Developing a Solar Eclipse Simulation for Greater Good

Joshua S. Vega, WB2JSV
New Jersey Institute of Technology, Newark, N.J.
jsv28@njit.edu

Dr. Nathaniel A. Frissell, W2NAF
New Jersey Institute of Technology, Newark, N.J.
nathaniel.a.frissell@njit.edu

Joshua D. Katz, KD2JAO
New Jersey Institute of Technology, Newark, N.J.
jk369@njit.edu

Dr. Joseph D. Huba
U.S. Naval Research Laboratory, Washington, D.C.
huba@nrl.navy.mil

Abstract

This paper presents our methodology for simulating the upcoming total solar eclipse that will be taking place on August 21, 2017. By taking advantage of a high-performance distributed computing cluster as well as a number of third-party scientific computing libraries we were able to efficiently simulate a large number of HF amateur radio contacts before, during, and after the upcoming eclipse. The data generated from the simulations allows us to peek into how the amateur radio community and radio propagation as a whole will be affected in preparation for the actual eclipse.

Keywords: eclipse, ionosphere, propagation, simulation

Introduction

The ionosphere is a region of Earth's atmosphere composed of electrons and other charged particles. The ionosphere is generally considered to be in the altitude range of 50 km to 1000 km and encompasses the mesosphere and thermosphere. It is this ionosphere that is responsible for HF propagation (as well as some VHF propagation). The ionosphere is composed of four layers during the day and two at night. The lowest layer is the D layer. It it only present during the day and is responsible for MF and low frequency HF absorption. The next layer is the E layer and reflects the lower HF frequencies (during an event known as sporadic E, the E_s layer can reflect frequencies as high as mid-to-low VHF). Following the E layer is the F layer. During the day however, the F layer splits into the lower F_1 and the higher F_2 layers. It is the F_2 layer that is predominantly responsible for long-distance HF propagation.

On August 21, 2017 a total solar eclipse will travel across the continental United States from the West Coast to the East Coast, making it the first to do so since 1918 [1]. Such an event is expected to have a large, but short-lived, effect on the ionosphere. By combining amateur radio and academia, the Ham Radio Science Citizen Investigation (HamSCI) organization is working to study the effects of the eclipse

by using the vast number of amateur radio operators spread throughout the country in order to generate massive amounts of data. HamSCI is accomplishing this by organizing the Solar Eclipse QSO Party (SEQP) [2]. The SEQP is a contest-like event that will take place before, during, and after the eclipse where points will be awarded based in part on the scientific merit of the data generated by each contact.

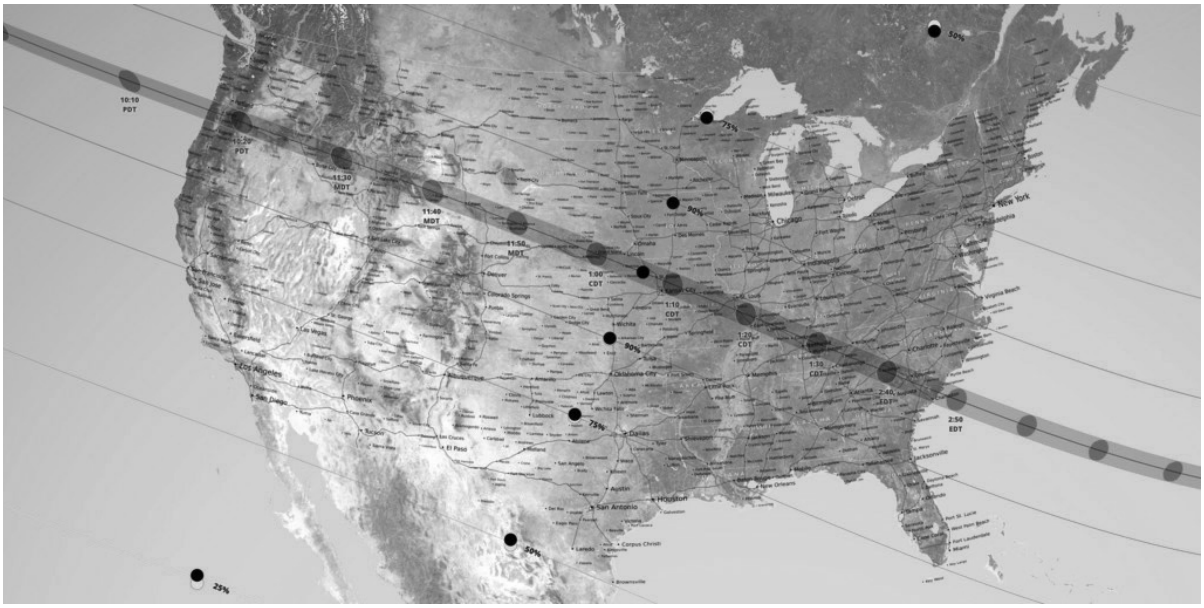


Figure 1: Path of totality for the August 21, 2017 total solar eclipse[1].

