

Interpreting FIPS Density, Temperature, and Pressure

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

Introduction

The purpose of this tutorial is to enable the user to interpret the density, temperature, and pressure (NTP) moments from observations by the Fast Imaging Plasma Spectrometer (FIPS) on MESSENGER. These parameters are useful for a variety of investigations of Mercury's plasma environment. For example, these moments were used to determine the average characteristics of the plasma sheet (Raines et al., 2011; Gershman et al., 2014) and to distinguish the plasma sheet from the magnetotail lobes (Sun et al., 2016). The derived moments were also used to assess the pressure balance across the dayside magnetopause to understand the nature of dayside magnetic reconnection at Mercury (Gershman et al., 2013).

The NTP moments are archived at the NASA Planetary Data System (PDS) for portions of each of MESSENGER's passes through the magnetosphere. Due to the FIPS sensor's limited field of view and variable orientation with respect to the plasma flow, the calculated quantities are valid only under certain conditions, as explained in Raines et al. (2011), Gershman et al. (2013), and in the [DDR SIS](#). To summarize, the primary conditions are that:

1. There are sufficient counts from the sensor to produce a well-defined spectrum;
2. The plasma is subsonic, such that $v_{bulk}/v_{th} < 0.5$ where v_{bulk} is the plasma bulk flow velocity and v_{th} is the plasma thermal velocity; and
3. The plasma is sufficiently isotropic, such that $0.5 < T_{\perp}/T_{\parallel} < 5$ where T_{\perp} is the plasma temperature perpendicular to the magnetic field and T_{\parallel} is the plasma temperature along the magnetic field.

Under these conditions, the NTP moments can be computed from FIPS observations despite the limitations of the FIPS sensor's field of view. While these assumptions typically hold inside Mercury's magnetosphere and close to the subsolar region of the magnetosheath, users of the NTP calculations are strongly encouraged to evaluate the validity of these assumptions, especially for studies involving features near the time scale of the data averaging used in the NTP calculations and/or inside the magnetosheath. This tutorial provides instructions on how to test the validity of the NTP calculations by comparison with energy spectrograms.

In the subsequent sections, all IDL command line calls will be in green. References to code within specific programs will be in blue (the name of the program will be *italicized*). All command line calls must be executed in the main tutorial directory.

Tutorial Structure

This tutorial is comprised of two examples to walk the user through the steps of downloading and loading FIPS data products from NASA PDS, comparing NTP calculations against FIPS observations, and incorporating deviations from the required conditions, such as a significant bulk flow, into the comparison. Routines in the IDL programming language and sample data required to reproduce the results are included with this tutorial. Both examples also use a software model of the FIPS sensor, which is described in a section prior to the examples. A brief description of each example is given below.

Example 1: Valid NTP fit

This example steps the user through loading the NASA PDS data and comparing NTP moments with FIPS observations to assess whether the moment calculations are valid. This example also uses the FIPS model to assess the robustness of the NTP moments.

Example 2: NTP misfit

This example steps the user through incorporating deviations from the conditions assumed for calculating the NTP moments into the comparison between the NTP calculation and FIPS observations to better interpret FIPS observations.

Each example uses routines in the IDL programming language. To follow the commands in the examples directly, the IDL code must be executed from the local directory of this tutorial. Once per IDL session, SPICE kernels must be initialized with the commands

```
@spice_idl_startup.idl  
cspice_furnsh, kdir+'msgr_1306_v02.bc'
```

The former command initializes all SPICE kernels that apply throughout the MESSENGER mission (reference frames, planetary barycenters, FIPS field of view, etc.), and the latter command initializes the monthly high-resolution kernel for MESSENGER position/pointing.

The file structure of the local directory must be as follows (files in italics, directories not in italics):

- FIPS_NTP_Tutorial (the local directory)
 - └ *Tutorial_Interpreting_FIPS_NTP.pdf* (this file)
 - └ *code_idl_startup.idl*
 - └ *spice_idl_startup.idl*
 - └ *ntp_tutorial_example1.idl*
 - └ *ntp_tutorial_example2.idl*
 - └ *fips_model_example.idl*
 - └ *mfips_e2e_model_test1.idl*
 - └ *mfips_e2e_model_test2.idl*
 - └ data
 - └ ... (NASA PDS FIPS data files used in the examples)
 - └ spice
 - └ ... (SPICE kernels used in FIPS timing and orientation)
 - └ code
 - └ ... (IDL routines developed for use in this tutorial)
 - └ mfips_e2e_model

- L ... (FIPS end-to-end model for generating hypothetical FIPS observations)

Acquiring the data for this tutorial

The FIPS observations are available at the PDS, and, for this tutorial, we will retrieve from the archive all data files required to interpret sample NTP calculations from the FIPS data acquired on 3 June 2013.

- 1) Go to the NASA PDS Planetary Plasma Interactions node. (<https://pds-ppi.igpp.ucla.edu/>)
- 2) Under “Quick Search” (left hand side of the webpage), select “Mercury”. Under “Instrument”, select “Energetic Particle and Plasma Spectrometer(EPPS)”, which the FIPS sensor is a part of.
- 3) For this tutorial the user will need several data products:
 - a) The raw proton counts from the FIPS sensor, which are located in the data set [MESS-E/V/H/SW-EPPS-2-FIPS-RAWDATA-V2.0](#). The raw counts recorded by FIPS are located under “DATA” => “FIPS_SCAN”. For the examples in this tutorial, we use observations from 3 June 2013, found under “2013” => “152_181_JUN” => [FIPS R2013154EDR V1](#). The data (.DAT) file contains the raw FIPS counts, and the format (.FMT) file describes the structure of the data file. We have included the data file with this tutorial, although the user is welcome to download it independently.
 - b) The FIPS proton flux observations are contained in the data set [MESS-E/V/H/SW-EPPS-3-FIPS-CDR-V1.0](#). These measurements are located under “DATA” => “FIPS_SCAN”. For the examples in this tutorial, we use observations found under “2013” => “152_191_JUN” => [FIPS R2013154CDR V3](#).
 - c) The derived NTP calculations are included in the data set [MESS-E/V/H/SW-EPPS-3-FIPS-DDR-V2.0](#), and are located under “DATA” => “FIPS_NTP”. For the examples in this tutorial, we use calculations from “2013” => “152_191_JUN” => [FIPS NTP 2013154 DDR V02](#).
- 4) Both the proton counts and flux are organized by FIPS energy channel. The energy per charge (E/Q) corresponding to each channel depends on the FIPS scan type and mission phase. To organize the counts/flux by the corresponding energy, we must retrieve the appropriate FIPS E/Q stepping table found at [MESS-E/V/H/SW-EPPS-3-FIPS-CDR-V1.0](#) => “CALIB”. The *FIPS_EQ.FMT* file within the “CALIB” directory describes the columns of each stepping table. For the examples here, we make use of the “FIPA_E2012045CDR_V1” table.
- 5) To model how incorporating a deviation from the required conditions to calculate the NTP moments (such as a significant bulk flow) would affect FIPS observations, we require specific information of FIPS position and pointing. This information is provided by the NASA NAIF SPICE kernels. To access the required pointing information, this tutorial requires the installation of ICY SPICE (the IDL-specific version of SPICE) and the downloading of MESSENGER kernels.
 - a) To install SPICE, follow the [“Toolkit”](#) link on the [NASA NAIF SPICE homepage](#). Following the instructions on that webpage and the [“Tutorials”](#) page, the user should download and install the ICY SPICE toolkit.

- b) To download the appropriate kernels, follow the “Data” link on the NASA NAIF SPICE homepage, selecting “PDS Archived SPICE Data Sets” and “[PDS SPICE Archives](#)”. From the PDS SPICE Archives, select the MESSENGER “[link](#)” => “[data/](#)”. Under the following directories, download the kernels:

“fk/”: *msgr_dyn_v600.tf*
msgr_v220.tf
“ik/”: *msgr_epps_v100.ti*
“lsk/”: *naif0011.tls*
“pck/”: *pck00010_msgr_v10.tpc*
“sclk/”: *messenger_2548.tsc*
“spk/”: *de405.bsp*
msgr_de405_de423s.bsp
msgr_040803_150430_150430_od431sc_2.bsp

In addition to these general kernels, we also need the specific pointing kernel for 3 June 2013. In the “ck/” directory, download *msgr_1306_v02.bc*.

Summary of FIPS data products required for this tutorial:

- EDR (raw data): FIPS counts at each time scan
 - *FIPS_R2013154EDR_V1.dat*
- CDR (calibrated data): FIPS counting rates at each time scan
 - *FIPS_R2013154CDR_V3.tab*
- DDR (derived data products): FIPS calculated NTP during select portions of each MESSENGER orbit
 - *FIPS_NTP_2013154_DDR_V02.tab*
- Miscellaneous:
 - *FIPA_E2012045CDR_V1.txt* – details of the E/Q steps of the FIPS sensor

Summary of SPICE kernels required for this tutorial:

- Reference frame: MESSENGER dynamics (MSO, etc.)
 - *msgr_dyn_v600.tf*
- Reference frame: MESSENGER orbiter and instruments
 - *msgr_v220.tf*
- Instrument field of view:
 - *msgr_epps_v100.ti*
- Leapseconds:
 - *naif0011.tls*
- Planetary constants:
 - *pck00010_msgr_v10.tpc*
- Spacecraft clock coefficients:
 - *messenger_2548.tsc*
- Planetary barycenters:
 - *de405.bsp*
 - *msgr_de405_de423s.bsp*
- Whole-mission reference spacecraft trajectory:

- *msg_040803_150430_150430_od431sc_2.bsp*
- High-resolution pointing for specific months:
 - *msg_1306_v02.bc*

FIPS Sensor Model

In order to test the robustness of the NTP calculations or to diagnose which NTP assumptions may have been violated in determining the moments, we turn to a model of the FIPS sensor response. The model takes user-specified plasma moments and returns the phase space density (PSD) that FIPS would observe from the corresponding Maxwellian distribution. We can therefore use the model to generate distributions that incorporate deviations from the NTP assumptions and compare the modeled FIPS response with the actual observations. This section walks the user through the structure, input, procedure, and output of the sensor model as well as a simple example to develop a physical understanding of the sensor.

