

Ion Velocity Distributions: Beam Fitting Software

Lynn B. Wilson III

October 9, 2012

Contents

1	Introduction	1
1.1	Placement/Location of IDL Routines	1
2	IDL Startup	2
2.1	Starting/Initializing IDL	2
2.2	IDL Structures	5
3	Plot Descriptions	6
4	IDL Routines	8
4.1	IDL Routine Outline	8
4.2	Process Outline	8
5	Prompt Information	11
5.1	Prompt Information: Plotting Parameter Initialization	11
5.2	Prompt Information: Changing Plotting Parameters	16
5.3	Prompt Information: Creating Core Mask	20
5.4	Prompt Information: Finding the Beam Peak	24
5.5	Prompt Information: Creating Beam Mask	27
5.6	Prompt Information: Fitting to the Beam Peak	31
6	Future Changes/Additions	39
A	Particle Data Structures in IDL	40
B	Unit Conversions	41
C	Bi-Maxwellian Distribution Functions	42
D	Fluid Moment Definitions	43
E	Statistics Definitions	45
<i>References</i>		47

1 Introduction

This software package is intended to allow a user to take an array of ion velocity distribution functions (as IDL structures) and interactively fit beams to bi-Maxwellian distributions. All the routines in this package require either the **UMN Modified Wind/3DP**¹ or **THEMIS TDAS**² IDL libraries. The routines each have a detailed man page that explains the usage, purpose, inputs, and keywords. The man page also contains a list of routines that are called in any given program/function, as well as the routine which called said program/function. **Make sure your IDL paths are set correctly so that the routines and their dependencies can be found and compiled when called.**

The routines make no assumptions about the type of distribution observed until the user reaches the stage of fitting to a model function, which happens to be a bi-Maxwellian here. The end results of this package are a series of postscript (PS) files containing the different plots produced through the process and an ASCII file containing the results of each fit. Note that the user can produce a plot of the phase space density³ without going through the rest of the fitting routines. The process involved in the analysis will be described below.

1.1 Placement/Location of IDL Routines

You should have a specific directory where you run IDL to avoid making IDL search your entire computer for called routines to compile. If not, you should at least have two special places, one for the **THEMIS TDAS** library and one for the **UMN Modified Wind/3DP** library. Many of the routines in the **UMN Modified Wind/3DP** library overlap with those in the **THEMIS TDAS** library, so be careful how you use these together. For instance, do not place the `~/wind_3dp_pros/` folder in the `~/tdas_?_??/idl/` directory. The two libraries should be in separate locations, but you can alter your IDL path specifications to allow IDL to find both sets of routines from one place⁴.

¹Found at: http://tetra.space.umn.edu/wiki/doku.php/umn_wind3dp

²Found at: <http://themis.ssl.berkeley.edu/software.shtml>

³technically the instrument measures E/q and we assume a mass for the incident particles so that we are really producing **velocity** space densities

⁴I discuss this in Section 2.1, but I apologize for limiting the discussion to Unix/Linux/Mac OSs. There are similar methods for forcing IDL to look at specific directories on Windows machines, but I am not familiar with them.

2 IDL Startup

Below, we will very briefly explain how to start IDL so that the software can be compiled and used correctly. All routines are found in the `~/wind_3dp_pros/LYNN_PRO/beam_fitting_routines/` directory of the **UMN Modified Wind/3DP** IDL package⁵.

2.1 Starting/Initializing IDL

I am going to assume that you know how to obtain the ion velocity distribution functions (DFs) as IDL structures for the instrument you are interested in examining. Appendix A lists many of the necessary structures tags for each individual IDL structure. If you are not familiar with these structures, there are detailed crib sheets found in `~/wind_3dp_pros/wind_3dp_crbs/` directory in the **UMN Modified Wind/3DP** IDL library explaining how to obtain either Wind/3DP [Lin et al., 1995] or THEMIS ESA [McFadden et al., 2008a,b] DFs⁶.

If you are looking at THEMIS data, then you want to use the TDAS start-up initialization software (*i.e.* type `themis` in a Unix terminal at the command line). If you are looking at Wind data, then use the **UMN Modified Wind/3DP** start-up initialization software⁷.

Note that if you use the TDAS software, you need to include `comp_lynn_pros.pro` in the `~/tdas_?_?/idl/` directory. You will also need to adjust the IDL paths in `setup_themis_bash` to allow IDL to look at the following directories in the `~/wind_3dp_pros/` directory: `LYNN_PRO/`, `THEMIS_PRO/`, `Coyote_Lib/`, and `rh_pros/`. Follow the same format as the supplied version of `setup_themis_bash` for each directory and add them to the end of TDAS IDL path specification.

The following should be placed in your `.bash_profile` or `.bashrc`. Here is an example of how to initialize your IDL paths using IDL's bash setup routine:

```
##### IDL setup #####
## Let IDL's bash script define default IDL paths
source /Applications/itt/idl/bin/idl_setup.bash
## Add current working directory path and any subdirectory paths
## onto IDL's default paths
IDL_PATH=$IDL_PATH:+.:+Applications/itt/idl/lib
## The second part is in case the idl_setup.bash file is not
## located or its paths point to the wrong place...
```

⁵Found at: http://tetra.space.umn.edu/wiki/doku.php/umn_wind3dp

⁶e.g. see `my_3DP_moments_save-files.txt` for 3DP or `themis_esa_*_crib.txt` for ESA

⁷see bash profile examples for `widl` and `widl64` at: http://tetra.space.umn.edu/wiki/doku.php/umn_wind3dp

The following is an example of how to use a bash script to initialize IDL specific to the TDAS software:

```
#####
## Set up TDAS, THEMIS
#####
function themis {
    ## We have to unset the IDL_PATH to avoid TPLOT conflicts... kludgy.
    ## If you have personal IDL routines that you normally include (as long as
    ## they don't include any TPLOT routines!), you may add them to the IDL_PATH
    ## after we clobber it below.

    unset IDL_PATH

#####
## Define the TDAS path
## [change this for your machine]
#####
export TDAS_UMN=/Users/lbwilson/Desktop/Old_or_External_IDL/TDAS/tdas_7_00

#####
## Define the ITT IDL path
## [change this for your machine]
#####
IDL_LOC='/Applications/itt/idl/'
source ${IDL_LOC}/bin/idl_setup.bash

#####
## Define TDAS IDL setup paths
## [change this for your machine]
#####
source $TDAS_UMN/idl/themis/setup_themis_bash
idl

#####
## Reset defaults
#####
source $HOME/.bash_profile
}
```

It should be placed in your `.bash_profile` or `.bashrc`. Note I have added comments specifying where you need to alter specific directory paths that will be specific to your machine.

The following shows you how to alter the file `~/tdas_?-??/idl/themis/setup_themis.bash`. Here are some examples of how to alter the default IDL paths in that file:

```
#####
## Configure the directory paths below accordingly for your machine
#####

## Location where the TDAS IDL code is installed
if [ ${IDL_BASE_DIR:-0} == 0 ] ; then
    export IDL_BASE_DIR ; IDL_BASE_DIR=/Users/lbwilson/Desktop/0ld_or_External_IDL/TDAS/tdas_7_00/idl
fi

## Location of my IDL routines
if [ ${IDL_LYNN_PRO_DIR:-0} == 0 ] ; then
    export IDL_LYNN_PRO_DIR ; IDL_LYNN_PRO_DIR=/Users/lbwilson/Desktop/swidl-0.1/wind_3dp_pros/LYNN_PRO
fi

## Location of my THEMIS-specific IDL routines
if [ ${IDL_THEMIS_PRO_DIR:-0} == 0 ] ; then
    export IDL_THEMIS_PRO_DIR ; IDL_THEMIS_PRO_DIR=/Users/lbwilson/Desktop/swidl-0.1/wind_3dp_pros/THEMIS_PRO
fi

## Location of Coyote Libraries [David Fanning's Routines]
if [ ${IDL_COYOTE_DIR:-0} == 0 ] ; then
    export IDL_COYOTE_DIR ; IDL_COYOTE_DIR=/Users/lbwilson/Desktop/swidl-0.1/wind_3dp_pros/Coyote_Lib
fi

## make sure IDL_PATH is initialized before we add THEMIS paths to it
export IDL_PATH; IDL_PATH=${IDL_PATH}:'<IDL_DEFAULT>'

## Set path for all IDL source code:
IDL_PATH=$IDL_PATH':'+$IDL_BASE_DIR':'+$IDL_LYNN_PRO_DIR':'+$IDL_THEMIS_PRO_DIR':'+$IDL_COYOTE_DIR
```

where you would need to change the following partial directory path:

```
/Users/lbwilson/Desktop/swidl-0.1/
```

to a path specific to your computer that points to the location of the `~/wind_3dp_pros/` directory. The corresponding partial path would also need to be changed in `comp_lynn_pros.pro` before calling. In `comp_lynn_pros.pro`, the routines with the following directory paths can be commented out as well:

```
/Users/lbwilson/Desktop/idllibs/*
/Users/lbwilson/Desktop/swidl-0.1/IDL_stuff/*
```

since they are idiosyncratic to my computer and not included in the ***UMN Modified Wind/3DP*** IDL library. Note that the * is use to represent a wild card flag here indicating all subsequent subdirectory path extensions beyond what is shown.

