Data Manual

Data product description for Supra-Thermal and Energetic Particle Spectrometer (STEPS-ASPEX)

STEPS-ASPEX
December, 2024

Doc Id: ASPEX_STEPS_DATA_MAN_V_1.2

Version Control Sheet

Version	Date	Change/Revision
1.0	26/11/2024	Initial version. Includes details of data products from STEPS.
1.1	24/12/2024	Revisions after peer review suggestions. 1. Description of file naming convention 2. Overview of data processing software 3. Sample code for reading STEPS file
1.2	31/12/2024	Added a Data Quality section

Contents

1. Introduction

Aditya Solar wind Particle EXperiment (ASPEX) payload on-board ADITYA-L1 mission is a particle spectrometer developed by Physical Research Laboratory, Ahmedabad to monitor and analyse the heliospheric ions from solar wind as well as energetic particles from different directions. ASPEX is configured to have two different subsystems namely SWIS (Solar Wind Ion Spectrometer) and STEPS (Supra-Thermal and Energetic Particle Spectrometer).

This document provides a detailed description of the data products generated from the STEPS subsystem. It outlines the features of the STEPS data pipeline software, the definitions of different levels of data generated by STEPS, the calibration database, and related components. The STEPS subsystem generates up to Level-2, which are fully calibrated and ready for use scientific data. Users are encouraged to directly utilize these Level-2 files for their research. However, users familiar with data processing techniques can employ the calibration database and software modules available through PRADAN to create custom Level-2 data products tailored to specific requirements. All the data products of STEPS are written in Common Data Format (CDF) format (https://cdf.gsfc.nasa.gov/).

2. Data Access

Indian Space Science Data Centre (ISSDC) is the nodal agency for archival and dissemination of ASPEX data sets and a user is expected to download data from their server (https://pradan.issdc.gov.in/al1/) . A user has to first register on their website to be able to download the data sets. The data sets are then accessible via a PRADAN interface. General information regarding data download, terms of use, etc are available on the ISSDC website.

3. STEPS Data Pipeline Software

STEPS Data Pipeline Software generates higher levels of data products from Level-0 raw data. The software architecture is built entirely in Python, supported by an SQLite3 database, Bash scripts, and cron jobs. This integrated framework facilitates the automated generation of data products at various levels with proper version control, efficient database organization, and seamless automation of data retrieval and distribution. STEPS Data Pipeline Software has four modules along with the calibration database to generate higher levels of data products from Level-0 raw files. The modules are

1. steps_l1_gen
2. steps_sv_gen
3. steps_l1a_gen
4. steps_flux_gen

Following is the flowchart depicting the data flow.

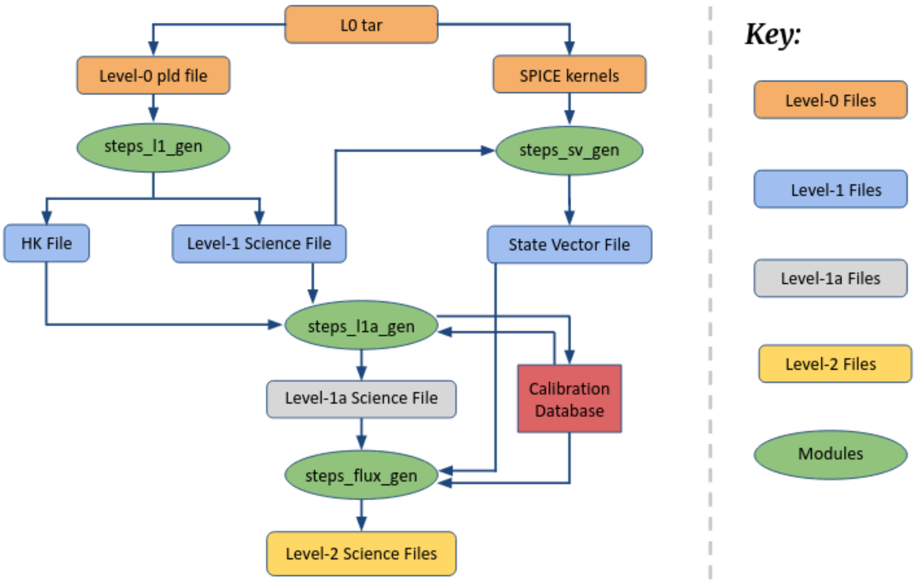

Figure 1: Overview of STEPS Data Pipeline Software workflow

Detailed descriptions of the software modules are provided in the following sections.