Program Structure

The FIPS model involves multiple modules to operate. Briefly, the core programs include:

- *run_input.idl*: this example script compiles the programs used in the model and runs an input generator (Example 1 and Example 2 use the commands within this script, but not the script itself. We recommend the user copy this file to their working directory to adapt for their personal use of the FIPS model);
- *make_model_input(...).pro*: these programs generate the user-specified plasma distributions to run through the sensor model and saves them in a .DAT file; it also sets basic parameters of the model (such as model resolution);
- *run_model.idl*: this example script calls the model runs on the generated distributions (Example 1 and Example 2 use the commands within this script, but not the script itself; see note above);
- *run_model.pro*: this program loads the distributions from the .DAT file, and for each distribution, runs the model. After running each distribution through the model, this program saves the output and generates relevant plots;
- *model5.pro*: this program is the actual FIPS model. Using specific pointing information from SPICE, the program loads in obstructions to FIPS field of view and samples across its field of view to build a “modeled” phase space density.

For more details about any given program, see the procedural description below and the comments within the program file.

Input

The primary input to the FIPS model are user-specified plasma parameters, from which the model generates plasma distributions to run through the sensor. The input generator, *make_model_input(...).pro*, defines the plasma distributions by:

- (1) proton density n (cm^{-3}),
- (2) thermal velocity v_{th} (km/s),
- (3) bulk velocity vector \vec{v}_{bulk} (km/s), and
- (4) the MET, to specify FIPS pointing.

The model uses these parameters to generate a drifting Maxwellian distribution whereby the bulk velocity vector is the drift velocity. This velocity can either be specified in MSO coordinates, which is most useful when trying to analyze FIPS observations scientifically, or in FIPS spherical coordinates, which is most useful to develop a physical understanding of the sensor. The Mercury Solar Orbiter (MSO) coordinate system is a right-handed coordinate system with its origin at Mercury's center; X_{MSO} points sunward, Y_{MSO} points duskward, and Z_{MSO} is directed northward (i.e., parallel to the north celestial pole). FIPS spherical coordinates are a spherical coordinate system with the sensor at the origin; the zenith direction ($\theta = 0^\circ$) is aligned with FIPS boresight; the azimuthal direction (ϕ) describes the angle around the boresight vector; and the radius corresponds to the magnitude of the \vec{v}_{bulk} vector.

NOTE: When specifying the bulk flow in the *make_model_input(...).pro* files, the convention is that the vector in MSO coordinates specifies the flow direction (i.e., parallel to the flow) while the vector in FIPS spherical specifies the flow origin (i.e., antiparallel to the flow). This convention is chosen based off the usefulness of MSO and FIPS spherical for generating science and intuition, respectively.

Regardless of the input coordinate system, *make_model_input(...).pro* can accept ranges of each parameter or single values of each. If ranges of density, etc. are specified, the program constructs all possible combinations of plasma distributions to run through the sensor model. The *make_model_input(...).pro* programs are examples of input generators for the sensor model; users are welcome to define their own input generators. Several variations of *make_model_input(...).pro* are included in this tutorial.

Procedure

After defining the plasma distributions in *make_model_input(...).pro*, *run_model.pro* simulates the FIPS sensor response due to a given plasma distribution using the routine *model5.pro*. The routine steps through each FIPS energy channel and each angular bin in FIPS spherical coordinates (resolution specified in *make_model_input(...).pro*). For each energy channel (corresponding to a proton with speed v) and angular bin, the program first constructs a probability distribution function $f(\vec{v})$ by assuming a drifting Maxwellian distribution:

$$f(\vec{v}) = \frac{1}{(v_{th}\sqrt{2\pi})^3} e^{-\epsilon}$$

$$\epsilon = \sum_i \frac{(v_i - v_{bulk,i})^2}{2v_{th}^2}$$

$$v_{th} = \sqrt{\frac{kT}{m_p}}$$

where k is Boltzmann's constant, m_p is the proton mass, and T is the temperature.

From the probability distribution function, the program uses the FIPS pointing provided by the SPICE kernels to determine the number of particles that the sensor is expected to observe within a given energy and angular bin, taking into account obstructions in the sensor's field of view. After stepping through each energy and angular bin, the program constructs the final distribution in units of counts and computes the simulated phase space density. This process repeats for each plasma distribution.

Output

After simulating FIPS observations of each plasma distribution, *run_model.pro* saves the model output into an IDL savefile and a PostScript plot. The IDL savefile contains an array of structures, where each array element corresponds to a single plasma distribution run through the model and each structure has the following fields:

- MET: the mission elapsed time (MET) of the pointing information used;
- UTC: the UTC time of the pointing information used;
- N_INPUT: the input density (cm^{-3}) used in the model run;
- V_BULK_INPUT_MSO: the bulk flow vector (km/s, km/s, km/s) used in the model run in MSO coordinates pointing along the direction of the bulk flow;
- V_BULK_INPUT_FIPS_SPH: the bulk flow vector (km/s, deg, deg) used in the model run in FIPS spherical coordinates pointing towards the origin of the bulk flow (i.e., antiparallel to the direction of bulk flow);
- VTH_INPUT: the thermal speed (km/s) used in the model run;
- B_INPUT: the magnetic field used in the model run (set to the null vector);
- EQTAB: the E/Q of each energy channel (keV);
- V: the speed (km/s) corresponding to each E/Q;
- CDIST: the modeled counts (s^{-1}) as a function of speed/energy;
- FDIST: the modeled PSD ($\text{s}^3 \text{m}^{-6}$) as a function of speed/energy;
- HVEL: a 2D array of counts on the FIPS MCP (used only for visualization);
- MAG: the magnetic field direction (set to the null vector).

The generated PostScript plots have the format as shown in Figure 1. The MET used for the FIPS pointing, the unit vector of the FIPS boresight (zenith direction) in MSO coordinates, and the user-specified plasma distribution parameters (n , v_{th} , \vec{v}_{bulk} in FIPS spherical coordinates) are specified in the title. The top panel is an angular distribution of counts in FIPS field of view. The plot is in FIPS spherical coordinates; rings correspond to zenith angles (15° , 35° , 55° , and 75° from the center ring outwards) and the “clock” angle around the plot is the azimuthal angle (with zero degrees to the right). The outlined blue areas mark obstructions in the field of view – the large obstruction at $\phi \sim 300^\circ$ is the sunshade and the small obstruction at $\phi \sim 200^\circ$ is a solar panel. The symbols mark the location of the Sun (“Sun”) and the origin of the bulk flow velocity (“v_in”) in FIPS field of view (FOV). If the plasma does not have a bulk flow into the sensor’s FOV, the symbol is plotted in the lower-left corner of the plot.

NOTE: throughout the mission, the solar panel within FIPS field of view moves; this movement is accurately captured in the blue outline but is not captured within calculating the obstructions in the FIPS model, explaining why the white region of low (zero) counts is not coincident with the blue outline. The difference in placement is negligible in calculating the FOV-accumulated PSD.

The bottom panels are the FOV-accumulated counts (left) and phase space density (right) as functions of energy and speed, respectively. For comparisons with the observed FIPS PSD and the NTP calculations we will use the modeled PSD in the bottom-right panel, which is stored as the FDIST structure variable in the model output savefile.

FIPS End to End Model Output
 2013-06-03T00:13:19 MET: 278705856
 FIPS pointing (MSO): [0.261, 0.825, 0.501]
 n: 29.0 V_Bulk: 9 Zenith: 35.5 Azimuth: 123.1 V_Thermal: 209

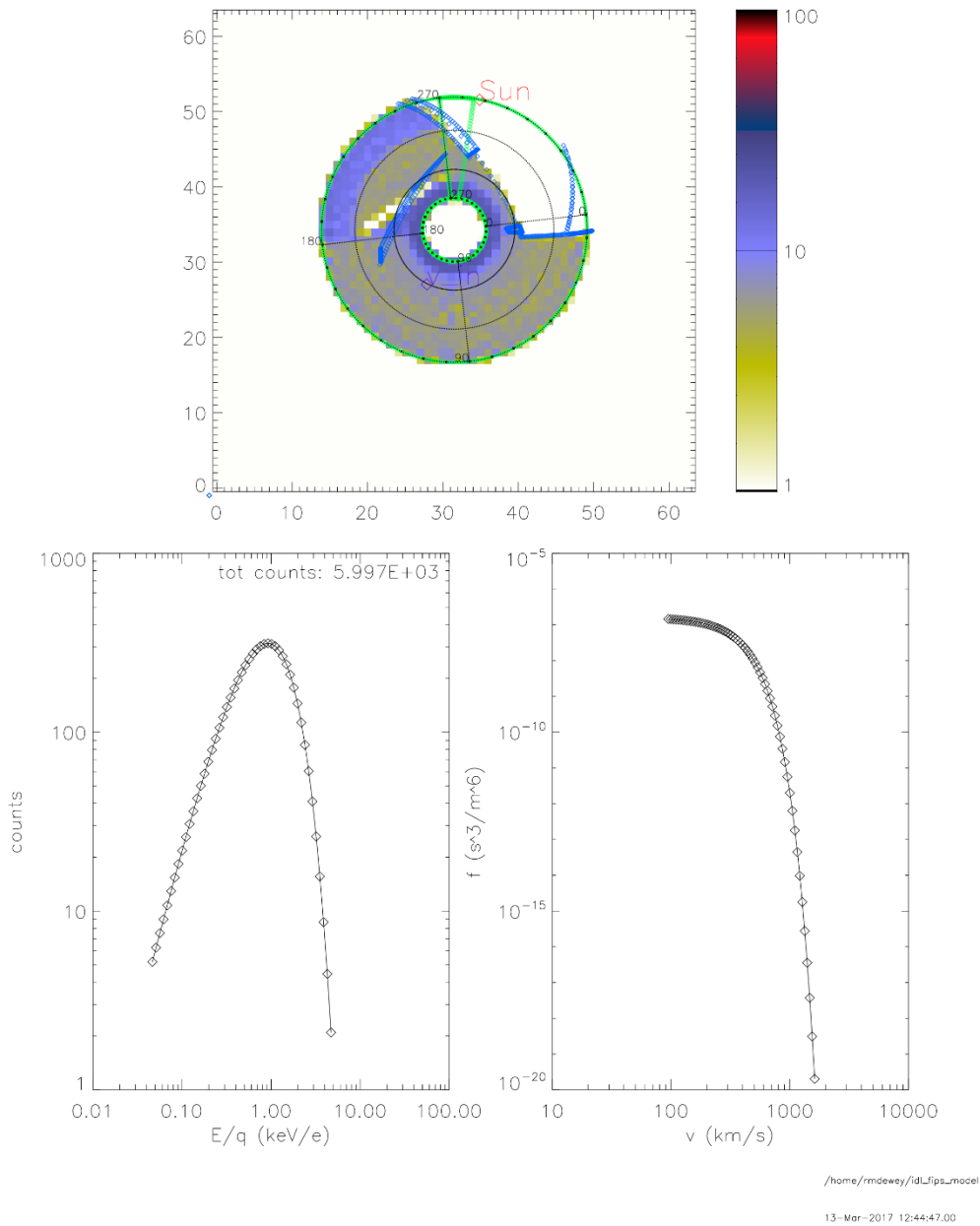


Figure 1. Output from the FIPS model of a subsonic flow at 06-03-2013T00:13:19. The input plasma distribution assumed a thermal velocity of $v_{th} = 209$ km/s, a proton density of $n = 29.0$ cm⁻³, and a bulk velocity of magnitude $v_{bulk} = 10$ km/s incident at $\theta = 36^\circ$ and $\phi = 123^\circ$ in FIPS spherical coordinates. See text for description of the individual panels.

