

All Sky Flux Map and Energy Resolved Pitch Angle Tutorial

Authors: Patrick J. Tracy (ptracy@umich.edu), Jim M. Raines (jraines@umich.edu)

Introduction

The purpose of this tutorial is to enable the user to construct 2D angular flux maps (AFM) and energy-resolved pitch angle (ERPA) distributions from observations by the Fast Imaging Plasma Spectrometer (FIPS) on MESSENGER. These data products are useful for a variety of investigations of the Mercury plasma environment. Studies of the plasma mantle and sodium plasma population have made extensive use of the AFM to investigate plasma isotropy or bulk flow directions (Gershman et al. 2015; DiBraccio et al. 2015). Work done with the magnetic cusp at Mercury utilized both the AFM and the ERPA distribution. With the combination of the directional and pitch angle information, details of the energization and precipitation processes in the cusp could be elucidated (Raines et al. 2014).

These products are provided in the NASA PDS data at a fixed time cadence, but investigators often want to produce them over intervals specific to their particular analysis, to improve the signal-to-noise ratio and simplify the interpretation. This tutorial provides the instructions needed to acquire data products from NASA PDS and calculate the AFM and ERPA distribution. Routines in the IDL programming language and sample data required to reproduce the results are included with this tutorial. These IDL code examples should be run from the “tutorial_code” directory contained with the tutorial files

Acquiring the data for this tutorial

The data from the FIPS sensor for the MESSENGER mission can be found at the NASA Planetary Data System (PDS). For this tutorial, we will acquire all the data files required to create the AFM and ERPA distributions from the FIPS data acquired on May 26th, 2012.

1) Go to the NASA PDS website (<https://pds.nasa.gov>)

2) Enter the search refinements (on left hand side of screen) of Planet= Mercury, Type= Data Set, Investigation = MESSENGER, and Instrument= Plasma analyzer. This narrows the data set down to the EPPS suite (Energetic Particle and Plasma Spectrometer), of which the Fast Imaging Plasma Spectrometer (FIPS) is a part.

3) For this tutorial the user will need the raw PHA information from the FIPS sensor, this can be found in the data set [MESSENGER E/V/H/SW EPPS CALIBRATED FIPS V1.0](#). In this folder the user will need the pulse height analyzed (PHA) data or PHA word data. A PHA word is the name given to full event words measured by FIPS. These come directly from the instrument and contain all measured quantities for a particular ion event (E/q, time-of-flight, and X/Y position on the MCP detector). In ground processing these quantities are converted to physical units. The PHA data can be found by following the link to the Planetary Plasma Interactions Website and navigating the folder system found there to the file(s)

“Top/DATA/FIPS_PHA/2012/122_152_MAY/FIPP_P2012147CDR_V3/FIPP_P2012147CDR_V3.TAB”

“Top/DATA/FIPS_PHA/2012/122_152_MAY/FIPP_P2012147CDR_V3/FIPS_PHA_CDR.FMT”.

This file will be downloaded for use in this tutorial and contains the incident velocity information for PHA measurements by the FIPS sensor.

The full static link to the data file used in this tutorial is

http://pds-ppi.igpp.ucla.edu/search/view/?f=yes&id=pds://PPI/MESS-E_V_H_SW-EPPS-3-FIPS-CDR-V1.0/DATA/FIPS_PHA/2012/122_152_MAY/FIPP_P2012147CDR_V3&o=1

4) The computation of the flux map for a given time period requires the time-dependent attitude information for the FIPS sensor. This attitude information, relative to the Mercury Solar Orbital (MSO) coordinate system is contained as a rotation matrix that rotates from FIPS instrument coordinate system to the MSO frame. In the MSO Cartesian frame, the X-direction points toward the Sun, the Z-direction points normal to Mercury’s orbital plane (positive toward the north celestial pole), and the Y-direction completes the right-handed coordinate system (positive duskward). One rotation matrix is provided for each scan of the FIPS sensor (about 10 seconds).

These rotation matrices are stored in the DDR data set, [MESSENGER E/V/H/SW EPPS CALIBRATED FIPS DDR V2.0](#), and can be found by following the link to the Planetary Plasma Interactions Website and navigating the path to the following files:

“Top/ DATA/ FIPS_ROTMSO/ 2012/ 122_152_MAY/FIPS_ROTMSO_2012147_DDR_V02/FIPS_ROTMSO_2012147_DDR_V02.TAB”

“Top/ DATA/ FIPS_ROTMSO/ 2012/ 122_152_MAY/FIPS_ROTMSO_2012147_DDR_V02/FIPS_ROTMSO_DDR.FMT.”

This file will be downloaded for use in this tutorial and contains the rotation information needed to create the all sky flux map.

Full static link

http://pds-ppi.igpp.ucla.edu/search/view/?f=yes&id=pds://PPI/MESS-E_V_H_SW-EPPS-3-FIPS-DDR-V2.0/DATA/FIPS_ROTMSO/2012/122_152_MAY/FIPS_ROTMSO_2012147_DDR_V02&o=1

5) The computation of the AFM and ERPA further requires some ancillary data for the FIPS sensor. The general location for this data is in the CDR data location, [MESSENGER E/V/H/SW EPPS CALIBRATED FIPS V1.0](#). From this location, one follows the link to the Planetary Plasma Interactions Website and navigate the folder system found there to the “Top/CALIB/” location. From this location, there are several files to download.

-Top/CALIB/FIPA_F2009228CDR_V1/FIPA_F2009229CDR_V1.TAB, this file contains the mapping from MCP pixel to incident ion direction for all MCP pixels within FIPS field of view. Also contained is the solid angle for each MCP pixel.

- *Top/CALIB/FIPA_F2009228CDR_V1/FIPS_FOVPIXEL.FMT* , format file explaining each column of the previous file

-*Top/CALIB/FIPA_E2012045CDR_V1.TXT*, file containing the details of the E/q stepping table of the FIPS sensor.

-*Top/CALIB/FIPS_EQ.FMT* , format file explaining each column of the of E/q stepping table file

6) The calculation of the Energy Resolved Pitch Angle further requires magnetic field measurements acquired by the Magnetometer. These can be found in the following directory of the NASA PDS website. [MESS-E/V/H/SW-MAG-4-SUMM-CALIBRATED-V1.0](http://pds-ppi.igpp.ucla.edu/search/view/?f=yes&id=pds://PPI/MESS-E_V_H_SW-MAG-4-SUMM-CALIBRATED-V1.0)

In this directory, navigate to the following files:

-*Top/ DATA/ MSO/ 2012/ 122_152_MAY/ MAGMSOSCIAVG12147_01_V08/ MAGMSOSCIAVG12147_01_V08.TAB*”, magnetic field data

-*Top/ DATA/ MSO/ 2012/ 122_152_MAY/ MAGMSOSCIAVG12147_01_V08/ MAGMSOSCIAVG12147_01_V08.LBL*”, format file for magnetic field data

Full Static Link

http://pds-ppi.igpp.ucla.edu/search/view/?f=yes&id=pds://PPI/MESS-E_V_H_SW-MAG-4-SUMM-CALIBRATED-V1.0/DATA/MSO/2012/122_152_MAY/MAGMSOSCIAVG12147_01_V08&o=1

7) In order to have access to the scan type and accumulation times for every FIPS scan in a data period, one also needs the rate-based data. The rates contain less information than the PHAs, but have information at every time scan of FIPS. The PHA data (*FIPP_P*.TAB*) downloaded above have a lot of information about each PHA, including the scan it was recorded in. However, if a given scan has no recorded PHAs, then information about that scan’s time and time duration are not available in the PHA data file. To get this complete time information we require the *FIPS_R*.TAB* from NASA PDS. For the purposes of this tutorial only the columns pertaining to scan time and scan type from this file will be utilized.

