is used to tell RAM-SCB what and how it should simulate. The `PARAM` interface, identical to the one used by the SWMF, is simple. Any line of text in the file not preceeded by a # symbol is ignored, so comment your file to your heart's content. Any text preceeded by #, however, is a interpreted as a `PARAM` command. A generic command takes the form,

```
#COMMANDNAME
Parameter1    ParameterName
Parameter2    ParameterName
...
ParameterN    ParameterName
```

The exact values of the parameters are command specific; the parameter names are simply tab-delimited from the parameters and are not needed (keeping them is recommended to keep your file clear and easily understood!) Here are a few commands in action:

```
#STARTTIME
2006 iYear
7 iMonth
19 iDay
0 iHour
```

```
0 iMinute
0 iSecond

#OUTERBOUNDARY
LANL NameBoundPlasma
T89C NameBoundMag

#VARIABLEDT
T    DoVariableDt

#STOP
-1   MaxIteration
300   tSimulationMax
```

First, take note that the parameter values can take many forms: rational numbers, integers, logical values, or strings of text. Remember that the trailing descriptions are not read by RAM-SCB; they are there to help you remember what each value does. In order, this list of `PARAM` commands sets the start date and time of the simulation, sets the plasma and magnetic field outer boundary conditions, turns on variable timestepping, and tells the code to run for unlimited iterations but only 300 seconds. All commands are described in detail in Chapter 9.

Once your `PARAM` file is customized, execute RAM-SCB.

```
./ram_scb.exe
```

RAM-SCB will begin; a lot of information will be written to screen that updates the user of its progress. Experienced Unix users know all of the tricks to capture that output; the following copies the output to file while preserving the output to screen:

```
./ram_scb.exe | tee runlog.txt
```

5.3 Required Inputs

Ring current models have three basic requirements: plasma fluxes at the outer boundary, magnetic field, and electric field throughout the equatorial plane. RAM-SCB fulfills these requirements slightly differently than other ring current models thanks to its sophisticated self-consistent magnetic field. Electric field may be specified either in the equatorial plane directly or as an ionospheric potential that is mapped along SCB field lines. Magnetic field is supplied as a shell surrounding the SCB domain, which consists of the body containing all field lines passing through the RAM equatorial domain. Plasma fluxes remain a simple specification about the outer boundary of RAM. In RAM-SCB, there are many ways to specify these; refer to Chapter 9 for commands that select inputs.

Plasma fluxes can be provided one of two ways: the first is coupling with the SWMF, covered in Chapter 7. The second is to use "GeoMlt" files, a product of LANL geosynchronous observations. These files can be obtained on request from the data caretakers; sample files are provided with the distribution. These files come in pairs per day; an ion and electron file exists for each day for which there is data coverage. Because the source observations do not readily differentiate between different ion species, the $K_P/F_{10.7}$ dependent empirical relationship of *Young et al.,* [1982] is used internally to divide up the Hydrogen flux in the input files. The K_P and $F_{10.7}$ indices are packaged in the RamIndices.txt input file; the user need not worry about them for historical simulations.

Convection electric potential can be specified in a plethora of ways. The simplest is to use the K_P dependent Volland-Stern electric field. This requires no additional input files beyond RamIndices.txt. The Weimer 2000 empirical model can also be used; like Volland-Stern, it is an internal calculation. Because

this model depends on solar wind conditions, users must supply an input file that contains this data from the OMNI database. The Weimer 2000 potential is mapped along SCB field lines to the equatorial plane automatically. Finally, SWMF electric fields can be used; this is covered later. Though it is possible to supply Weimer 2000 and SWMF potentials in the equatorial plane without SCB mapping, this approach requires extensive work by the user as is not recommended. See the entry on the command `#EFIELD` for additional details on file formats and obtaining these inputs.

Magnetic field boundary conditions can be supplied via three sources: a simple dipole model, from one of the Tsyganenko empirical models, or from the SWMF. When using a dipole, it is possible to use a dipole with or without the SCB calculation. Tsyganenko 89 (T89) input files are provided with the distribution and require no additional input from the user. Input from more complex Tsyganenko models require substantial work to acquire, they are not recommended for most users. See the entry for the `#OUTERBOUNDARY` command in Chapter 9 for more details.