FIPS Model Tests

To verify that the model operates correctly on the user's machine, we have included two test runs of the model and have stored the output savefile and plots in the 'data/' directory. The first test generates plasma flows of various Mach numbers (supersonic, transonic, and subsonic) and steps the bulk flow vector through FIPS field of view. On the command line, the user should execute the command

@mfips_e2e_model_test1.idl

This command will generate the plasma flows and run them through the FIPS sensor to produce the output savefile and plots. The savefile should be identical to *mfips_e2e_model_data_test1.sav* and the plots should be identical to *mfips_e2e_model_plots_test1.ps*.

The second test generates a flow in MSO coordinates from behind the FIPS sensor with varying Mach number. As described in the example below, when the flow is supersonic and the bulk flow vector is outside the field of view, no counts are recorded in the FIPS FOV. As the plasma becomes hotter and the Mach number decreases, plasma begins to enter FIPS FOV. On the command line, the user should execute the command

@mfips_e2e_model_test2.idl

The savefile should be identical to *mfips_e2e_model_data_test2.sav* and the plots should be identical to *mfips_e2e_model_plots_test2.ps*.

FIPS Model Example

For a brief example of how to use the FIPS model and for building some physical intuition of the sensor, we will simulate flows at various Mach numbers and at various points in the field of view. In *make_model_input_fips_frame_example.pro* we have specified 10 flows in FIPS spherical coordinates by combining different Mach numbers ($M = v/v_{th}$) and flow directions:

- (1) Highly supersonic ($M = 10$) with a bulk flow inside FIPS FOV
 $v_{th} = 10 \text{ km/s}$, $v_{bulk} = 100 \text{ km/s}$, $\hat{v}_{bulk} = (35^\circ, 270^\circ)$, $n = 50.0 \text{ cm}^{-3}$
- (2) Supersonic ($M = 2$) with a bulk flow inside FIPS FOV
 $v_{th} = 50 \text{ km/s}$, $v_{bulk} = 100 \text{ km/s}$, $\hat{v}_{bulk} = (35^\circ, 270^\circ)$, $n = 50.0 \text{ cm}^{-3}$
- (3) Transonic ($M = 1$) with a bulk flow inside FIPS FOV
 $v_{th} = 100 \text{ km/s}$, $v_{bulk} = 100 \text{ km/s}$, $\hat{v}_{bulk} = (35^\circ, 270^\circ)$, $n = 50.0 \text{ cm}^{-3}$
- (4) Subsonic ($M = 0.5$) with a bulk flow inside FIPS FOV
 $v_{th} = 200 \text{ km/s}$, $v_{bulk} = 100 \text{ km/s}$, $\hat{v}_{bulk} = (35^\circ, 270^\circ)$, $n = 50.0 \text{ cm}^{-3}$
- (5) Highly subsonic ($M = 0.1$) with a bulk flow inside FIPS FOV
 $v_{th} = 1000 \text{ km/s}$, $v_{bulk} = 100 \text{ km/s}$, $\hat{v}_{bulk} = (35^\circ, 270^\circ)$, $n = 50.0 \text{ cm}^{-3}$
- (6) Highly supersonic ($M = 10$) with a bulk flow outside FIPS FOV
 $v_{th} = 10 \text{ km/s}$, $v_{bulk} = 100 \text{ km/s}$, $\hat{v}_{bulk} = (175^\circ, 270^\circ)$, $n = 50.0 \text{ cm}^{-3}$
- (7) Supersonic ($M = 2$) with a bulk flow outside FIPS FOV
 $v_{th} = 50 \text{ km/s}$, $v_{bulk} = 100 \text{ km/s}$, $\hat{v}_{bulk} = (175^\circ, 270^\circ)$, $n = 50.0 \text{ cm}^{-3}$
- (8) Transonic ($M = 1$) with a bulk flow outside FIPS FOV
 $v_{th} = 100 \text{ km/s}$, $v_{bulk} = 100 \text{ km/s}$, $\hat{v}_{bulk} = (175^\circ, 270^\circ)$, $n = 50.0 \text{ cm}^{-3}$
- (9) Subsonic ($M = 0.5$) with a bulk flow outside FIPS FOV
 $v_{th} = 200 \text{ km/s}$, $v_{bulk} = 100 \text{ km/s}$, $\hat{v}_{bulk} = (175^\circ, 270^\circ)$, $n = 50.0 \text{ cm}^{-3}$
- (10) Highly subsonic ($M = 0.1$) with a bulk flow outside FIPS FOV
 $v_{th} = 1000 \text{ km/s}$, $v_{bulk} = 100 \text{ km/s}$, $\hat{v}_{bulk} = (175^\circ, 270^\circ)$, $n = 50.0 \text{ cm}^{-3}$

The spatial maps of FIPS observations for each of these flows are:

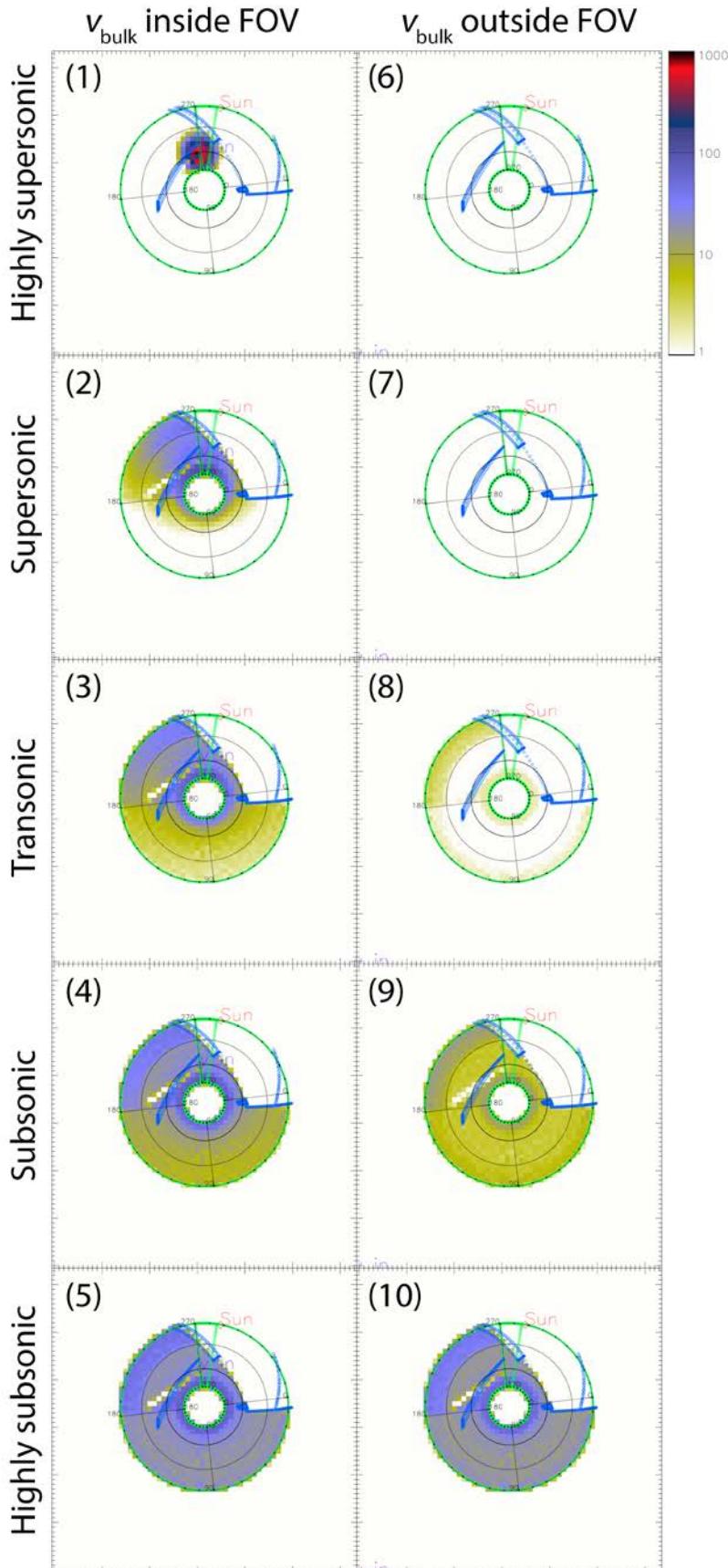


Figure 2. Spatial maps of FIPS counts within its field of view for ten hypothesized plasma flows. The left column has a bulk flow into the FIPS FOV and the right column has a bulk flow parallel to FIPS FOV. From top to bottom, the plasma becomes more subsonic. As the plasma becomes more subsonic, it becomes sufficiently isotropic to view in FIPS FOV regardless of the bulk flow direction.

As one would expect, the Mach number and bulk flow direction have significant influence on FIPS observations. FIPS can only detect supersonic flows if they are within its FOV, while strongly subsonic flows are sufficiently isotropic to be observed independent of the bulk flow direction. We can clearly see the effect of Mach number in the right column where the bulk flow is coming from behind the sensor; only for subsonic flows does FIPS observe enough plasma for accurate NTP calculations, hence the Mach number condition on the NTP calculations (see Introduction). Since the FIPS model integrates observations over the FOV to produce the modeled PSD, the modeled PSD is degenerate (non-unique) for unique pairs of v_{bulk} and v_{th} . To illustrate, both (5) and (10) flows are sufficiently subsonic ($v_{bulk}/v_{th} < 0.5$) but have bulk flows originating from nearly opposite directions. The difference in the modeled PSD is negligible between the two. Likewise, two supersonic flows can produce nearly identical modeled PSD if they are both within FIPS FOV (e.g., they are in different quadrants). In order to disentangle FOV effects when comparing with observations, one needs to compare the spatial PSD map between the model and FIPS observations. We direct the reader to the All Sky Flux Map and Energy Resolved Pitch Angle Tutorial to construct spatial maps in their own data analysis. Given the focus on evaluating moments-

based calculation of NTP, disentangling FOV effects is beyond the scope of this tutorial.