The directory for the files to grab are located in [MESSENGER EPPS CALIBRATED DATA](http://pds-ppi.igpp.ucla.edu/search/view/?f=yes&id=pds://PPI/MESS-E_V_H_SW-MAG-4-SUMM-CALIBRATED-V1.0/DATA/MSO/2012/122_152_MAY/MAGMSOSCIAVG12147_01_V08&o=1)

In this directory navigate to the following files:

-*Top/DATA/FIPS_SCAN_2012/122_152_MAY/FIPS_R2012147CDR_V3/FIPS_R2012147CDR_V3.TAB*” (contains the ion counting rates for each time step of FIPS)

-*Top/DATA/FIPS_SCAN_2012/122_152_MAY/FIPS_R2012147CDR_V3/FIPS_SCAN_CDR.FMT*” (contains the format file for the counting rates data)

Summary of data products acquired:

- *FIPA_E2012045CDR_V1.txt* – details of the E/Q steps of the FIPS sensor
- *FIPA_F2009229CDR_V1.tab* – mapping from MCP pixels of FIPS to incident ion angles in FIPS frame
- *FIPP_P2012147CDR_V3.tab* - list of ions (PHAs) recorded by the FIPS sensor
- *FIPS_EQ.fmt* – format file for the E/q steps file for FIPS (*FIPA_E2011047CDR_V1.txt*)

- *FIPS_FOVPIXEL.fmt* –format file for the MCP pixel to incident ion angles file (*FIPA_F2009229CDR_V1.tab*)
- *FIPS_PHA_CDR.fmt* – format file for the list of ions recorded by FIPS (*FIPP_P2012147CDR_V3.tab*)
- *FIPS_R2012147CDR_V3.tab* – FIPS counting rates at each time scan
- *FIPS_ROTMSO_2012147_DDR_V02.tab* – list of rotation matrices from the FIPS sensor to the MSO frame
- *FIPS_ROTMSO_DDR.fmt* – format file for list of rotations from the FIPS sensor to the MSO frame (*FIPS_ROTMSO_2012147_DDR_V02.tab*)
- *FIPS_SCAN_CDR.tbl* –format file for *FIPS_R2012147CDR_V3.tab*
- *MAGMSOSCIAVG12147_01_V08.tbl* –format file for magnetic data from MESSENGER (*MAGMSOSCIAVG12147_01_V08.tbl*)
- *MAGMSOSCIAVG12147_01_V08.tab* – 1 second magnetic field data averages from the Magnetometer

Once all of these data product are acquired, it is possible to proceed to the actual analysis procedure.

These data products are read in and conveniently stored as arrays of structures in the IDL programming language. The “*read_in_tutorial_data.pro*” along with the .sav file templates (named **_template.sav*, for each data product) can accomplish the reading in and convenient storage of the data products listed above. The data products, the “*read_in_tutorial_data.pro*”, and the file reading templates are provided as part of this tutorial.

Calculation of flux map from PHA and ROTMSO data products

As noted above, flux map data products are available at the PDS, with a fixed time resolution. See the FIPS_FLUXMAP data product at the DDR data location where the ROTMSO data product was located for examples. The time and solid angle averaging required to properly create these products must be recomputed for time resolutions different than the default resolution available at the PDS. The purpose of this tutorial is to illustrate the creation of these data products for an accumulation of FIPS measurements over an arbitrary time period.

```
# MESSENGER/FIPS Decoded PHA Data
# E/Q Table used: FIPA_E2012045CDR_V1.txt
# SPICE kernels used: naif0011.tls, messenger_2548.tsc
# created: Tue Sep 15 2015
#
# Met          UTC_Time  Quality Scantype Pri/Dec Step  E/Q    X    Y    Zenith Azimuth  TOF TOFns  Wedge  Strip Zigzag  M/Q  Weight
246477866.000 2012-147T00:00:00.859 0 2 0 18 2.403 19 35 52 189 80 52.1 -9999 -9999 -9999 1.000e+00 2.228e-10
246477866.000 2012-147T00:00:00.859 0 2 0 22 1.636 17 41 66 211 72 46.9 -9999 -9999 -9999 1.000e+00 7.922e-10
246477866.000 2012-147T00:00:00.859 0 2 0 23 1.485 25 45 52 245 73 47.5 -9999 -9999 -9999 1.000e+00 9.704e-10
246477866.000 2012-147T00:00:00.859 0 2 0 26 1.113 32 49 60 278 59 38.4 -9999 -9999 -9999 1.000e+00 0.000e+00
246477866.000 2012-147T00:00:00.859 0 2 0 26 1.113 22 46 62 237 73 47.5 -9999 -9999 -9999 1.000e+00 1.951e-09
246477866.000 2012-147T00:00:00.859 0 2 0 26 1.113 29 32 -9999 -9999 73 47.5 -9999 -9999 -9999 1.000e+00 0.000e+00
246477866.000 2012-147T00:00:00.859 0 2 0 26 1.113 33 42 34 287 72 46.9 -9999 -9999 -9999 1.000e+00 1.298e-09
246477866.000 2012-147T00:00:00.859 0 2 0 27 1.011 36 38 24 326 76 49.5 -9999 -9999 -9999 1.000e+00 1.711e-09
246477866.000 2012-147T00:00:00.859 0 2 0 27 1.011 32 46 49 278 76 49.5 -9999 -9999 -9999 1.000e+00 0.000e+00
246477866.000 2012-147T00:00:00.859 0 2 0 27 1.011 29 37 -9999 -9999 72 46.9 -9999 -9999 -9999 1.000e+00 4.259e-09
246477866.000 2012-147T00:00:00.859 0 2 0 27 1.011 33 37 -9999 -9999 76 49.5 -9999 -9999 -9999 1.000e+00 2.352e-09
246477866.000 2012-147T00:00:00.859 0 2 0 27 1.011 36 44 45 301 75 48.8 -9999 -9999 -9999 1.000e+00 0.000e+00
246477866.000 2012-147T00:00:00.859 0 2 0 27 1.011 31 45 45 273 73 47.5 -9999 -9999 -9999 1.000e+00 0.000e+00
246477866.000 2012-147T00:00:00.859 0 2 0 27 1.011 43 43 59 329 74 48.2 -9999 -9999 -9999 1.000e+00 0.000e+00
246477866.000 2012-147T00:00:00.859 0 2 0 27 1.011 25 37 30 209 77 50.1 -9999 -9999 -9999 1.000e+00 1.822e-09
246477866.000 2012-147T00:00:00.859 0 2 0 28 0.918 32 40 24 281 73 47.5 -9999 -9999 -9999 1.000e+00 1.964e-09
246477866.000 2012-147T00:00:00.859 0 2 0 28 0.918 33 44 42 285 73 47.5 -9999 -9999 -9999 1.000e+00 0.000e+00
246477866.000 2012-147T00:00:00.859 0 2 0 28 0.918 21 49 -9999 -9999 73 47.5 -9999 -9999 -9999 1.000e+00 0.000e+00
246477866.000 2012-147T00:00:00.859 0 2 0 28 0.918 39 44 51 314 74 48.2 -9999 -9999 -9999 1.000e+00 0.000e+00
246477866.000 2012-147T00:00:00.859 0 2 0 28 0.918 32 48 56 278 81 52.7 -9999 -9999 -9999 1.000e+00 0.000e+00
246477866.000 2012-147T00:00:00.859 0 2 0 28 0.918 36 39 28 320 73 47.5 -9999 -9999 -9999 1.000e+00 0.000e+00
---
```