All files listed above must be placed in their respective input files in the run directory. Other input files required by advanced options (e.g. virtual satellites) should be placed in the top level of the run directory. Each run directory requires its own set of inputs.

5.4 Restarting Simulations

It is often desirable to split a simulation up into several parts. For example, you may want to split a very long simulation up into separate executions, or change parameters at certain parts of a storm, or perhaps a simulation was interrupted undesireably. Restart files allow you to continue a simulation seamlessly.

Throughout a simulation, restart files are being written to the `restartOUT` folder of the run directory. The files are written at a set frequency and at the end of a successful run (both of these options are configuarble through the `PARAM` interface.) Only the most recent restart is saved, users can periodically pull these files out of `restartOUT` if they prefer a back log of restarts. These files contain the full distribution functions as well as the start time, current time, and current iteration at the point that the restart was written.

To restart a run, first move restart files from `restartOUT` to `restartIN`. Then, either create a new `PARAM.in` file or edit the existing one such that no `#STARTTIME` command is present (remember, the restart file knows this information already), the run time exceeds the current time listed in the restart file, and the command `#RESTART` is included. This tells RAM-SCB to restart the simulation rather than start anew. Restarts are carefully implemented to preserve files in the output directories and continue the simulation as if there was no interruption.

For an example of restarting a simulation, run Test 2 and inspect the `PARAM.in` files as well as the contents of `restartIN` and `restartOUT`.

5.5 Output

RAM-SCB, by default, has a rich output set. This can be expanded by activating other output file types via the `PARAM` interface, be sure to review the commands listed in Chapter 9. Table 5.1 summarizes the output that can be generated by RAM-SCB. Visualization of these files is covered in Chapter 6.

Type	Extension	Format	Default?	Contents
Log file	`*.log`	ASCII	Yes	Dst, integrated values
Pressure files	`*.dat`	ASCII	Yes	$\perp$ and $\parallel$ partial pressures
E-Field files	`*.in`	ASCII	No	Equatorial electric potential
Boundary files	`*.dat`	ASCII	No	Fluxes at the outer boundary
RAM flux files	`*.nc`	NetCDF	Yes	Full equatorial flux information
Virtual Satellites	`*.nc`	NetCDF	No	Satellite specific values
Field integral files	`*.dat`	ASCII	Yes	h, I geometric integrals
Magnetic field file	`*.dat`	ASCII	No	Full 3D SCB field
Potential file	`*.dat`	ASCII	No	Ionospheric electric potential
3D pressure file	`*.dat`	ASCII	No	Full 3D anisotropic pressure

Table 5.1: List of available output files.

Chapter 6

Output Visualization

RAM-SCB output can be visualized in a number of different ways depending on the user's tastes and preferences. However, a standard library for opening, manipulating, and plotting exists as a sub-module of Spacepy. For 3d visualization of the domain, some basic tools are provided to enable the use of VTK-powered tools such as ParaView.

6.1 SpacePy

Primary support for visualization of RAM-SCB input and output files is provided through the open-source SpacePy library. The latest versioned release of SpacePy can be obtained from the Python Package Index using

```
pip install spacepy
```

Alternatively, the latest stable development version of SpacePy can be obtained from the SpacePy git repository. For documentation and examples, please see https://spacepy.github.io/autosummary/spacepy.pybats.ram.html

6.2 VTK: The Visualization Toolkit

Basic support for VTK-based visualization is provided in the `Scripts/viz` subdirectory.

6.2.1 VTK Data from RAM-SCB Restart Files

'convertRAMrestart.py' batch converts NetCDF restart files to '.vtp' and '.vtu' files

6.2.2 Visualization using VTK/ParaView

Convenience routines to make VTK files with the seed locations are in `makeCustomSource.py`.
A standard view of the RAM-SCB domain can be generated using Paraview's scripting capability through the `pvpython` interpreter packaged with ParaView. Running `visualizeRAM.py` with `pvpython` will generate output PNG files showing RAM quantities on the equatorial plane and a cutaway of the SCB field.
Other packages such as PyVista, MayaVi, VisIt, etc. can all be used with the VTK files generated by `convertRAMrestart.py`.

