Space Hazards Induced Near Earth by Large Dynamic Storms

SHIELDS Framework User Manual

Vania Jordanova and the SHIELDS Team

March 27, 2017

Contents

List of Abbreviations

DA	Data Assimilation
EPIC	Embedded Particle in Cell
LANL	Los Alamos National Laboratory
LDRD	Laboratory Directed Research and Development
PIC	Particle-in-Cell
PTM	Particle Tracing Model
RAM-SCB	Ring current-Atmosphere interactions Model with Self-Consistent Magnetic field
RIM	Ridley Ionosphere Model
SCE	Spacecraft Charging Environment
SHIELDS	Space Hazards Induced near Earth by Large Dynamic Storms
SWMF	Space Weather Modeling Framework
UMich	University of Michigan
WPI	Wave-Particle Interactions

1. Introduction

Our society is increasingly dependent on technologies susceptible to harmful conditions in space, i.e. “space weather”. Predicting space weather hazards remains a big space physics challenge due to the complex multi-scale nature of the magnetosphere. SHIELDS (Space Hazards Induced near Earth by Large Dynamic Storms) addresses this challenge. SHIELDS is an end-to-end model of the magnetosphere driven by the dynamic solar wind which specifies one of the most harmful space weather hazards - the spacecraft surface charging environment (SCE). A thorough understanding of SCE is needed to strengthen spacecraft design when it comes to hazard mitigation. The primary SCE source is the hot (10s of keV) plasma injected from the magnetotail into the inner magnetosphere during storms and substorms - magnetospheric reconfiguration events. SHIELDS simulates the dynamics of these hot particles on both macro- and micro-scale and provides a framework for understanding the near-Earth space environment where operational satellites reside.

1.1 Acknowledgments

The development of SHIELDS has been the very definition of a team effort. From inception to execution, no piece of SHIELDS would be here if it were not for the collaborative efforts of many scientists. A project as large as SHIELDS cannot succeed without institutional support, and the Laboratory Directed Research and Development (LDRD) program at LANL has been supportive throughout the development of SHIELDS and its eventual public release.

1.2 SHIELDS in Brief

The approach taken by SHIELDS to achieve its goal of developing a better model of the SCE is to leverage the best available modeling capabilities, identify their shortcomings, and to develop new capabilities to address these shortcomings. The backbone of SHIELDS is the Space Weather Modeling Framework (SWMF) (*Toth et al.*, 2012), a suite of models that was developed primarily at UMich and is currently maintained by the UMich Center for Space Environment Modeling. The SWMF offers capabilities for modeling the global dynamics of the magnetosphere and heliosphere, and it contains modules which allow for – among many other things – the inclusion of ionospheric electrodynamics and specialized inner magnetosphere models.

The Ring Current-Atmosphere Interactions Model with Self-Consistent Magnetic Field (RAM-SCB) (*Jordanova et al.*, 2010) is a large-scale kinetic model of inner magnetospheric plasmas that is currently being developed and maintained at LANL. RAM-SCB couples directly with the SWMF in order to improve its representation of the inner magnetosphere plasma environment. The work of SHIELDS research has resulted in the development of many new and exciting features that enable the coupled SWMF—RAM-SCB model to provide a much better specification of the SCE:

1. An improved model of wave-particle interactions
2. An embedded particle-in-cell (PIC) model to improve reconnection physics
3. A dynamic particle tracing model (PTM) to propagate distribution functions
4. An integrated data assimilation (DA) capability to improve fidelity to measurements

Details of these new capabilities are discussed in Section 3.

2. Quick Start

Future releases of SHIELDS may have different instructions, so users should make sure to refer to the version of this document that accompanies their particular release of the framework.

2.1 Obtaining the Source Code

The Space Weather Modeling Framework (SWMF) can be obtained from the University of Michigan's Center for Space Environment Modeling, http://csem.engin.umich.edu/tools/swmf/. The SWMF is freely available but it is subject to a license that does not allow for redistribution (it is not export controlled, however).

The Ring Current-Atmosphere Interactions Model with Self-Consistent Magnetic Field (RAM-SCB) can be freely downloaded from https://github.com/lanl/RAM-SCB. RAM-SCB is open source software, and there are no limitations on its use, modification, or redistribution.