3.1 steps_l1_gen

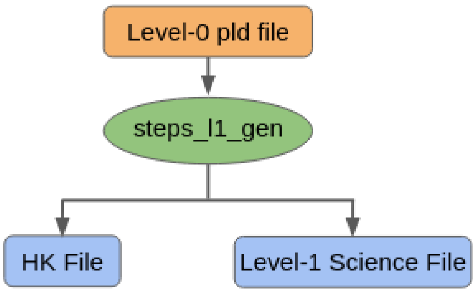

Figure 2: Flowchart of steps_l1_gen

Function: Conversion of Level-0 data file of STEPS (binary) into Level-1 science and House Keeping parameter (HK) files (CDF).

Process:
First, the raw data is read from the multiple dumps or a single payload data file. Redundant and bad data frames are then removed to ensure data integrity. The Universal Time Coordinated (UTC) information is extracted from the ground level header of the file, which is used to sort the valid data frames according to UTC. After sorting, the data is segregated into predefined variables based on the STEPS data packet structure. Finally, the cleaned, sorted data is written into two separate CDF files: one for Level-1 science data and the other for housekeeping data.

3.2 steps_sv_gen

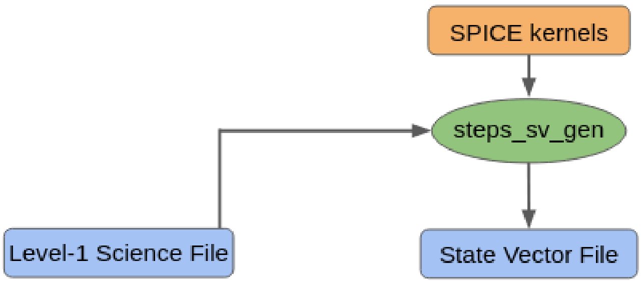

Figure 3: Flowchart of steps_sv_gen

Function: Generates position and velocity of spacecraft and look-angle information of instruments on STEPS sub-system.

Processes

The process involves reading the Meta-kernel file and Level-1 Science file. The Meta-kernel contains the names of the SPICE kernels relevant to the STEPS subsystem, which are used to retrieve the spacecraft's positional and velocity data. UTC time information is extracted from the Level-1 Science file to synchronize the data. Using the SPICE kernels, the position and velocity of the spacecraft are calculated with respect to the Geocentric Solar Ecliptic (GSE) coordinate system. Additionally, the angles subtended by the boresight of individual detector units of STEPS with respect to the X, Y, and Z axes of both the GSE and Radial Tangential Normal (RTN) coordinate systems are computed. Finally, a State vector file is created, and the computed position, velocity, and angle values are stored in that file for further use.

3.3 steps_l1a_gen

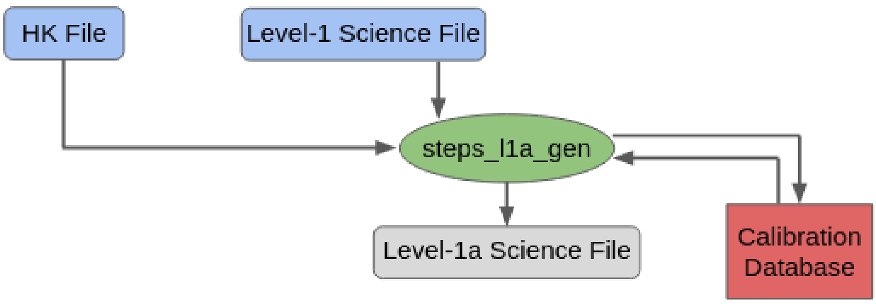

Figure 4: Flowchart of steps_l1a_gen

Function: Conversion of ADC channels into incident energy using calibration database

Processes

The process begins by reading the Level-1 Science and HK files. The data frames are then split into separate variables based on whether the calibration is "ON" or "OFF." For calibration "OFF" data frames, the deposited energy corresponding to the ADC channels is calculated using the parameters from the calibration database, with temperature variations also accounted for. This deposited energy is then converted to incident energy using fitting parameters from the calibration database. For calibration "ON" data frames, the instrument's gain is checked, and if a significant change is detected, the calibration database is updated accordingly.