Chapter 7

RAM-SCB and SWMF

RAM-SCB may be used as a component to the Space Weather Modeling Framework. Under this mode, RAM-SCB receives initial and boundary conditions from the Framework's other components and returns plasma properties to create a two-way coupled system.

Currently the extant documentation for using RAM-SCB as the IM component in SWMF is out of date. For the time being, we recommend that interested users contact the RAM-SCB developers for assistance.

Chapter 8

Included Scripts

RAM-SCB comes packaged with many helpful scripts that aid the user. These scripts are written either in Perl or Python with the standard libraries only. This allows them to be run on many systems without needing to install additional software as both languages are ubiquitous in Unix-like environments. The subdirectory `Scripts` contains several helpful scripts that are unique to RAM-SCB while the subdirectory `Share/Scripts` contains scripts designed for SWMF-like applications but often useful for RAM-SCB as well. For visualization scripts, please see Chapter 6.

While all RAM-SCB scripts are listed here, only a handful of SWMF scripts are described. Be sure to explore the `Share/Scripts` directory before you constructing your own solutions to common problems (e.g. endian conversions, PARAM-checking, etc.)

8.1 Config.pl (SWMF Script)

`Config.pl` is part of the SWMF *Config* system for installing and pre-configuring RAM-SCB in both stand-alone and component modes. Use of `Config.pl` is covered extensively in Chapter 3.2.

8.2 DiffNum.pl (SWMF Script)

`DiffNum.pl` is a powerful, quantitative re-write of the popular `diff` utility. It compares two files, finds and quantifies differences in any numerical entries, and, if any are found, lists the differences and raises an exception. The main purpose of `DiffNum.pl` is to find and

Usage:

```
DiffNum.pl [options] File1 File2
```

Common options include `-a=VALUE` and `-r=VALUE`, which allow the user to ignore absolute and relative differences less than `VALUE` and `-t` which turns off the comparison of text.

8.3 CatLog.py (RAM-SCB Script)

A common problem in both RAM-SCB and many SWMF modules is many fractured, separate log files from a single simulation that required several restarts. Often, these log files overlap in time because a simulation did not complete and restarting results in re-simulating a small portion of the run. Manually concatenating these log files together into a single seamless, monotonic file can be time consuming.

Usage:

```
CatLog.py [options] log1 log2 [log3] [log4]...[logN]
```

Files 2 through N will be appended to the first file. Unix wild-card characters can be used to get file-globbing effects. If the headers of any of the trailing log files does not match the leading file, it is discarded. If the leading file includes a wild-card character, the files are arranged and appended in alpha-numeric order. Available options include `-debug` (print debug information), `-rm` (remove all but first log file), and `-nocheck` (deactivate checking for overlapping entries.) See `CatLog.py -h` for examples.

8.4 Updating Input Files

8.4.1 updateRamIndices.py

This script retrieves definitive F10.7 flux data and Kp data and appends to the main RamIndices input file used by RAM-SCB.

8.4.2 provisionalIndices.py

This script will update the RamIndices file *in a run directory* with provisional and predicted indices. This facilitates running RAM-SCB in a nowcasting or forecasting mode.

8.4.3 rtUpdateParam.py

This script will update a `PARAM.in` file to enable a restart. The intended purpose was for realtime operations, but is more broadly useful. It provides an automated way of, if necessary, replacing the `STARTTIME` block with a `RESTART` directive, and then further updating the stop time for the simulation.

Chapter 9

Complete List of Input Commands

The content of this chapter is generated from the PARAM.XML file. The XML file can be read with an editor and can be used for creating PARAM.in files by copying small parts from them.

The transformation of the XML format into LaTex is done with the share/Scripts/XmlToTex.pl script. This script generates index terms for all commands, which are used to create an alphabetical index at the end of this chapter.