2.2 IDL Structures

The IDL structures may require minor modification if you are using data from the THEMIS spacecraft [Angelopoulos, 2008]. Once you retrieve an array of burst data structures using [thm_part_dist_array.pro](#), it is important to note that the THETA and PHI (see Appendix A for definition) angle bin tags are defined in the DSL coordinate system. It is also important to note that some of the structure tags differ from the 3DP structure tags (e.g. VSW is not found). The **UMN Modified Wind/3DP** IDL library uses the VSW tag for the bulk flow velocity while the TDAS software uses VELOCITY.

The routine, [modify_themis_esa_struct.pro](#), is a vectorized routine that adds the appropriate structure tags necessary for using THEMIS ESA IDL structures with the **UMN Modified Wind/3DP** software. Then one can take that modified array of IDL structures and pass it to the vectorized routine, [rotate_esa_thetaphi_to_gse.pro](#), which rotates the THETA and PHI angles to GSE coordinates. This routine will also add the corresponding GSE magnetic field and bulk flow velocities (using TPILOT handles with the **MAGF_NAME** and **VEL_NAME** keywords) to the input array of structures so that everything is in the same coordinate basis. **Be careful! Both of these routines modify the input structures so you may wish to make copies.**

3 Plot Descriptions

This section will explain/describe the basic anatomy of the plots produced by the routines described below. Figure 1 shows the basic anatomy of the ion velocity distribution functions (DFs) that will be shown herein. The basic setup of this example will reflect all the DFs presented. Note that the crosshairs (horizontal red line and vertical blue line) in the contour plot can move and the cuts of the DF shown in the panel below will always represent the DF along the corresponding color-coded crosshair. Meaning, if the red line in the crosshair was shifted vertically 100 km/s, then the red line in the cuts plot below would change to indicate the cut along this line through the DF.

The example shows the contours projected onto the plane containing the average bulk flow

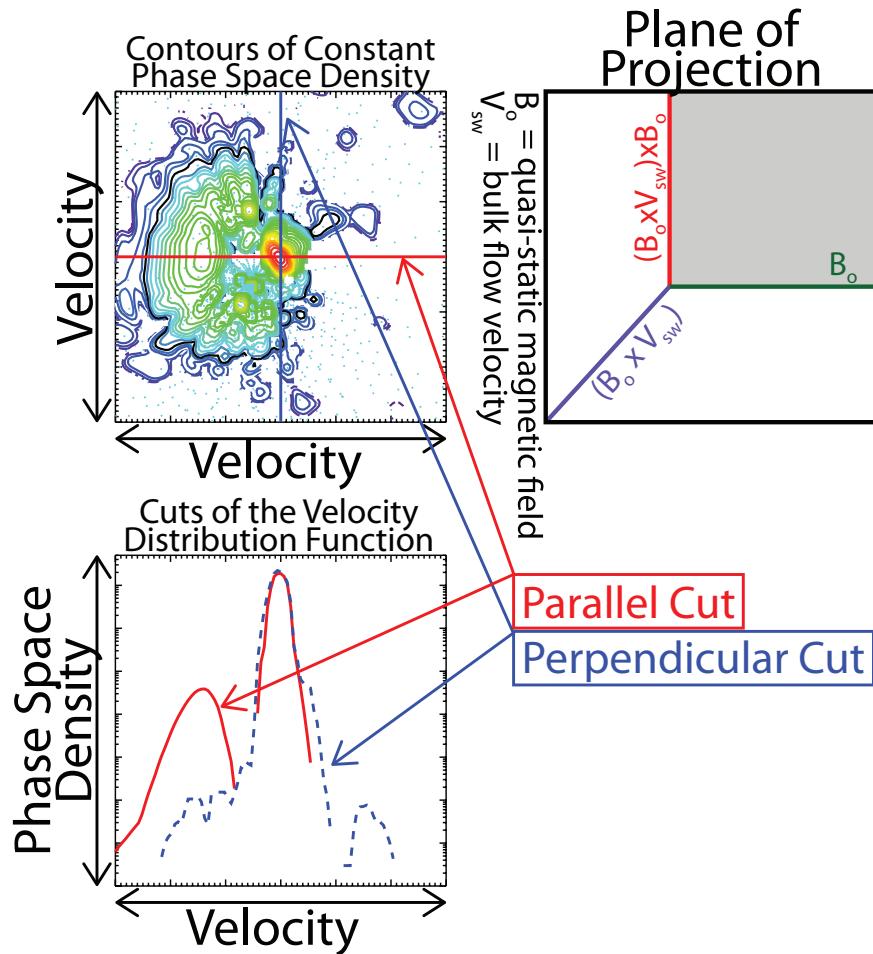


Figure 1: An example ion velocity distribution observed by the THEMIS IESA electrostatic analyzer. The figure shows the contour plot, cuts of the distribution, and plane of projection for the contour plot (indicated by the shaded region in upper-right-hand corner panel). In the contour plot, red(purple) contours correspond to the highest(lowest) phase space densities for this distribution. The color scale for the contours will be indicated by the range of phase space densities in the cuts plot directly below. The cut lines shown in the bottom panel are color-coded and correspond to the crosshairs in the contour panel. The velocity will be shown in 1000's km/s and the phase space densities in $\text{s}^3\text{cm}^{-3}\text{km}^{-3}$.

velocity (\mathbf{V}_{sw}) and quasi-static magnetic field (\mathbf{B}_o). All the DFs shown herein will lie in the same plane, but the routines allow the user to use any of the three planes shown in the upper-right-hand panel of Figure 1. These distributions do not assume gyrotropy. For more examples of these types of plots, see *Wilson III et al.* [2009, 2010, 2012].

An additional advantage of this software is that it allows the user to interactively change the estimate of the bulk flow velocity. Some previous studies have plotted foreshock ion DFs in the spacecraft frame, however I would strongly urge you not to try this. Figure 2 contours of constant phase (velocity) space density versus velocity relative to the directions defined by the shadowed planes shown to the right. As you can see, the interpretation of the distributions might change as a consequence of not being in the bulk flow frame (right-hand column of contours). Note that the plane is also dependent on your definition of \mathbf{V}_{sw} , which if inaccurate, can cause the triangulation routines to project data onto a plane that does not contain a significant fraction of the core or beam components (e.g. compare top row of contour plots in Figure 2).

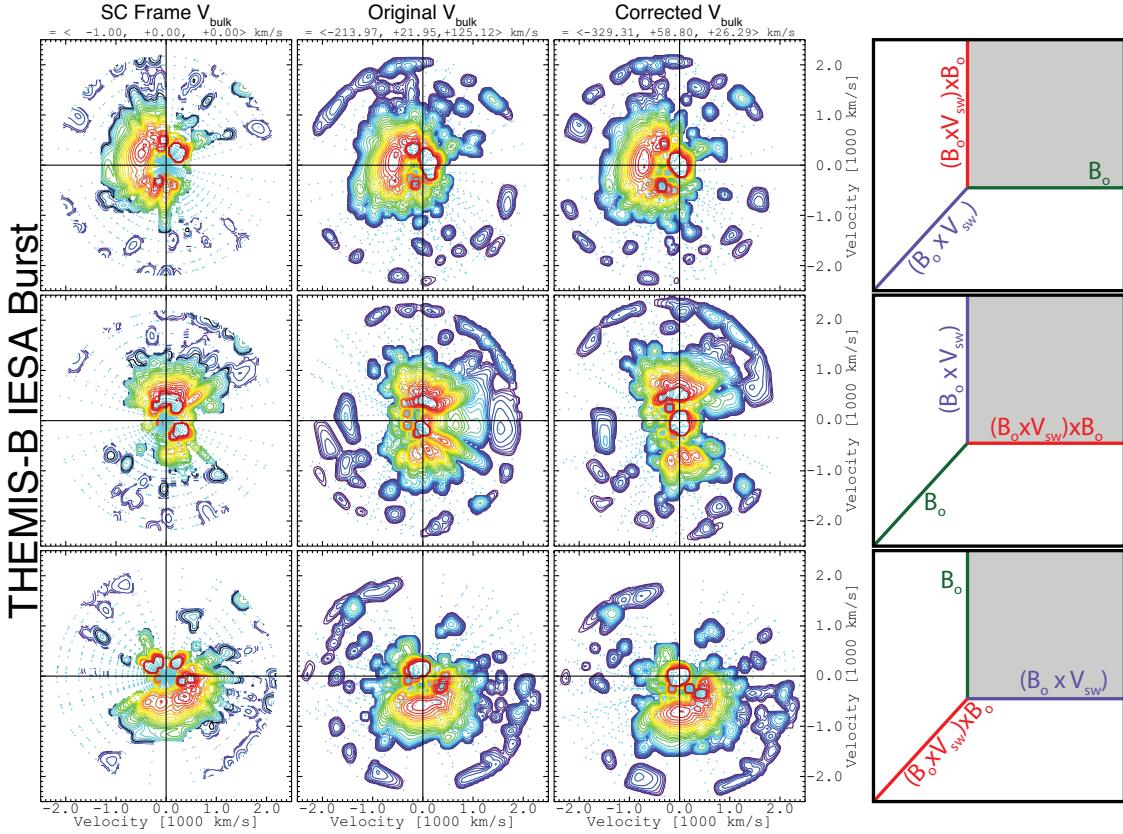


Figure 2: An example THEMIS IESA Burst particle velocity distribution observed upstream of a bow shock crossing. The three columns of contour plots correspond to three different rest frames defined by \mathbf{V}_{bulk} at the top of each column. Each contour plot shows contours of constant phase space density (uniformly scaled from 1×10^{-14} to $1 \times 10^{-8} \text{ s}^3 \text{cm}^{-3} \text{km}^{-3}$, where red is high) versus velocity projected onto three different planes defined by the shaded region in the coordinate axes shown in right-hand column. The velocity axes range from $\pm 2500 \text{ km/s}$ and the crosshairs show the location of the origin.