3.4 steps_flux_gen

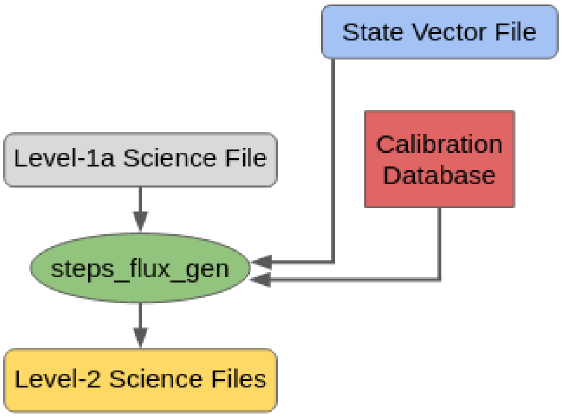

Figure 5: Flowchart of steps_flux_gen

Function: Generation of direction differentiated flux in pre-defined energy bins

Processes

The process includes reading the Level-1a Science file, State vector file, and necessary calibration files. The counts are converted to flux by dividing by the instrument's geometric factor, integration time, and energy bin width. Poisson error is also calculated and included in the final data product. Furthermore, the state vector information is appended to the final Level-2 data product.

4. File naming convention

A general file naming convention is followed for all data products related to STEPS. Common file name structure for STEPS L1 and L2 data products are:

`AL1_AST91_LN_TT_YYYYMMDD_Vmn.cdf`

File naming convention is explained is explained as follows:

AL1 : Spacecraft identifier (Aditya-L1)

AST91 : Payload identifier (ASPEX-STEPS, 91 is the identifier as used by L0 team to identify ASPEX data products)

LN : Data product level, N = 1 for level 1 and N = 2 for level 2 data products

TT: Description of file content. In Level-1, 'TT' can be either SC, HK, or SV. 'SC' refers to the Level-1 science file containing raw counts from all STEPS packages with respect to the ADC channels. 'HK' represents the housekeeping file, which includes parameters such as temperatures of detector units, voltage monitors, calibration information, and more. 'SV' corresponds to the state vector file, containing information like the position and velocity of the Aditya-L1 spacecraft, as well as the look-angle of different STEPS detector units. In Level-2, 'TT' can take values of PS, EP, NP, and IM, each corresponding to the detector units of STEPS and containing differentiated flux directions from the respective units. Note that 'SR' and 'SP' are not included in Level-2, as they face saturation, making the data unsuitable for scientific analysis; however, they are provided in Level-1.

5. Level-1 Data Product

STEPS Level-1 data consists of structured raw payload data frames, timestamped with UTC after processing to eliminate any corrupted data packets present in the Level-0 data.

Following table provides an overview of L1 data product description for STEPS.

#	Title	Description
1	L1 data products	STEPS - Three CDF files per day SC: Raw counts vs ADC HK: Instrument health parameters SV: State vector information **The CDF files are compressed using gzip with a compression level of 6*
2	Associated files	One meta file for each L1 CDF file
3	Science ready	No
4	Data Dissemination Unit	Individual L1 CDF file

Table 1: Overview of the Level-1 data product generated for STEPS

5.1 Data file Description

5.1.1 Level-1 Science File

File name: The file naming convention follows the description mentioned in section 3. It is as mentioned below:

`AL1_AST91_L1_SC_YYYYMMDD_Vmn.cdf`

Frequency of generation: 1 file per day, per observation ID.