9.1 Input Commands for the RAM_SCB: IM Component

List of IM commands used in the PARAM.in file

9.1.1 Base

#STARTTIME command

```
#STARTTIME
2006            iYear
7                       iMonth
19                      iDay
0                       iHour
0                       iMinute
0                       iSecond
0.0                     fracSecond
```

The STARTTIME command sets the integer year, month, day, hour, minute and second at which the simulation begins. This command is only used in standalone mode and only for the first session.

#STOPTIME command

```
#STOPTIME
2006            iYear
7                       iMonth
19                      iDay
6                       iHour
0                       iMinute
0                       iSecond
0.0                     fracSecond
```

The STOPTIME command sets the integer year, month, day, hour, minute and second at which the simulation stops. This command is only used in standalone mode.

#EVENT command

```
#EVENT
110105  Event
```

Sets the short name description of the run

#VERBOSE command

```
#VERBOSE
```

If present, RAM-SCB will output additional information about performance and progress during the simulation

#RESTART command

```
#RESTART
```

If present, RAM-SCB will restart a simulation using the information stored in the restart files that the user must place in the `restartIN` folder in the run directory. Start time and simulation conditions are read from the restart files. This command can not be used with `#STARTTIME`.

#DESCRIPTION command

```
#DESCRIPTION
Simulation of the Sep. 1st, 2005 storm.
```

The StringDescription string can be used to describe the simulation for which the parameter file is written. The #DESCRIPTION command and the StringDescription string are saved into the restart file, which helps in identifying the restart files. It is often added to NetCDF files to help describe the output.

9.1.2 Ram

#MAXTIMESTEP command

```
#MAXTIMESTEP
100.0   MaxHalfStep
```

Set the maximum half time step that can be taken by RAM. Even if the CFL limit allows for a larger time step, the code will not surpass the limit set here. Note that this is the *half* time step taken in the time-splitting scheme; a full timestep is two times the half time step.

#MINTIMESTEP command

```
#MINTIMESTEP
1.0     MinHalfStep
```

Set the minimum half time step that can be taken by RAM. Even if the CFL limit specifies a smaller time step, the code will not surpass the limit set here. Note that this is the *half* time step taken in the time-splitting scheme; a full timestep is two times the half time step.

#VARIABLEDT command

```
#VARIABLEDT
T       DoVariableDt
```

Select between a variable timestep that is governed by the CFL number or a semi-static timestep that only decreases during periods of high geomagnetic activity as determined by the Kp index. The former will allow the timestep to grow as large as the CFL number will permit, while the latter will use a timestep of 5 seconds during low (less than 5) Kp or 1 second during higher Kp.

#SPECIES command

```
#SPECIES
4                               nS
_H _O He _e             NameVar
F                               FixedComposition
```

Defines the species to run in RAM. See the RAMSpecies_README for more information.

#NITROGEN_PERCENT command

```
#NITROGEN_PERCENT
0       NitrogenPercent
```

If using nitrogen in the species this command will convert a percent of the oxygen in the simulation into nitrogen.

#USEWPI command

```
#USEWPI
F       DoUseWPI
F       DoUseBASdiff
F       DoUseKpdiff
F       DoUseEMIC
```

Flag to turn on pitch angle diffusion. The default setting for the Kp-based version uses Kp=0 coefficients. Turning this flag on interpolates Kp-dependent diffusion coefficents for Kp in the range (0-4). Different options are available if selecting DoUseBASDiff. EMIC wave pitch angle diffusion is used for ions wave-particle interaction.

#USEFLC command

```
#USEFLC
F       DoUseFLC
F       DoWriteFLCDiffCoeff
Flag to turn on the field line curvature scattering of ions and write out the pitch angle diffusion coeff
```

#COULOMB command

```
#COULOMB
F       DoUseCoulomb
```

Flag to turn on Coulomb interactions. By default no Coulomb interactions are used.

#RAMLIMITER command

```
#RAMLIMITER
1.5     BetaLim
```

Set the beta factor for the MC limiter implemented in RAM. A value of 1 is equivalent to using the Min-Mod limiter; a value of 2 is equivalent to the superbee limiter. 1.5 is the default; many codes that implement this limiter use a value of 1.5.