The code to generate these plots is contained within the script *fips_model_example.pro*. To execute the script from the local directory:

```
@fips_model_example.idl
```

Or to follow the line-by-line method, we first compile the FIPS model:

```
@mfips_e2e_model/compile_mfips_e2e_model.idl
```

Next, we generate the input plasma distribution parameters (this MET corresponds to 06-03-2013T00:13:16.40, the epoch time in Example 2):

```
mets = 12542396D  
ydoy = 2013154D  
make_model_input_fips_frame_example, mets=mets, ydoy=ydoy
```

Within this program, we have specified the distribution parameters using:

```
;; density  
n = calc_values(range=[50., 50.], incr=1.0)  
  
;; velocity components in FIPS spherical coordinates  
;; -----  
v_mag = calc_values(range=[100.,100.], incr=100.) ;; speed  
zen = calc_values(range=[35.,175.], incr=140.) ;; zenith angle  
azi = calc_values(range=[270.,270.], incr=10.) ;; azimuthal angle  
  
;; thermal velocity  
;; -----  
vth = [10.,50.,100.,200.,1000.]
```

The program will save the plasma distributions in *model_input.dat*. We then run the model:

```
run_model, inputfile='model_input.dat', fsw_v=7, ydoy=ydoy
```

This program will feed each distribution into the sensor model and generate the savefile and plots, as described above.

Example 1: Valid NTP fit

This example steps the user through loading the NASA PDS data (acquired above) and comparing the FIPS CDR observations with NTP calculations to assess whether the moment calculations are valid. This example also uses the FIPS sensor model to demonstrate how to test the robustness of the NTP calculation.

To determine the validity of an NTP calculation, we need to compare the phase space density (PSD) derived from the NTP moments with the PSD observed by FIPS. To do so, we must:

1. Initialize IDL routines;
2. Load the FIPS EDR counts, CDR flux, and DDR NTP calculations;
3. Derive (“recover”) the PSD from the NTP moments;
4. Convert the CDR flux to PSD and determine the uncertainty in the PSD from EDR counts;
5. Compare the CDR PSD with the “recovered” PSD.

In addition to these five steps, if we wish to determine the sensitivity of the NTP calculation to different bulk flows, we must:

6. Generate a modeled PSD by simulating bulk flows;
7. Compare the modeled PSD with FIPS observations;
8. Iterate steps 6 and 7.

The script `ntp_tutorial_example1.idl` follows these eight steps for a 60s-accumulated NTP calculation centered at 06-03-2013T08:13:12.40. During this interval, MESSENGER was located at [1.83, -0.12, 0.01] R_M in MSO coordinates (where R_M is Mercury’s radius, 2440 km); the spacecraft was positioned at a local time of 11.7 hours, a geographic latitude of 0.4° , and a distance of $1.83 R_M$ from the planet’s center. Given the local time close to noon, small latitude, and distance in the range between the typical subsolar magnetopause standoff ($\sim 1.5 R_M$) and bow shock ($\sim 2.1 R_M$) distances, we expect the spacecraft to be in the magnetosheath close to the subsolar point. Indeed, magnetic field and plasma observations shown in Figure 3 confirm that MESSENGER was located in the dayside magnetosheath.

MESSENGER: PLASMA OVERVIEW

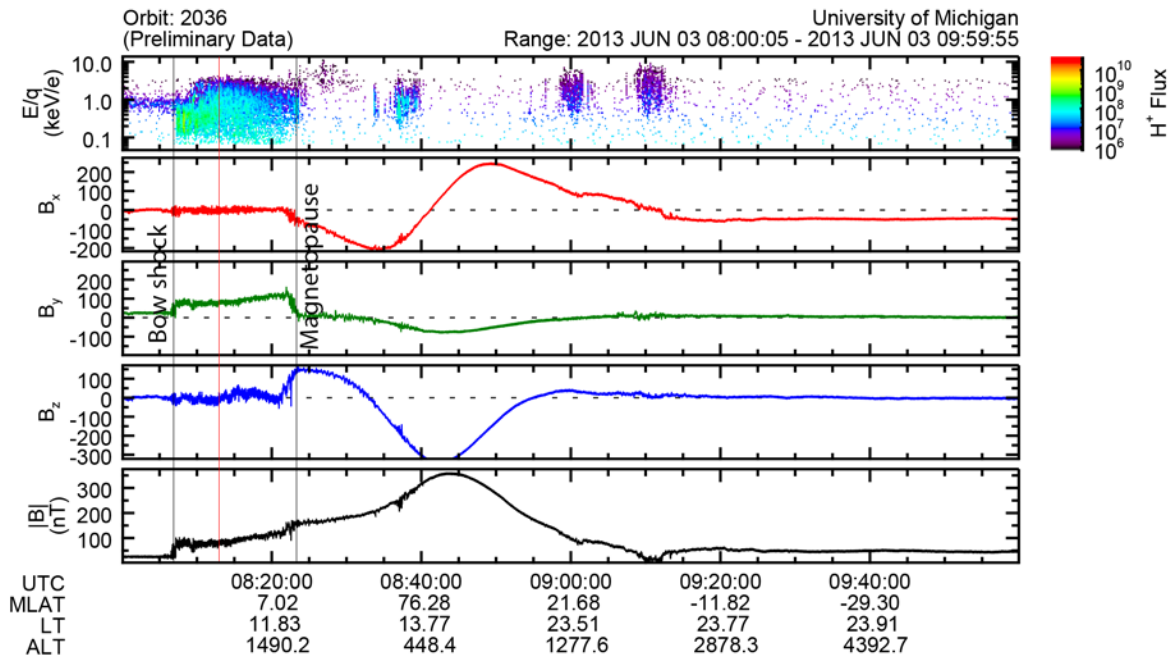


Figure 3: Overview of the 3 June 2013 08:00 orbit. Top to bottom: FIPS proton flux, vector magnetic field components, and magnetic field strength. Tick marks are labeled with the UTC time, magnetic latitude (MLAT), local time (LT) and altitude (ALT). The approximate bow shock and magnetopause crossings are marked in black; the approximate midpoint of the NTP calculation is marked in red. The NTP interval for this example falls well within the magnetosheath.

In the magnetosheath close to the subsolar point, we expect the shocked solar wind to be sufficiently isotropic and subsonic so that the NTP assumptions are valid. To confirm the validity of the NTP calculation, we follow the steps outlined above in the `ntp_tutorial_example1.idl` script. Below, we walk through each line in the `ntp_tutorial_example1.idl` script. Executing the script with the command `@ntp_tutorial_example1.idl` will perform all of these commands.

1. Initialize routines

To compile each of the IDL routines, use:

```
@code_idl_startup.idl
```

If the user has not done so already, the SPICE kernels need to be initialized (once per IDL session). To initialize the default kernels and the June 2013 kernel, see the Tutorial Structure section above.

2. Load the FIPS EDR, CDR, DDR data

Next, we load the NASA PDS data:

```
EDR = READ_FIPS_EDR()
CDR = READ_FIPS_CDR()
DDR = READ_FIPS_DDR()
```

Each of these programs uses the templates described in the .FMT files to open the respective data file, read in the relevant observations, and return the data in a structure of arrays. Within the EDR and CDR calls, the programs load the FIPS energy per charge (E/Q) table to associate each observation with the appropriate energy. Each of these calls returns the corresponding data products spanning the entire calendar day 3 June 2013.

The returned structures have the following fields:

EDR:

- MET: the mission elapsed time (MET);
- JD: the MET expressed as the Julian Date (JD);
- Counts: an L by M array of counts during each FIPS scan, where L is the number of scans and M is the number of energy channels (in s^{-1});
- EQTAB: an L by M array of energies corresponding to the counts array (in keV).

CDR:

- MET: the mission elapsed time (MET);
- JD: the MET expressed as the Julian Date (JD);
- Flux: an L by M array of flux during each FIPS scan, where L is the number of scans and M is the number of energy channels (in $s^{-1} cm^{-2} keV^{-1} Sr^{-1}$);
- EQTAB: an L by M array of energies corresponding to the flux array;
- PSD: an L by M array of phase space density (PSD) calculated from the flux observations by using:

$$F(v) = \frac{m_p}{v^2} J(v)$$

where $F(v)$ is the phase space density, $J(v)$ is the flux, v is the speed, and m_p is the species mass (here, protons). The PSD is reported in ($s^3 m^{-6}$).

DDR:

- MET_Start: the MET time corresponding to the beginning of the accumulation used in the NTP calculation;
- MET_Stop: the MET time corresponding to the end of the calculation; any scans with MET between the MET_Start and MET_Stop were used in the NTP calculation;
- JD_Start: MET_Start expressed as Julian Date;
- JD_Stop: MET_Stop expressed as Julian Date;
- N: calculated density n (in cm^{-3});
- T: calculated temperature T (in MK);
- P: calculated pressure (in nPa);
- Quality: a quality flag for the NTP data (see the corresponding .FMT file for description).

3. Derive “recovered” PSD from DDR NTP

To derive the PSD from the NTP calculation (the “recovered” PSD or the “DDR PSD”), we start by first specifying the DDR start and stop Julian Dates of interest. For the date and time 06-03-

2013T08:13:12.40, we find the DDR NTP calculation that includes our Julian Date of interest using the following commands:

```
JD_Interest = JULDAY(06,03,2013,08,13,12.40)
DDR_Index = (WHERE(DDR.JD_Start LE JD_Interest AND DDR.JD_Stop GE JD_Interest))[0]
```

Equipped with the index of the DDR NTP calculation of interest, we use the density and temperature of that NTP calculation to determine the “recovered” PSD:

```
F_DDR = NTP_to_PSD(DDR.N[DDR_Index], DDR.T[DDR_Index])
```

This function calculates the PSD from a non-drifting (stationary) Maxwellian with density n and temperature T :

$$F_{DDR}(v) = n \left(\frac{m_p}{2\pi kT} \right)^{3/2} e^{-\frac{1}{2} \frac{m_p v^2}{kT}}$$

where k is Boltzmann’s constant, m_p is the proton mass, v is the speed, T is the temperature, and n is the density. Note that we do not have to make use of the FIPS viewing model for this non-drifting Maxwellian case.

The function *NTP_to_PSD.pro* returns a structure of arrays with the following fields:

- PSD: the “recovered” PSD (in $s^3 m^{-6}$);
- Flux: the “recovered” flux (in $s^{-1} cm^{-2} keV^{-1} Sr^{-1}$), calculated using the same formula used to convert CDR flux to CDR PSD but of course using the “recovered” PSD;
- Speed: the speed (in km/s) of the PSD and flux.