In preparation for the eclipse and SEQP, a simulation program was needed that would predict how radio propagations would change throughout the event. In addition, the goal of the project was to develop a tool that could be used in the future to help understand and visualize radio wave propagation further such as band openings and optimal directional anetenna azimuths. The development of this simulation program proved to be a major undertaking and required several months to complete.

Design

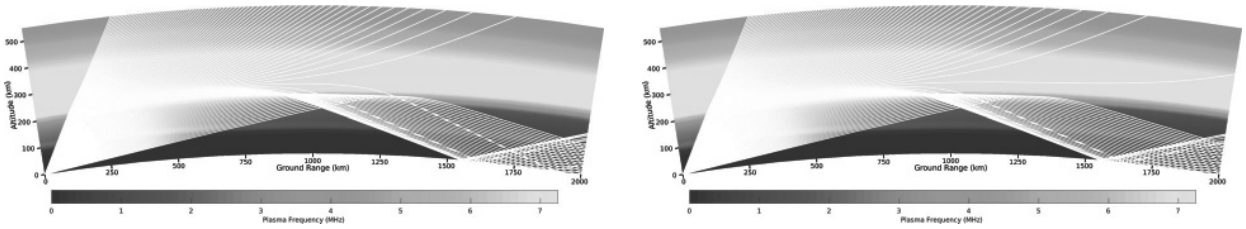
In designing the simulation program, there were several features the program needed to provide. Each of these features brought a different challenge that had to be overcome. While some compromises had to be made, the majority of the necessary features were implemented to a satisfactory standing. One of the most important of these criteria was integrating the PHaRLAP package [3]. Written in FORTRAN with a MATLAB wrapper, PHaRLAP is a very well known and capable scientific ionospheric raytracer that can be used to trace radio signal paths through the ionosphere from a transmitter to a receiver. PHaRLAP serves as the core around which the rest of the project was designed and written.

The purpose of creating the eclipse simulation is to be able to compare raytraces through a "normal" ionosphere to raytraces through an ionosphere that has been affected by an eclipse. In order to create an "eclipsed" ionospheric model, the normal model needed to be modified in a way that would resemble an eclipse passing through. This functionality was provided by Magda Moses, KM4EGE as a MATLAB

[1] Image created by NASA's Scientific Visualization Studio (https://svs.gsfc.nasa.gov/4518)

function that was used to augment an existing ionosphere [4]. The function outputs an attenuation factor from 0.0 to 1.0. The value of the attenuation factor is dependent on the timestamp of the simulation and the coordinate at which it is to be calculated. The values are based on numerical models developed from the 1999 total solar eclipse that crossed over much of Europe [5].

One of the more defining features of the simulation program is the ability to generate two-dimensional raytraces through a three-dimensional ionospheric model. Unfortunately, PHaRLAP does not natively support this functionality. In order to circumvent this limitation, two-dimensional "slices" were interpolated from the three-dimensional ionosphere grid. When using the IRI2016, this proved to be trivial as the ionospheric model return by PHaRLAP's wrapper places data points at regular intervals. MATLAB's `griddedInterpolant` provides an efficient way to interpolate these types of values at individual points. By sampling points along the azimuth between the transmit and receive stations at a predefined regular interval, a PHaRLAP-compatible slice was created. Figure 2 shows a raytracing comparison between PHaRLAP's internal slicing mechanism and the one described here. This method of slicing was also compatible with the eclipse attentuation function described in the previous paragraph.



(a) Raytrace using PHaRLAP's native implementation.