#RAMGRID command

```
#RAMGRID
20      nR
25      nT
35      nE
72      nPa
```

- nR: Number of radial grid points used in RAM
- nT: Number of toroidal grid points used in RAM
- nE: Number of energy bins used in RAM
- nPa: Number of pitch angle bins used in RAM

9.1.3 Scb

#TIE_SCB_TO_RAM command

```
#TIE_SCB_TO_RAM
```

If this is present then the SCBTimeStep will be read as number of RAM iterations to do, not seconds

#PRESS_MODE command

```
#PRESS_MODE
SKD             PressMode
```

Method used to extrapolate RAM pressure to all of SCB grid in events where the SCB grid extends beyond the RAM domain. ***Current options for SCB pressure extrapolation:***

1. **SKD**: Uses Spence-Kivelson empirical formula
2. **ROE**: Uses Roederer empirical formula
3. **EXT**: Uses linear extrapolation
4. **FLT**: Uses a flat extension
5. **BAT**: Uses BATS-R-US equatorial pressure

#SCB_FIELD command

```
#SCB_FIELD
0.0             ConstZ
0.2             ConstTheta
```

- ConstZ: Groups points along the midnight axis using the equation [zetaVal+constZ*SIN(zetaVal)]
- ConstTheta: Groups points at the equator using the equation [thetaVal + constTheta*sin(2.*thetaVal)]

#SCB_CONVERGENCE command

```
#SCB_CONVERGENCE
0.9             ConvergenceDistance
```

Minimum accuracy that needs to be reached regardless of other factors (between 0 and 1)

#SCBSMOOTHING command

```
#SCBSMOOTHING
1               PressureSmoothing
11              SavitzyGolayIterations
```

Method used to smooth pressure for SCB from RAM ***Current options for SCB pressure smoothing:***

1. Savitzy-Golay Filter
2. B-Spline smoothing
3. Gaussian Filter
4. 1+3

#SCB_GHOSTCELLS command

```
#SCB_GHOSTCELLS
0       psiChange
4       theChange
```

The number of ghost cells to add to the SCB force balancing SOR

#SCBFLAGS command

```
#SCBFLAGS
F       Isotropic
F       ReduceAnisotropy
T       BetaExtrapolation
F       AzimuthalOffset
F       EmptyLossCone
T       AdaptiveMesh
```

- Isotropic: Whether to use isotropic pressure mappings in SCB (does not effect RAM)
- ReduceAnisotropy: Whether to change the anisotropy to a marginally mirror-stable
- BetaExtrapolation: Whether to extrapolate Beta Euler potential values to the outside boundary or use the previous point (Currently unused)
- AzimuthalOffset: Whether to set equidistance at most problematic time (if T) or keep it at midnight (if F)
- EmptyLossCone: Whether to use a filled loss cone (if F) or a more realistic empty loss cone using M. Liemohn's 2004
- AdaptiveMesh: Whether to use Mesh refinement in magnetic flux, so that one has equidistant magnetic flux surfaces

#SCBDETAILS command

```
#SCBDETAILS
F       SORDetail
T       EnergyDetail
F       ForceBalanceDetail
F       PressureDetail
```

- SORDetail: Whether to compute extra Sucessive Over Relaxation information
- EnergyDetail: Whether to compute DPS Dst and other energy information
- ForceBalanceDetail: Whether to compute global force balance before and after SCB runs, this will cause additional output
- PressureDetail: Whether to compute the full 3D pressure profile before an SCB run, this will cause additional output

#SCBGRID command

```
#SCBGRID
71      nTheta
45      nPsi
73      nZeta
```

- nTheta: Number of theta (polar) grid points used in SCB
- nPsi: Number of psi (radial) surfaces used in SCB
- nZeta: Number of zeta (azimuthal) surfaces used in SCB

9.1.4 Components

#COMPONENT_TIMESTEPS command

```
#COMPONENT_TIMESTEPS
300.0   SCBTimeStep
300.0   BCTimeStep
300.0   EFTimeStep
```