4. Convert CDR flux to PSD and determine uncertainty

As described above, the *READ_FIPS_CDR.pro* function already calculates the CDR PSD from the flux using the equation:

$$F(v) = \frac{m_p}{v^2} J(v).$$

This PSD is specified for each FIPS scan while the NTP calculation spans multiple FIPS scans. To find a representative PSD for the NTP calculation interval, we average the observed PSD within the NTP interval. To find the FIPS scans that fall within the NTP interval, we identify the relevant array indices using the command

```
CDR_Index = WHERE(CDR.JD GE DDR.JD_Start[DDR_Index] AND CDR.JD LE DDR.JD_Stop[DDR_Index])
```

We then average the CDR PSD across these scans using the command

```
F_CDR = MEAN(CDR.PSD[CDR_Index,*], DIM=1)
```

We average the PSD instead of summing it because PSD is not an extrinsic property, i.e., increasing the number of scans should not increase PSD.

To determine the uncertainty associated with this averaged PSD, we propagate the Poisson errors from the EDR counts. Since the PSD in a given E/Q channel is linearly related to counts, we know that the fractional error from Poisson statistics in the counting $\left(\text{Fractional error} = \frac{\sqrt{\text{counts}}}{\text{counts}}\right)$ gives the fractional error on the PSD. Again, since the NTP calculation spans multiple FIPS scans, we must combine these scans together to calculate the uncertainty. Hypothetically, if a FIPS scan were to last as long as the NTP calculation (60s), the counts reported at the end of the scan would be the total counts accumulated during the scan (separated by E/Q channel). Therefore, to propagate the Poisson error, we add together all EDR counts in the FIPS scans during the NTP calculation interval. To find the EDR FIPS scans that fall within the NTP interval, we use the command

```
EDR_Index = WHERE(EDR.JD GE DDR.JD_Start[DDR_Index] AND EDR.JD LE DDR.JD_Stop[DDR_Index])
```

This array should be the same as the CDR_Index array. Next, we total the counts across these scans, keeping energy channels distinct, and compute the fractional error with the commands

```
EDR_Counts = TOTAL(1.*EDR.Counts[EDR_Index,*],1)
Frac_err = SQRT(EDR_Counts)/(EDR_counts>1.)
```

The logical operator >1. is used to avoid NaN values in the fractional error.

5. Compare CDR PSD with the “recovered” PSD

After computing the “recovered” PSD from the NTP calculation, we are almost ready to compare the FIPS-observed CDR PSD with the NTP-derived “recovered” PSD to determine the validity of the NTP calculation. We must first convert the CDR E/Q into speed for easier comparison with “recovered” PSD using the command

```
Speed_CDR = CONVERT_keV(ENERGY=CDR.EQTAB[CDR_Index[0],*])
```

In this call, it is assumed that the FIPS scan type (which can change the E/Q table) does not change across the NTP calculation so that the E/Q table at the first time step within the calculation interval is representative of the entire interval. The *CONVERT_keV.pro* function takes either an energy (in keV) or speed (in km/s) as the input parameter and returns the other. The input (energy or speed) is specified with the ENERGY or SPEED keywords, respectively.

The PSD comparison is visualized using the command

```
W = FIPS_Comparison_Plotter(CDR, DDR, DDR_Index, Speed_CDR, F_CDR, Frac_Err, F_DDR, $
/BUFFER, FILENAME='NTP_Tutorial_Example1_Step5.png')
```

This command call produces the following plot:

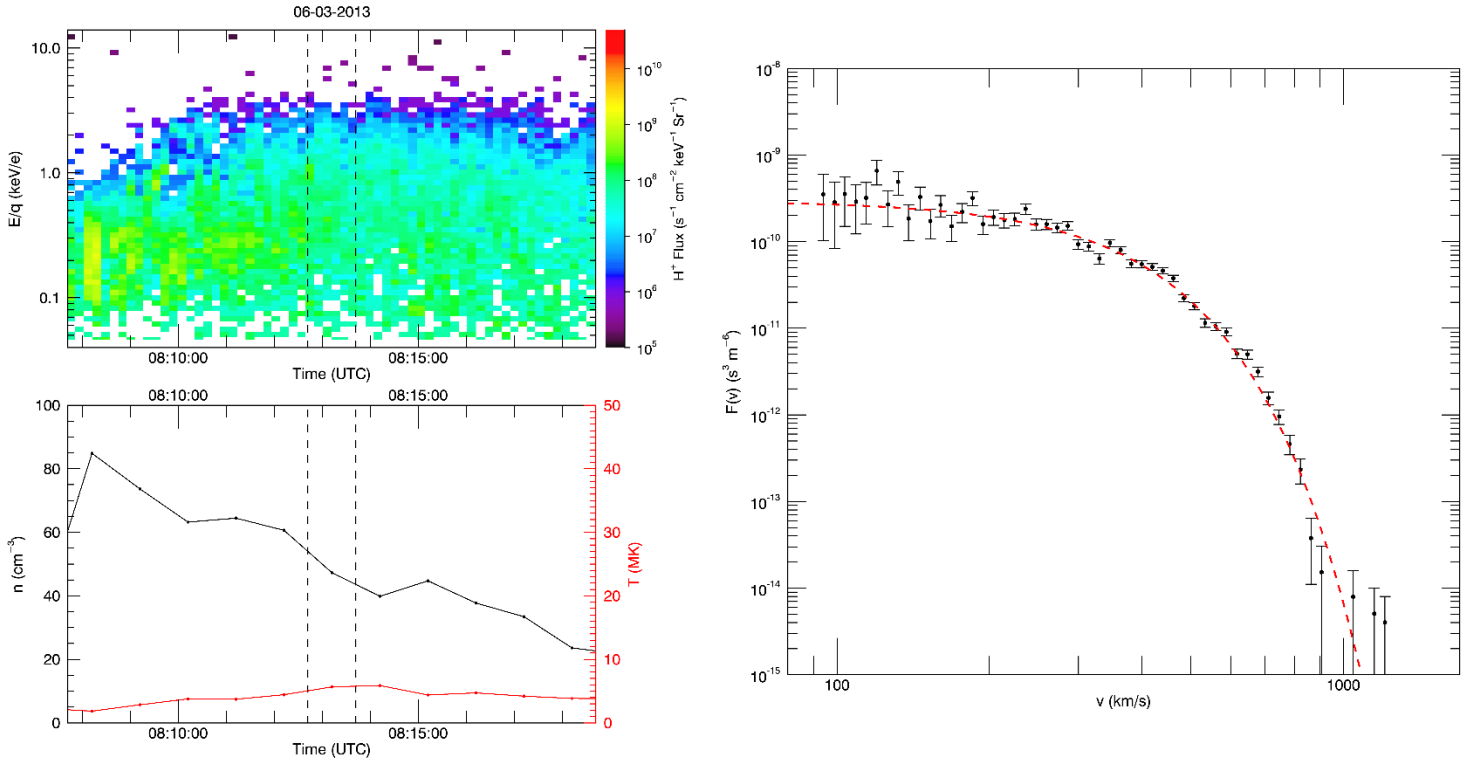


Figure 4: Comparison of CDR PSD with “recovered” PSD from DDR NTP. (top left) A time series of the CDR flux. The accumulation interval of the NTP calculation is bounded by dashed lines. (bottom left) A time series of the DDR density (black) and temperature (red). The symbols correspond to the midpoint of the NTP calculation time range. (right) Phase space density observed by FIPS (black, with error bars from Poisson statistics) and as derived from the NTP calculation (red, dashed). The close alignment between the observed and “recovered” PSD throughout the entire velocity range indicates that the calculated NTP are valid for this interval.

The top-left panel is a time series of the CDR flux, and the bottom-left panel is a time series of the DDR NTP calculations. In both panels, the vertical dashed lines indicate the JD_Start and JD_Stop for our NTP interval. The panel on the right compares the CDR PSD (black) with the “recovered” PSD (red, dashed). The close agreement between FIPS observations (CDR) and the NTP calculation (“recovered”) indicates that the NTP calculation for this interval is valid.

To test statistically the comparison between the “recovered” and observed PSD (and later, involve the modeled PSD), we suggest a modified chi-squared (χ^2) test. We define χ^2 to be:

$$\chi^2 = \sum_i \left(\frac{F_{CDR,i} - F_{test,i}}{\delta F_{CDR,i}} \right)^2$$

where F_{CDR} is the observed PSD, F_{test} is the test PSD (either the “recovered” or modeled PSD, or some combination of the two), and δF_{CDR} is the uncertainty of the observed PSD, each at energy (speed) step i . We calculate δF_{CDR} from the propagated Poisson errors, multiplying the fractional error by the observed PSD to get δF_{CDR} (see step 4).

For the comparison above, $\chi^2 = 96.1$, calculated by the following command:

```
Chi = FIPS_Distance_Measure(F_CDR, Frac_Err*F_CDR, ddr=F_DDR, specify_speed=speed_cdr)
```

We have tested this distance measure against several NTP calculations from various orbits and have found that while it tends to agree with conclusions of the fit by eye, it is not a robust measure. Given the logarithmic scaling of the comparison, small deviations (by eye) such as those at ~ 600 km/s above can significantly increase χ^2 . Deviations of a small number of single points, such as those near ~ 100 km/s in addition to those at ~ 600 km/s, can therefore be responsible for the bulk of the χ^2 value, i.e., the offset by a few points can dramatically increase χ^2 . Although not mathematically confirmed rigorously, we believe the NTP moment calculations tend to minimize χ^2 . Given the limitations of this distance measure, we do not include a specific χ^2 value below which the test PSD is “valid”, but use it for comparison between different test phase space densities.

6. Generate “modeled” PSD

The close comparison between the observed and “recovered” PSD in step 5 suggests the calculated n , T , and pressure are accurate for this interval, but to determine the robustness of the NTP calculation to different bulk flows, we must incorporate the FIPS model. In other words, the NTP calculation assumes no bulk flow (i.e., a non-drifting Maxwellian), so to determine what magnitude of bulk drift could exist without affecting the NTP calculation, we simulate flows in the FIPS model.