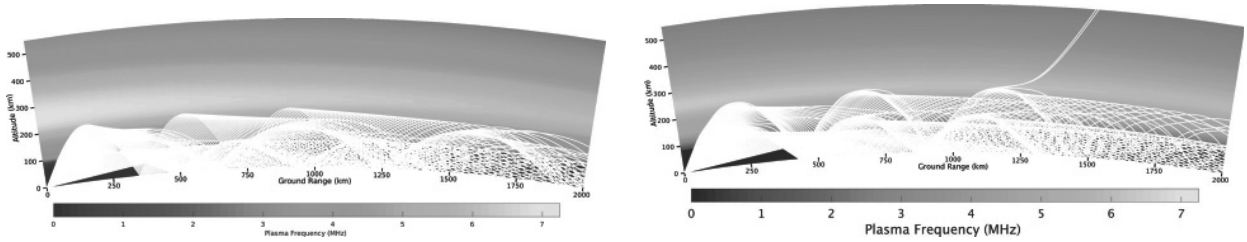
(b) Raytrace using the interpolation method described earlier in this paper.

Figure 2: Simulated raytraces for QSO from AA3B toward PY1NB using IRI2016 at 21 MHz.

Late in the development of the eclipse simulation, the opportunity arose to use the SAMI3 ionospheric model developed at the U.S. Naval Research Laboratory in place of the IRI2016 model [6]. The SAMI3 model that was used had been created specifically for the upcoming eclipse and provided both "normal" and "eclipsed" versions [7]. Unlike the attenuation function that was used with the IRI2016, SAMI3's method of calculating the eclipse not only takes into account the simulation timstamp and the coordinates of the data point, but also the altitude of the data point. Therefore, different altitudes at the same ground coordinate are attenuated by different amounts. This produces a model that more closely parallels the real-world expection [8]. Figure 3 compares raytraces conducted with the IRI2016 and SAMI3 models.

Integrating SAMI3 proved to be somewhat challenging. Unlike the IRI2016 model which was presented as a regular grid, the SAMI3 model's data points are based on a horizontal geomagnetic coordinate system causing the data to be stored in a three-dimensional sparse format with ground coordinate values varying slightly at each altitude. This introduced two primary challenges: converting from the horizontal coordinate system to the World Geodetic System 1984 (WGS84), and interpolating a sparse set of data. The coodinate system conversion was solved with a simple `if-else` statement where if the longitude value was more than 180 degees, 360 degrees would be subtracted from it prior to storing. This converted the original (0, 360] degrees longitude coordinate system to a PHaRLAP-compatible (-180, 180] degrees longitude coordinate. The second challenge was not such a simple change. The solution was to use MATLAB's `scatteredInterpolant` interpolator in place of the `griddedInterpolant` that was

previously used with the IRI2016. Although `scatteredInterpolant` provided the necessary functionality, it came with a significant increase in runtime and resource usage. Attempts were made to provide the same or similar functionality in a more efficient manner but ultimately the decision was made to use `scatteredInterpolant`.



(a) Raytrace using IRI2016 (default parameters).

(b) Raytrace using SAMI3.

Figure 3: Simulated raytraces for QSO from AB5XM toward WZ7I at 7 MHz.

One of the more annoying (as opposed to frustrating) challenges of integrating SAMI3 was actually the first step: loading the SAMI3 files into MATLAB. When using MATLAB's text file I/O routines, the SAMI3 model for a single timestamp could take several minutes to be loaded. By contrast, the same exact values were loaded in Python 3 in mere seconds. Ultimately, the solution was to load the SAMI3 files in Python 3, save the values into .mat files using SciPy's `scipy.io.savemat`, and then load the .mat files directly into MATLAB (in a matter of seconds). Although this method requires two individual steps, the overall runtime of the simulation decreased dramatically. It should also be noted that since the SAMI3 files do not change, the Python 3 conversion step needs only to be performed when the files are first received.

Execution

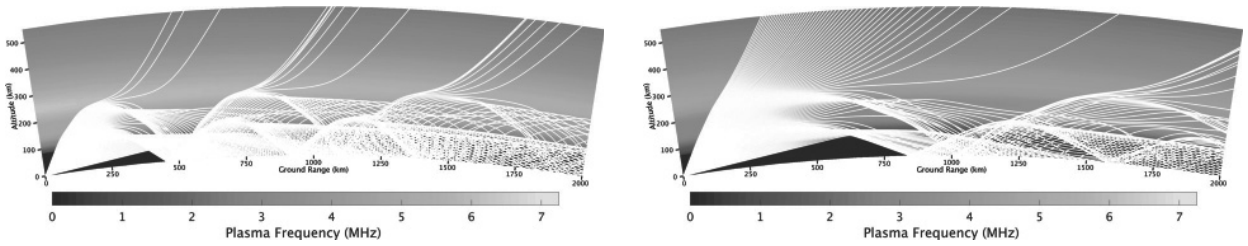
Although the simulation program was initially written for execution on a single machine, it was quickly apparent that simulating the entire SEQP on a single machine would take far too long to complete. Thankfully, The New Jersey Institute of Technology's Academic and Research Computing Systems department (NJIT-ARCS) provides several cluster computers available through the Tartan High Performance Computing Initiative to those performing research through the university [9]. One of these clusters, Kong, was used for the SEQP simulation.