Figure 1: A sample of the PHA data file retrieved from the Planetary Data System

Figure 1 shows the sample PHA data that will be used in the construction of this data product. The key columns for our work are the time (MET or UTC) the energy per charge of an ion (E/q), the zenith and

azimuth angle of ions entering the instrument, and the phase space density associated with that ion measurement. In the data table, the column labeled “weight” corresponds to the phase space density associated with each ion measurement.

The format of this document resembles a recipe which is easily translated into other programming languages. Pseudo code will be utilized below to explain the analysis process.

Conversion from FIPS coordinates to MSO

The measurements from FIPS are a list of ions that FIPS has measured in the form of PHAs. This tutorial focuses on the processing of proton measurements, although the procedure can be applied to other heavy ions as well. Assuming the mass and charge for a given ion measurement (a row in the data

shown in Figure 1) one converts the E/q measurement for that ion to a velocity via the relationship $\frac{E}{q} = \frac{1}{2} \frac{m}{q} v^2$. This yields a velocity magnitude, which, along with the zenith and azimuth directions, can be converted to a velocity vector for each proton. In the data set shown above, protons can be distinguished from heavy ions by the value in the mass per charge data field; protons will have a value of 1.0 in that column.

This velocity vector is in the frame of the FIPS sensor, which is continuously changing as the spacecraft attitude changes. In the FIPS frame of reference, the zenith direction is along the sensor boresight vector (with some arbitrary azimuth component in the plane normal to the zenith direction). Due to its rotation, the FIPS frame of reference is not suitable for creating a flux map that is summed over multiple instrument accumulation times (scans). A given ion measurement may appear in different locations in the FIPS frame between two different scan times, but because the spacecraft rotated between those time points, the ions observed could have come from exactly the same velocity direction in the ambient space environment at Mercury.

Therefore a frame that is independent of the current spacecraft attitude is needed for accumulation of the individual ion measurements. That frame is the MSO frame. The ROTMSO data product provides the transformation between the FIPS coordinate frame and MSO frame for each measurement time step of the FIPS sensor.

```
# MESSENGER/FIPS EPPS --> MSO Frame Rotation Matrices
# created: Fri Dec 4 2015
#
# INDEX MET MATRIX_ROW_0 MATRIX_ROW_1 MATRIX_ROW_2
#
0 246477866.000 2.9010854221e-01 -9.2067380169e-01 2.6114514089e-01 9.5100612839e-01 3.0782962090e-01 2.8779650224e-02 -1.0688487970e-01 2.4000140701e-01 9.6487043023e-01
1 246477877.000 2.9010855919e-01 -9.2067721888e-01 2.6113307493e-01 9.5136778349e-01 3.0790280043e-01 2.5468039953e-02 -1.0361442931e-01 2.4104509779e-01 9.649478513e-01
2 246477887.000 2.9010770531e-01 -9.2067346094e-01 2.6114721939e-01 9.5164625108e-01 3.0624989731e-01 2.2454995644e-02 -1.0065186944e-01 2.4201523079e-01 9.6503771207e-01
3 246477899.000 2.9011288051e-01 -9.2067507010e-01 2.6113584943e-01 9.5202196205e-01 3.0542853614e-01 1.9172409316e-02 -9.7409899556e-02 2.4304490245e-01 9.6511164477e-01
4 246477909.000 2.9010336979e-01 -9.2067891167e-01 2.6113287128e-01 9.5234899540e-01 3.0459522045e-01 1.5510456992e-02 -9.4188246722e-02 2.4407350044e-01 9.6517173648e-01
5 246477919.000 2.9010628615e-01 -9.2067738514e-01 2.6113501341e-01 9.5263162978e-01 3.0385040625e-01 1.2957946760e-02 -9.127668531e-02 2.4500629161e-01 9.6521631089e-01
6 246477930.000 2.9010330122e-01 -9.2067759849e-01 2.6113757748e-01 9.5293304078e-01 3.0302487051e-01 9.7235676018e-03 -8.8083451418e-02 2.4602578471e-01 9.6525261868e-01
7 246477941.000 2.9011210106e-01 -9.2067287643e-01 2.6114444863e-01 9.5321897953e-01 3.0221106787e-01 6.4590900705e-03 -8.4504278639e-02 2.4704235404e-01 9.6527785574e-01
8 246477951.000 2.9010509922e-01 -9.2067640214e-01 2.6113980884e-01 9.5347241740e-01 3.0145882704e-01 3.5951146569e-03 -8.2032837692e-02 2.4794644375e-01 9.6529429471e-01
9 246477962.000 2.9010126355e-01 -9.2067892082e-01 2.6113517890e-01 9.5373901574e-01 3.0063552538e-01 4.1318645987e-04 -7.8886923768e-02 2.4893494295e-01 9.6530213539e-01
10 246477973.000 2.9010547131e-01 -9.2067708456e-01 2.6113415789e-01 9.5396684700e-01 2.998362231e-01 -2.4650333276e-03 -7.6043645770e-02 2.4982891865e-01 9.6532930707e-01
11 246477983.000 2.9010394789e-01 -9.2067845882e-01 2.6113382652e-01 9.5421257435e-01 2.990757999e-01 -5.6157510439e-03 -7.298971569e-02 2.5080625321e-01 9.6538267280e-01
12 246477994.000 2.9010090959e-01 -9.2067919169e-01 2.6113461717e-01 9.5444512317e-01 2.9825813813e-01 -8.7515658487e-03 -6.9828140147e-02 2.5177749906e-01 9.65426272165e-01
13 246478004.000 2.9010125698e-01 -9.2067802068e-01 2.6113835976e-01 9.5464575284e-01 2.9751854924e-01 -1.1584448343e-02 -6.7027958975e-02 2.5265528908e-01 9.6523186728e-01
14 246478015.000 2.9010741866e-01 -9.2067762897e-01 2.6113289557e-01 9.5485431773e-01 2.9671124421e-01 -1.46836895140e-02 -6.3960280392e-02 2.5360430020e-01 9.6515114242e-01
15 246478025.000 2.9010700359e-01 -9.2067659376e-01 2.6113706539e-01 9.5503606263e-01 2.9597344567e-01 -1.7488240001e-02 -6.1188606379e-02 2.5446871942e-01 9.6514331852e-01
16 246478036.000 2.9010673277e-01 -9.2067644259e-01 2.611377682e-01 9.552584016e-01 2.9516270499e-01 -2.055655093e-02 -5.8152195929e-02 2.5540914003e-01 9.6508263548e-01
17 246478047.000 2.9010975266e-01 -9.2067865550e-01 2.6113394907e-01 9.5540631801e-01 2.9434867900e-01 -2.3614035438e-02 -5.5123917944e-02 2.5633897184e-01 9.6501386069e-01
18 246478057.000 2.9010521145e-01 -9.2067833142e-01 2.6113287104e-01 9.5555948765e-01 2.9361641541e-01 -2.6371892380e-02 -5.2393051619e-02 2.5717855791e-01 9.6494256694e-01
19 246478068.000 2.9009932440e-01 -9.2068061857e-01 2.6113068078e-01 9.5572079416e-01 2.9280314608e-01 -2.9396531529e-02 -4.9395086367e-02 2.5803994282e-01 9.6485974647e-01
20 246478078.000 2.9009759710e-01 -9.2068225265e-01 2.6112750490e-01 9.5585756950e-01 2.9200861139e-01 -3.2128244456e-02 -4.6687251053e-02 2.5921727539e-01 9.6476950724e-01
```

