Space Hazards Applications, LLC

SHELLS-hires Algorithm and Software

Version 1.0

SHELLS-hires Algorithm

Document Version Summary

|  |  |  |  |
| --- | --- | --- | --- |
| Version | Date | Authors | Revision Description |
| 1.0 | 09/2020 | J.C. Green | Initial Release (describes the POES/MetOp data pre-processing) |
|  |  |  |  |
|  |  |  |  |

# Introduction

## Purpose of this document

The purpose of this document is to describe the processing steps and software used to create the SHELLS-hires model of the electron radiation belts.

## Who should use the document

This document is intended for those interested in how the SHELLS-hires model was developed and how to use the software created for that development.

# Observing System Overview

## Products generated

There are a number of different steps required to both develop and run the SHELLS-hires model and thus a number of different products are generated along the way. Here we describe the products generated when creating the low altitude electron flux data used as input to develop the model. Other products will be added to this section as the model development progresses.

### Input Data Preparation

To create and run the SHELLS model requires an input dataset of low altitude electron fluxes. To generate this dataset of electron fluxes, a number of additional intermediate data products are produced that are each described here.

#### L binned electron flux data files

The first step to producing the electron flux used to create the SHELLS model is to transform the 2 second time series of data from the POES/MetOp satellites to a new times series with average flux in .25 L shell bins from 1 to 8 for each satellite pass through the radiation belts. Since the satellites are in polar orbits, they make 4 passes through the radiation belts each orbit. The reason for creating these intermediate data files is to decrease the size and complexity of the POES/MetOp data. With these new smaller files the steps that follow can be repeated and modified more quickly as the model is developed. The original datafiles are extremely large because they contain data from both the TED and MEPED instruments as well other ancillary information. In addition, the satellites spend a large portion of the orbit over the polar cap region where there are typically no high energy electron fluxes. The processing code used to create these L binned data (make\_monthly\_Lbins\_data.py) can be used to produce a data file with L binned averages of any of the processed POES data values. The variables to include in the L binned data file can be chosen as an input to the code. For this work, L binned data files are created that contain the median time for each pass along with L\_IGRF, lat, lon, MLT, mep\_ele\_tel90\_flux\_e1, mep\_ele\_tel90\_flux\_e2, mep\_ele\_tel90\_flux\_e3, mep\_ele\_tel90\_flux\_e4, Btot\_sat, meped\_alpha\_90\_sat all averaged, and the North/South direction (N=1, South = 0) in .25 L bins. Lastly, the hourly Kp value is added for each pass interpolated to the median pass time. (The processing code can add other hourly omni values to the datafiles if requested but they are not used at this time.)

#### Cumulative distribution function files

Once the monthly L binned data files are created, they are used as input to generate both yearly and multi-year files with information about the cumulative distribution functions (cdfs) of the data values. For the SHELLS-hires model cdfs are created for the 90 degree electron flux measurements binned by hemisphere, satellite direction, L, and longitude. There are 2 hemisphere bins (North and South) and 2 satellite direction bins (Northward and Southward). The L shells are binned from 1 to 8 in .25 L increments and the longitudes are divided into 10 degree bins from 0 to 360. The files that are created contain the percentile level for each data value as well as the data for each percentile.

#### SAR mapped electron flux data files

In order to create and run the SHELLS neural network requires input electron fluxes with orbital variations removed. One data product generated is POES/MetOP electron flux data that has been manipulated in order to remove orbital variations using the SAR technique. The final product is monthly netcdf files. Each record in these files contains electron fluxes for each satellite pass through the radiation belts from L=1-8 in .25 L bins mapped to the reference longitude of 20 degrees.

### SHELLS Output products

TBD

## Instrument Characteristics

### POES/MetOp MEPED instruments

The primary input dataset used here is the electron fluxes from the Medium Energy Proton and Electron Detector (MEPED) carried by the POES satellites operated by NOAA and the MetOp satellites operated by EUMetSat. This constellation consists of satellites in highly inclined polar orbits at ~850 km. Since July 1979, 11 POES satellites and 3 MetOp satellites have launched that carried a version of MEPED, making the dataset an excellent source for understanding long term and near real time variations in the space environment. For the SHELLS-hires project, we focus our initial analysis on data from 5 satellites (NOAA 15, 18, 19 and MetOp-01 (or MetOp-B) and MetOp-02 (or MetOp-A)) only because our immediate interest is in comparing the data with that from the Van Allen Probes.