4 IDL Routines

The routines explained herein are found in the `~/beam_fitting_routines/` subdirectory of the `~/wind_3dp_pros/LYNN_PRO/` directory of the **UMN Modified Wind/3DP** IDL package⁸. Below, the manner in which the routines are called will be discussed and how to format the input IDL structure will be discussed as well.

4.1 IDL Routine Outline

The main wrapping routine is `wrapper_beam_fit_array.pro`. The routine, `beam_fit_1df_plot_fit.pro`, is the main wrapping routine for the plotting and fitting. The contour plotting routine is `beam_fit_contour_plot.pro`.

The routine, `beam_fit_keywords_init.pro`, initializes the plotting parameters used in the contour and cuts plots.

The wrapping routine for the beam fitting and prompting is `beam_fit_fit_wrapper.pro`. This routine calls `beam_fit_fit_prompts.pro` and `df_fit_beam_peak.pro`.

The main prompting routines are: `beam_fit_gen_prompt.pro`, `beam_fit_prompts.pro`, `beam_fit_options.pro`, and `beam_fit_fit_prompts.pro`.

The two common blocks are: `beam_fit_keyword_com.pro` and `beam_fit_params_com.pro`.

The routines that control, define, and interact with the common blocks are:
`beam_fit_struc_common.pro`, `beam_fit__get_common.pro`, `beam_fit__set_common.pro`,
`beam_fit_set_defaults.pro`, and `beam_fit_unset_common.pro`.

4.2 Process Outline

The routines start by initializing some basic parameters (e.g. velocity range, phase space density range, number of contour levels, etc.) used by the contour plotting routines. Then the routine asks if the user would like to print the index list, prompting them for an integer associated with a specific DF. Once entered, the routine begins the main program loop and calls the wrapping routine for plotting and fitting. This process goes through the following outline:

1. Plot original DF in velocity frame of reference defined by **VSW** structure tag (1st window: **Entire Distribution [Core Bulk Frame]**)
 - (a) Prompt user to see if they wish to change anything (e.g. bulk flow velocity, range of phase space densities plotted, etc.)
 - (b) If yes, then user changes appropriate parameter and the routines replot the DF after each change
 - (c) If no, a PS file of the plot is created and we move on
2. Prompt the user to use either the cursor or command line to estimate the radius of a velocity circle that would encompass the **core** of the DF
 - (a) The routine will plot the circle on the DF (1st window: **Entire Distribution [Core Bulk Frame]**) before using the circle to create a mask that will remove all data inside that radius
 - (b) If the user is satisfied that the circle encompasses all or enough of the **core**, the mask will be applied leaving only the **halo** (2nd window: **Halo Distribution [Core Bulk Frame]**)
 - (c) If the user is satisfied that the mask adequately removed the **core**, then the user is prompted to change any of the plotting parameters⁹

⁸Found at: http://tetra.space.umn.edu/wiki/doku.php/umn_wind3dp

⁹here it is often wise to change the `zrange` option to make the `beam` easier to observe

- (d) once satisfied, another PS file is created with only the **halo** plotted
3. Prompt the user to use either the cursor or command line to estimate the parallel(horizontal- or X-axis) and perpendicular(vertical- or Y-axis) velocity offset of the center of the **beam** peak (3rd window: **Beam Cuts [Core Bulk Frame]**)
 - (a) The routine will replot the results and then ask if the crosshairs (i.e. red horizontal and blue vertical lines) look to be centered on the **beam** peak
 - (b) If the user is not satisfied, they can try again, otherwise they are prompted to change any of the plotting parameters again prior to saving a PS file
 4. Prompt the user to estimate a velocity radius that would encompass the **beam** (4th window: **Beam Distribution [Beam Bulk Frame]**)
 - (a) use either the cursor or command line to estimate the velocity radius
 - (b) once satisfied, the routine applies a mask removing all the data outside this radius and asks if the mask removed enough of the **halo**
 - (c) the routine transforms into the **beam** frame of reference and asks again whether the user is satisfied with their choice
 - (d) once satisfied, the user is asked to change any of the plotting parameters
 - (e) once finished, the routine produces two PS files showing only the **beam**: one in the **core** bulk frame and one in the **beam** bulk frame
 5. Prompt the user asking whether to perform the fit in the **core** bulk frame or the **beam** bulk frame¹⁰ (4th window: **Beam Distribution [Beam Bulk Frame]**)
 6. Begin fitting process (Levenberg-Marquardt least-squares fit)
 - (a) the routine performs moment analysis on the DF in the frame of reference chosen by the user in the last step
 - (b) the initial model results are calculated from the moment analysis results and shown as dash-dotted lines in the cut plots (bottom panel of 4th window)
 - (c) prompt the user to constrain the model function parameters¹¹, assume user said yes, then:
 - i. ask user if they wish to tie the density (n_b) to the thermal speeds ($V_{T_{b\perp}}$ and $V_{T_{b\parallel}}$) and peak (A_b) of the bi-Maxwellian $\Rightarrow n_b = A_b \pi^{3/2} V_{T_{b\perp}}^2 V_{T_{b\parallel}}$
 - ii. ask user if they wish to limit the range of parallel ($V_{o_{b\parallel}}$) and perpendicular ($V_{o_{b\perp}}$) drift velocity values
 - iii. ask user if they wish to limit the range of values for $V_{T_{b\perp}}$ and $V_{T_{b\parallel}}$
 - (d) routine will fit with user specified constraints, re-plot the data with model fit results in the cut plots (bottom panel of 4th window), and then ask the user the following:
 - i. ask user if they wish to change the minimum percentage of A_b they wish to use for the data used by the fitting routines

¹⁰The **core** bulk frame is more common and affects the **beam** less. This is because the transformation to the **beam** frame can often be inaccurate since we have only specified the **beam** velocity in the **core** bulk frame in one plane. If the **beam** is not well defined in the plane of projection the user chose, then the transformation to the **beam** bulk frame will be result in a skewed contour plot and cuts that do not resemble Gaussian curves.

¹¹user should probably say yes here

- ii. ask if the user wishes to continue to tie the density or if they wish to start tying the density to the amplitude and thermal speeds
 - iii. ask user if they wish to change, constrain, or limit any of the model parameters
 - iv. if any of the above were changed/altered, then the routine re-plots, re-fits, and re-plots and starts the prompting over otherwise a PS file is created and routine returns to main wrapping routine
7. Prompt user to see whether they wish to add the fit results to the output ASCII file
 8. Move onto next DF

5 Prompt Information

This section will provide more details about most of the prompts the user will encounter. As a general rule, informational outputs will be surrounded by:

prompting outputs will be surrounded by:

Often times the routines will inform the user what display window they are working with, which is important for cursor routines. These outputs will be surrounded by:

These prompts are generated by `beam_fit_gen_prompt.pro`, which is called by other wrapping routines. In most cases, there will be information provided as to the type of input expected (i.e. string vs. float) and what the format should be. In many cases, if you enter an incorrect format, the general prompting routine will catch the error and prompt you again. However, incorrect input format can break the code in some places so please pay attention to instructions.

When you are prompted and the last part of the prompt contains (*y/n*), then the only acceptable inputs are **y**, **n**, or **q**. An input of **q** will often result in quitting the current program but can also be used to exit a loop, exit one of the prompting routines, or to stop changing something. Pay attention to both the prompt and the information provided before the prompt for input format information etc.

5.1 Prompt Information: Plotting Parameter Initialization

The first set of prompts are initialized by `beam_fit_keywords_init.pro`, and are as follows:

This prompt asks the user whether they would like to use the default value¹² for the velocity range limit, **VLIM**. Depending on the situation, I type **n** and enter either 25d2 or 15d2 (double precision

¹²derived from maximum energy of input data structures

inputs for 2500 and 1500). If I want to examine something close to the center of the contour plot in the bulk flow frame, I use 1500 km/s. If I want to examine the entire distribution, then I use 2500 km/s¹³. Let's say I type `n` then we get the following:

```
=>=>=>=>=>=>=>=>=>=>=>=>=>=>=>=>=>=>=>=>=>=>=>=>=>=>=>
Enter a value > 32.84 [km/s] and < c [~300,000 km/s]:  15d2
=>=>=>=>=>=>=>=>=>=>=>=>=>=>=>=>=>=>=>=>=>=>=>=>
```

where we enter a floating point or double precision value (I typically use `25d2` or `15d2`).

