SEP Modeling Challenge: Research to Operations SHINE 2019

Measurements for Challenge SEP Events

Organized by Kathryn Whitman and Ian Richardson

Kathryn.Whitman@nasa.gov

July 2019

Approach to Calculating SEP Values for this Challenge

- **SEP measurements depend on:**
  - **The instruments with which they are taken**
  - **Choices made during post-processing**
- Values are reported here for multiple spacecraft and data sets:
  - GOES-15 corrected EPEAD fluxes from the west-facing detector
    - Differential channels
    - Integral channels
  - GOES-13 corrected EPEAD fluxes from the west-facing detector
    - Differential channels
    - Integral channels
  - ESA's SEPEM (RSDv2) calibrated data set extending from 1974 – 2015
  - The SEPEM data set with a background-subtraction applied using a method developed at SRAG
- **Measurements display a range of values and model performance should be considered in the context of that range**

SEP Definitions for this Challenge

- **SEP event definitions are selected to align with those used operationally by the Space Radiation Analysis Group (SRAG) when supporting the International Space Station (ISS) Mission Control**
  - **Threshold:** >10 MeV proton flux exceeds 10 pfu (1/[cm^2 s sr])
    - **Consequence:** Awareness that proton levels are rising and provide support to Mission Control if an EVA is planned
  - **Threshold:** >100 MeV proton flux exceeds 1 pfu
    - **Consequence:** 24/7 support of Mission Control as proton levels can cause increased radiation dose behind shielding
  - SRAG currently monitors GOES-13 >10 MeV and > 100 MeV integral channels
  - Event end when 3 consecutive data points (15 minutes) fall below 0.85*threshold
    - Definition is somewhat arbitrary and was defined to keep the SRAG alarm code from retriggering as the proton flux decays slowly away around the threshold levels
    - >100 MeV fluxes below 1 pfu will not cause significant increase in dose behind shielding, so the part of the event above these levels is the most important to quantify

For Modelers: Data Preparation Package for SHINE 2019

- Kathryn Whitman developed a series of codes to help modelers calculate the values requested for the SHINE 2019 SEP Modeling Challenge session

➢ All codes in repository: https://github.com/ktindiana/operational-sep

- **operational_SEP_SHINE_wrapper.py**
  - Runs operational_sep_quantities.py for all SHINE events for all combinations of GOES-13, GOES-13, and SEPEM data types
  - Allows users to specify model info and runs operational_sep_quantities.py for model
  - Makes comparison plots with compare_data_model.py and saves to file
- **operational_sep_quantities.py**
  - Calculates all values requested for shine session for GOES and SEPEM measurements (https://shinecon.org/shine2019/session2019.php#session19)
  - **Can calculate the same values for any model that outputs integral or differential flux time series**
- **compare_data_model.py**
  - Make comparison plots between measurements and model results

operational_SEP_SHINE_wrapper.py

Analysis and Plots for All SHINE Events: operational_SEP_SHINE_wrapper.py

- Runs operational_sep_quantities.py for all GOES and SEPEM experiments
  - All values in the tables in this powerpoint template are saved in output files
- Runs operational_sep_quantities.py for your model
  - All values requested for SHINE are saved in output files
- Makes plots of measurements and model for easy visualization of model performance by running compare_data_model.py

Modify operational_SEP_SHINE_wrapper.py and operational_sep_values.py for your Model

Values that should be changed in the operational_SEP_SHINE_wrapper.py for your model

```
#!!!!!!!!!!!!!!!!!!! EDIT HERE !!!!!!!!!!!!!!!!!!!!!!
model_start_dates = ['2012-03-07','2012-05-17']
model_end_dates = ['2012-03-14','2012-05-20']

#Specify model file names associated with the SEP events as a list
#---expect files to be in data folder---
model_file_names = ['MODEL/sample_model_mar2012.csv',
                    'MODEL/sample_model_may2012.csv']

model_name = 'MODEL'
model_flux_type = 'integral'

#List all dates available for your model (comma separated, no spaces). e.g.:
#    '2012-03-07,2012-05-17'
#May or may not be the same as model_start_dates. Should be date on the day
#the SEP flux crossed threshold.
sep_dates = '2012-03-07,2012-05-17'
#!!!!!!!!!!!!!!!!!!! END EDITS !!!!!!!!!!!!!!!!!!!!!!
```