- SCBTimeStep: Amount of time (in seconds) between calls to SCB. If set to 1.0 it will call SCB on every RAM time step
- BCTimeStep: Amount of time (in seconds) between reading boundary flux files (or from SWMF). If set to 1.0 will update on every RAM time step
- EFTimeStep: Amount of time (in seconds) between updating electric fields. If set to 1.0 it will update every RAM time step

#PLASMAPSHERE command

```
#PLASMAPSHERE
F                       DoUsePlasmasphere
Carpenter       PlasmasphereModel
Analytic        TauCalculation
```

Determines whether to use a plasmasphere and which plasmasphere model to use if desired

#FLUX_CAP command

```
#FLUX_CAP
1e10    ElectronFluxCap
1e8             ProtonFluxCap
```

Sets a flux cap at the boundary for both electrons and protons

#OUTERBOUNDARY command

```
#OUTERBOUNDARY
LANL    NameBoundPlasma
DIPL    NameBoundMag
MAXW    NameDistribution
```

Set the outer boundary conditions for RAM-SCB. Based on these settings, different files are expected in the respective input directories to suppy magnetic field and energy flux at the outer boundary of the codes.
Current options for plasma boundary conditions:

1. **LANL**: Use fluxes calculated from LANL Geosynchronous measurements. Composition is determined from the Young et al. empirical relationship based on Kp and F10.7.
2. **SWMF**: Use fluxes calculated from the Space Weather Modeling Framework. If in coupled mode, values are calculated on-the-fly and no input files are required.

Current options for magnetic field boundary conditions:

1. **DIPS**: Use a simple dipole to constrain the SCB field.
2. **DIPL**: Use a simple dipole throughout; no SCB calculation.
3. **SWMF**: Use the magnetic field from the Space Weather Modeling Framework. If RAM-SCB is in coupled mode, these values are calculated and obtained on-the-fly rather than read from input files.
4. **T89D**: Use the Tsyganenko 89c empirical model. As this field depends only on Kp, input files are provided in the RAM-SCB distribution.
5. **T96D**: Use the Tsyganenko 1996 empirical model.
6. **T02D**: Use the Tsyganenko 2002 empirical model.
7. **T04D**: Use the Tsyganenko 2004 empirical model.

Current options for plasma distribution shape:

1. **MAXW**: Assume a Maxwellian distribution.
2. **KAPA**: Assume a Kappa distribution.

#EFIELD command

```
#EFIELD
IESC            NameEfield
F                       UseEfInd
```

Set the source for the convective electric field in RAM-SCB. The choice made will set the type of input file required at runtime.

Current options for the electric field:

1. **IESC**: SWMF IE-component electric field mapped to the equatorial plane via RAM-SCB field lines.
2. **VOLS**: K_P-based Volland-Stern empirical electric field (internal VS calculation).
3. **WESC**: Weimer 2001 empirical electric field mapped to the equatorial plane via RAM-SCB field lines (internal W2K calculation).

The parameter UseEfInd turns the use of induced electric field on or off. Default is no induced electric field.

The **VOLS** and **WESC** are *internal* calculations and do not require these additional files, but carry the requirements of their respective underlying models. The Volland-Stern model requires the K_P index, which is provided for historical simulations. The Weimer 2000 empirical model requires upstream solar wind conditions, which can be obtained from the OMNI database. This data must be placed into the run directory in a file named *omni.txt*.

9.1.5 Input

#OMNIFILE command

```
#OMNIFILE
omni.txt        NameOmniFile
```

The WESC (Weimer electric field traced along SCB field lines) electric field selection calculates Weimer's empirical electric field on-the-fly. To do this, solar wind inputs are required from the Omni database. The ascii file that contains these inputs should either be called "omni.txt" and be located in the run directory (default behavior) or this command should be used to point the code in the correct location.