The next prompt asks the user if they wish to use a different number of contour levels (i.e. the number of lines shown on the contour plots), defined by the variable **NGRID**.

```
=>=>=>=>=>=>=>=>=>=>=>=>=>=>=>=>=>=>=>=>=>=>=>=>=>=>=>
Do you wish to use the default value of NGRID [= 30]? (y/n):  y
=>=>=>=>=>=>=>=>=>=>=>=>=>=>=>=>=>=>=>=>=>=>=>=>=>=>=>=>
```

I recommend using the default value of `30` because including more clutters the plots and less can reduce resolution. The input is an integer value.

The next prompt asks the user about smoothing the contour and the cuts of the DF. My general rule is to not smooth the contours but to smooth the cuts. However, each person will have their own preference. Note, however, that smoothing the contour plots may result in artificially round beam-like features appearing in the contour plots that do not represent that actual data. The prompt is as follows:

```
-|-|-|-|-|-|-|-|-|-|-|-|-|-|-|-|-|-|-|-|-|-|-|-|-|-|-|-|-|-|-|-|-|-|-|-|-|-|-|-|-|-|-|
You will now be asked if you wish to smooth the contour and
cuts of the particle velocity distribution functions (DFs). I
recommend smoothing the cuts but not the contours any more than the
minimum amount designated by SMOOTH. The SM_CUTS keyword determines
if the routines smooth the cuts and the SM_CONT keyword determines
if the routines smooth the contours. The NSMOOTH defines the number
of points to use in SMOOTH [Width parameter in routine] for each.
-|-|-|-|-|-|-|-|-|-|-|-|-|-|-|-|-|-|-|-|-|-|-|-|-|-|-|-|-|-|-|-|-|-|-|-|-|-|-|-|-|
```

```
=>=>=>=>=>=>=>=>=>=>=>=>=>=>=>=>=>=>=>=>=>=>=>=>=>=>=>=>
Do you wish to use the default value of SM_CONT [= FALSE]? (y/n):  y
=>=>=>=>=>=>=>=>=>=>=>=>=>=>=>=>=>=>=>=>=>=>=>=>=>=>=>=>
```

```
=>=>=>=>=>=>=>=>=>=>=>=>=>=>=>=>=>=>=>=>=>=>=>=>=>=>=>=>=>
Do you wish to use the default value of SM_CUTS [= FALSE]? (y/n):  n
=>=>=>=>=>=>=>=>=>=>=>=>=>=>=>=>=>=>=>=>=>=>=>=>=>=>=>=>
```

```
=>=>=>=>=>=>=>=>=>=>=>=>=>=>=>=>=>=>=>=>=>=>=>=>=>=>=>=>=>
Do you wish to smooth the cuts of the DFs (y/n):  y
=>=>=>=>=>=>=>=>=>=>=>=>=>=>=>=>=>=>=>=>=>=>=>=>=>=>=>=>
```

¹³Note that these values only apply to ion detectors that have upper energy limits near ~30 keV.

Note that the information output shown above will be printed between each prompting command, but I did not show that here to save space.

The next prompt asks the user about the number of points to smooth (I recommend 4, but more points can be useful for noisy data) over:

```
=>=>=>=>=>=>=>=>=>=>=>=>=>=>=>=>=>=>=>=>=>=>=>=>=>=>=>
Do you wish to use the default value of NSMOOTH [= 03]? (y/n): n
=>=>=>=>=>=>=>=>=>=>=>=>=>=>=>=>=>=>=>=>=>=>=>=>=>=>=>=>
=>=>=>=>=>=>=>=>=>=>=>=>=>=>=>=>=>=>=>=>=>=>=>=>=>=>=>=>
Enter an integer > 2 and < 10: 4
=>=>=>=>=>=>=>=>=>=>=>=>=>=>=>=>=>=>=>=>=>=>=>=>=>=>=>
```

The next prompt asks the user which plane¹⁴, relative to the quasi-static magnetic field (defined by **MAGF** structure tag) and the bulk flow velocity (defined by **VSW** structure tag), they wish to project contours of constant phase space density onto:

```
-|-|-|-|-|-|-|-|-|-|-|-|-|-|-|-|-|-|-|-|-|-|-|-|-|-|-|-|-|-|-|-|-|-|-|-|-|-|-|-|
You will now be asked which plane you wish to enter project
the particle velocity distribution functions (DFs) onto. If we
assume we have two input vectors, V1 and V2, then the options are:
'xy' : horizontal axis parallel to V1 and plane normal to vector
        defined by (V1 x V2) [DEFAULT]
'xz' : horizontal axis parallel to (V1 x V2) and vertical axis
        parallel to V1
'yz' : horizontal axis defined by (V1 x V2) x V1 and vertical
        axis parallel to (V1 x V2)
-|-|-|-|-|-|-|-|-|-|-|-|-|-|-|-|-|-|-|-|-|-|-|-|-|-|-|-|-|-|-|-|-|-|-|-|
=>=>=>=>=>=>=>=>=>=>=>=>=>=>=>=>=>=>=>=>=>=>=>=>=>=>
Enter 'xy', 'xz', or 'yz' for the desired plane of projection: xy
=>=>=>=>=>=>=>=>=>=>=>=>=>=>=>=>=>=>=>=>=>=>=>=>=>=>
```

The next prompt asks the user to define the range of phase space densities to shown in the contour and cut plots. This input, associated with **DFRA**, defines the minimum and maximum value of the contour levels and the Y-Axis range for the cut plots. The outputs are as follows:

```
-|-|-|-|-|-|-|-|-|-|-|-|-|-|-|-|-|-|-|-|-|-|-|-|-|-|-|-|-|-|-|-|-|-|
You will now be asked whether you wish to enter a plotting
range for the phase (velocity) space density plots. The input
format should be (format = x.xxESee), where x is any integer
0-9, E = e in IDL [= x 10^(See)], S = sign of exponent, and
ee = exponent values. This is different from the inputs for
DFMIN and DFMAR, which correspond to the keywords MIN_VALUE and
MAX_VALUE, respectively, used by PLOT.PRO.
```

¹⁴In nearly all publications, the plane chosen will correspond to the **xy** option.

```

An example input would be: 1.05e-12
-|-|-|-|-|-|-|-|-|-|-|-|-|-|-|-|-|-|-|-|-|-|-|-|-|-|-|-|-|-|-|-|-|-|-|-|-
=>=>=>=>=>=>=>=>=>=>=>=>=>=>=>=>=>=>=>=>=>=>=>=>=>
Do you wish to use the default value of DFRA = 4.666E-14 - 5.498E-06 [cm^(-3) km^(-3) s^(3)]? (y/n): n
=>=>=>=>=>=>=>=>=>=>=>=>=>=>=>=>=>=>=>=>=>=>=>=>=>=>=>

```

Note that if I had selected **y**, the routines would have used **MIN.PRO** and **MAX.PRO** to determine the ranges for the contour and cut plots. However, with this input, I am allowed to define these ranges. Here is an example input/output:

```

=>=>=>=>=>=>=>=>=>=>=>=>=>=>=>=>=>=>=>=>=>=>=>=>=>=>=>=>
Enter a new value for Min. DF plot range (format = x.xxxESee): 1d-13
=>=>=>=>=>=>=>=>=>=>=>=>=>=>=>=>=>=>=>=>=>=>=>=>=>=>=>=>=>
=>=>=>=>=>=>=>=>=>=>=>=>=>=>=>=>=>=>=>=>=>=>=>=>=>=>=>=>=>
Enter a new value for Max. DF plot range (format = x.xxxESee): 5d-6
=>=>=>=>=>=>=>=>=>=>=>=>=>=>=>=>=>=>=>=>=>=>=>=>=>=>=>=>=>=>

```

which corresponds to a plot range of 1×10^{-13} to $5 \times 10^{-8} \text{ cm}^{-3} \text{ km}^{-3} \text{ s}^{+3}$.

The next prompt asks the user to define the lower and upper bounds on the data. These inputs will have similar results to the **MIN_VALUE** and **MAX_VALUE** keywords used by **PLOT.PRO** or other built-in plotting routines. Here is an example input/output:

```

-|-|-|-|-|-|-|-|-|-|-|-|-|-|-|-|-|-|-|-|-|-|-|-|-|-|-|-|-|-|-|-|-|-|-|-|-|-|
You will now be asked whether you wish to enter a default
value for the lower and upper bound to be used in the phase
(velocity) space density plots. The input format should be
(format = x.xxxESee), where x is any integer 0-9, E = e in IDL
[= x 10^(See)], S = sign of exponent, and ee = exponent values.