The MEPED instruments measure high energy electron and proton flux with both telescope and omnidirectional dome detectors  [*Green*, 2013; *Evans and Greer*, 2000]. Our initial analysis considers only the electron telescope measurements but the method could also be applied to those from the proton telescopes. Additionally, the telescope detectors measure the electron flux in two directions: roughly radially outward from the Earth (0 degree detector) and perpendicular to the radial direction (90 degree detector). For the SHELLS model, we focus on data from the 90 degree detector. (Data from the 0 degree detectors are more severely impacted by contaminating proton flux that should be evaluated and removed prior to applying the SAR technique.) The MEPED data are currently made available online by the NOAA National Centers for Environmental Information (NCEI) at <https://www.ngdc.noaa.gov/stp/satellite/poes/dataaccess.html>. Several versions of data are available: raw, processed corrected, and processed uncorrected data. The raw data gives the original instrument counts not translated into meaningful particle flux units. The processed corrected data [*Peck et al.*, 2015] gives the data translated from counts into flux with an attempt to correct or remove the known proton contamination in the electron detectors. Unfortunately, this data is only available up to 2014 and does not include the full time period of the Van Allen Probes era that we are interested in. This data is also not suitable for any real time products since it does not update as new data is collected. The processed uncorrected data used here, provides the data in flux units (#/cm2-s-sr) at four integral energy channels: >40 keV (e1), >130 keV (e2), >287 keV (e3), and >612 keV (e4). This format of the data includes data beginning in 2012 and is also updated in real time as data is received by the ground stations.

### Van Allen Probes ECT

Coming soon …

# Algorithm Description

## Algorithm Overview

SHELLS is a system developed for creating a model of Earth’s electron radiation belts. The model uses a neural network to create a mapping from low altitude POES/MetOP electron flux measurements to the high altitude Van Allen Probes electron measurements. Once the mapping is established, the electron flux throughout the magnetosphere can be projected using only the POES/MetOp data. [*Claudpierre and O’Brien* , 2020] describes the original proof of concept version of the SHELLS model. Our goal is to improve upon this proof of concept by increasing the time and spatial resolution of the model and adding forecast capabilities and uncertainty estimates.

In order to increase the time resolution of the SHELLS model first requires some manipulation of the POES/MetOP data that is the input to the neural network. The reason for modifying the input data is that the electron fluxes vary with the longitude of the satellite. The longitude dependence is caused by the fact that Earth’s magnetic field strength varies as a function of L and longitude. High fluxes are typically measured in regions where the magnetic field strength is low and vice versa. To remove these longitude variations we use the Statistical Asynchronous Regression (SAR) [*O’Brien et al*, 2001a] technique to map the measured fluxes to one consistent longitude at each L. The SAR method works by establishing the connection between the statistical distributions of measured fluxes at different locations. The assumption of the method is that fluxes at different locations are highly correlated such that the percentile flux levels will be the same at different locations. For example, the method would predict that if a satellite at L=6 measures fluxes at the 90% level at 85 degrees longitude, the fluxes will be at the 90% level simultaneously at all other longitudes. To map measurements from one L and longitude to a consistent longitude requires knowledge of the cumulative distribution of the fluxes at each L and longitude. Section 3.2.1 describes the code that establishes the mapping procedure in more detail.

A description of the neural network algorithm will be added as it is developed.

## Processing Outline

This section describes the python code used to manipulate the POES/MetOp data and create files needed as input for developing the neural network mapping between POES/MetOp and Van Allen probes.

### Creating POES/MetOp input electron flux data

There are 4 steps to creating the input electron fluxes as outlined below.