Values that should be changed in operational_sep_quantities.py for your model

```
######FOR USER DATA SETS######
#(expect the first column contains date in YYYY-MM-DD HH:MM:SS format)
#Identify columns containing fluxes you want to analyze
user_col = arr.array('i',[1,2,3,4,5,6])

#DELIMETER between columns; for whitespace separating columns, use " " or ""
user_delim = ","

#DEFINE ENERGY BINS associated with user file and columns specified above as:
#   [[Elow1,Ehigh1],[Elow2,Ehigh2],[Elow3,Ehigh3],etc]
#Use -1 in the second edge of the bin to specify integral channel (infinity):
#   [[Elow1,-1],[Elow2,-1],[Elow3,-1],etc]
user_energy_bins = [[300,-1],[100,-1],[60,-1],[50,-1],[30,-1],[10,-1]]
############################
```

INSTRUCTIONS: To Run the Wrapper Code (Mac)

Do steps 1 – 3 the first time you run the code to ensure you have the correct libraries.

For Macs, I prefer using Homebrew. If you don't have homebrew, you may install with:
`/usr/bin/ruby -e "$(curl -fsSL https://raw.githubusercontent.com/Homebrew/install/master/install)"`

1. Install python3.
   - If you use homebrew, you may use this command in the terminal:
     `brew install python`
2. Install additional libraries matplotlib, wget, scipy
   - ```
     pip3 install matplotlib
     pip3 install wget
     pip3 install scipy
     ```
3. If you try to run the code and get an error related to certificates, you may need to also run:
   `pip3 install certifi`
4. Run the code in terminal while inside the directory containing the code (try python if python3 doesn't work):
   `python3 operational_SEP_SHINE_wrapper.py`
     ```

INSTRUCTIONS: To Run the Wrapper Code (Windows)

This is one way to run the code on Windows. Do steps 1 – 2 the first time you run the code to ensure you have the correct libraries.

1. Install python3.
   - Download anaconda navigator from https://www.anaconda.com/ (this downloads python3 plus a bunch of packages)
2. Install additional libraries matplotlib, wget, scipy
   - In Anaconda Prompt (which comes with anaconda navigator, NOT in command prompt)
     ```
     pip install matplotlib
     pip install wget
     pip install certify
     ```
   - If pip install doesn't work, try: `conda install matplotlib`, etc
3. Run the code in Anaconda Prompt while inside the directory containing the code:
   ```
   py operational_SEP_SHINE_wrapper.py
   ```

operational_sep_quantities.py

PROVIDES:

Time to cross thresholds: >10 MeV exceeds 10 pfu; >100 MeV exceeds 1 pfu

Time to Peak Flux for >10 MeV, >100 MeV

Peak Flux for >10 MeV, >100 MeV

Duration >10 MeV, >100 MeV

Event fluence (total integrated event intensity) for >10 MeV, >100 MeV and energy spectrum

Code to Calculate SEP Values: operational_sep_quantities.py
Also for use with your time profile model

- A user-friendly python 3 code was developed by Kathryn Whitman (Kathryn.Whitman@nasa.gov) to calculate the values requested for the SHINE 2019 SEP Modeling Challenge.
- **This code is sent along with this presentation to all SEP modelers so that the exact choices and methodology for determining the SEP values is transparent to all.**

➢ If the code is run with GOES-13 integral channels, the output will exactly match that generated by the SRAG operational alarm code.

- **Additional intention of this code:** For **models that produce SEP time profiles**, this code can be used to calculate exactly the same values in exactly the same way as data.
- Users may input their own data set (e.g. model output or data with special post-processing, such as background subtraction)
- Users may define their own threshold, e.g. >50 MeV, 10 pfu, to control event start and end
- Note: The code is not smart enough to handle multiple SEP events in a row – please contact Katie if you want to use the code for a specific event and run into problems; Updates and modifications will be ongoing and made public

Description: Code to Calculate SEP Operational Quantities

- The code, **operational_sep_quantities.py**, does the following for a single SEP event:
    1. Creates “data” and ”output” directories
    2. Automatically downloads GOES-8 to GOES-15 data for requested time periods into “data”
        - User must match spacecraft to requested time period
    3. To use SEPEM data, the user must first manually download and unzip file (URL in code) into ”data” directory. The code will then break up the data set into yearly files for faster reading.
    4. Searches for data gaps (e.g. -999, assumes any negative flux is a data gap) and fills in the missing data by applying a **linear interpolation with time**
        - Zero flux is treated as a valid value as model code may have real zero values
    5. If user selects differential channels, code will estimate >10 and >100 MeV fluxes using **power-law interpolation across the energy bins**

Description: Code to Calculate SEP Operational Quantities

- The code, **operational_sep_quantities.py**, does the following for a single SEP event:

6. Calculates threshold crossing times, peak flux, peak time, rise time (onset to peak), event end, and duration
    - Calculates all for >10 MeV, 10 pfu and >100 MeV, 1 pfu thresholds; will additionally calculate for a user-input threshold
    - **An event ends when three consecutive data points fall below 0.85*threshold**
    - Peak flux is the highest flux measured between event start and end times

7. Calculates event-integrated fluence of original channels (integral or differential) as well as event-integrated fluence corresponding to >10 MeV and >100 MeV fluxes (estimated if the original channels were differential)

8. Fluences and event values are saved to files in “output” directory

9. Shows useful plots with start and end times identified (**close plots for program to continue**)

INSTRUCTIONS: To Run the Code

RUN CODE FROM COMMAND LINE, e.g.:

```
python3 operational_sep_quantities.py --StartDate 2012-05-17 --EndDate '2012-05-19 12:00:00' --Experiment GOES-13 --FluxType integral
--showplot
```

RUN CODE FROM COMMAND FOR USER DATA SET:

```
python3 operational_sep_quantities.py --StartDate 2012-05-17 --EndDate '2012-05-19 12:00:00' --Experiment user --ModelName MyModel
--UserFile MyFluxes.txt --FluxType integral --showplot
```

RUN CODE IMPORTED INTO ANOTHER PYTHON PROGRAM, e.g.:

```
import operational_sep_quantities as sep
start_date = '2012-05-17'
end_date = '2012-05-19 12:00:00'
experiment = 'GOES-13'
flux_type = 'integral'
model_name = '' #if experiment is user, set model_name to describe data set
user_file = ''
showplot = True
detect_prev_event = True
threshold = '100,1' #default; modify to add a threshold to 10,10 and 100,1
sep.run_all(start_date, end_date, experiment, flux_type, model_name, user_file, showplot, detect_prev_event, threshold)
```

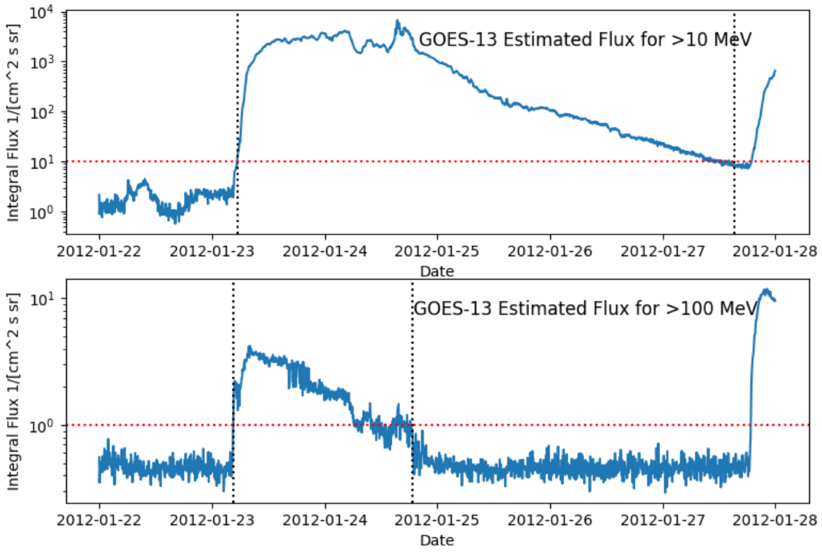
Example Run and Output – January 23, 2012

```
kwhitman$ python3 operational_sep_quantities.py --StartDate 2012-01-22 --EndDate 2012-01-29 --Experiment GOES-13 --FluxType differential --showplot
Checking that paths exist: data and output
Checking that the requested data is present on your computer.
Reading in data files for GOES-13.
Extracting fluxes for dates: 2012-01-22 00:00:00 to 2012-01-29 00:00:00
Checking for bad data values and filling with linear interpolation with time.