```

```

An example input would be: 1.05e-12
-|-|-|-|-|-|-|-|-|-|-|-|-|-|-|-|-|-|-|-|-|-|-|-|-|-|-|-|-|-|-|-|-|-|-|-|-|-|-|-
=>=>=>=>=>=>=>=>=>=>=>=>=>=>=>=>=>=>=>=>=>=>=>=>=>=>=>=>
Do you wish to use the default value of DFMIN = 1.000E-18 [cm^(-3) km^(-3) s^(3)]? (y/n): n
=>=>=>=>=>=>=>=>=>=>=>=>=>=>=>=>=>=>=>=>=>=>=>=>=>=>=>=>=>
=>=>=>=>=>=>=>=>=>=>=>=>=>=>=>=>=>=>=>=>=>=>=>=>=>=>=>=>=>
Enter a new value for DFMIN (format = x.xxxESee): 9d-14
=>=>=>=>=>=>=>=>=>=>=>=>=>=>=>=>=>=>=>=>=>=>=>=>=>=>=>=>=>
=>=>=>=>=>=>=>=>=>=>=>=>=>=>=>=>=>=>=>=>=>=>=>=>=>=>=>=>=>
Do you wish to use the default value of DFMAX = 1.000E-18 [cm^(-3) km^(-3) s^(3)]? (y/n): n
=>=>=>=>=>=>=>=>=>=>=>=>=>=>=>=>=>=>=>=>=>=>=>=>=>=>=>=>=>
=>=>=>=>=>=>=>=>=>=>=>=>=>=>=>=>=>=>=>=>=>=>=>=>=>=>=>=>=>
Enter a new value for DFMAX (format = x.xxxESee): 1d-5

```

Note that you want these values to be outside the range of values you chose for **DFRA**, but not too far away. These inputs will prevent cut lines from racing off your cut plots to some unknown values. They are also useful for preventing your contour plots from being overwhelmed by noise.

The next prompt asks the user if they wish to use a default fill-value, which will be used to replace MISSING data points or NaNs. Here is an example input/output:

The next prompt asks the user about the percentage of A_b (the peak amplitude of the beam) to allow the fitting routine to use when iteratively reducing the parameters to the best fit results. I tend to recommend using the default at first and changing the value later, if necessary, when fitting. Here is an example input/output where I choose not to use the default:

where I have chosen to use 2% of A_b .

5.2 Prompt Information: Changing Plotting Parameters

The next prompt asks the user to specify the index of the IDL structure the user wants to examine. Here is an example input/output:

The program will start working on plotting the data and the following input/output will be shown:

At this point, I have chosen to change something and have entered `beam_fit_options.pro`. The routine input/output goes as follows:

```
Enter command (type '?' for help or 'q' to leave) :?  
+++++  
----- WRAPPER_BEAM_FIT_ARRAY COMMANDS -----  
q      quit [Enter at any time to leave program]  
  
zrange    change the DF range of the contour and cut plots  
zfloor     change the Min. DF-value to allow in plots  
zceil      change the Max. DF-value to allow in plots  
vrange     change the velocity range of the plots  
nsmooth    change the number of points to smooth in the plots  
plane      change the plane of projection in contour plots  
sm_con     if set, then smooth [NSMOOTH pts] contour plots  
sm_cut     if set, then smooth [NSMOOTH pts] cut plots  
vbulk      change the bulk flow velocity estimate  
vcmax      change the magnitude/radius of the 'core' velocity circle  
vbmax      change the magnitude/radius of the 'beam' velocity circle  
vbeam      change the X- and Y-component of the 'beam' velocity  
vb_reg     define a rectangular velocity region around 'beam'  
  
zfill      change the DF-value used for MISSING points  
zperc      change the lowest % of the peak DF to use in fits  
  
next       switch to next particle distribution  
prev       switch to previous particle distribution  
index      choose an index for the particle distribution  
-----  
+++++
```

where these options are defined as follows:

1. `q` : under these circumstances, entering this value will result in leaving the current prompting routine but not quitting
2. `zrange` : enter this to change **DFRA**¹⁵
3. `zfloor` : enter this to change **DFMIN**
4. `zceil` : enter this to change **DFMAX**
5. `vrage` : enter this to change **VLIM**
6. `nsmooth` : enter this to change **NSMOOTH**
7. `plane` : enter this to change **PLANE**

¹⁵see Section 5.1 for definition/reference of each of these capital letter names

8. `sm_con` : enter this to change **SM_CONT**
 9. `sm_cut` : enter this to change **SM_CUTS**
 10. `vbulk` : enter this to change **VSW** structure tag
 11. `vcmax` : enter this to define **VCMAX**¹⁶
 12. `vbmax` : enter this to define **VBMAX**¹⁷
 13. `vbeam` : enter this to define { **V_BX**, **V_BY** }
 14. `vb_reg` : *obsolete option*
 15. `zfill` : enter this to change **FILL**
 16. `zperc` : enter this to change **PERC_PK**
 17. `next` : enter this to switch to the next DF (increases the IDL structure array index by one)
 18. `prev` : enter this to switch to the previous DF (decreases the IDL structure array index by one)
 19. `index` : enter this to switch to define the index of the corresponding DF you wish to examine.

At this point, however, the only things you may want to change are `zrange`, `zfloor`, `zceil`, `vrange`, `nsmooth`, `sm_con`, `sm_cut`, or `vbulk`. Change `vbulk` only if the peak of the distribution does not look centered at the origin, otherwise type `q` to leave and then type `n` when it prompts you asking whether you want to change any of these parameters. So we type `q` to leave and then:

Do you wish to change any of the plot ranges or the VSW estimate?

[e.g. VLIM, DFMIN, DFMAX, DFRA, VSW, etc.]

To change any of these type 'y', otherwise type 'n': n

at which point Figure 3 will be produced.

¹⁶velocity radius used to create mask that removes core

¹⁷velocity radius used to create mask that isolates beam

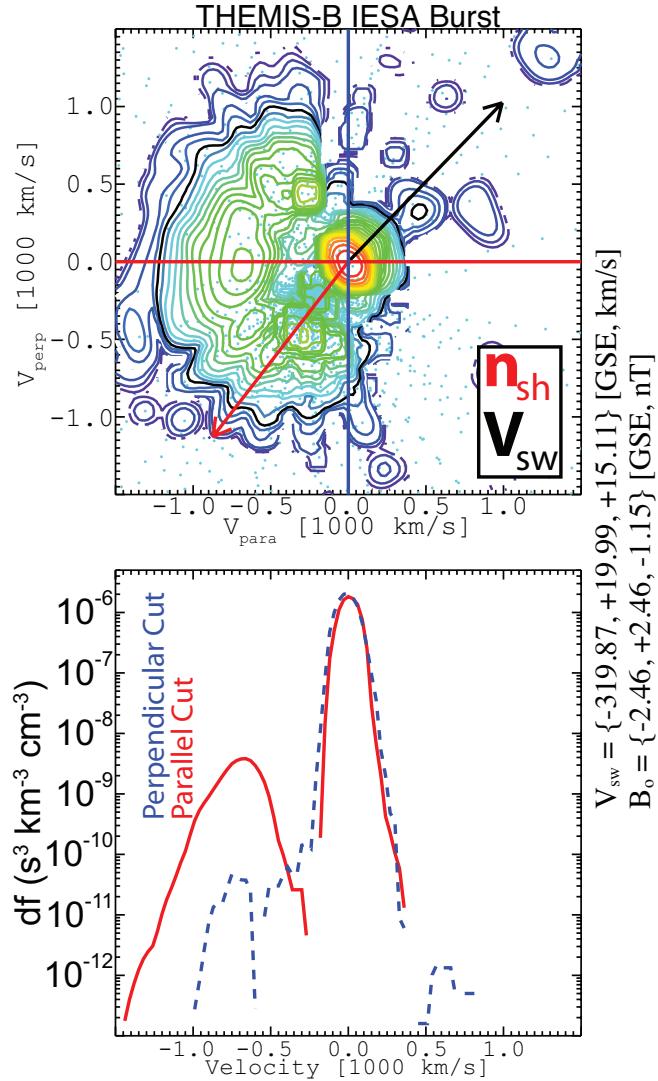


Figure 3: The figure is same DF as that shown in Figure 1, but I have added a shock normal vector (red n_{sh} with associated arrow) and the projection of the bulk flow velocity (black arrow), both with a label box. I have provided my best estimate of \mathbf{V}_{sw} [km/s] and \mathbf{B}_o [nT] to the right of the figure. This is the first output of the routines.

5.3 Prompt Information: Creating Core Mask

The next step involves prompting the user to determine a velocity radius that would encompass the entire **core** for a mask, **VCMAX**. After the mask is applied, the remaining distribution will be only the **halo**. Here is an example input/output:

at which point a circle will be produced onto the current display (1st window: **Entire Distribution [Core Bulk Frame]**) with a radius of 500 km/s. Then the routines will prompt the user for the following:

at which point the region inside the circle will be removed/masked and a new plot (2nd window: **Halo Distribution [Core Bulk Frame]**) with only the **halo** will be shown. The user will then be asked if their estimate was good enough:

¹⁸I do not recommend trying to use the cursor here. You will be asked to select a rectangle and the routine will try to estimate a circular radius from that.