Figure 2: Excerpt of ROTMSO data product. Each row of the data file provides the complete rotation matrix between FIPS sensor frame and the MSO frame.

The process of converting a given ion measurement to a velocity vector in MSO is illustrated below in pseudocode

```

-----
//Define variables
theta= zenith angle corresponding to the MCP pixel of the ion measurement
phi= azimuth angle corresponding to the MCP pixel of the ion measurement
v_unit_FIPS_vec= - [sin(theta)*cos(phi) , sin(theta)*sin(phi) , cos(theta) ]
//need the minus sign in front as the zenith and azimuth angles correspond to the look direction of the
//instrument which is the negative of the velocity direction for an observed ion.

//rotate to MSO coordinate frame
met_pha= time stamp for pha (mission elapsed time, see Figure 1)
rot_met_arr=array of MET times found in ROTMSO data product (see Figure 2)
row_index= where ( met_pha == rot_met_arr ) //find correct rotation matrix from ROTMSO
Q= 3x3 rotation matrix, formed from the 3 matrix rows found in the ROTMSO data product at the row
corresponding to met_pha
v_MSO_vec=Q * v_unit_FIPS_vec // rotate vector, via matrix multiplication, yields velocity vector in
MSO

// calculate the polar angles of this MSO velocity vector (MSO frame)
theta_MSO=acos(v_MSO_vec_z / norm(v_MSO_vec) ) // theta = acos (z/r)
phi_MSO=atan(v_MSO_vec_y / v_MSO_vec_x ) // phi = atan(y/x)
-----

```

Notice that not all the protons have valid azimuth and zenith angles; this is an instrument detection issue. Only protons with valid angles can be used for this analysis.

The process illustrated in this pseudo code is implemented in the IDL programming language as the “convert_fips_angles_to_mso_angle.pro” function included with the tutorial.

Accumulating PHAs into a flux map

With a velocity vector in MSO calculated for each PHA, it is now possible to accumulate those PHAs into the desired flux map (AFM). The key part of the construction of this flux map is not only properly accumulating the ions that FIPS observes, but also accumulating all the possible directions in which FIPS *could* have seen an ion. There is an important difference between FIPS not looking in a given direction and FIPS looking in a given direction and not observing ions. This methodology is of the utmost importance in the construction of this flux map. This process will again be described in pseudo code again, but with a little more involved preface describing the variables and flow structure.

This pseudo code will loop over each time step covered in our sample data period. In particular, the interval covered is from **2012-147T16:00:00** to **2012-147T16:19:59** (see *FIPP_P2012147CDR_V3.tab*). Here the time format is “yyyy-doyThh:mm:ss”, where DOY is the day of year. This time range corresponds to an observation of a relatively isotropic population in Mercury’s foreshock.

Each time step requires the proper mapping of PHAs from the FIPS frame to the MSO frame, along with a computation of which portion of the MSO angular space is observed by FIPS at that time step. The mapping of the PHAs velocity direction to the MSO frame is done using the zenith and azimuth angles given in the PHA list in FIPP_P*.tab. The calculation of the MSO angular space that is visible to FIPS is done by mapping the entire FIPS field of view, which consists of the look direction of each MCP pixel, to the MSO angular coordinates.

Normalizations (or appropriately weighted averages) must be performed to account for multiple MCP pixels falling in a given MSO angular bin and to account for the intermittent observation of some MSO directions as FIPS rotates through a time interval. If FIPS did not look in a given direction for some portion of the overall time period, then only the fluxes observed when FIPS was looking in that direction are included in the computation of the average flux over the interval.

The nature of this normalization lends itself to the three loop structure seen below. At each time step a loop over all PHAs in that time step determines the observed flux. A separate loop over the look direction of all MCP pixels determines the spatial averaging required to map the flux from the FIPS frame to the MSO frame. And a loop over all time steps encompasses the previous two loops and averages over the multiple time scans of FIPS over the measurement period.

Recall that FIPS observes ions over a range of E/q values. In computing the AFM, we must integrate the E/q dimension to compute the flux in each angular bin in the MSO frame. Furthermore, the flux that is computed in this tutorial is actually the average flux per solid angle. This is done so that an isotropic distribution still shows up as a uniform in an AFM that has angular bins of unequal solid angle.

The process illustrated in the pseudo code below is also shown in IDL code in the “flux_map_calculation.pro” file, which reads data from an IDL save file named “FIPS_tutorial_data.sav” using the routine “read_in_tutorial_data.pro”. The details of creating the IDL save file are described in the read routine itself.

//Define variables

PHA= structured array with each PHA as an element and each field as a property associated with a given PHA. In particular, PHA[i].met would represent the met of the ith PHA, and PHA.met would represent a list of the mets for all PHAs.

PHA.met = mission elapsed time when PHA was recorded

PHA.utc_time = UTC formatted time string

PHA.scantype= indicates scan mode that FIPS was in

PHA.step = E/q step number of the scan that PHA was observed on

PHA.epq = energy per charge of current PHA

PHA.X = X index of MCP pixel that recorded the PHA

PHA.Y = Y index of MCP pixel that recorded the PHA

PHA.zenith = zenith angle of PHA in FIPS frame

PHA.azimuth = azimuth angle of PHA in FIPS frame

PHA.mpq = mass per charge of PHA (amu/e), used to determine whether it is a proton or not.
PHA.weight = phase space density associated with PHA

MCP = structured array with each MCP pixel as an element and each field a property of the MCP pixel
MCP.X = the X index assigned to a given pixel
MCP.Y = the Y index assigned to a given pixel
MCP.polar = polar angle of given MCP pixel (same as zenith angle)
MCP.azimuth = azimuth angle of given MCP pixel
MCP.SA = solid angle of given MCP pixel
MCP.qual = quality of pixel, 0 = in field of view, 1 = obstructed by S/C and therefore not a valid pixel

EQ_data = structured array where each element corresponds to a row of the table given in
"FIPA_E2011047CDR_V1.txt".