For this epoch, the NTP moments calculated $n = 47.27 \text{ cm}^{-3}$ and $T = 5.664 \text{ MK}$ (from DDR), resulting in $v_{th} = \sqrt{kT/m_p} = 216.2 \text{ km/s}$. The conditions on the NTP calculation require $v_{bulk}/v_{th} < 0.5$ so $v_{bulk} \lesssim 100 \text{ km/s}$ for the NTP assumptions to be met. To test how this bulk flow would affect the PSD comparison, we simulate flows with bulk speed from ~ 0 km/s to ~ 100 km/s incident on FIPS from various direction, using the same density and temperature (thermal speed) reported in the DDR records. At the subsolar magnetosheath, we expect some flow in the antisunward direction, and some flow around the planet. We try the following parameters and their possible combinations in *make_model_input_example1.pro*:

```
;; density
ns = [47.27]

;; velocity components in MSO
;; -----
vzs = [0.]
vys = [0., -20., -40., -80.]
vxs = [0., -20., -40., -80.]

;; thermal velocity
;; -----
vths = [216.2]
```

These conditions range from $v_{bulk} = 0$ km/s to $v_{bulk} = 113$ km/s, with the flow direction rotating from purely antisunward to purely dawnward.

To run the model with these conditions, we first compile the FIPS model using the command

```
@mfips_e2e_model/compile_mfips_e2e_model.idl
```

Next, we generate the input plasma-distribution parameters:

```
mets = (DDR.MET_Start[DDR_Index] + DDR.MET_Stop[DDR_Index])/2D  
ydoym = 2013154D  
make_model_input_example1, mets=mets, ydoym=ydoym
```

Within this program, we have specified the distribution parameters above. The program will save the plasma distributions in as *model_input.dat*. We then run the model by executing the command

```
run_model, inputfile='model_input.dat', fsw_v=7, ydoym=ydoym
```

The PostScript plots and the IDL savefile have been included in the 'data/' directory. Given the 16 plasma distributions this script will run through the FIPS model, executing the code in this step can take several minutes.

7. Compare modeled PSD with observed and "recovered" PSD

After generating the modeled PSD, we need to compare each of these distributions with the "recovered" and observed PSD. First, we restore the modeled PSD from the IDL savefile with the command

```
RESTORE, 'data/mfips_e2e_model_data_example1_step6.sav'
```

This command restores the savefile provided with the tutorial. If the user ran the model in step 6, to restore the savefile produced on the user's machine, use the command

```
RESTORE, 'mfips_e2e_model_data.sav'
```

(This command will restore the savefile from the most recent model run.)

To compare the phase space densities, we make use of the *FIPS_Model_Comparison_Plotter.pro* routine. As described in more depth in the program's documentation, the plotter receives the observed, "recovered", and modeled PSD, and plots all three against each other. The plotter operates with different modes to either plot all three PSDs independently, combine the modeled and "recovered" PSD, or combine multiple modeled PSDs together. As an example of the plotter, the command below plots the index-0, 5, 10, and 15 modeled PSD against the "recovered" and observed PSD as shown in Figure 5 below.

```
W = FIPS_Model_Comparison_Plotter (Speed_CDR, F_CDR, Frac_Err, F_DDR, Model_Data, $
'alone', [0,5,10,15], /BUFFER, FILENAME='NTP_Tutorial_Example1_Step7.png')
```

The index-0, 5, 10, and 15 modeled PSD correspond to plasma distributions each with unit vector flow direction $\hat{v}_{bulk} = [-\frac{1}{\sqrt{2}}, -\frac{1}{\sqrt{2}}, 0]$ in MSO coordinates and bulk speeds of ~ 0 km/s, ~ 28 km/s, ~ 57 km/s, and ~ 113 km/s, respectively. These flows correspond to the solid rainbow lines (blue to red, respectively) in Figure 5. We notice that the $v_{bulk} \approx 0$ km/s distribution (blue) matches the “recovered” PSD (red, dashed) nearly perfectly, as we would expect by modeling a non-drifting Maxwellian with the NTP moments in the FIPS model. As the bulk speed is increased, the modeled PSD starts to diverge from the “recovered” PSD for speeds greater than ~ 200 km/s.

Without a rigorous distance measure to test the goodness of fit of the modeled PSD to either the “recovered” or observed PSD, it is difficult to tell exactly what bulk flow causes too much misfit, but the χ^2 parameter defined above appears to agree with the visual trends. For the index-0, 5, 10, and 15 modeled PSD compared with the observed PSD, the χ^2 values are 94.7, 104.0, 142.2, and 348.2, respectively. For comparison, $\chi^2 = 96.1$ between the “recovered” and observed PSD. Discussed above, despite relatively similar looking curves in log-log space, the χ^2 exhibit a substantial dynamic range. To compute the distance measures for these indices we use the commands

```
Chi0 = FIPS_Distance_Measure(F_CDR, Frac_Err*F_CDR, model=Model_Data[0].FDIST)
Chi5 = FIPS_Distance_Measure(F_CDR, Frac_Err*F_CDR, model=Model_Data[5].FDIST)
Chi10 = FIPS_Distance_Measure(F_CDR, Frac_Err*F_CDR, model=Model_Data[10].FDIST)
Chi15 = FIPS_Distance_Measure(F_CDR, Frac_Err*F_CDR, model=Model_Data[15].FDIST)
```

Between (1) the deviations for speeds greater than ~ 500 km/s between the $v_{bulk} \approx 113$ km/s modeled distribution (red, solid) and the observed PSD and (2) the sharp change in χ^2 , it appears that the NTP calculations are generally insensitive to bulk flows $v_{bulk} < 113$ km/s. In other words, during this 3 June 2013 interval, there may have been subsonic flows (i.e., the plasma may have been best represented as a drifting Maxwellian) but these flows are likely $v_{bulk} < 113$ km/s if present. To determine exactly how χ^2 changes with a different bulk flows, the number and precision of modeled plasma distributions can be expanded in step 8.

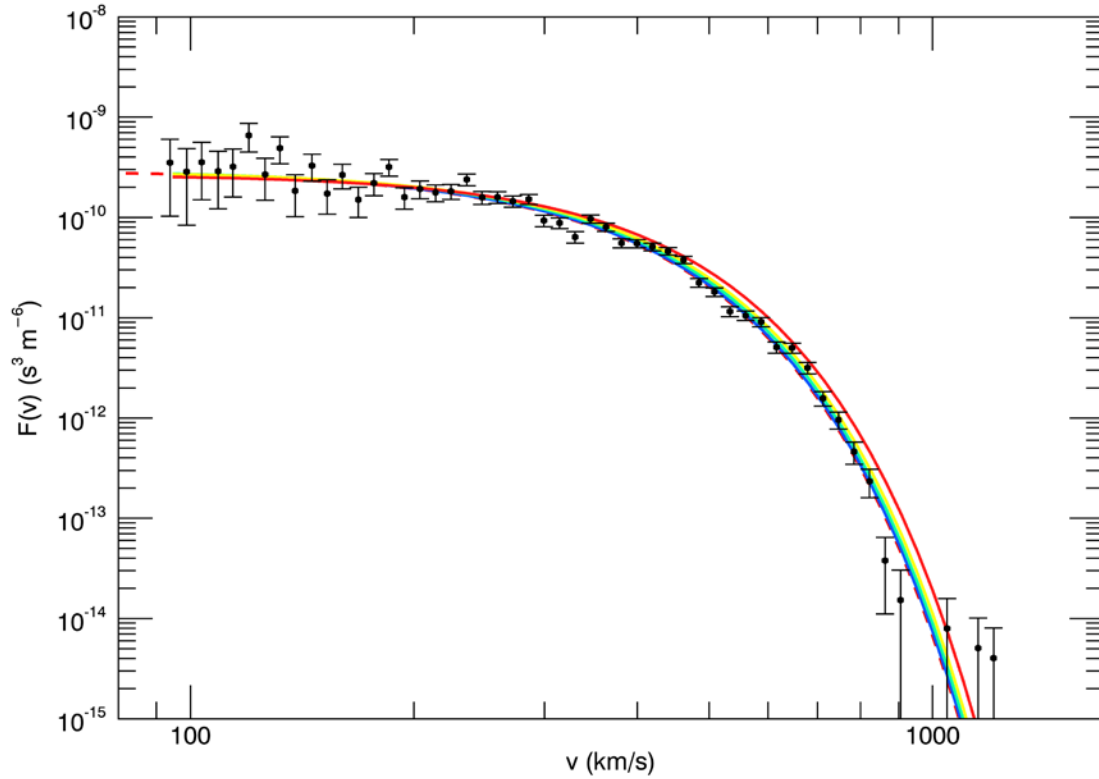


Figure 5. Observed PSD (black, with error bars), “recovered” PSD (red, dashed), and modeled PSD (rainbow, solid). The modeled PSD corresponds to a drifting Maxwellian distribution with $n = 47.27 \text{ cm}^{-3}$, $v_{th} = 216.2 \text{ km/s}$, and variable \vec{v}_{bulk} (see text).

8. Iterate steps 6 and 7

In step 7, we tested the robustness of the NTP calculation to several subsonic flows of the same density and temperature and found that the bulk flow may have been non-zero (but less than $\sim 113 \text{ km/s}$) without affecting the NTP calculation significantly. We encourage the user to explore the sensitivity of other parameters by iterating steps 6 and 7. For example, the user could vary the density and temperature in addition to a bulk flow to determine which density and temperature ranges replicate the “recovered” or observed PSD closely. The next step would be to create new plasma distributions to use in the FIPS model (as per step 6) and compare them with the phase space densities (as per step 7).

Example 2: NTP Misfit

This example steps the user through incorporating deviations from the NTP assumptions into the comparison between the NTP calculation and FIPS observations to inform the uncertainties of the moment calculations and to better understand FIPS observations at a specific epoch. This example builds on Example 1.

Example 1 walks the user through comparing an NTP calculation with FIPS observations for an interval during which the NTP assumptions appear valid. Not all NTP calculations fit the FIPS observations so well. Example 2 walks the user through an example where the “recovered” PSD does not match FIPS observations at all speeds. It follows the same initial 5 steps described in Example 1, but modifies the later steps by incorporating deviations from the NTP assumptions into modeling the hypothetical response of the FIPS sensor. Therefore, in addition to the steps outlined in Example 1, we:

6. Generate a modeled PSD by incorporating deviations from the NTP assumptions;
7. Compared the model PSD with FIPS observations;
8. Iterate steps 6 and 7.

To illustrate this procedure, we focus on a 60s-accumulated NTP calculation centered at epoch 06-03-2013T00:13:16.40 in *ntp_tutorial_example2.idl*. During this interval, MESSENGER was located at [1.83, -0.08, 0.02] R_M , corresponding to a local time of 11.8 hours, a geographic latitude of 0.8°, and a distance of 1.77 R_M from the planet’s center. Similar to the interval used in Example 1, the plasma and magnetic field observations confirm that MESSENGER was located in the subsolar magnetosheath at that time.