¹⁹ Typing `q` at this point will return the user to `wrapper_beam_fit_array.pro`.

```

<><><><><><><><><><><><><><><><><><><><><><><><><><>
-|-|-|-|-|-|-|-|-|-|-|-|-|-|-|-|-|-|-|-|-|-|-|-|-|-|-|-|-|-|-|-|-|-|-|-|-|-|-|-|
[Type 'q' to quit at any time]
-|-|-|-|-|-|-|-|-|-|-|-|-|-|-|-|-|-|-|-|-|-|-|-|-|-|-|-|-|-|-|-|-|-|-|-|-|-|-|-|
=>=>=>=>=>=>=>=>=>=>=>=>=>=>=>=>=>=>=>=>=>=>=>=>=>=>=>
Did your estimate of Vcmax (500.00 km/s) remove all [or enough] of the core (y/n)? y20
=>=>=>=>=>=>=>=>=>=>=>=>=>=>=>=>=>=>=>=>=>=>=>=>=>=>=>=>=>=>

```

Now the user will be allowed to change some of the plot parameters before saving. Here is an example input/output:

```

<><><><><><><><><><><><><><><><><><><><><><><><><><><>
Working in Window #: 02 [Title: 'Halo Distribution [Core Bulk Frame]']
<><><><><><><><><><><><><><><><><><><><><><><><><><><><><><>
-|-|-|-|-|-|-|-|-|-|-|-|-|-|-|-|-|-|-|-|-|-|-|-|-|-|-|-|-|-|-|-|-|-|-|-|-|-|-|
[Type 'q' to leave this loop at any time]

Do you wish to change any of the plot ranges or the VSW estimate?
[e.g. VLIM, DFMIN, DFMAX, DFRA, VSW, etc.]
-|-|-|-|-|-|-|-|-|-|-|-|-|-|-|-|-|-|-|-|-|-|-|-|-|-|-|-|-|-|-|-|-|-|-|-|-|-|-|
=>=>=>=>=>=>=>=>=>=>=>=>=>=>=>=>=>=>=>=>=>=>=>=>=>=>=>=>
To change any of these type 'y', otherwise type 'n': y
=>=>=>=>=>=>=>=>=>=>=>=>=>=>=>=>=>=>=>=>=>=>=>=>=>=>=>=>=>

```

After removing the **core**, I recommend changing the plot range for the phase space densities, especially if the previous range was dependent upon the **core** range of values. Here is an example input/output:

```

-|-|-|-|-|-|-|-|-|-|-|-|-|-|-|-|-|-|-|-|-|-|-|-|-|-|-|-|-|-|-|-|-|-|-|-|-|-|
You will now be asked whether you wish to enter a plotting
range for the phase (velocity) space density plots. The input
format should be (format = x.xxxESee), where x is any integer
0-9, E = e in IDL [= x 10^(See)], S = sign of exponent, and
ee = exponent values. This is different from the inputs for
DFMIN and DFMAX, which correspond to the keywords MIN_VALUE and
MAX_VALUE, respectively, used by PLOT.PRO.

An example input would be: 1.05e-12
-|-|-|-|-|-|-|-|-|-|-|-|-|-|-|-|-|-|-|-|-|-|-|-|-|-|-|-|-|-|-|-|-|-|-|-|-|-|-|
=>=>=>=>=>=>=>=>=>=>=>=>=>=>=>=>=>=>=>=>=>=>=>=>=>=>=>=>
Do you wish to keep the current value of DFRA = 1.000E-13 - 5.000E-06 [cm^(-3) km^(-3) s^(3)]? (y/n): n

```

²⁰Typing *n* here will allow the user to try estimating **VCMAX** again

```
=>=>=>=>=>=>=>=>=>=>=>=>=>=>=>=>=>=>=>=>=>=>=>=>=>=>=>
```

```
=>=>=>=>=>=>=>=>=>=>=>=>=>=>=>=>=>=>=>=>=>=>=>=>=>=>=>=>
```

```
Enter a new value for Min. DF plot range (format = x.xxxESee):  1d-13
```

```
=>=>=>=>=>=>=>=>=>=>=>=>=>=>=>=>=>=>=>=>=>=>=>=>=>=>=>=>
```

```
=>=>=>=>=>=>=>=>=>=>=>=>=>=>=>=>=>=>=>=>=>=>=>=>=>=>=>=>
```

```
Enter a new value for Max. DF plot range (format = x.xxxESee):  1d-8
```

```
=>=>=>=>=>=>=>=>=>=>=>=>=>=>=>=>=>=>=>=>=>=>=>=>=>=>=>=>
```

Immediately after entering these values, the routines will re-plot the data and ask the user if they approve. Here is an example input/output:

```
<><><><><><><><><><><><><><><><><><><><><><><><><><>
```

```
Working in Window #:  02  [Title:  'Halo Distribution [Core Bulk Frame]']
```

```
<><><><><><><><><><><><><><><><><><><><><><><><><><><><>
```

```
-|-|-|-|-|-|-|-|-|-|-|-|-|-|-|-|-|-|-|-|-|-|-|-|-|-|-|-|-|-|-|-|-|-|-|-|-|-|-|-|-|-|-|
```

```
[Type 'q' to leave this loop at any time]
```

```
Do you wish to keep this change?
```

```
-|-|-|-|-|-|-|-|-|-|-|-|-|-|-|-|-|-|-|-|-|-|-|-|-|-|-|-|-|-|-|-|-|-|-|-|-|-|-|-|-|-|
```

```
=>=>=>=>=>=>=>=>=>=>=>=>=>=>=>=>=>=>=>=>=>=>=>=>=>=>=>=>
```

```
To keep change type 'y', otherwise type 'n':  y
```

```
=>=>=>=>=>=>=>=>=>=>=>=>=>=>=>=>=>=>=>=>=>=>=>=>=>=>=>=>
```

If I had typed *n*, I would have been sent back to the prompt shown above that asks whether I want to change anything. Now we are given that prompt again, allowing us to change something else, but let us type *n* and move on. At this point Figure 4 is created.

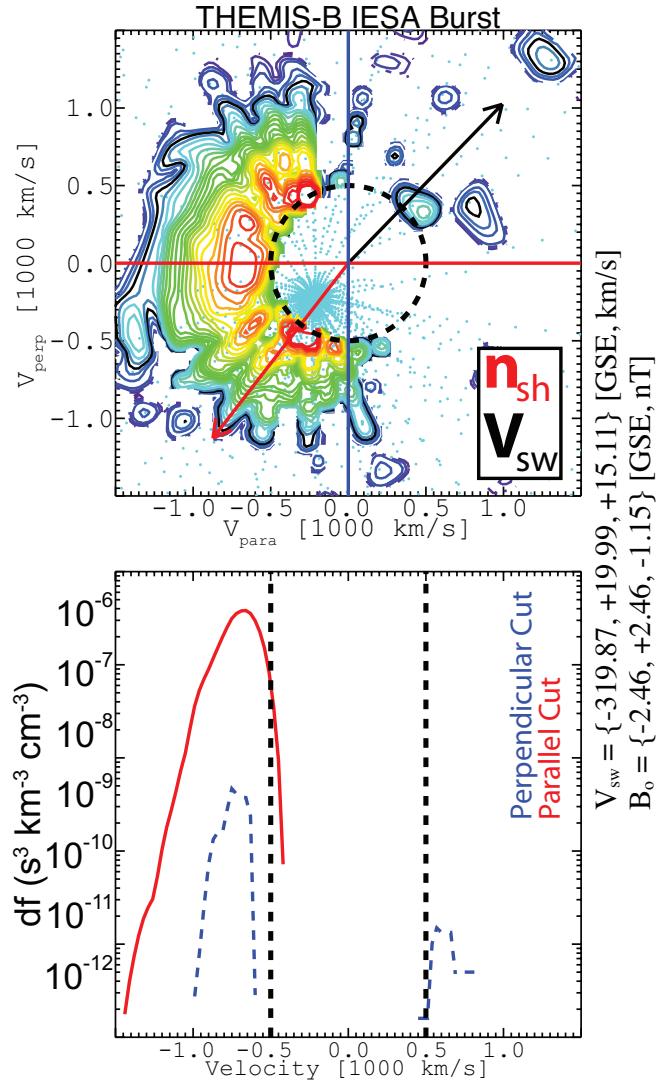


Figure 4: The figure is same DF as that shown in Figures 1 and 3, but the core has been removed/masked and the plot range has changed.

5.4 Prompt Information: Finding the Beam Peak

The next step involves finding the center of the **beam** peak, { **V_BX**, **V_BY** }. The user can use either the cursor (click a single point) or they can guess by entering values on the command line. In some cases I have found it useful to get an initial guess with the cursor then refine the guess with the command line. It all depends how close you wish to be at this stage because these estimates will change at the fitting stage later on. Here is an example input/output:

You can define the values of {V_BX, V_BY} using either the command line input (format = XXXX.xxx) or your mouse cursor.