EQ_data.step = step number of FIPS scan
EQ_data.epq = energy per charge of step
EQ_data.dshv = deflection system high voltage index
EQ_data.accum = accumulation time for step
EQ_data.H_thresh = time of flight threshold for proton identification (not used in this tutorial)

ROTMSO = structured array with each time step as an element and each field a property of the rotation
from FIPS frame to MSO frame at the current time step
ROTMSO.met = mission elapsed time for current time step
ROTMSO.Q = rotation matrix from FIPS Cartesian to MSO Cartesian at the current time step

RATES = structured array of timing data from the rate measurements of the FIPS sensor
RATES.MET = mission elapsed time for each scan during time interval
RATES.utc_time = UTC time string for the current scan MET time
RATES.qual = the quality of the current scan period (see format file for details)
RATES.scantype = indicates the mode of FIPS for the scan determines the E/Q stepping and
accumulation times (see format file for details)

```
//Define unit conversion  
amu2kg=1.661E-27 // kg/amu  
e2C= 1.602E-19 // Coulombs/electron charge
```

```
//Define proton info  
m=1.0 //amu  
q=1.0 // electron charge
```

```
// only select PHAs that are recorded as protons  
H_ind = find_where (PHA.mpq == 1.0) //find H+ based on mass per charge value (amu/e)  
H_PHA= PHA[H_ind] //only take subset of all PHAs that are proton
```

```
//find max and min pha times from pha list
```



```

min_pha_time=min(H_PHA.met, min_ind)
max_pha_time=max(H_PHA.met, max_ind)

//use ROTMSO data product to find list of scan times that cover this time range
time_arr_ind=find_where( (ROTMSO.met >= min_pha_time) & (ROTMSO.met <= max_pha_time) )
time_arr=rotmso[time_arr_ind].met // array of unique met time steps

N= n_elements( time_arr) //total number of unique met time steps

//preallocate total flux and viewing time arrays
flux_tot_arr=zeros(18,36) //set 10 deg wide bins in MSO coordinates (in both zenith and azimuth)
viewtime_tot_arr=flux_tot //preallocate viewtime array that is same size as "flux_tot"
MSO_zenith_low_bin_edges=[0,10,20,..., 170] //define lower bin edges
MSO_azimuth_low_bin_edges=[0,10,20,30,... 350]

for n=0:N-1 //loop over time steps
    t_n = time_arr[n]
    flux_arr_SA_wtd=zeros(18,36) //preallocate zero-filled solid angle weighted flux array for
    current time step

    // find index of all PHAs that are observed in current time step
    time_ind = find_where( H_PHA.met == t_n)

    //record the scan type for each time interval for later use (each PHA with the same met time will
    //have the same scan type). One scan type per time step
    scan_type=H_PHA[time_ind[0]].scantype //just take first PHA of time step

    //This won't work if there are no PHAs observed in a given time step. That step will still need to
    //be accounted for in the normalization, but will not get looped over in the current scheme for
    //looping over time. Need to look at RATES data instead to find scan type (this is illustrated in
    //the actual IDL code, but is an unnecessary complication for this pseudocode)

    M= n_elements(time_ind) // total number of PHAs in current time step

    //Look up E/q table for current time step and calculate "delta_v" terms. These are used in the
    //Riemann integration scheme that computes the flux

    valid_steps=find_where(EQ_data.accum > 1 ) // some invalid steps in scantype=0 have
    accumulation //times of 1 ms

    //central velocity value for each bin
    v_table=sqrt(2*EQ_data[valid_steps].epq*1000.0/ (m*amu2kg/(q*e2C) ) )
    //unit conversion for kV to V, and amu to kg, electron charge to Coulombs

```

```

delta_epq=0.06 // relative full width of each E/q bin
v_low= sqrt(2*EQ_data[valid_steps].epq*(1-delta_epq/2)*1000.0/
          (m*amu2kg/(q*e2C) ) )
v_high= sqrt(2*EQ_data[valid_steps].epq*(1+delta_epq/2)*1000.0/
          (m*amu2kg/(q*e2C) ) )

//define high edge of bins to be equal to low edge of next higher epq bin (fill gaps
//between bins as they are not adjacent and we want to integrate over gap)
v_high_new=v_high //preallocate
v_high_new[1:end]=v_low[0:end-1]

dv=v_high_new - v_low //calculate width of each bin in velocity

for m=0:M-1 //loop over number of PHAs in current time step
    // need current E/q to vel table to compute the contribution of each PHA to the flux
    v=sqrt(2*H_PHA[time_ind[m]].epq*1000.0/ ( H_PHA[time_ind[m]].mpq *amu2kg/e2C)
    //find velocity of current PHA, use v= sqrt(2 * E/q / (m/q) ) with appropriate unit
    //conversions

    //compute flux contribution from current PHA
    flux=H_PHA[time_ind[m]].weight * v^3 * dv

    //find which MSO bin the PHA falls in
    // use function based on FIPS to MSO coordinate transformation discussed in the
    //previous section
    [theta_MSO, phi_MSO] = convert_FIPS_angles_to_MSO_angles(
        H_PHA[time_ind[m]].zenith, H_PHA[time_ind[m]].azimuth)

    // find the zenith bin by rounding theta_MSO down to the nearest 10, then finding the
    //MSO bin whose left edge matches.
    MSO_zenith_ind= find_where( MSO_zenith_low_bin_edges == round(theta_MSO, 10) )

    // find azimuth bin by same method as zenith
    MSO_azimuth_ind=find_where(MSO_azimuth_low_bin_edges == round(phi_MSO,10) )

    // Look up the solid angle for the current pixel by its X, Y pixel values, which are
    //given in the FIPA_F*.tab file, which was load into the "MCP" struct array.
    pixel_ind= find_where(MCP.X == H_PHA[time_ind[m]].X &
        MCP.Y == H_PHA[time_ind[m]].Y )

    // add flux from PHA to appropriate bin, weighted by solid angle
    flux_arr_SA_wtd[MSO_zenith_ind, MSO_azimuth_ind]=
        flux_arr_SA_wtd[MSO_zenith_ind, MSO_azimuth_ind] +
        flux*MCP[pixel_ind].SA

```

```

endfor //end of loop over PHAS

//Now we need to account for how the MSO angular space was observed by FIPS during this
//time period. Loop over every MCP pixel so we can normalized each MSO angular bin by
//the total solid angle that observed it during the time step.

valid_pixel_ind=find_where(MCP.qual == 0)
L = n_elements(valid_pixel_ind)

//preallocate solid angle array for mapping of MCP pixels to MSO
SA_arr=zeros(18,36) //create array same size as flux_arr

for l=0:L-1 // loop over valid MCP pixels for current time step

    //map current MCP pixel to MSO cords
    [theta_MSO, phi_MSO]=convert_FIPS_angles_to_MSO_angles(
        MCP[valid_pixel_ind[l]].polar, MCP[valid_pixel_ind[l]].azimuth)

    // find the zenith bin by rounding theta_MSO down to the nearest 10, then finding the
    //MSO bin whose left edge matches.
    MSO_zenith_ind= find_where( MSO_zenith_low_bin_edges == round(theta_MSO, 10) )

    // find azimuth bin by same method as zenith
    MSO_azimuth_ind=find_where(MSO_azimuth_low_bin_edges == round(phi_MSO,10) )

    //add solid angle contribution from pixel to array
    SA_arr[MSO_zenith_ind, MSO_azimuth_ind]=SA_arr[MSO_zenith_ind,
        MSO_azimuth_ind] + MCP_pixel[valid_pixel_ind[l]].SA

endifor //end of loop over MCP pixels

//normalize flux array by solid angle (accounts for that weighting by solid angle we did earlier)
flux_arr= flux_arr_SA_wtd ./ SA_arr //divide element by element

//This division will yield some 0/0 errors as FIPS does not have a full 4*pi steradian field of view.
//It is permissible to overwrite these Not A Number (NaN) values with zero as it will not
//invalidate the normalization scheme. At the end we will check for bins that were never
//observed and represent those differently than bins that were observed, but saw no ions.

zero_SA_ind=find_where(SA_arr == 0) //using 1D indexing for a 2D array here
flux_arr[zero_SA_ind] = 0 // set NaN values to zero (works out to the same as not including
//them in weighted average)