File generation: The level-1 science file is created using the *steps_l1_gen* module within the STEPS data pipeline software. This module processes raw binary files (.pld) extracted from Level-0 tar files to create Level-1 outputs. Multiple Level-0 tar files are combined to produce daily datasets corresponding to a specific observation ID.

File description: The Level-1 Science File contains raw data frames and their corresponding UTC timestamps from the six detector units of the STEPS instrument: Sun Radial (SR), Parker Spiral (PS), Earth Pointing (EP), Intermediate (IM), North Pointing (NP), and South Pointing (SP). This file provides time-series data representing Counts versus ADC channel numbers for the detector units.

Variables: 14, metadata is stored in CDF.

Associated Files: 1 Meta file. The parameters of the meta-file are described in section 7.

5.1.2 HouseKeeping parameter File

File name: The file naming convention follows the description mentioned in section 3. It is as mentioned below:

`AL1_AST91_L1_HK_YYYYMMDD_Vmn.cdf`

Frequency of generation: 1 file per day, per observation ID.

File generation: The housekeeping parameter file is created using the *steps_l1_gen* module along with the above mentioned level-1 science file. This module processes the raw binary files (.pld) extracted from the Level-0 tar files and generates housekeeping parameter files as part of its output. Multiple Level-0 tar files are combined to produce daily datasets for a specific observation ID.

File description: The Housekeeping parameter file contains temperatures of different units of STEPS, gain modes, calibration pulses information, DAC values, the trigger counts in both high gain and low gain modes etc.

Variables: 113, metadata is stored in CDF.

Associated Files: 1 Meta file. The parameters of the meta-file are described in section 7.

5.1.3 State Vector File

File name: The file naming convention follows the description mentioned in section 3. It is as mentioned below:

`AL1_AST91_L1_SV_YYYYMMDD_Vmn.cdf`

Frequency of generation: 1 file per day, per observation ID

File generation: The *steps_sv_gen* module is responsible for generating the state vector file. It uses SPICE kernels (.bc and .bsp files) included in the Level-0 tar package, along with the Level-1 science file, as inputs. The state vector file provides information corresponding to the time duration covered in the science file.

File description: The state vector file includes the spacecraft's position and velocity in the GSE coordinate system, the orientation of the detector boresight in both GSE and RTN coordinates, and the angles between the spacecraft, the Sun, and the Earth in GSE coordinates.

Variables: 18, metadata is stored in CDF

Associated Files: 1 Meta file. The parameters of the meta-file are described in section 7.

6. Level-2 Data Product

Level-2 is the highest level of data product generated by the STEPS data pipeline software. The Level-2 file of STEPS will contain direction differentiated flux with incident energy information. The final data product will be given in logarithmically arranged predefined energy bins.

Following table provides an overview of L2 data product description for STEPS.

#	Title	Description
1	L2 data products	STEPS - Four CDF files per day PS/EP/IM/NP: Differential directional flux from PS/EP/IM/NP units **The CDF files are compressed using gzip with a compression level of 6*
2	Associated files	1 meta-file and 1 preview image for each L2 CDF file
3	Science ready	Yes
4	Data Dissemination Unit	Individual L2 CDF file

Table 2: Overview of the Level-2 data product generated for STEPS

6.1 Data file Description

6.1.1 Level-2 Science File

File name: The file naming convention follows the description mentioned in section 3. It is as mentioned below:

`AL1_AST91_L2_<unit>_YYYYMMDD_Vmn.cdf`

where `<unit>` can be PS, EP, IM, and NP depending on the detector unit.

Frequency of generation: 4 files per day

File generation: Level-2 science files are produced using the *steps_flux_gen* module within the STEPS pipeline software. This module takes as input an intermediate data product 'Level-1a files' derived from Level-1 files, calibration database and state vector file. The intermediate data product, created using the *steps_l1a_gen* module, is not archived at the ISSDC.

File description: The Level-2 science file provides differential flux data for four different directions. Each file includes the differential flux, incident energies, counting errors, detector look angles, and the spacecraft's position and velocity.

Users are advised to use data from the IM, PS-OUT, NP, and EP-OUT detectors for their scientific studies. Data from the inner detectors (PS-INN and EP-INN) should only be used after consulting with the payload team.