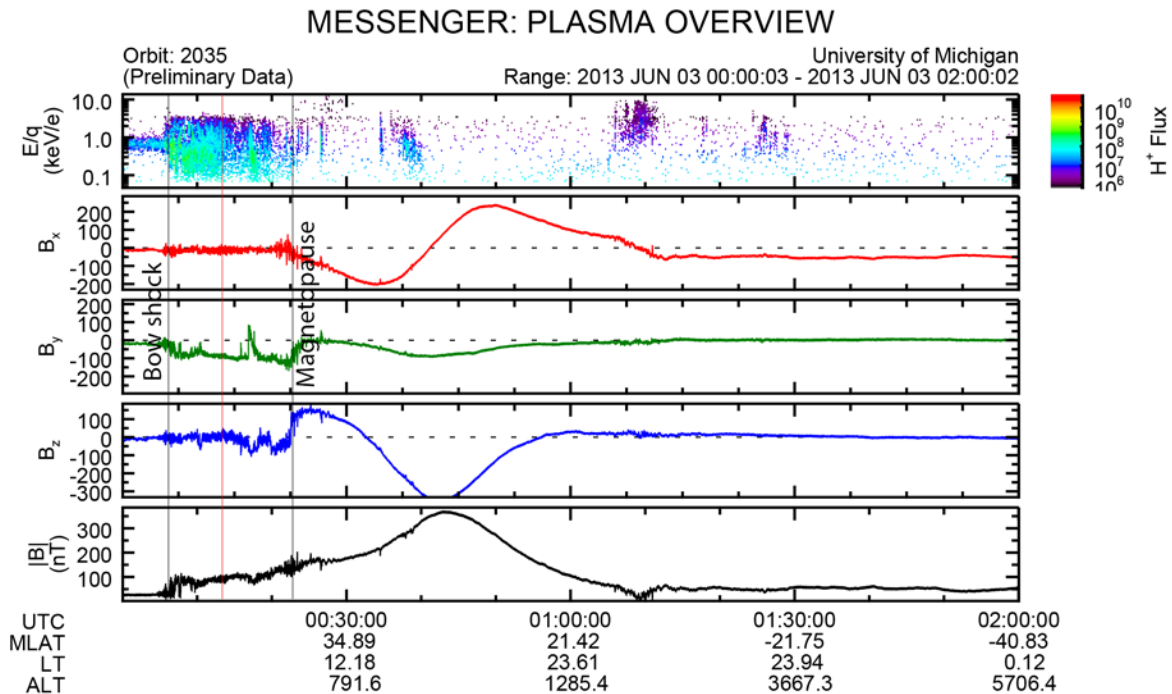


Figure 6: Overview of the 3 June 2013 00:00 orbit in the same format as Figure 3.

We expect the subsolar magnetosheath plasma to be sufficiently stagnant and thermalized for the NTP assumptions to be valid, but when we follow steps 1-5 above, we see that the “recovered” PSD fails to

fit FIPS observations for all energies (speeds), as shown in Figure 7. To reproduce this plot, follow steps 1-5 in *ntp_tutorial_example2.idl*. The NTP calculation matches PSD observations for speeds > 300 km/s, but we see significant disagreement by up to approximately an order of magnitude at low < 300 km/s speeds. The χ^2 value between the “recovered” and observed PSD is 190.7, greater than that between the “recovered” and observed PSD in Example 1.

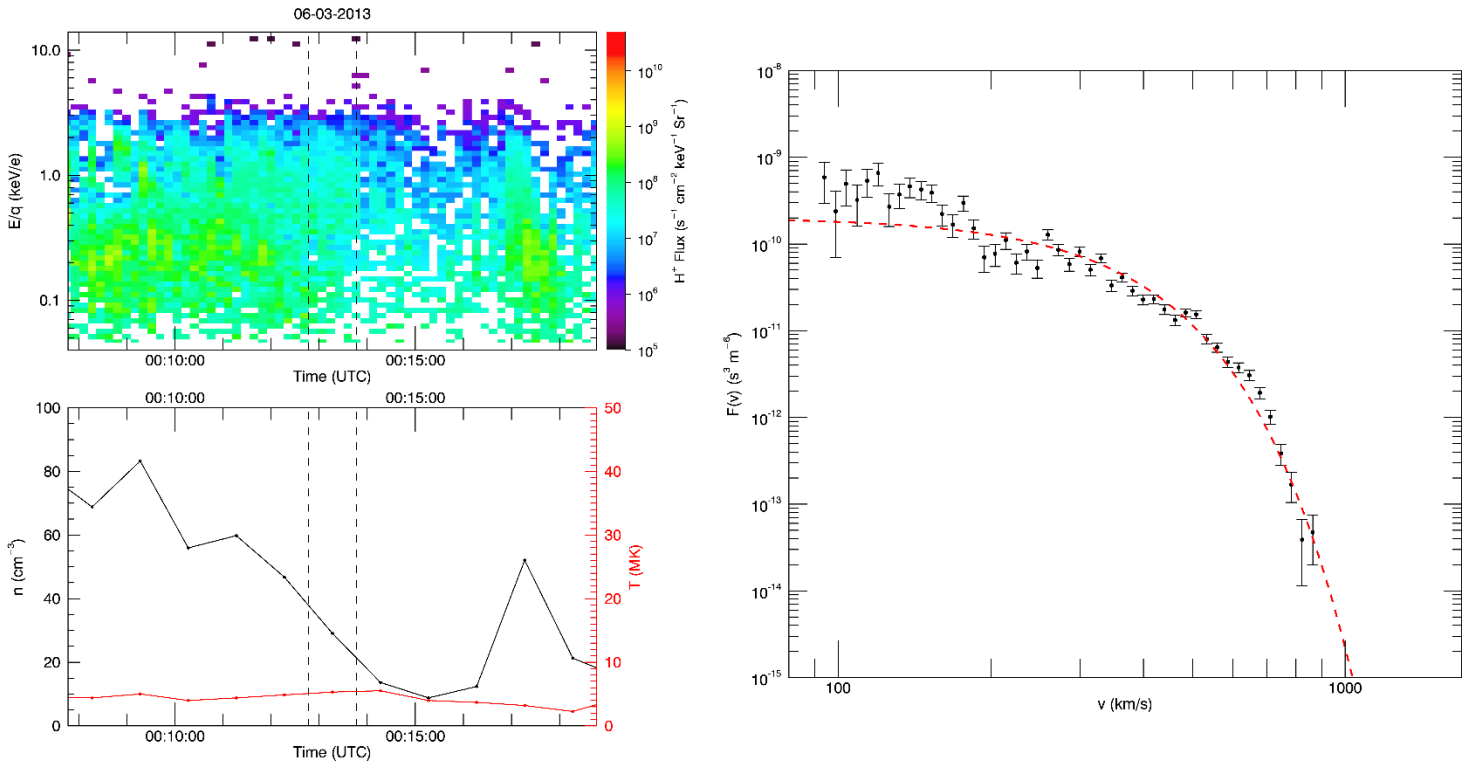


Figure 7: Comparison of CDR PSD with “recovered” PSD from DDR NTP, in the same format as Figure 4.

Several of the NTP assumptions, such as a negligible bulk flow, may have been violated during this interval, producing the misfit between the NTP calculation and FIPS observation. In order to incorporate deviations from the NTP assumptions into our comparison, we model the PSD that FIPS would hypothetically observe given a specific plasma distribution. We encode deviations from the NTP assumptions within this plasma distribution.

6. Generate a modeled PSD

Whereas Example 1 modeled plasma distributions to test the robustness of the (valid) NTP calculation, Example 2 models plasma distributions to determine which violations of the NTP assumptions cause the apparent misfit.

Here, we can start by adding a second flow to the “recovered” PSD. When we added a bulk flow to the DDR density and temperature in Example 1, the low speed portion of the PSD was relatively unchanged for bulk velocities ranging from ~0 km/s to ~100 km/s (see Figure 5). To reproduce the low-speed enhancement in Figure 7, we may therefore have more success by superposing a second flow in addition

to the DDR non-drifting Maxwellian. For clarity, a phase space density produced from the FIPS model will be referred to as a modeled PSD (or modeled flow), whereas the non-drifting Maxwellian distribution produced directly from the NTP measurements is referred to as the “recovered” PSD.

To model the low-speed deviations in the PSD in Figure 7 by adding a second flow, we need to guess a range of plasma distribution parameters to serve as the initial trials of the FIPS model. Since the “recovered” PSD underestimates at speed < 300 km/s, we expect that the bulk speed cannot be too large; a large bulk speed could produce a local maxima far from $v \sim 0$, depending on the thermal spread. Since the PSD appears fairly uniform across this low-speed interval, the plasma population is not likely to be a supersonic beam of plasma within the FIPS FOV; a supersonic beam would produce a strong maximum at the drift speed velocity and drop off quickly at both lower and higher speeds. Since the magnetosheath is composed of the shocked solar wind, we expect some flow in the anti-sunward direction. We also expect some motion around the planet as the flow diverts at the subsolar magnetopause. As an initial guess, we try the following parameters and their possible combinations in the *make_model_input_example2_step6.pro*:

```
;; density
ns = [5., 10., 25.]

;; velocity components in MSO
;; -----
vzs = [-100., 0., 100.]
vys = [-100., 0., 100.]
vxs = [-100.]

;; thermal velocity
;; -----
vths = [100., 300.]
```

These conditions will range from highly subsonic (Mach number ~ 0.3) to transonic (Mach number ~ 1), under various bulk flow directions and densities.

To run the model with these conditions, follow step 6 in *ntp_tutorial_example2.pro*. The PostScript plots and the IDL savefile have been included in the ‘data/’ directory. Given the 54 plasma distributions this script will run through the FIPS model, executing the code in this step can take several minutes.

7. Compare modeled PSD with observed and “recovered” PSD

After generating the modeled PSD, we need to compare each of these distributions with the “recovered” and observed PSD. After we restore the modeled PSD from the IDL savefile, we use the *FIPS_Model_Comparison_Plotter.pro* to compare the various PSDs. Since we are combining two plasma flows in our comparison (the “recovered” PSD and a modeled PSD), we use a different mode of the plotter than in Example 1 step 7. Following the documentation in the program, we use the “combined” mode of the plotter.