SHIELDS-specific extensions to the SWMF and RAM-SCB are now distributed with the codes they extend, so they will require no additional downloads. Postprocessing tools and the AEGIS tool can be found on the SHIELDS web page, http://www.lanl.gov/projects/shields/index.php.

2.2 Library Dependencies of RAM-SCB

At present (although this may change in future releases), RAM-SCB depends on a handful of external libraries that must be present in order to compile and run. These libraries are:

The NetCDF Library (both C and Fortran)
Many systems have NetCDF pre-installed, but the Fortran version is not as common. The source code for NETCDF can be found at http://www.unidata.ucar.edu/downloads/netcdf/index.jsp. This library is used for creating portable, self-describing data files. NetCDF itself requires the presence of the HDF5 library. If HDF5 is not present on your system, it will be necessary to download and install ZLIB, a prerequisite for HDF5, and the HDF5 source code itself. These can be found at http://www.zlib.net/ and https://support.hdfgroup.org/HDF5/.

The NCAR Graphics Library
The NCAR Graphics Library is used to provide visualization capabilities for RAM-SCB, allow for images to be output during run-time. The source code and interactive installer for this library can be found at http://www.ncl.ucar.edu/Download/.

The Princeton Spline Library
The Princeton Spline Library (PSPLINE) is required for the interpolation of scattered grid data in RAM-SCB. The source code can be found at http://w3.pppl.gov/ntcc/PSPLINE/. The PSPLINE library is subject to terms of use that are very permissive.

2.3 Compiling RAM-SCB Dependencies

In order to properly compile the libraries on which RAM-SCB depends, it is necessary that its dependencies be compiled in a particular order, as some of them depend on others. If the libraries described above are compiled following their instructions in the following order, the dependences should be properly configured:

1. zlib
2. HDF5
3. NetCDF
4. NCAR Graphics Library
5. PSPLINE

Any libraries that are already present on your system (subject to caveats given above) can be skipped, so long as the installation instructions for the code that depends on them are modified appropriately.

2.4 Installing SHIELDS

1. Unzip the SWMF source code
2. Unzip the RAM-SCB source code
3. Move RAM-SCB into the IM subdirectory of the SWMF directory
4. Install SWMF using standard directions provided in the SWMF manual
5. Configure SWMF to use BATSRUS, RAM-SCB, and the Ridley_serial models
6. Compile using standard method described in the SWMF manual

3. SHIELDS Components

3.1 Stand-Alone Components

Space Weather Modeling Framework (SWMF)
A full description of the SWMF and its available components can be found in the SWMF manual, available at http://csem.engin.umich.edu/tools/swmf/documentation/index.php (the manual may also be built from source code included in the **doc** directory of the SWMF software distribution). SHIELDS makes use of three components of the SWMF: the BATS-R-US global magnetosphere, the Ridley Ionosphere Model (RIM), and the Embedded Particle-in-Cell (EPIC) model. The EPIC capability was specifically developed in response to SHIELDS project needs, but it is now a fully validated and publicly available module within SWMF.

Ring Current-Atmosphere Interactions Model with Self Consistent Magnetic Field (RAM-SCB)
A full description of RAM-SCB can be found in the RAM-SCB Manual, available in the **doc** directory of the RAM-SCB software distribution. It is possible to run RAM-SCB in a “stand-alone” mode rather than as a component of SWMF, and instructions on how to do this will be provided later in Section 5.

3.2 SHIELDS-Specific Components

The Wave-Particle Interactions (WPI) Model
One of the most important factors controlling the evolution of energetic particle distributions is the interaction between these particles and electromagnetic waves. The effects of these interactions are included in RAM-SCB by specifying diffusion coefficients, which are in effect numerical expressions that quantify how distributions of charged particles evolve in the presence of waves. However, these coefficients are frequently determined from long-period statistics or simplified parametrizations. In SHIELDS, a new approach has been taken wherein waves are self-consistently modeled using PIC simulations to determine event-specific diffusion coefficients. This approach offers a much greater fidelity to the actual physical dynamics and eliminates a known source of systematic uncertainty.