```

```

//Now we have the correct average flux map for the current time step. We still need to weight
//that flux by the time duration of the current time step so we can do a weighted average in
//time over the entire time range being analyzed.

//Look up the accumulation time for the current scan time. Need to look at the FIPA_E*.txt file,
//at the "Accum" columns that have a number equal to the current scan type in their title. The
//values for the Survey (scan type 0) and Burst mode (scan type 2) are hardcoded below and will
//work for the majority of FIPS data.

//This won't work if there are no PHAs in the current scan, which can be an issue in time of low
//data volume. In attached IDL code we use the rate data to circumvent this issue.
if scantype == 0 // survey mode
    accum_time = 950 //ms
elseif scantype == 2
    accum_time = 50 //ms
else
end

flux_tot_arr = flux_tot_arr + flux_arr * accum_time

// for every angular direction in MSO that had at least one MCP pixel observe it (has a nonzero
//value in the SA_arr) increment the total viewing time for the MSO angular bin by the
//accumulation time of the current scan
nonzero_SA_ind = find_where(SA_arr > 0.0) // using 1D indexing for a 2D array here, syntax may
vary

viewtime_tot_arr[nonzero_SA_ind] = viewtime_tot_arr[nonzero_SA_ind] + accum_time

endfor // end of loop over time steps

final_flux = flux_tot_arr ./ viewtime_tot_arr //element by element division

```

Note: MSO angular bins that were never observed over the entire time period of study will yield a divide-by-zero error in this last step. Bins that have a "viewtime_tot_arr" value of zero will be given a special color on the flux map to indicate that they were never observed. This relates back to the emphasis that a region of space that was observed by FIPS, but no ions were seen is much different than a region of space that was not observed by FIPS, resulting in no ion being recorded.

With the data set collected all that remains is to plot it up in a flux map format. An example is shown after the next section, but the actual plotting routines are left to the user.

Calculation of Energy-Resolved Pitch Angle map (ERPA)

The calculation of the ERPA is very similar to that of the angular flux map, it involves the accumulation of PHAs in two dimensions. In the case of the flux map, those were the theta and phi angles of the ion velocity in MSO. For the ERPA, the two dimensions are the ion energy and its angle relative to the magnetic field. The proper normalization over time and solid angle for the ERPA distribution is almost identical to that of the flux map, but is performed over the pitch angle dimension of the data coordinates. The computation of the pitch angle will also represent an additional step in the calculation, compared to the case for the AFM. The E/q measurement is not directional and does not have to be remapped in the same way as the directional components of FIPS measurements.

Furthermore, instead of the integrated flux reported in each bin, the ERPA reports the average phase space density in each bin of E/q and pitch angle.

The process illustrated in the pseudo code below is also shown in IDL code in the “erpa_map_calculation.pro” file. It requires that data to be saved in an IDL save file called “FIPS_tutorial_data.sav” according to the details in the “read_in_tutorial_data.pro” IDL file.

//Define variables, most are the same as in the flux map tutorial

PHA= structured array with each PHA as an element and each field as a property associated with a given PHA. In particular, PHA[i].met would represent the met of the ith PHA, and PHA.met would represent a list of the mets for all PHAs.

PHA.met = mission elapsed time when PHA was recorded

PHA.utc_time = UTC formatted time string

PHA.scantype= indicates scan mode that FIPS was in

PHA.step = E/q step number of the scan that PHA was observed on

PHA.epq = energy per charge of current PHA

PHA.X = X index of MCP pixel that recorded the PHA

PHA.Y = Y index of MCP pixel that recorded the PHA

PHA.zenith = zenith angle of PHA in FIPS frame

PHA.azimuth = azimuth angle of PHA in FIPS frame

PHA.mpq = mass per charge of PHA (amu/e), used to determine whether it is a proton or not.

PHA.weight = phase space density associated with PHA

MCP = structured array with each MCP pixel as an element and each field a property of the MCP pixel

MCP.X = the X index assigned to a given pixel

MCP.Y= the Y index assigned to a given pixel

MCP.polar = polar angle of given MCP pixel (same as zenith angle)

MCP.azimuth = azimuth angle of given MCP pixel

MCP.SA = solid angle of given MCP pixel

MCP.qual = quality of pixel, 0 = in field of view, 1= obstructed by S/C and therefore not a valid pixel

EQ_data=structured array where each elements corresponds to a row of the table given in "FIPA_E2011047CDR_V1.txt".
 EQ_data_0.step = step number of FIPS scan
 EQ_data_0.epq = energy per charge of step
 EQ_data_0.dshv= deflection system high voltage index
 EQ_data_0.accum = accumulation time for step
 EQ_data_0.H_thresh= time of flight threshold for proton identification (not used in this tutorial)

//This is a new data item not in the previous flux map tutorial

MAG=structured array where each element corresponds to a row of the data file MAGMSOSCI AVG12147_01_V08.TAB. 1 second time resolution.
 MAG.met = mission elapsed time of magnetic field vector
 MAG.bx = x component of measured magnetic field vector (MSO coordinates)
 MAG.by= y component of measured magnetic field vector (MSO coordinates)
 MAG.bz=z component of measure magnetic field vector (MSO coordinates)

//Define unit conversion
 amu2kg=1.661E-27 // kg/amu
 e2C= 1.602E-19 // Coulombs/electron charge

//Define proton info
 m=1.0 //amu
 q=1.0 // electron charge

// only select PHAs that are recorded as protons
 H_ind =find_where (PHA.mpq == 1.0) //find H+ based on mass per charge value (amu/e)
 H_PHA= PHA[H_ind] //only take subset of all PHAs that are proton

//find total number of unique time steps in list of PHAs
 time_arr= unique(H_PHA.met) // array of unique met time steps
 N= n_elements(time_arr) //total number of unique met time steps

//preallocate zero-filled total phase space density (PSD) and viewing time arrays
 PSD_tot_arr=zeros(18,60) //set 10 deg wide bins in pitch angle, and 60 E/q bins
 viewtime_tot_arr=zeros(18,60) //preallocate viewtime array, only for angular dimension
 PA_low_bin_edges=[0,10,20,..., 170] //define lower bin edges