1. Create monthly L binned data files for each POES/MetOp satellite
2. Create yearly cumulative distributions for each POES/MetOp satellite
3. Create multi-year cumulative distributions for each POES/MetOp satellite
4. Create SAR mapped electron flux data files

#### Create monthly L binned data files

The code to create the monthly L binned data files is make\_monthly\_Lbins\_data.py and can be called from the command line or as a function:

make\_monthly\_Lbins\_data.py

*PURPOSE: To create monthly Lbin files of the POES/MetOp electron data that are faster to work with then the full NGDC netcdf files  
   
INPUTS:***:param***: sdate - (datetime) start date to create files***:param***: edate - (datetime) end date to create files***:param***: sat (str) - satellites name i.e. 'm02'***:param***: config (str) - the config file name (with paths) or the top directory of POES data***:param***: odirec (str) - directory to put the Lbin data (default ./Lbindata)***:param***: vars list(str) - variables to get, i.e. ['mep\_ele\_tel90\_flux\_e1' , 'mep\_ele\_tel90\_flux\_e2']***:param***: omvars list(str) - omni variables to get, i.e. ['Kp\*10', 'Dst']*

:**param***: mplots (0 or 1) – flag to make quality check plots or not  
  
OUTPUTS:  
Creates netcdf files called /odirec/yyyy/poes\_Lbin\_sat\_yyyymm.nc*

*USAGE (command line):*

*This example will create monthly L bin files and plots for the metop2 satellite and the year 2013 using the configfile to define the location of the POES data and /SHELLS/*

*Example:  
python make\_monthly\_Lbins\_data -s 2013-01-01 -e 2013-12-31 -sat m02 -d /SHELLS/configfile.ini -od /SHELLS/ -pt*

*DEPENDENCIES:*

*poes\_utils.py – module with many different poes processing routines*

*data\_utils.py – module for downloading and reading the omni data*

#### Create yearly cumulative distributions for each satellite

Once the monthly L binned data files are created, they are used to create files with the cumulative distributions of the electron flux for each satellite over a year using make\_yearly\_cdf.py.

*PURPOSE: This program takes monthly Lbin files and turns them into yearly cumulative distributions as a function of L, lon, hemisphere, sat direction, and Kp  
  
INPUTS:***:param***: syear (int)- The start year (YYYY) to create yearly cdfs for***:param***: eyear (int)- The end year (YYYY) to create yearly cdfs for***:param***: sat (str)- The satellite to create yearly cdfs for***:param***: odir (str)- The directory for putting the output files***:param***: evars list(str)- A list of the variables to create cdfs for  
  
NOTE!!! This is currently only set up to work well with the default variables.  
It will need to be updated in the future to work with any variable by automatically defining the cdfs bins from the data. Right now the bins are hard coded with what works best for particle fluxes, mag field, and pitch angles.*

*OUTPUTS  
netcdf files called odir/sat/poes\_cdf\_sat\_YYYY.nc'*

*The output files contain the cdf percentiles for each variable, the value for each percentile, and the total counts in each bin. The data are binned by hemisphere (N==0, S==1), satellite direction (Northward = 0 Soutward =1), L (0 to 8 in .25 increments), longitude(0 to 360 in 10 degree bins) and Kp (0-10).  
  
For example:*

*data = nc4.Dataset(‘odir/sat/poes\_cdf\_sat\_YYYY.nc')*

*data['mep\_ele\_tel90\_e1’][0,0,8,0,1,:] would give the percentile value for fluxbins (0 to 8 in .1 increments)in the Northern hemisphere when the satellite is moving northward at L=3-3.25, lon =0-10 and Kp=1-2.  
  
data['mep\_ele\_tel90\_e1\_sar][0,0,8,0,1,:] would give the flux values for percentile bin (0 to 1 in .1 increments)in the Northern hemisphere when the satellite is moving northward at L=3-3.25, lon =0-10 and Kp=1-2.  
  
data['mep\_ele\_tel90\_e1\_n][0,0,8,0,1] would give the number of values (0 to 1 in .1 increments)in the Northern hemisphere when the satellite is moving northward between L=3-3.25, lon =0-10 and Kp=1-2.*

*USAGE (command line)*  
python make\_yearly\_cdf -s 2013 -e 2015 -sat m02 -d /SHELLS/Lbindata -od /SHELLS/cdfdata