The Particle Tracing Model (PTM)
The SHIELDS PTM extension allows for test particles to be propagated through the electric and magnetic fields that are produced by SWMF and RAM-SCB. This code is designed to make use of fields that are defined on a discrete (but not necessarily uniform) Cartesian grid. Because data from the SWMF is typically defined on an adaptive mesh grid, PTM also includes a pre-processing script to interpolate SWMF data onto a compatible Cartesian grid. Particle tracing results can be post-processed to determine pitch-angle resolved distribution functions and fluxes. An advanced application of PTM is to dynamically specify the outer boundary condition for RAM-SCB; this will be discussed later in Section 5.

The Data Assimilation (DA) Architecture

In order to improve RAM-SCB model fidelity when in situ measurements are available, a novel data assimilation framework has been developed. Called the "Reduced Basis Ensemble Kalman Filter" method (described in *Godinez et al.*, 2016), this approach has been demonstrated to provide orders of magnitude improvement in model fluxes. It should be noted, however, that SHIELDS with data assimilation requires significantly more computational resources than without, as the assimilation process requires multiple versions of the program to be run simultaneously to create the necessary ensemble of results. The use of DA in the coupled RAM-SCB/SWMF framework is currently experimental.

The Spacecraft Charging Model

Data that is output from SHIELDS can be used to drive a variety of different models for estimating environmental effects on spacecraft. We have currently implemented an empirical model of spacecraft charging based on measurements from the LANL geosynchronous spacecraft [*Thomsen et al.*, 2013]. A future release of SHIELDS will include a curvilinear particle-in-cell (CPIC) model for determining the charging of realistic spacecraft.

An Easy-to-use Graphical Interface for SHIELDS (AEGIS)

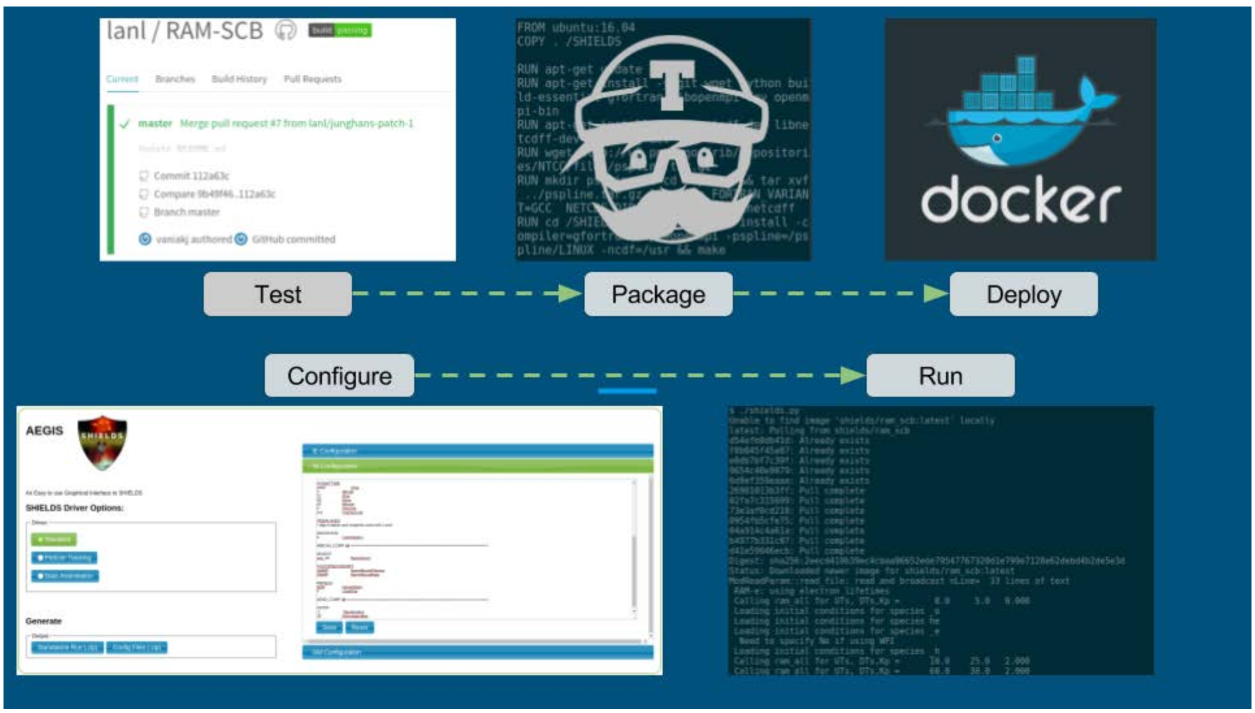
Because of the many complex models that comprise the SHIELDS framework, it can be difficult to properly configure the various input files that are needed to achieve a desired modeling result. In order to streamline the setup process and to reduce accidental errors, we have developed a graphical interface for SHIELDS that allows users to select their components of interest, set relevant parameters, and generate input files and execution scripts. This tool is available on the SHIELDS web site (http://www.lanl.gov/projects/shields/index.php) and more in-depth documentation and usage examples can be found there.