//estimate the time duration of each scan (take differences between time arr)
 //the PHA met time represents the time at the end of the data accumulation
 scan_delta_t_arr=time_arr //preallocate
 scan_delta_t_arr[1:end]=time_arr[1:end]-[time_arr[0:end-1]
 scan_delta_t_arr[0]=scan_delta_t_arr[1] //assume the first scan length is the same as the second

```

for n=0:N-1 //loop over N time steps
    t_n = time_arr[n]
    PSD_arr_SA_wtd=zeros(18,60) //preallocate solid angle weighted PSD array for current time
    //step; 18 pitch angle bins by 60 E/q bins

    // find index of all PHAs that are observed in current time step
    time_ind = find_where( H_PHA.met == t_n)

    //record the scan type for each time interval for later use (each PHA with the same met time will
    //have the same scan type). One scan type per time step
    scan_type=H_PHA[time_ind[0]].scantype //just take first PHA of time step

    //This won't work if there are no PHAs observed in a given time step. That step will still need to
    //be accounted for in the normalization, but will not get looped over in the current scheme for
    //looping over time. Need to look at RATES data instead to find scan type (this is illustrated in
    //the actual IDL code, but is an unnecessary complication for this pseudocode)

    M= n_elements(time_ind) // total number of PHAs in current time step

    //Look up E/q table for current time step and calculate "delta_v" terms. These are used in the
    //Riemann integration scheme that computes the flux

    valid_steps=find_where(EQ_data.accum > 1 ) // some invalid steps in scantype=0 have
    accumulation //times of 1 ms, invalid in scantype=2 have accumulation = 0 ms

    //central velocity value for each bin
    v_table=sqrt(2*EQ_data[valid_steps].epq*1000.0/ (m*amu2kg/(q*e2C) ) )
    //unit conversion for kV to V, and amu to kg, electron charge to Coulombs

    delta_epq=0.06 // relative full width of each E/q bin
    v_low= sqrt(2*EQ_data[valid_steps].epq*(1-delta_epq/2)*1000.0/
        (m*amu2kg/(q*e2C) ) )
    v_high= sqrt(2*EQ_data[valid_steps].epq*(1+delta_epq/2)*1000.0/
        (m*amu2kg/(q*e2C) ) )

    //define high edge of bins to be equal to low edge of next higher epq bin (fill gaps
    //between bins as they are not adjacent and we want to integrate over gap

    v_high_new=v_high //preallocate
    v_high_new[1:end]=v_low[0:end-1]
    dv=v_high_new - v_low //calculate width of each bin in velocity

    //average magnetic field vectors (1-sec resolution) over the current FIPS scan.

```

```

//define start and stop time to average magnetic field over (approximate)
t_start=t_n-scan_delta_t_arr[n] //start time
t_stop=t_n //stop time

//find mag vectors in time range
mag_ind=find_where(MAG.met < t_stop & MAG.met > t_start)
b_ave_vec=average_B_vectors(t_start, t_stop, MAG[mag_ind] )

for m=0:M-1 //loop over number of PHAs in current time step
    // need current E/q to vel table to compute the contribution of each PHA to the average
    //PSD
    v=sqrt(2*H_PHA[time_ind[m]].epq*1000.0/ ( H_PHA[time_ind[m]].mpq amu2kg/e2C)
    //find velocity of current PHA, use v= sqrt(2 * E/q / (m/q) ) with appropriate unit
    //conversions

    //use step field from PHA to find the E/Q index of the PHA from the E/Q table
    // information (same as DSHV index)

    dshv_index=EQ_data.dshv[find_where(EQ_data.step == H_PHA[time_ind[m]].step )]

    //compute flux contribution from current PHA
    flux=H_PHA[time_ind[m]].weight * v^3 * dv //use if integrating over E/Q dimension
    // to compute the pitch angle distribution

    //find which pitch angle bin the PHA falls in
    // use function based on FIPS to MSO coordinate transformation discussed in the
    //previous section
    [theta_MSO, phi_MSO] = convert_FIPS_angles_to_MSO_angles(
        H_PHA[time_ind[m]].zenith, H_PHA[time_ind[m]].azimuth)

    //find unit velocity vector for ion in MSO
    v_MSO_vec= - [sin(theta_MSO)*cos(phi_MSO) ,
        sin(theta_MSO)*sin(phi_MSO) , cos(theta_MSO) ]

    // find angle between current PHA and magnetic field (pitch angle)
    PA=acos( dot(v_MSO_vec, b_ave_vec) / (norm(v_MSO_vec) * norm(b_ave_vec) )
        )*180/pi //degrees

    // find pitch angle bin by rounding PA down to the nearest 10, then finding PA bin
    // whose left edges matches
    PA_bin_ind=find_where(PA_low_bin_edges == round (PA, 10) )

    // Look up the solid angle for the current pixel by its X, Y values
    //Need to use the FIPA_F*.tab file

```



```

pixel_ind=find_where(MCP.X == H_PHA[time_ind[m]].X &
                    MCP.Y == H_PHA[time_ind[m]].Y )

// add PSD from PHA to appropriate bin, weighted by solid angle
PSD_arr_SA_wtd[PA_bin_ind, dshv_index]=
    PSD_arr_SA_wtd[PA_bin_ind, dshv_index] +
    H_PHA[time_ind[m]].weight*MCP[pixel_ind].SA

endfor //end of loop over PHAs

//Now we need to account for how the pitch angle space was observed by FIPS during this
//time period. Loop over every MCP pixel so we can normalized each pitch angle bin by
//the total solid angle that observed it during the time step.

valid_pixel_ind=find_where(MCP.qual == 0)
L = n_elements(valid_pixel_ind)

//preallocate solid angle array for mapping of MCP pixels to pitch angle
SA_arr=zeros(18,60)

for l=0:L-1 // loop over valid MCP pixels for current time step

    //map current MCP pixel to MSO cords
    [theta_MSO, phi_MSO]=convert_FIPS_angles_to_MSO_angles(
        MCP[valid_pixel_ind[l]].polar, MCP[valid_pixel_ind[l]].azimuth)

    //find unit velocity vector for ion in MSO
    v_MSO_vec= - [sin(theta_MSO)*cos(phi_MSO) ,
        sin(theta_MSO)*sin(phi_MSO) , cos(theta_MSO) ]

    // find angle between current PHA and magnetic field (pitch angle)
    PA=acos( dot(v_MSO_vec, b_ave_vec) / (norm(v_MSO_vec) * norm(b_ave_vec) )
        )*180/pi //degrees

    // find pitch angle bin by rounding PA down to the nearest 10, then finding PA bin
    // whose left edge matches
    PA_bin_ind=find_where(PA_low_bin_edges == round (PA, 10) )

    //add solid angle contribution from pixel to array. Add to every E/q bin at a given
    // pitch angle
    SA_arr[PA_bin_ind,*]=SA_arr[PA_bin_ind,*] + MCP_pixel[valid_pixel_ind[l]].SA

endfor //end of loop over MCP pixels