#INDICES_FILE command

```
#INDICES_FILE
RamIndices.txt          NameIndicesFile
```

Name of the KP and F10.7 indices file needed for RAM (included in default installation)

#BOUNDARY_FILE_PATH command

```
#BOUNDARY_FILE_PATH
IM/input_ram/           BoundaryPath
```

Location where the flux boundary files can be found (needed by RAM)

#QINDENTON_FILE_PATH command

```
#QINDENTON_FILE_PATH
IM/input_scb/           QinDentonPath
```

Location where the QinDenton files can be found (Used for the T##X magnetic field models)

#INITIALIZATION_FILE_PATH command

```
#INITIALIZATION_FILE_PATH
IM/input_ram/           InitializationPath
```

Location where the RAM Initialization.nc file can be found (included in default installation)

#TSO7_DIRECTORY command

```
#TS07_DIRECTORY
IM/input_scb/           TS07Directory
```

If using the TS07 magnetic field model, the code will look for the TS07 inputs in this directory

9.1.6 Output

#WRITE_BOUNDARY command

```
#WRITE_BOUNDARY
```

If present, the RAM boundary flux will be written to file each time it is updated

#WRITE_POTENTIAL command

```
#WRITE_POTENTIAL
```

If present, the RAM electric potential will be written to file each time it is updated

#SAVERESTART command

```
#SAVERESTART
3600.0          DtSaveRestart
T               DoSaveFinalRestart
```

Configure when restart files are saved. The first parameter sets the frequency, in seconds, that restart files are saved during a simulation. The second parameter toggles saving restarts at the successful completion of a simulation. Defaults are shown.

#TIMEDRESTART command

```
#TIMEDRESTART
F               TimedRestartFiles
```

If true the restart files will be appended with the simulation date and time.

#LOGFILE command

```
#LOGFILE
60.0    DtWrite
```

Specify the write frequency of the RAM-SCB log file. Default value is shown. Note that frequent writes for any output file may limit the maximum timestep taken by RAM-SCB.

#SAVEFLUX command

```
#SAVEFLUX
3600    DoSaveFlux
```

Specify the write frequency of equatorial flux values for all pitch angles and energies to NetCDF file. These files are powerful research tools but large and may be undesirable for long simulations.

#SATELLITE command

```
#SATELLITE
60.0            DtOutput
2               nSatellite
satellite1.dat  NameTrajectoryFile
satellite2.dat  NameTrajectoryFile
F               DoUseVAPini
```

The solution of RAM-SCB can be extracted along a satellite's path as the simulation progresses by using the #SATELLITE command. Simply set the number of virtual satellites to be included in the simulation, the time frequency to write the output in seconds (minimum/default is 10), and the location of each corresponding Trajectory File. For each Trajectory File given, RAM-SCB will produce a NetCDF file containing the solution along the satellite's trajectory at a time frequency of sixty seconds. Note that reducing DtOutput to small values can reduce the maximum RAM-SCB time step as a result of ensuring that the file is written every DtOutput.

Satellite Trajectory Files contain the trajectory of the satellite. They should have the following format:

```
#START
 2004  6  24   0   0  58   0  2.9  -3.1 - 3.7
 2004  6  24   0   1  58   0  2.8  -3.2 - 3.6
```

The file containing the satellite trajectory should include data in the following order:

```
yr mn dy hr min sec msec x y z
```

with the position variables in units of the body radii or the length scale normalization. Note that this is the same format as BATS-R-US trajectory files with one important feature: the position is assumed to be in SM coordinates. If the input coordinates are in another coordinate system, use the #COOR command to specify what system is used in the file. RAM-SCB will convert from that system to SM.

The maximum number of satellite files allowed is 100. The maximum number of lines in a given satellite file (past the #START command) is 100,000. Both of these can be changed by editing the value of MaxRamSat and MaxRamSatLines (respectively) in ModRamSats.f90

Each satellite listed under the #SATELLITE command will produce a NetCDF file in the output_ram folder. Simulation meta data, such as run parameters, start times, etc., are contained within this file as well as energy and pitch angle grid information. Magnetic field, flux, spacecraft position, and other values are saved for every DtOutput for which the satellite is within the 3D Equilibrium code's domain. Each record written takes about ten kilobytes of hard drive space.