In addition to the descriptions provided here, detailed technical documentation for each SHIELDS component can also be found accompanying their source code.

4. Configuring the Framework

This section describes how to configure SHIELDS for different types of simulations.

Because of the computational complexity of the multiscale, multiphysics problem that SHIELDS solves, SHIELDS is designed to run at large scale on the world's most powerful supercomputers. For such advance applications, SHIELDS can be built from source following the instructions detailed in the user manual. However, a critical part of the SHIELDS design philosophy is to allow both users and developers to get working with SHIELDS on any system without having to deal with complicated software dependencies. As such, SHIELDS has been packaged in a Docker container, allowing installation and deployment with a single command on any laptop, desktop, or high-performance computing system with Docker installed.



If a developer wishes to experiment with a working version of SHIELDS, they can jump into the shell where they will find a compiled version of SHIELDS along with all dependencies and source necessary for recompilation by simply typing:

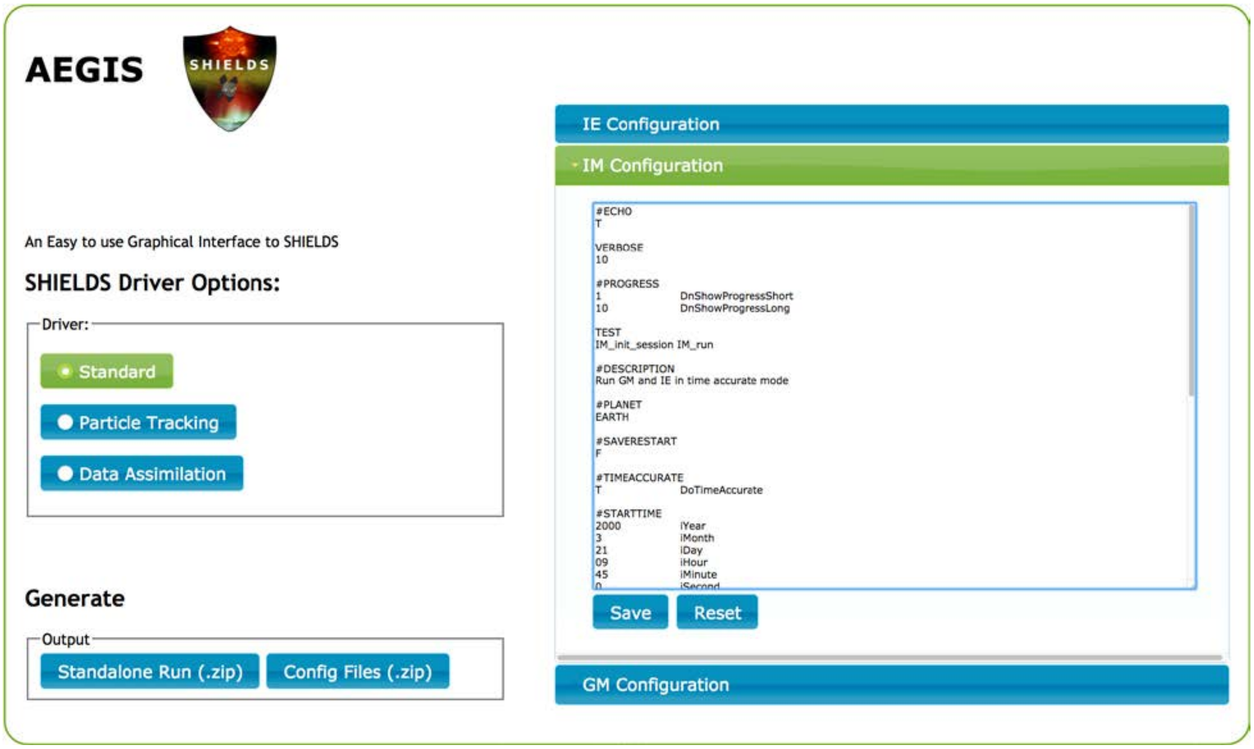
docker run -it shields/ram_scb /bin/bash