After ensuring that the original implementation was functioning correctly when executed on a single compute node of the cluster, changes were made to introduce the clustered functionality into the program. Using MATLAB's Parallel Computing Toolbox, portions of the work were re-written to be distributed amongst several compute nodes. By doing so, multiple compute nodes would work concurrently on a single input value, with each node taking a small portion of the work, before being aggregated and written to the output file. This design is common in the world of parallel computing and much of the scheduling and data distribution is done automatically by MATLAB without requiring developers intervention. However, during the initial test runs, the time needed to initialize was found to be far longer than expected. When executed with as little as four compute nodes, the program could take anywhere between 45 minutes to an hour before the simulation code actually began executing (by comparison, initialization was completed in less than a minute when using the original single machine implementation).

Such long initialization times were deemed excessive and unacceptable for short turnover between simulations. An alternative parallel execution method was necessary. With MATLAB providing no other suitable method for parallel execution, focus was turned toward the job scheduling engine used by Kong, Son of Grid Engine (SGE), where a feature known as array jobs was discovered. An SGE array job is a single job that spawns multiple sub-jobs with each sub-job executing on its own independent compute node. Using this feature, the input data was partitioned into 15 minute buckets (each bucket contained all the transmit-recieve pairs for a period of 15 minutes) with each sub-job's compute node taking a single time bucket and calculating all values within the selected bucket. This method of execution has proven to be the most efficient way to run the simulations. Using this method of execution, 2.8 million transmit-receive pairs were computed across six HF bands in just under 10 hours.

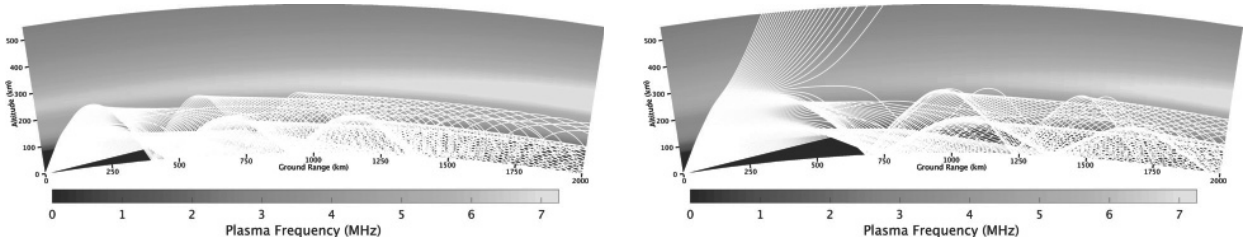
Results

Figures 4 and 5 below show just a couple samples raytraces outputted by the simulation. In addition to images such as the ones below, the simulation program also outputs a comma-separated values (CSV) file for each time bucket. These CSV files contain the information generated by PHaRLAP for each ray. The values can then be used in other ways to help understand the effects of the eclipse on propagation. Figure 6 is an example of one use for the output. It shows all the connecting rays (any ray that is able to make it from the transmitter to receiver) across a basemap of the United States.



(a) Raytrace through "normal" ionosphere. (b) Raytrace through "eclipsed" ionosphere.

Figure 4: Simulated raytraces for QSO from AC4PA toward WE9V at 7 MHz.



(a) Raytrace through "normal" ionosphere. (b) Raytrace through "eclipsed" ionosphere.

Figure 5: Simulated raytraces for QSO from AA4V toward PY1NB at 7 MHz.