#OUTPUT_FREQUENCY command

```
#OUTPUT_FREQUENCY
300.0   DtPressureFileWrite
300.0   DthIFileWrite
3600.0  DtEFieldFileWrite
300.0   DtMAGxyzFileWrite
```

Specify the frequency of various output writes for RAM and SCB. Default frequencies are shown.

- DtPressureFileWrite: How often RAM equatorial pressures should be written (in seconds)
- DthIFileWrite: How often SCB hI files should be written (in seconds)
- DtEFieldFileWrite: How often electric field files should be written
- DtMAGxyzFileWrite: How often x, y, and z from SCB should be written

9.1.7 Testing

#TEST command

```
#TEST
CON_session::do_session         StringTest
```

A space separated list of subroutine and function names to be tested. Only subroutines containing the 'call CON_set_do_test(...)' statement can be tested. The first argument is the name of the subroutine, usually defined as the string parameter 'NameSub'. Default is an empty string.
This feature and its conventions has been adopted from the Space Weather Modeling Framework.

Chapter 10

Bibliography

1. Jordanova, V. K., L. M. Kistler, J. U. Kozyra, G. V. Khazanov, and A. F. Nagy. Collisional losses of ring current ions. *J. Geophys. Res., 101*, 111, 1996.
2. Jordanova, V. K., Y. S. Miyoshi, S. Zaharia, M. F. Thomsen, G. D. Reeves, D. S. Evans, C. G. Mouikis, and J. F. Fennell. Kinetic simulations of ring current evolution during the Geospace Environment Modeling challenge events, *J. Geophys. Res., 111*, A11S10, October 2006. doi:10.1029/2006JA011644.
3. Jordanova, V. K., S. Zaharia, and D. T. Welling. Comparative study of ring current development using empirical, dipolar, and self-consistent magnetic field simulations. *Journal of Geophysical Research (Space Physics)*, 115(A14):A00J11, December 2010. doi: 10.1029/2010JA015671.
4. Rasmussen, C. E., S. M. Guiter, and S. G. Thomas, Two-dimensional model of the plasmasphere: refilling time constants, *Planet. Space Sci., 41*, 35, 1993.
5. Stern, D. P., The motion of a proton in the equatorial magnetosphere, *J. Geophys. Res., 80*, 595, 1975.
6. Volland, H., A semiempirical model of large-scale magnetospheric electric fields, *J. Geophys. Res., 78*, 171, 1973.
7. Weimer, D. R., An improved model of ionospheric electric potentials including substorm perturbations and application to the geospace environment modeling November 24, 1996, event, *J. Geophys. Res., 106(A1)*, 407, 2001.
8. Welling, D. T., V. K. Jordanova, S. G. Zaharia, A. Glocer, and G. Toth. The effects of dynamic ionospheric outflow on the ring current. *Journal of Geophysical Research (Space Physics)*, 116, March 2011. doi:10.1029/2010JA015642.
9. Welling, D. T., V. K. Jordanova, A. Glocer, G. Toth, M. W. Liemohn, and D. R. Weimer. The two-way relationship between ionospheric outflow and the ring current. *Journal of Geophysical Research: Space Physics*, May 2015. ISSN 21699380. doi:10.1002/2015JA021231.
10. Young, D. T., H. Balsiger, and J. Geiss. Correlations of magnetospheric ion composition with geomagnetic and solar activity. *J. Geophys. Res., 87*, 9077, 1982.
11. Zaharia, S., V. K. Jordanova, M. F. Thomsen, and G. D. Reeves. Self-consistent modeling of magnetic fields and plasmas in the inner magnetosphere: Application to a geomagnetic storm. *Journal of Geophysical Research (Space Physics)*, 111, A11S10, October 2006. doi:10.1029/2006JA011619.
12. Engel, M. A., Morley, S. K., Henderson, M. G., Jordanova, V. K., Woodroffe, J. R., and Mahfuz, R. (2019). Improved simulations of the inner magnetosphere during high geomagnetic activity with the RAM-SCB model. Journal of Geophysical Research: Space Physics, 124, 4233–4248. doi:10.1029/2018JA026260