Do you wish to use the cursor to estimate {V_BX, V_BY} (y/n): y

Working in Window #: 03 [Title: 'Beam Cuts [Core Bulk Frame]']

Select a point on the contour plot at the center of the 'beam' peak region of the DF. The routine will ask you if you like your estimate and if not, allow you to try again. The routine will then use the two values to estimate velocity components in the bulk flow frame. This will then be converted to GSE coordinates to be used as the 'beam' velocity in the spacecraft frame of reference. Note that you should try to select points as close as possible to the center of the 'beam' peak for the best results.

Working in Window #: 03 [Title: 'Beam Cuts [Core Bulk Frame]']

Please use the cursor to select a single point that identifies the point of interest. Make sure you press the mouse button only once and do not click on the window to try and move it prior to selecting this set of coordinates.

at which point the user will click one point on the contour plot (3rd window: **Beam Cuts [Core Bulk Frame]**). Note that you must not try to move the display window or click on another display window that is not currently set. The routines automatically force the

current window to the front and the window outputs inform you which window you should be interacting with. Once you click on a point in the contour plot, the following input/output will occur:

You selected the following point (x,y) in DATA coordinates: (-682.45, -30.000)

Then, the routines will re-plot the output (3rd window: **Beam Cuts [Core Bulk Frame]**) with the crosshairs centered on the point corresponding to your choice of { **V_BX**, **V_BY** }. Then the routines will check to see if you are satisfied with the results. Here is an example input/output:

Your estimate for V_BX is: -682.45 km/s

Do the crosshairs look centered on the peak of the 'beam' (y/n)? **y**²²

Once the user has selected a point they are satisfied with, the routine will ask whether the user wants to change any of the plot parameters from Section 5.2. Rarely have I found this necessary at this point, so type **n** and Figure 5 is created.

²¹At this point, typing **n** will return you to try using the cursor again.

²²At this point, typing **n** will return you to the first prompt of this section.

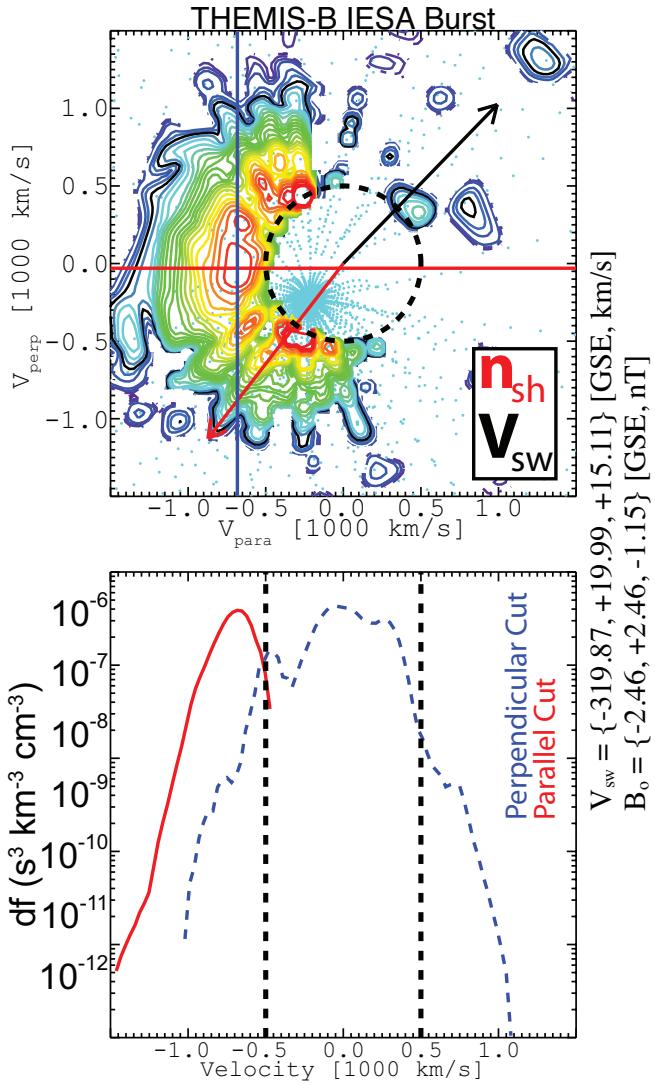


Figure 5: The figure is same DF as that shown in Figure 4, but the crosshairs are now centered on the beam peak. Note that the cuts (bottom panel) of the DF have changed accordingly.

5.5 Prompt Information: Creating Beam Mask

The next step involves prompting the user to determine a velocity radius, **VBMAX**, that would encompass the entire **beam** for a mask to remove everything but the **halo**. After the mask is applied, the remaining distribution will be only the **beam**. Here is an example input/output:

```
-|-|-|-|-|-|-|-|-|-|-|-|-|-|-|-|-|-|-|-|-|-|-|-|-|-|-|-|-|-|-|-|-|-|
```

You will now be asked whether you wish to enter a single velocity radius of a circle that would encompass the entire 'beam' or to use the mouse cursor to define a region that would encompass the entire 'beam'. The circle will be centered on the crosshairs [at (-682.45, -30.00) km/s] you determined in the previous step.

Thus, if you choose the 2nd option, the velocity radius of the circle will be found from the average distance between the four corners selected and the center.

```
-|-|-|-|-|-|-|-|-|-|-|-|-|-|-|-|-|-|-|-|-|-|-|-|-|-|-|-|-|-|-|
```

```
-|-|-|-|-|-|-|-|-|-|-|-|-|-|-|-|-|-|-|-|-|-|-|-|-|-|-|-|-|-|
```

You can define the value of VBMAX using either the command line input (format = XXXX.xxx) or your mouse cursor.

```
-|-|-|-|-|-|-|-|-|-|-|-|-|-|-|-|-|-|-|-|-|-|-|-|-|-|-|-|-|-|
```

```
=>=>=>=>=>=>=>=>=>=>=>=>=>=>=>=>=>=>=>=>=>=>=>
```

To use the cursor to estimate VBMAX type 'c', otherwise type 'r': *r*²³

```
=>=>=>=>=>=>=>=>=>=>=>=>=>=>=>=>=>=>=>=>=>=>=>=>
```

At this point I should comment that the current method for finding the region encompassing the **beam** will leave a slight excess outside of the circle. You can see this in Figures 6a and 6b where the contours exist outside of the black dashed circle. Now the user will be prompted to input a new value for **VBMAX**. Here is an example input/output:

```
-|-|-|-|-|-|-|-|-|-|-|-|-|-|-|-|-|-|-|-|-|-|-|-|-|-|-|-|-|-|
```

Enter a value for VBMAX that is larger than 260.46 km/s

and less than 75% of 2170.53 (or 1627.90) km/s.

Note: The remaining contours of the end result will be outside the circle projected onto the contour plot due to rotation, translation, and averaging technique used by the beam masking routine. However, I recommend choosing a value that is slightly larger than the beam for more points to use in the fitting routine.

```
-|-|-|-|-|-|-|-|-|-|-|-|-|-|-|-|-|-|-|-|-|-|-|-|-|-|-|-|-|
```

```
=>=>=>=>=>=>=>=>=>=>=>=>=>=>=>=>=>=>=>=>=>=>
```

²³Similar to **VCMAX**, I do not recommend using the cursor here. It is much faster to just enter estimates by hand. You will have multiple opportunities to change your guess.

Your estimate for VBMAX is: 425.00 km/s

The routine will then re-plot (4th window: **Beam Distribution [Beam Bulk Frame]**) the DF including the **halo** with a new dashed circle centered at { **V_BX**, **V_BY** } with a radius corresponding to your estimate of **VBMAX**. Then, you will be prompted to verify that the circle encompasses enough of the beam. Here is an example input/output:

Now routine will isolate the **beam** (removing the excess **halo**) and then re-plot (4th window: **Beam Distribution [Beam Bulk Frame]**) the DF in the **core** rest frame with the crosshairs through the **beam** peak. You will be asked again if you are satisfied with your estimate of **VBMAX**. Here is an example input/output:

The routine will then transform into the **beam** rest frame and re-plot (4th window: **Beam Distribution [Beam Bulk Frame]**) the DF with crosshairs and dashed circle centered on the origin. Then the routine will ask if you are still pleased with your results²⁴. Here is an example in-

²⁴Note that this is the last place to change your estimate of **VBMAX** prior to plotting.

put/output:

At this point, you are given the same prompt that asks whether you want to change any of the plot parameters from Section 5.2. Again, rarely do I find this to be necessary. If you completely screwed up your estimate of **VBMAX** or { **V_BX**, **V_BY** }, you can change it here but the routine will not (currently) re-plot the DFs already created with your new estimates. After typing **n**, Figures 6a and 6b will be created.