After comparing each modeled PSD with the “recovered” and observed PSD, we find that the modeled distribution that best accounts for the discrepancies corresponds to the 27th of the plasma distributions modeled. The IDL call to create the comparison between this modeled flow and the observed plasma data is shown in Figure 7 (this also serves as an example of this mode of the plotter):

```
W = FIPS_Model_Comparison_Plotter (Speed_CDR, F_CDR, Frac_Err, F_DDR, Model_Data, $  
  'combined', [26], /BUFFER, FILENAME='NTP_Tutorial_Example2_Step7.png')
```

This command produces Figure 8 and corresponds to the following modeled plasma distribution parameters (in MSO coordinates):

$$\begin{aligned}n &= 10.0 \text{ cm}^{-3} \\v_{th} &= 100 \text{ km/s} \\\vec{v}_{bulk} &= [-100, 0, 0] \text{ km/s}\end{aligned}$$

Superposing this flow on the hot, stagnant plasma distribution (the “recovered” PSD) improves the fit at low speeds. There is, however, some discrepancy at a velocity of ~200 km/s, which leads to an increased χ^2 at 339.4. Physically, the modeled PSD corresponds to a transonic flow in the antisunward direction. Considering the distance between MESSENGER’s location during the NTP interval and the magnetopause (see Figure 3) and how close the spacecraft was to the subsolar point (local time of noon, equatorial latitude), a transonic antisunward flow (modeled PDS) superposed on a hot, stagnant flow (“recovered” PSD) is physically reasonable.

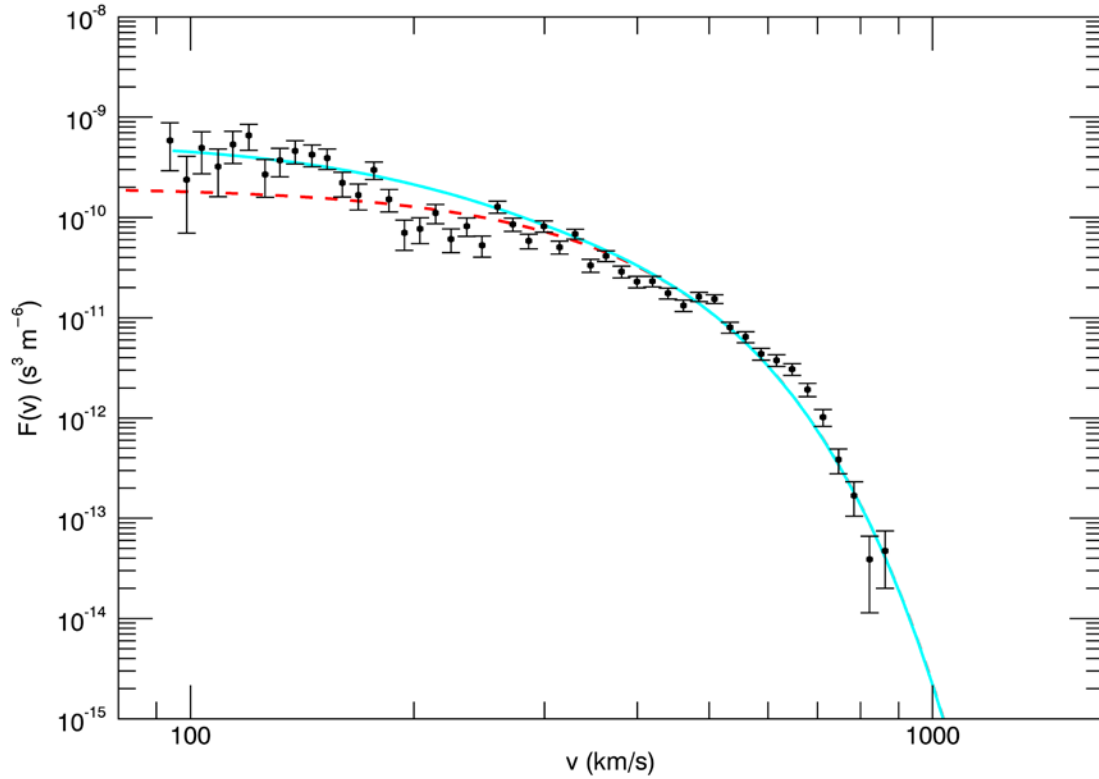


Figure 8. Observed PSD (black, with error bars), “recovered” PSD (red, dashed), and “recovered” PSD combined with the additional modeled secondary population (cyan, solid). The secondary population corresponds to a drifting Maxwellian distribution with $n = 10.0 \text{ cm}^{-3}$, $v_{th} = 100 \text{ km/s}$, and $\vec{v}_{bulk} = [-100, 0, 0] \text{ km/s}$.

8. Iterate steps 6 and 7

If the plasma distributions from step 6 do not adequately fit the discrepancies between the “recovered” and observed PSD, or to investigate any degeneracies with the modeled PSD, the next step would be to create new plasma distributions to use in the FIPS model (as per step 6) and compare them with the phase space densities (as per step 7).

For example, replacing the “recovered” PSD from the NTP calculations with a drifting Maxwellian with properties:

$$\begin{aligned} n &= 40.0 \text{ cm}^{-3} \\ v_{th} &= 165 \text{ km/s} \\ \vec{v}_{bulk} &= [-300, 0, 0] \text{ km/s} \end{aligned}$$

And replacing the secondary flow with:

$$\begin{aligned} n &= 10.0 \text{ cm}^{-3} \\ v_{th} &= 50 \text{ km/s} \\ \vec{v}_{bulk} &= [-100, 0, 0] \text{ km/s} \end{aligned}$$

Produces the following fit:

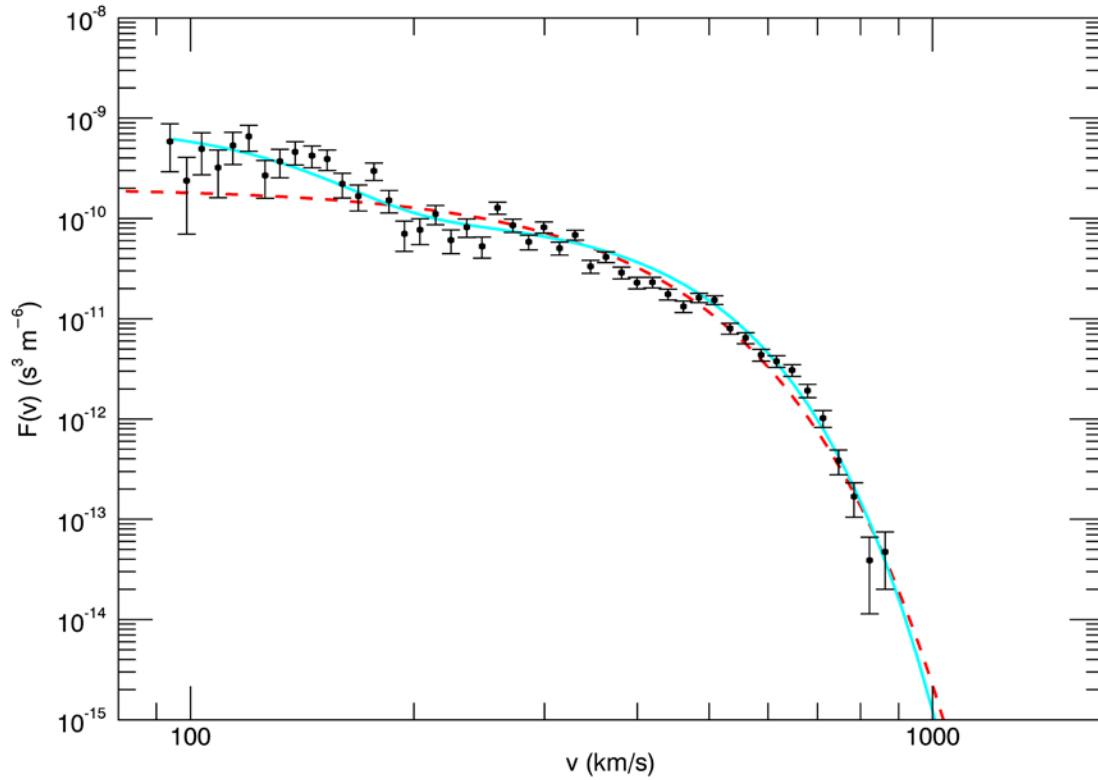


Figure 9. Observed PSD (black, with error bars), “recovered” PSD (red, dashed), and two modeled flows superimposed (cyan, solid). See text for the plasma distributions of the two modeled flows.

These two modeled flows superimposed (cyan) improve the comparison between the modeled PSD and the observed PSD at speeds between ~ 200 km/s to ~ 300 km/s compared to the modeled flows in Figure 8. The χ^2 value reflects this improved fit, with $\chi^2 = 173.3$ here. Given the position of the spacecraft in the magnetosheath (again), it is not clear that these two flows are physically reasonable, but this is a scientific judgment that must be made on a case-by-case basis in the context of a wider scientific analysis of the data. In this example, we have not performed this wider analysis.

Summary

Due to the limited field of view and accommodation on the MESSENGER spacecraft, the FIPS sensor can only derive the plasma density, temperature, and pressure (NTP) moments under specific assumptions about the plasma distribution it samples. These NTP calculations were archived at the PDS for portions of each MESSENGER orbit, although the assumptions may not be valid for each calculation individually. This tutorial steps the user through two case studies of how to access FIPS data, compare the NTP calculations against FIPS observations, and in the case of discrepancy, how to model plasma distributions that may violate one or more of the NTP assumptions. With the general procedure and sample code developed in this tutorial, we have provided the user the necessary framework to evaluate the validity of the NTP calculations for any epoch of the MESSENGER mission.

References

- Gershman, D. J., J. A. Slavin, J. M. Raines, T. H. Zurbuchen, B. J. Anderson, H. Korth, D. N. Baker, S. C. Solomon (2013), Magnetic flux pileup and plasma depletion in Mercury's subsolar magnetosheath, *J. Geophys. Res. Space Physics*, **118**, 7181-7199, doi:10.1002/2013JA019244.
- Gershman, D. J., J. A. Slavin, J. M. Raines, T. H. Zurbuchen, B. J. Anderson, H. Korth, D. N. Baker, S. C. Solomon (2014), Ion kinetic properties in Mercury's pre-midnight plasma sheet, *J. Geophys. Res. Lett.*, **41**, 5740-5747, doi:10.1002/2014GL060468.
- Raines, J. M., J. A. Slavin, T. H. Zurbuchen, G. Gloeckler, B. J. Anderson, D. N. Baker, H. Korth, S. M. Krimigis, R. L. McNutt Jr. (2011), MESSENGER observations of the plasma environment near Mercury, *Planet and Space Sci.*, **59**, 2004-2015, doi:10.1016/j.pss.2011.02.004.
- Sun, W. J., S. Y. Fu, J. A. Slavin, J. M. Raines, Q. G. Zong, G. K. Poh, T. H. Zurbuchen (2016), Spatial distribution of Mercury's flux ropes and reconnection fronts: MESSENGER observations, *J. Geophys. Res. Space Physics*, **121**, 7590-7607, doi:10.1002/2016JA022787.