Variables: 23, metadata is stored in CDF.

Associated Files: 1 Meta file. The parameters of the meta-file are described in section 7.

7. Meta file information

Each CDF file produced by the STEPS Data Pipeline software is accompanied by a meta-file containing relevant information about the file. The table below outlines the various parameters included in the meta-file.

- ***Identification Details***

Parameters		Definition
mission		Aditya-L1
Payload	name	ASPEX STEPS
Payload	description	Description about ASPEX/STEPS
proposal_id		Contains proposal id of the particular CDF file
observation_id		Contains observation id of the particular CDF file
version		Version of CDF file (Refer section 7.1)

- ***Observation Details***

Parameters		Definition
Time_Coordinates	start_date_time	Start time of observation
Time_Coordinates	stop_date_time	End time of observation
Time_Coordinates	exposure_time_in_sec	Total duration of data contained in the CDF file in seconds
Time_Coordinates	integration_time_in_sec	Integration time of data contained in the CDF file in seconds. Default integration time is 1 second.
Event_Flag	assessment	0 or 1
Event_Flag	description	0 corresponds to 'Normal data' and 1 corresponds to 'Event Data'. The event flag will be generated automatically and this is yet to be implemented in the Data pipeline software.

- ***File Details***

Parameters	Definition
file_name	Filename according to file naming convention. Refer section 4.
file_size_in_kb	Total file size in Kilobytes.
md5_checksum	Checksum for data integrity

creation_date_time	Date and time at which CDF file is created
content_type	Following are identifier for content type parameter Level-1: SC/HK/SV Level-2: PS/EP/IM/NP

- ***Processing Details***

Parameters		Definition
total_input_files_used		Total number of files used by the software to create the CDF file
Input_Files_List	input_fiename	Files used by the software to create the CDF file
used_software_version		Version of the software used
level		Level of CDF file (Level1/Level2)
Quality	assessment	Value between 1-10
Quality	detailed_description	Refer section 7.2

- ***Remarks***

Parameters	Definition
remarks	General remarks regarding usage of STEPS data

Table 3: Meta file information

7.1 Data Product Versions

Each data product generated by the STEPS pipeline software is assigned a version number, represented in the CDF filenames using two variables, m and n, where both can range from 0 to 9. The first version of a data product is assigned the version number 01, and subsequent versions increment accordingly (e.g., 02, 03, etc.). The version number may be updated under several conditions, including modifications to the data processing software, updates in calibration methodologies, changes to SPICE kernels, or other significant alterations.

7.2 Data Quality

In Level 1 and Level 2, the data quality is assessed based on the health parameters of the STEPS unit that collected the data. If the instrument's health is good, the data quality is considered good; if the health is poor, the data quality is considered poor. The instrument's health is evaluated using the High Voltage Monitor (HVM) of a detector, with their ideal operational ranges established during calibration tests. The following parameters and their ideal ranges are used to determine data quality:

Parameters	Ideal Range (Volts)	Description
HVM_PS_Si	0.9 - 1	Voltage monitor of PS
HVM_EP_Si	0.9 - 1	Voltage monitor of EP
HVM_IM	0.475 - 0.575	Voltage monitor of IM
HVM_NP	0.475 - 0.575	Voltage monitor of NP

Table 4: HK Parameters used to quantify data quality

Data quality is evaluated based on the percentage of time, calculated over a single day, that the specified parameters remain within their ideal ranges. The table below outlines the quality definitions for Level-1 and Level-2 data products of STEPS.

Quality Level	Description	Daily coverage
1-6	Poor quality data	< 20%
7	Average quality data	20-50 %
8	Good quality data	50-70 %
9	Very good quality data	70-90 %
10	Excellent quality data	> 90%

Table 5: Description of the data quality levels for STEPS. These definitions are tentative and may be modified once year long data is reprocessed and compared with existing measurements at L1 point.