The SHIELDS application is kept automatically up to date with the latest SHIELDS developments. As changes are made to the SHIELDS code base, they are automatically built and tested using Travis Continuous Integration. If all tests pass, a new container is built reflecting the code changes and pushed to Dockerhub. SHIELDS releases are automatically tagged with the corresponding change to the source code. If you wish to target a specific release of SHIELDS you can specify the commit tag and the correct image will be downloaded automatically, for example:

docker run -it shields/ram_scb:112a63c9 /bin/bash

We warmly encourage pull requests from SHIELDS developers.

For application users who wish to generate and run simple parameter files we recommend AEGIS – “An Easy to use Graphical Interface to SHIELDS”. AEGIS provides a user-friendly mechanism to configure simple input problems for SHIELDS; along with a Python driver *AEGIS.py* that runs SHIELDS using your custom input parameters.



To use AEGIS simply modify the desired parameters and select “Standalone Run” where you will be prompted to download a (.zip) file. After extracting the (.zip) file either double click on *AEGIS.py*, or run from the command line using Python.

AEGIS.py utilizes the SHIELDS Docker application and will automatically download and run latest version of SHIELDS. It is necessary that users of AEGIS have Docker pre-installed. As the simulation executes output files are automatically written within the extracted AEGIS directory.

5. Some Example Cases

In this section, we will provide some concrete examples of how to use different elements of the SHIELDS framework. Because SHIELDS is a framework rather than a single simulation code, it is necessary to specify which components of the framework are being used and to prepare the required inputs for those elements. Complex combinations of framework elements may require the construction of special-purpose scripts (the preparation of which can be greatly facilitated by using the AEGIS tool discussed in Section 4). The examples in this section are sufficiently simple that users who are comfortable with Unix shells and basic command-line utilities should have little difficulty in achieving the described tasks.

5.1 Stand-Alone RAM-SCB

In its most elementary configuration, the SHIELDS framework allows for the simulation of only the inner magnetosphere plasma environment using RAM-SCB along with an external boundary condition that can be provided by a user-specified time series. Although the SWMF framework allows for RAM-SCB to be run in stand-alone mode, it is more straightforward to treat RAM-SCB as decoupled from the larger framework.

In order to prepare this simulation, it will be necessary to edit the file “PARAM.IN.RAM” in the **`Examples`** subdirectory.

5.1.1 Specifying Boundary Conditions

In the Examples subdirectory, there is a file titled “RAM-FLUXES_20130317.dat”. This file contains time series of data from the LANL geo satellites for the March 17-18, 2013 geomagnetic storm. The simulation is expecting an input file named “RAM-FLUXES.dat”. The easiest way to prepare the simulation is to create a symbolic link to the desired input file:

ln –s RAM-FLUXES_20130317.dat RAM-FLUXES.dat

The preparation of input files for RAM-SCB is described in the RAM-SCB manual (see Section 2 for directions on where to find this document).

5.1.2 Input File Configuration

5.1.3 Validating Results

These steps have been automated: type “make shields_ex1” to prepare this simulation run. The output of this test can be compared to your results either by comparing output graphics or by `diff`ing the outputs.

5.2 Coupled SWMF and RAM-SCB

The following example demonstrates the full coupling between RAM-SCB and the SWMF. In order to prepare this simulation, it will be necessary to edit the file "PARAM.IN.SWMF" in the `Examples` subdirectory.