```

```

//normalized PSD array by solid angle (accounts for that weighting by solid angle we did earlier)
PSD_arr= PSD_arr_SA_wtd / SA_arr //element by element division.

//This division will yield some 0/0 errors as FIPS does not have a full 4*pi steradian field of view.
//It is permissible to overwrite these Not A Number (NaN) values with zero as it will not
//invalidate the normalization scheme. At the end we will check for bins that were never
//observed and represent those differently than bins that were observed, but saw no ions.

zero_SA_ind=find_where(SA_arr == 0 ) // 1D indexing of 2D array
PSD_arr[zero_SA_ind] = 0 // set NaN values to zero (works out to the same as not including
//them in weighted average). Corresponds to setting whole rows of PSD_arr_SA_wtd to zero

//Now we have the correct average PSD map for the current time step. We still need to weight
//that PSD by the time duration of the current time step so we can do a weighted average in
//time over the entire time range being analyzed.

//Look up the time duration for the current scan time. Need to look at the FIPA_E*.txt file, at
//the "Accum" columns that have a number equal to the current scan type in their title
if scantype == 0 // survey mode
    accum_time = 950 //ms
elseif scantype == 2
    accum_time = 50 //ms
else
end

PSD_tot_arr = PSD_tot_arr + PSD_arr * accum_time

// for every pitch angle that had at least one MCP pixel observe it (has a nonzero value in the
//SA_arr), increment the total viewing time for the that pitch angle bin by the
//accumulation time of the current scan. Just a weighted average scheme over time.
nonzero_SA_ind = find_where(SA_arr > 0.0 )

viewtime_tot_arr[nonzero_SA_ind]= viewtime_tot_arr[nonzero_SA_ind] + accum_time

endfor // end of loop over time steps

final_PSD= PSD_tot_arr/ viewtime_tot_arr //element by element division.
-----

```

Note: Pitch angle bins that were never observed over the entire time period of study will yield a divide-by-zero error in this last step. Bins that have a "viewtime_tot_arr" value of zero will be given a special color on the pitch angle map to indicate that they were never observed. This relates back to the

emphasis that a region of space that was observed by FIPS, but no ions were seen is much different than a region of space that was not observed by FIPS, resulting in no ion being recorded.

With the data set collected all that remains is to plot it up in an Energy Resolved Pitch Angle format. If one wanted to compute the pitch angle distribution (not separated by E/q), one would replace all calculations of the PSD with calculation of the flux (as in the flux map pseudo code). This essentially integrates over the E/q dimension in the PSD_arr_SA_wtd array. All operations done to the PSD_arr_SA_wtd would be replaced by operation to the flux_arr_SA_wtd (as seen in the flux map pseudo code). There would only be 18 x 1 arrays involved in the calculation in this case.

A successful run of this code over the prescribed time period will allow the production of the data products seen below. From these plots we find the presence of a relatively isotropic population superposed with an anti-sunward solar wind in the AFM. The bulk solar wind flow direction would show up roughly in the center of the AFM, but is not observed by FIPS due to the heat shield of the MESSENGER spacecraft. All "unobserved" flow directions are shaded white in the plot. This includes directions that were not in the FIPS field of view, as well as directions blocked by obstacles such as the heat shield. The directions shown on the AFM indicate the vector direction of the observation, i.e. "Dusk" and "Dawn" indicates duskward and dawnward ion flow velocities, respectively, and "Sun" indicates flow velocity toward the Sun. The ERPA indicates that the observed distribution is fairly uniform in pitch angle, except for an enhancement around 40-50 degrees corresponding to the solar wind.

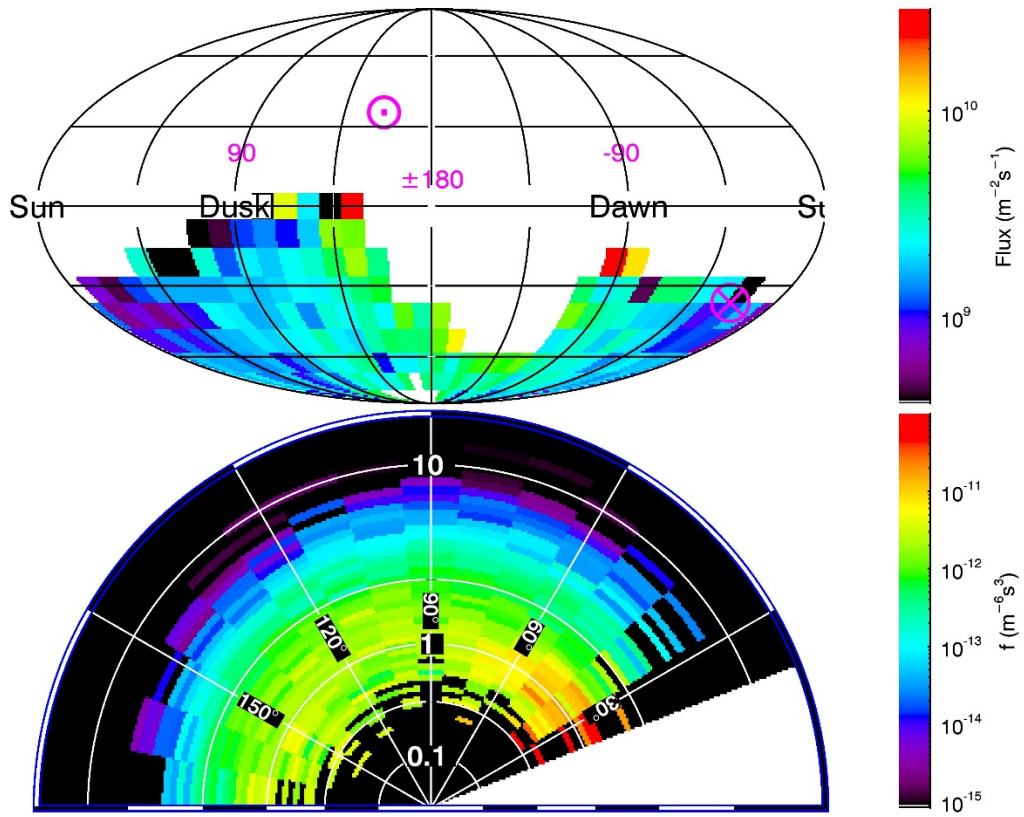


Figure 3: Angular flux map (top) and Energy Resolved Pitch Angle (bottom) plots for the time period 2012-05-26 16:00:00 to 2012-05-26 16:19:59. White areas indicates regions that are unobserved with the FIPS field of view. Also shown on the figure is the average vector magnetic field direction, the magenta circle with a dot is the +B direction, while the magenta circle with the X is the -B direction.

References

DiBraccio, G.A., Slavin, J.A., Raines, J.M., Gershman, D.J., Tracy, P.J., Boardsen, S.A., Zurbuchen, T.H., Anderson, B.J., Korth, H., McNutt Jr., R.L., Solomon, S.C. (2015), First observations of Mercury's plasma mantle by MESSENGER, *Geophys. Res. Lett.*, **42**, 9666-9675, doi:10.1002/2015GL065805.

Gershman, D.J., J.M. Raines, J.A. Slavin, T.H. Zurbuchen, T. Sunderg, S.A. Boardsen, B.J. Anderson, H. Korth, and S.C. Solomon (2015), MESSENGER observations of multiscale Kelvin-Helmholtz vortices at Mercury, *J. Geophys. Res. Space Physics*, **120**, 4354-4368, doi:10.1002/2014JA020903

Raines, J.M., D.J. Gershman, J.A. Slavin, T.H. Zurbuchen, H. Korth, B.J. Anderson, and S.C. Solomon (2014), Structure and dynamics of Mercury's magnetospheric cusp: MESSENGER measurements of protons and planetary ions, *J. Geophys. Res. Space Physics*, **119**, 6587-6602, doi:10.1002/2014JA020120.