7.3 Event Flag

Manually identifying event dates in a large dataset can be challenging for users. To simplify this process, an event flag will be introduced in the STEPS datasets to indicate transient events. A value of '1' will denote the presence of event data, while '0' will indicate normal data. Although this variable is already included in the XML file, with a ‘0’, the implementation of transient event identification and labeling is still pending and will be incorporated in a future revision.

8. Other Data Products

In addition to the regular science observation data generated by ASPEX Payload Operation Centre (ASPEX-POC) for STEPS, there are some more data sets which are generated and sent to ISSDC. These are discussed in subsequent sections.

8. 1 Outreach Data Product

Outreach product for STEPS will be a day-wise lightcurve (variation of Flux with Time) for some particular energy bands. Following table provides an overview of outreach data product description for STEPS.

#	Title	Description
1	Outreach Product	1 png file Contains light curve in selected energy bands
2	Associated files	1 Meta file
3	Purpose	Outreach, quick look of dataset

Table 6: Overview of the Outreach data product generated for STEPS

File name: The file naming convention is as mentioned below:

`OR_AL1_AST_YYYYMMDD.png`

where 'OR' corresponds to Outreach product followed by spacecraft and payload identifier and date of observation.

Frequency of generation: 1 file per day

File generation: The outreach product image file is generated by *steps_outreach_gen* module of the data processing routine. The image file requires level-2 data products from a previous day's observation.

File description: As of now, it is planned to provide a light curve of Parker Spiral Unit in some predefined energy bands as outreach product.

Associated files: 1 JSON Meta file as per ICD.

ZIP bundle: The above two files will be bundled together in a zip archive and will be transferred to ISSDC. Similar to the previous cases, a text file containing file integrity will also be generated and transferred

8.2 Calibration Data

The data collected during the payload's calibration operations will be processed and transferred to ISSDC similar to level 1 and level 2 data files as described previously. Since it is expected that calibration operations will have a different proposal ID, it is expected that dissemination of these files will be restricted only to the PI institute. In this context, it may be noted that the data files obtained directly from calibration operations are termed as 'raw calibration data files' and will follow the above suggested plan for generation, archival and dissemination.

These files may be processed to further generate 'derived calibration data files' which may be used to update the calibration database. The derived calibration data files may also be generated by ASPEX-POC even without a calibration operation. These files along with the software will be hosted separately by ISSDC and they are not a part of the regular data stream. All calibration data files along with a calibration index file will be bundled together and sent to ISSDC as a single zip file with appropriate versioning. The ISSDC team has agreed to host these files in the 'Other Downloads' section of their webpage along with the software. At present, we expect the update in the calibration database 1-2 times in a year.