#### Create multi-yearly cumulative distributions for each satellite

Yearly cdf files are created in order to ensure that the data each year is reasonable and to note any time changes. However, in order to improve the statistics it is better to use cdfs base on several years of data. For SHELLS-hires we use cdfs from 5 years of data. The code that will take a range of yearly cdf files and accumulate them together is called make\_multi\_year\_cdf\_vars.py

*PURPOSE: This program takes the yearly cdf files and turns them into total cdfs over many years***:param***: syear\_all (int) - YYYY the start year to combine cdf data***:param***: eyear\_all (int) - YYYY the end year to combine cdf data (will include the year)***:param***: sat (str) - the satellite name i.e. 'm02'***:param***: dataloc (str) - the directory of the yearly cdf files***:param***: evars (list(str))- the variables to create multi-year cdfs for (must be in the yearly files)***:param***: ofile(str) - a string to add onto the output file to identify it, i.e. could be 'V2'***:param***: plots(0 or 1) - 0 do not make plots (makes median plots and line plots of L vs lon for each variable/Kp)  
  
OUTPUTS: Creates a multi-year cdf file in dataloc/sat/ called poes\_cdf\_sat\_YYYY\_YYYY\_variable\_ofile.nc  
 NOTE: Files are created for each variable so the file is not so huge  
  
USAGE (from command line):  
python make\_multi\_year\_cdf -s 2014 -e 2019 -sat m02 -vars mep\_ele\_tel90\_flux\_e1 mep\_ele\_tel90\_flux\_e2 mep\_ele\_tel90\_flux\_e3 mep\_ele\_tel90\_flux\_e4 meped\_alpha\_90\_sat Btot\_sat -d ./cdfdata/ -o V2 -pt  
  
USAGE (as a function):  
import make\_multi\_year\_cdf\_vars as mcdf  
mcdf.make\_multi\_year\_cdf\_vars(2014, 2019, 'm02','./cdfdata/',['mep\_ele\_tel90\_flux\_e1', 'mep\_ele\_tel90\_flux\_e2','mep\_ele\_tel90\_flux\_e3', 'mep\_ele\_tel90\_flux\_e4', 'meped\_alpha\_90\_sat','Btot\_sat'],'V2',0)  
  
The two examples above will create cumulative distribution functions of data from 2014-2019 for the m02 satellite and the 6 variables listed using the yearly cdf files in the directory ./cdfdata. The commands will create netcdf files containing the cdf data for each variable with 'V2' appended on the end of the file name and no quality check plots.*

#### Create SAR mapped electron flux data files

Once the cdf data files are created for each year, they are used to create data files with the electron flux mapped to a reference longitude. The code to do this piece is called make\_training\_data\_var.py.

*PURPOSE: To create a datafile of electron flux mapped to one longitude with SAR to be used  
for developing the SHELLS neural network  
  
INPUTS:***:param***: sdate(datetime)- time to start processing data***:param***: edate(datetime)- time to end processing data***:param***: satlist(list(str))- i.e. ['n15','n18','n19','m01','m02']***:param***: varlist(list(str))- variables to process i.e. ['mep\_ele\_tel90\_flux\_e1', 'mep\_ele\_tel90\_flux\_e2',  
 'mep\_ele\_tel90\_flux\_e3', 'mep\_ele\_tel90\_flux\_e4']***:param***: cdf\_dir(str) directory where the cdf files are***:param***: Lbin\_dir(str) directory where the Lbin data files are***:param***: neur\_dir(str) directory for the output files***:param***: reflon(int) E longitude to map to (degrees)***:param***: syear\_all (int) The start year of the accumulated cdf file***:param***: eyear\_all The end year of the accumulated cdf file  
  
OUTPUTS: monthly pickle files with the SAR modified data to be used by the SHELLS neural network  
  
USAGE(command line)  
python make\_training\_data.py -s 2013-01-01 -e 2013-05-01 -sats n15 n18 n19 m01 m02 -cd ./cdfdata/ -ld ./Lbindat/ -nd ./neural\_data/ -l 20 -sy 2015 -ey 2018:*

## Physics of the problem

TBD

## Algorithm Output

### TBD

## References