HamSCI: Eclipse Simulator Documentation

Joshua S. Vega, WB2JSV

Last Updated: October 15, 2018

Contents

1 Introduction

HamSCI's eclipse simulator (colloquially known as "`eclipsesim`") is a software package developed by members of the HamSCI team at the New Jersey Institute of Technology used to simulate HF (3-30 MHz) through the ionosphere. Specifically, the package was developed in order to simulate the effects of the August 21, 2017 total solar eclipse on HF amateur "ham" radio communications. The bulk of the package is developed in MATLAB. However, the inputs and outputs are both in the comma-separated values (CSV) format in order to allow for easy interoperability with other analysis tools (such as those written in Python, R, or Julia).

This document is written in order to provide the reader with an understand of the inner workings of the simulation package as well as justify some of the design decisions that were made during development. In addition, this document is intended to be used as a helpful guide to using the package. Because of this, as the package undergoes future changes, so too should this document in order to maintain parity between the package and its documentation.

2 Dependencies

The `eclipsesim` package relies on a few third-party dependencies in order to implement some specialized functionality. This section provides a brief overview of the different dependencies and their function.

2.1 PHaRLAP

PHaRLAP[1]is a robust HF raytracing toolkit developed by Dr. Manuel Cervera of the Defence Science and Technology Group, Australia. It is used to compute the actual ray paths of radio transmission from transmitter to receiver. More information on how it is used is provided in section 4. Although the core of the toolkit is written in FORTRAN, a MATLAB application programming interface (API) is provided. PHaRLAP also provides a simple API for the IRI (see below) that is compatible with the ray path computation API.

2.2 IRI

The International Reference Ionosphere (IRI)[2]is a ionospheric model developed by the Committee on Space Research (COSPAR) and International Union of Radio Science (URSI). It is periodically updated to incorporate new measurements. It generates monthly averages for several ionospheric measurements including electron density, and electron and ion temperatures.

[1] *"The results published in this paper were obtained using the HF propagation toolbox, PHaRLAP, created by Dr. Manuel Cervera, Defence Science and Technology Group, Australia (manuel.cervera@dsto.defence.gov.au). This toolbox is available by request from its author."*

[2] `http://irimodel.org/`

2.3 SAMI3

SAMI3 is a three-dimensional global ionosphere model developed by the U.S. Naval Research Laboratory (NRL). In 2017, Joseph D. Huba and Douglas Drob published a paper detailing a modified SAMI3 model that simulated effects of the August 21, 2017 eclipse on the ionosphere [Huba and Drob, 2017]. This model was provided to HamSCI by Joseph Huba and have since been the *de facto* ionosphere model for eclipse simulations. For more information on SAMI3's data structure and how it has been integrated into the eclipse simulation, see Appendix B.

3 Usage

The `eclipsesim` package provides two methods of execution out-of-the-box: serially and in parallel. More information and specific instructions on each execution method is discussed in subsection 3.2 and subsection 3.3, respectively.

Regardless of the execution method used, `eclipsesim` is designed to be executed on a Linux-based machine. While limited success has been achieved on other operating systems (notably, Microsoft Windows), the execution scripts are developed with Linux in mind and may not execute as expected in incompatible environments. All instructions provided in this section are therefore intended for Linux-based systems and may require modification for other operating systems.

3.1 Setup

Before the eclipse simulation can be run, all external dependencies must be accessible to the simulation program. This section briefly describes what dependencies are required and how they must be configured. Some extension version of the simulation program may require additional dependencies. Please refer to Appendix A for extension-specific dependency information.

MATLAB: Since all computation is executed in MATLAB, it should come as no surprise that the most important dependency is MATLAB itself. The simulation is primarily designed to execute on version 2016a however later versions should also be compatible. When using older versions of MATLAB some compilation and runtime errors have been seen to occur due to breaking changes between older version and 2016a or later. Once MATLAB has been installed, it's important to ensure that the `matlab` program is available for execution through a modified `$PATH` variable.

PHaRLAP: As described in subsection 2.1, PHaRLAP is a vital part of the simulation program. Due to the restrictions on the distribution of PHaRLAP, it is not and should not be directly distributed with any of the eclipse simulation versions. Instead, it is recommended that prospective users of the simulation tool contant the author of PHaRLAP and receive a direct copy. Once a copy of PHaRLAP is acquired, it can be installed anywhere that is accessible to the simulation tool as long as the `PHARLAP_DIR` variable is updated appropriately in the execution script.

IRI: If the IRI will be used as the ionosphere model there is no need to install any additional dependencies as it is already integrated into PHaRLAP. However, keep in mind that with most version of the simulation tool being designed for SAMI3, some modification of the simulation code will be required.