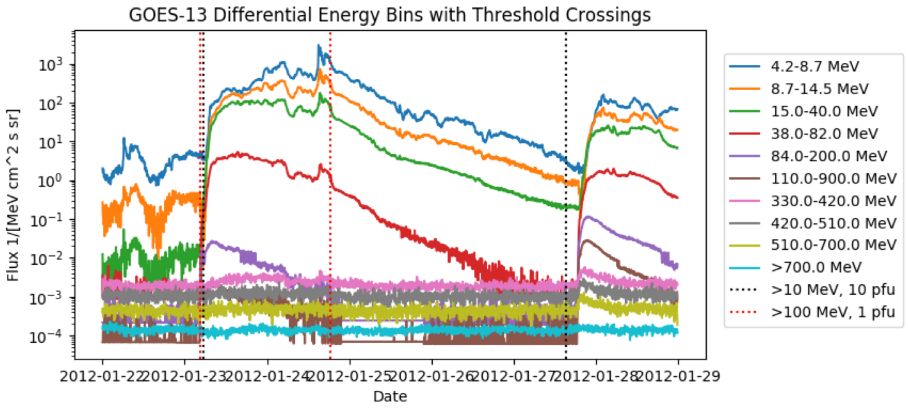
There is a data gap for time 2012-01-25 16:25:00 and energy 4.2 - 8.7 MeV. Filling in missing value with linear interpolation in time.
The first good value previous to gap is on 2012-01-25 16:20:00 with value 57.995
The first good value after to gap is on 2012-01-25 16:30:00 with value 58.943
Filling gap at time 2012-01-25 16:25:00 with interpolated flux 58.468999999999994
… (More output for corrections to other energy channels but left out to save space)
Finished checking for bad data.