9. Code for reading STEPS file

All data products are generated in CDF v3.9 format. To read these files, users must have CDF libraries version 3.9 or higher installed. Python version 3.6 or above (along with the CDF libraries) can be used to read these files. In this section, a sample code is given to plot a lightcurve using data from the STEPS instrument. A lightcurve is a graphical representation of the flux received by the instrument over time. By extracting the relevant time and flux data from the STEPS CDF files, the variations in the measured flux can be visualized. The following python code illustrates the process of generating a lightcurve from the STEPS data.

```
#——————————————————————————————————————————————————————————————#
# Python code for reading Level-2 STEPS data file
#——————————————————————————————————————————————————————————————#
# Importing relevant packages
errors_list = [] # List to collect all the errors
try:
	import numpy as np
	from datetime import datetime, timedelta
	from spacepy import pycdf
	import numpy as np
	import matplotlib.pyplot as plt
	import matplotlib.dates as mdates
	import matplotlib.lines as mlines
	from matplotlib.ticker import ScalarFormatter
except Exception as e:
	errors_list.append("***ERROR:  Required  module  not  found:  "  +  str(e)
	+"***\n")
if errors_list:
	for error in errors_list:
		print(error)
	sys.exit()

#——————————————————————————————————————————————————————————————#
# Function definitions
#——————————————————————————————————————————————————————————————#

def energy_limits(ene_val):
	ene_l = np.round(ene_val.T[0], 2)
	ene_h = np.round(ene_val.T[1], 2)
	ebin_mean = []
	for i in range(len(ene_l)):
		ebin_mean.append((ene_l[i]+ene_h[i])/2)
```

```
        return ene_l, ene_h, ebin_mean

#—————————————————————————————#
# Reading STEPS Level-2 PS file
#—————————————————————————————#
file_path = "AL1_AST91_L2_PS_20240701_V01.cdf" # Path of STEPS CDF File (update as
per your file location)
data = pycdf.CDF(file_path)
print("————————————————————")
print("Global Attributes")
print("————————————————————")
print(data.attrs)
print("————————————————————")
print("Variables")
print("————————————————————")
print(data)
print("————————————————————")
print("Local attributes - Epoch")
print("————————————————————")
print(data["Epoch"].attrs)
print("————————————————————")
print("Local attributes - Differential flux of PS_Outer")
print("————————————————————")
print(data["PS_Outer_allspecies"].attrs)
print("————————————————————")
print("Local attributes - Energy bins of PS_Outer")
print("————————————————————")
print(data["Ebin_outer_allspecies"].attrs)

# Storing data into different variables
epoch = data['Epoch'][:]
PS_Outer_flux = data['PS_Outer_allspecies'][:].T
PS_Outer_energy = data['Ebin_outer_allspecies'][:, :]
PS_Outer_energy_low, PS_Outer_energy_high, PS_Outer_energy_all =
energy_limits(PS_Outer_energy)

#—————————————————————#
# Plotting Light Curve
#—————————————————————#
fig = plt.figure(1, figsize=(18, 10))
gs = fig.add_gridspec(1, 1, hspace=0, wspace=0)
ax = gs.subplots(sharex=False, sharey=False)
marker_size = 1
legend_handles = []
colors = [
        '#FF4500', # Orange Red
        '#32CD32', # Lime Green
        '#1E90FF', # Dodger Blue
        '#FF69B4', # Hot Pink
        '#8A2BE2', # Blue Violet
        '#FFD700', # Gold
```

```
        '#40E0D0', # Turquoise
        '#FF6347', # Tomato
        '#9370DB', # Medium Purple
        '#3CB371', # Medium Sea Green
        '#FFB6C1', # Light Pink
        '#4682B4', # Steel Blue
        '#FFA07A', # Light Salmon
        '#7FFF00', # Chartreuse
        '#00CED1', # Dark Turquoise
        '#6495ED', # Cornflower Blue
        '#DC143C', # Crimson
        '#D2691E', # Chocolate
        '#BA55D3', # Medium Orchid
        '#F4A460' # Sandy Brown
]

for i in range(len(PS_Outer_flux)):
        scatter = ax.scatter(epoch, PS_Outer_flux[i], s=marker_size,
color=colors[i])
        legend_handles.append(mlines.Line2D([], [],
color=scatter.get_facecolor()[0], marker='o', markersize=6))

ax.tick_params(direction='inout', length=12, width=1.5, colors='k')
ax.set_yscale('log')
ax.set_ylabel('PS_Out \nIon Flux', fontsize=14, weight='bold')
ax.tick_params(direction='inout', length=12, width=1.5, colors='k')
[x.set_linewidth(1.5) for x in ax.spines.values()]
for label in (ax.get_xticklabels() + ax.get_yticklabels()):
        label.set_fontsize(14)

ax.legend(legend_handles, [f'{PS_Outer_energy_low[i]:1.2f} -
{PS_Outer_energy_high[i]:1.2f} MeV' for i in range(len(PS_Outer_flux))], loc='upper
right', fontsize=11, frameon=False, bbox_to_anchor=(1.13, 1.1), ncol=1,
columnspacing=1)

ax.grid(True, which='both', linestyle='--', linewidth=0.5, alpha=0.7)

box = ax.get_position()
ax.set_position([box.x0, box.y0, box.width*0.85, box.height])

ax.xaxis.set_major_formatter(mdates.DateFormatter('%b %d \n %H:%M'))
ax.grid(True, which='both', linestyle='--', linewidth=0.5, alpha=0.7)

fig.text(0.5, 0.04, 'UTC', ha='center', va='center', fontsize=18, weight='bold')
plt.show()
```