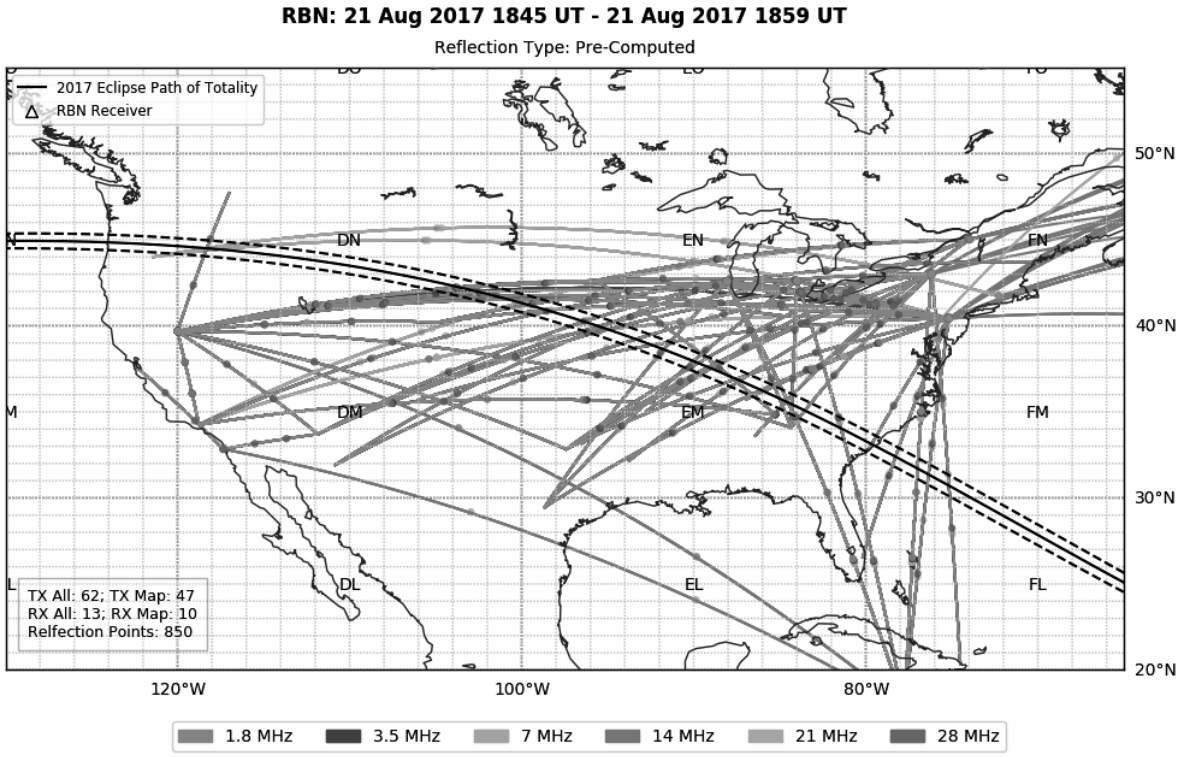


Figure 6: Sample map showing successfully raytraces.

Summary

This paper presented the steps and methodology taken in order to produce an eclipse simulation program capable of simulating a large number of transmit-receive pairs within a relatively short timeframe. This was accomplished by using basic parallelization techniques and a distributed cluster computer. In addition, the eclipse simulation program makes use of the U.S. Naval Research Laboratory's SAMI3 ionospheric model which provides both the "normal" and "eclipsed" ionospheres necessary to perform raytracing.

This simulation program is designed to aid in understanding radio propagation both during the eclipse as well in general. Along with the data collected from HamSCI's Solar Eclipse QSO Party, the data generated by the simulation program can be utilized to further improve our understanding of the ionosphere and the events that change it.

References

[1] D. Lu, "Here's every total solar eclipse happening in your lifetime. Is this year your best chance?," *The Washington Post*, 2017.

[2] "Solar eclipse QSO party." http://hamsci.org/seqp.

[3] M. Cervera, "PHaRLAP: Provision of High-frequency Raytracing LAboratory for Propagation studies," 2016. *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.*

[4] M. Moses, “eclipse2017_ionopath,” 2017. https://github.com/km4ege/eclipse2017_ionopath.

[5] S. Burujupalli, G. D. Earle, M. J. Ruohoniemi, and H. S. Dhillon, “Ray-tracing to study the ionosphere during eclipse,” 2017.

[6] J. D. Huba and J. Krall, “Modeling the plasmasphere with SAMI3,” *Geophysical Research Letters*, vol. 40, pp. 6–10, 2013.

[7] 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*, vol. 44, 2017.

[8] N. Jakowski, S. M. Stankov, V. Wilken, C. Borries, D. Altadill, J. Chum, D. Buresova, J. Boska, P. Sauli, F. Hruska, and L. R. Cander, “Ionospheric behavior over Europe during the solar eclipse of 3 October 2005,” *Journal of Atmospheric and Solar-Terrestrial Physics*, vol. 70, pp. 836–853, 2008.

[9] New Jersey Institute of Technology, “Tartan high performance computing initiative.” https://ist2.njit.edu/tartan-high-performance-computing-initiative/.