SAMI3: If SAMI3 will be used as the ionosphere model, then there are a few steps required in order to prepare it to work with the simulation tool. After acquiring the SAMI3 model it must be converted from the raw data format to the `*.m` file format which is far more efficient for MATLAB to load and parse. For detailed information about this step, refer to Appendix B. Once it has been properly converted, the SAMI3 `*.m` file(s) must be placed in a directory that is accessible to the simulation program. Be sure to update the `SAMI3_PATH` variable appropriately in the execution script.

3.2 Serial

The serial execution model is designed for testing or very short simulations. The serial execution model only executes the simulation for a single point in time. This can obviously be adjusted by modifying the bash execution script. However for this sample, it is assumed that only one time-step is being simulated. It is also assumed that all dependencies have been properly installed and configured and that the appropriate modifications have been made to the execution script.

To execute the simulation tool in serial mode, you'll be executing the `eclipsesim_local.sh` file in as bash script without any arguments (ensure that it has been configured to have the execution permissions bit enabled):

```
$ ./eclipsesim_local.sh
```

This script will the first job file available in the input directory (see subsection 3.4 for information on inputting data to the simulation tool). When the simulation is complete the terminal will be prepared for a new command and the outputted data will be available in the output directory (see subsection 3.5 for information on outputting data from the simulation tool). If an error occured during the simulation, MATLAB will halt and indicate where in the MATLAB code the error occurred.

3.3 Parallel

The parallel execution model is designed to be executed on a computing cluster running a Sun Grid Engine (SGE; now Oracle Grid Engine) compatible scheduling system. One such system is Son of Grid Engine (also SGE). When submitting a job, SGE adds it to the job queue. When a compute node with the requested resources is made available, the next job is assigned to be executed on said node. SGE also allows multiple compute nodes to be requested by one job using the array option. For more information about the many options made available to jobs, please refer to both the SGE documentation as well as any cluster-specific documentation.

To execute the simulation tool in parallel mode, you'll be executing the `eclipsesim_sge.sh` file as an argument to the `qsub` application. Depending on the specific variant of the simulation tool, there may or may not be specific eclipse and non-eclipse SGE execution scripts (such as `eclipsesim_sge_eclipse.sh` and `eclipsesim_sge_base.sh`, respectively). Below is an example of submitting a job to SGE:

```
$ qsub ./eclipsesim_sge.sh
```

This command will submit the job file to the SGE queue. Once the job has been selected for execution, it will ensure that MATLAB is loaded and then begin the process of preparing the simulation for execution (this step is identical to the equivalent step in the serial execution model). Due to the parallel nature of this execution model, no response will be printed to the terminal. Instead, a log file for each job submitted job (if the job is an array job, one log file will be generated for each sub-job) containing all output information for that individual job (or sub-job). For this reason it is not recommended that the simulation be tested or debugged in the parallel execution mode (unless it is directly related to the parallel nature of the execution).

3.4 Input

In order for the simulation tool to execute, it requires some input data. This data is usually in the form of a transmitting station location, receiving station location, frequency, and time of exchange. Because the simultion is divided into timestamps, the time of exchange is encoded into the input filename. This is done using the format ''`<YYYY>-<MM>-<DD> <HH>:<mm>:<ss>.csv`'' (e.g. "2017-08-21 16:00:00.csv" for August 21, 2017 at 4:00 PM UTC).

Within the input file, the remainder of the input data is stored. This is done by entering the data as rows into the file using the Comma-Separated Values (CSV) format. The first line of file should be ''`tx_call,rx_call,tx_lat,tx_lon,rx_lat,rx_lon,freq`'' (this may vary based on the simulation tool version; some versions do not use the `freq` column). Each row follows the following format:

Name	**Datatype**	**Description**	**Example**
`tx_call`	alphanumeric string	The name (or callsign) of the tx station	`AK6VS`
`rx_call`		The name (or callsign) of the rx station	`N6NC`
`tx_lat`	floating-point number	The latitude of the tx station	`37.71`
`tx_lon`		The longitude of the tx station	`-122.50`
`rx_lat`		The latitude of the rx station	`32.83`
`rx_lon`		The longitude of the rx station	`-117.33`
`freq`		The frequency of the transmission in MHz	`14.03`

Below is an example row from an input file:

```
AK6VS,N6NC,37.71,-122.50,32.83,-117.33,14.03
```

Once the correct input files have been created (or possibly generated), they should be placed into the `./jobs` sub-directory of the simulation tool's directory. If no such directory exists, the specific version of the simulation tool may use another method of inputting data. In such cases, please refer to A or to the accompanying `README` file for the simulation tool.

3.5 Output

4 Design

4.1 Rough Pass

4.2 Fine Pass

4.3 Plotting

5 Conclusion

References

J. D. Huba and D. Drob. SAMI3 prediction of the impact of the 21 August 2017 total solar eclipse on the ionosphere/plasmasphere system. *Geophysical Research Letters*, 44:5928–5935, 2017.

A Extensions

A.1 10-Hop

B SAMI3 Data Format