Converting differential flux to integral flux for >10MeV.
Converting differential flux to integral flux for >100MeV.
Calculating threshold crossings and SEP event characteristics.
Flux      Threshold    Time Crossed         Peak Flux            Peak Time            Rise Time  End Time            Duration
>10 MeV   10 pfu       2012-01-23 05:25:00  6740.972728430253   2012-01-24 15:30:00  1 day, 10:05:00    2012-01-27 15:15:00  4 days, 9:50:00
Calculating threshold crossings and SEP event characteristics.
Flux      Threshold    Time Crossed         Peak Flux            Peak Time            Rise Time  End Time            Duration
>100 MeV   1 pfu       2012-01-23 04:35:00  4.2245477057861   2012-01-23 08:05:00  3:30:00    2012-01-24 18:40:00  1 day, 14:05:00
Extracting fluxes for dates: 2012-01-23 05:25:00 to 2012-01-27 15:15:00
=====Calculating event fluence for event defined by >10 MeV, for 2012-01-23 05:25:00 to 2012-01-27 15:15:00
Extracting fluxes for dates: 2012-01-23 05:25:00 to 2012-01-27 15:15:00
Event-integrated fluence for >10.0 MeV: 5028965340.697082 1/[cm^2]
Event-integrated fluence for >100.0 MeV: 4867346.341940369 1/[cm^2]
Extracting fluxes for dates: 2012-01-23 04:35:00 to 2012-01-24 18:40:00
=====Calculating event fluence for event defined by >100 MeV, for 2012-01-23 04:35:00 to 2012-01-24 18:40:00
Extracting fluxes for dates: 2012-01-23 04:35:00 to 2012-01-24 18:40:00
Event-integrated fluence for >10.0 MeV: 4254464976.467603 1/[cm^2]
Event-integrated fluence for >100.0 MeV: 3486739.9315438424 1/[cm^2]
Plotting estimated integral fluxes with threshold crossings.
Plotting fluxes in original energy bins. Any bad data points were interpolated. Lines indicate event start and stop for thresholds.
Plotting event-integrated fluence spectrum.
```

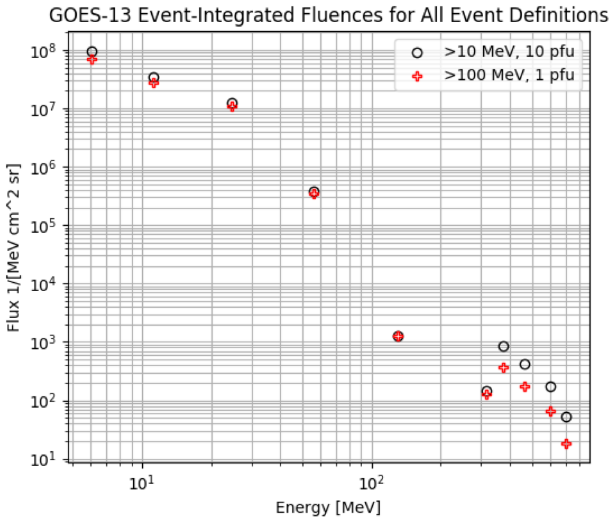
Example Run and Output – January 23, 2012



Integral fluxes estimated from the differential channels using power law interpolation across energy bins. Event start and end times for each threshold indicated by vertical black lines. Red line indicates 10 pfu threshold.



Plot of differential channels with the event start and end times for >100 MeV, 1 pfu threshold indicated by vertical black lines.



Event-integrated fluence of differential channels for the start and end times defined by >100 MeV, 1 pfu threshold. HEPAD channels could benefit from background-subtraction, especially when increase is small.

Output Files from Code – Fluence for Thresholds

Output files containing the **event-integrated fluence for each of the integral or differential energy bins**, as specified by user. There is a fluence file that corresponds to each threshold definition. For example, for GOES-13 spacecraft differential channels for Jan. 23, 2012 event:

>10 MeV, 10 pfu: fluence_GOES-13_differential_gt10_2012_1_23.csv
>100 MeV, 1 pfu: fluence_GOES-13_differential_gt100_2012_1_23.csv

	A	B	C	D
1	#Event defined by >10	10 pfu; start time 2012-01-23 05:25:00	end time 2012-01-27 15:10:00	
2	#Elow	Emid	Ehigh	Fluence 1/[MeV cm^2 sr]
3	4.2	6.044832504	8.7	97655914.41
4	8.7	11.2316517	14.5	34676630.66
5	15	24.49489743	40	12262732.19
6	38	55.82114295	82	377672.5668
7	84	129.614814	200	1279.650093
8	110	314.6426545	900	143.9770302
9	330	372.2902094	420	841.476201
10	420	462.8174586	510	415.3252485
11	510	597.4947699	700	172.30089
12	700	700	-1	52.21452675
13				

Output Files from Code – All Other Values for Thresholds

Output files containing the event-integrated fluence for the integral or differential energy bins, as specified by user. There is a fluence file that corresponds to each threshold definition. For example, for GOES-13 spacecraft differential channels for Jan. 23, 2012 event:

sep_values_GOES-13_differential_2012_1_23.csv

	A	B	C	D	E	F	G	H	I	J
1	#Energy Threshold [MeV]	Flux Threshold [pfu]	Start Time	Peak Flux 1/[MeV cm^2 s sr]	Peak Time	Rise Time	End Time	Duration	Fluence >10 MeV [cm^-2]	Fluence >100 MeV [cm^-2]
2	>10	10	1/23/12 5:25	6740.972728	1/24/12 15:30	1 day 10:05:00	1/27/12 15:15	4 days 9:50:00	5028965341	4867346.342
3	>100	1	1/23/12 4:35	4.224547706	1/23/12 8:05	3:30:00	1/24/12 18:40	1 day 14:05:00	4254464976	3486739.932
4										

Threshold definition

Time of first threshold crossing

Peak flux between start and end times

Time of Peak

Time from start to peak

End of event

End time – start time

>10 MeV and >100 MeV event-integrated fluences integrated between the start and end times for the threshold definition in the row. Units [cm^{-2}]

compare_data_model.py

Makes plots of:

Integral flux time profile

Fluence spectrum

Bar charts for timing and intensity

Plotting Code: compare_data_model.py

- Read in output from operational_sep_quantities.py
- Make useful visualizations to compare measurements with model results:
  - Time profile of >10 MeV and >100 MeV fluxes
  - Fluence spectrum
  - Bar charts – rise time, duration, peak flux, >10 MeV fluence, >100 MeV fluence