5.2.1 Specifying Boundary Conditions

In the Examples subdirectory, there is a file named "IMF_20130317.dat". This file contains solar wind data for the March 17-18, 2013 geomagnetic storm. The simulation will be looking for a file named "IMF.dat". The easiest way to prepare the simulation is to create a symbolic link to the desired input file:

```
>ln –s Examples/IMF_20130317.dat IMF.dat
```

The preparation of input files for the SWMF is described in the SWMF manual (see Section 2 for directions on where to find this document).

5.2.2 Input File Configuration

5.2.3 Validating Results

These steps have been automated: type "make shields_ex2" to prepare this simulation run. The output of this test can be compared to your results either by comparing output graphics or by `diff`ing the outputs.

5.3 Particle Tracing

In the `Examples` subdirectory, there is a file named "shields_20130317.cdf" which contains example output data from a simulation of the March 17, 2017 geomagnetic storm. In this example, we will use the PTM framework to process this data, create inputs for particle tracing, and generate particle distributions.

5.3.1 Preparing a PTM Run

The first step in preparing a run of PTM is to create files containing the electromagnetic fields used by the particle tracing framework. Pre- and post-processing for SHIELDS-specific components is handled by Python scripts. Let us assume that the location of the SHIELDS installation is recorded in the $SHIELDS environment variable. Then, on the command line,

```
>export PYTHONPATH=$SHIELDS/PTM/scripts
>python ptm_interpolate.py $SHIELDS/Examples/shields_20130317.cdf
```

The next step is to prepare the input files for a PTM run. There are a variety of ways to do this (see source code documentation), but the simplest is to specify configuration details in a text file and use the ptm_input.py script. An example input file would be:

```
#---PTM.TXT STARTS ON THIS LINE---
# Time
18          #hours
0           #minutes
0           #seconds

# Coordinates
GSM         # GSM, GSE, GEO, SM

# Position
-6.6        #X
0.0         #Y
0.0         #Z
#---PTM.TXT ENDS ON THIS LINE---
```

Supposing the preceding lines were saved as plain text in “ptm.txt”, the required input files for PTM could be generated using the following command:

```
>python ptm_input.py ptm.txt
```

There will be three files created by this script:

ptm_parameters_0000.txt
dist_velocity_0000.txt
dist_density_0000.txt

You are now prepared to run the particle tracing simulation, which can be executed as follows:

```
> python ptm_run.py
```

This script will check the environment for the necessary file structure, put the input files in the correct location, and run the PTM executable.

5.3.2 Analyzing Particle Tracing Data

Once the executable terminates, particle trajectories can be analyzed and post-processed to obtain distribution functions and particle fluxes:

```
> python ptm_postproc.py
```

Resulting data files and images can be found in the **`ptm_output/postproc`** directory.

5.3.3 Validating Results

These steps have been automated: type "make shields_ex3" to prepare this simulation run. The output of this test can be compared to your results either by comparing output graphics or by `diff`ing the outputs.

References

Godinez, H. C., Y. Yu, E. Lawrence, M. G. Henderson, B. Larsen, and V. K. Jordanova (2016), Ring current pressure estimation with RAM-SCB using data assimilation and Van Allen Probe flux data, *Geophys. Res. Lett.*, 43:11948-11956, doi:10.1002/2016GL071646.

Jordanova, V. K., S. Zaharia, and D. T. Welling, (2010), Comparative study of ring current development using empirical, dipolar, and self-consistent magnetic field simulations, *J. Geophys. Res., 115*, A00J11, doi:10.1029/2010JA015671.

Thomsen, M. F., V. K. Jordanova, and M. G. Henderson (2013), Statistical properties of the surface-charging environment at geosynchronous orbit, *Space Weather*, 11:237-244, doi:10.1002/swe.20049.

Toth, G., B. van der Holst, I. V. Sokolov, D. L. De Zeeuw, T. I. Gombosi, et al. (2012), Adaptive Numerical Algorithms in Space Weather Modeling, *J. Comp. Phys., 231*, 870, doi:10.1016/j.jcp.2011.02.006.