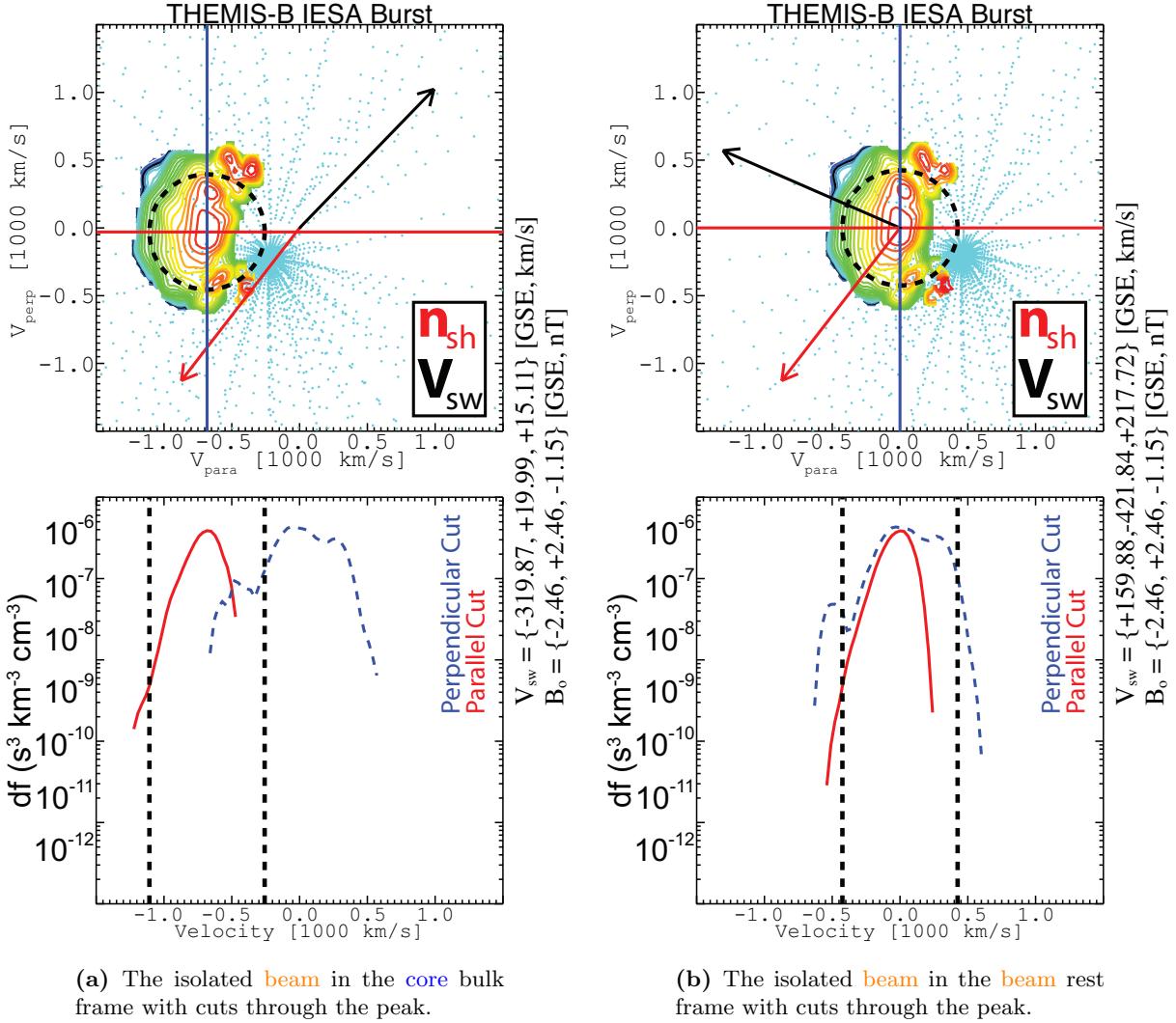


Figure 6: The isolated beam shown in two different rest frames.

5.6 Prompt Information: Fitting to the Beam Peak

The next step involves fitting the **beam** peak to a bi-Maxwellian velocity distribution function (see Section C). The routines use the Levenberg-Marquardt least-squares fit method to determine the open variables used in the fitting process. Because these DFs often have multiple peaks, non-Maxwellian features, non-gyrotropic asymmetries, etc. the wrapping routine (`beam_fit_wrapper.pro`) for the main fit routine (`df_fit_beam_peak.pro`) allows the user to constrain/limit the initial guesses produced by moment analysis (using `moments_3d_new.pro`²⁵) prior to performing the first iteration of the least-squares fit. Since the plane of projection may not contain the actual peak of the **beam**, the routine asks whether the user wishes to perform the analysis in the **core** or **beam** rest frames. Here is an example input/output:

```
-|-|-|-|-|-|-|-|-|-|-|-|-|-|-|-|-|-|-|-|-|-|-|-|-|-|-|-|-|-|-|-|-|-|-|-|-|-|  
You will now be asked whether you want to fit the beam peak in the  
'beam' bulk frame or 'core' bulk frame. The results should not  
differ drastically in regards to density and thermal speeds, but  
the fitting routine does appear to have less trouble when in the  
'beam' bulk frame.26
```

```
[Type 'q' to quit at any time]  
-|-|-|-|-|-|-|-|-|-|-|-|-|-|-|-|-|-|-|-|-|-|-|-|-|-|-|-|-|-|-|-|-|-|-|  
  
=>=>=>=>=>=>=>=>=>=>=>=>=>=>=>=>=>=>=>=>=>=>=>=>  
To fit in the 'core' bulk frame type 'c', otherwise type 'b': c  
=>=>=>=>=>=>=>=>=>=>=>=>=>=>=>=>=>=>=>=>=>=>=>=>=>=>
```

Now the routine will transform the DF into the **core** rest frame and perform moment analysis on the distribution that contains only the **beam**. The following will be printed to the screen and the routine will re-plot the results now with bi-Maxwellian model cuts from the moment analysis:

```
-|-|-|-|-|-|-|-|-|-|-|-|-|-|-|-|-|-|-|-|-|-|-|-|-|-|-|-|-|-|-|-|-|  
The initial Moment Analysis results are:
```

Beam Density [cm ⁻³]	=	2.6862
Beam Para. Thermal Speed [km/s]	=	236.4142
Beam Perp. Thermal Speed [km/s]	=	226.9503
Beam Para. Drift Speed [km/s]	=	-682.4457
Beam Perp. Drift Speed [km/s]	=	-30.0000
Beam Temp. Anisotropy [Tperp/Tpara]	=	0.0000

Do the results shown in the plot look good or is
further analysis necessary?

```
-|-|-|-|-|-|-|-|-|-|-|-|-|-|-|-|-|-|-|-|-|-|-|-|-|-|-|-|-|-|-|-|
```

²⁵This is an adaptation of `moments_3d.pro` and `moments_3du.pro` with more comments and a more detailed manual page.

²⁶This comment is outdated. There was a bug in `beam_fit_wrapper.pro` that caused this last statement to be true, but it is not the case now. If the projected plane is close to the center of the **beam** peak, the results in the two frames can be very similar.

In most cases, you will want to constrain the parameters used to construct the bi-Maxwellian model function. The parameters used to create the model bi-Maxwellian and, consequently, the parameters which are iteratively modified by the least-squares fitting routine are defined by the variable **PARAM**. **PARAM** is a six element array with the following values/definitions:

1. $\text{PARAM}[0]$ = Number density [Scalar float, cm^{-3}]
2. $\text{PARAM}[1]$ = Parallel thermal speed [Scalar float, km/s]
3. $\text{PARAM}[2]$ = Perpendicular thermal speed [Scalar float, km/s]
4. $\text{PARAM}[3]$ = Parallel drift speed [Scalar float, km/s]
5. $\text{PARAM}[4]$ = Perpendicular drift speed [Scalar float, km/s]
6. $\text{PARAM}[5]$ = *** Not Used Here ***

The routine that performs the Levenberg-Marquardt least-squares fit is `mpfit2dfun.pro`²⁷. It has a keyword, **PARINFO**, which is an array of structures (same number of elements as **PARAM**) that allows the user to limit, constrain, fix, or tie any value on input. The used structure tags for the j^{th} element of **PARINFO** with associated definitions are:

1. $\text{PARINFO}[j].\text{VALUE}$ = scalar [float] value of j^{th} parameter (should be equal to $\text{PARAM}[j]$)
2. $\text{PARINFO}[j].\text{FIXED}$ = scalar [byte] telling `mpfit2dfun.pro` to vary (FALSE) or not vary (TRUE) $\text{PARAM}[j]$
3. $\text{PARINFO}[j].\text{LIMITED}$ = two element array [byte] telling `mpfit2dfun.pro` whether the values in $\text{PARINFO}[j].\text{LIMITS}$ should define the lower(1st element) or upper(2nd element) bounds for the fit result associated with $\text{PARAM}[j]$
4. $\text{PARINFO}[j].\text{LIMITS}$ = two element array [float] telling `mpfit2dfun.pro` what the range of possible values can be for the fit result associated with $\text{PARAM}[j]$
5. $\text{PARINFO}[j].\text{TIED}$ = scalar [string] telling `mpfit2dfun.pro` how $\text{PARAM}[j]$ relates to $\text{PARAM}[k]$ and/or $\text{PARAM}[i]$, etc.

Note that the only way these routines will allow the user to use the TIED tag is by constraining the density to peak amplitude of the cuts and the thermal speeds. The user may alter any of the other structure tags (done through `beam_fit_fit_prompts.pro`) as they wish. Below, we show an example input/output interaction that led to the results shown in Figure 7:

```
=>=>=>=>=>=>=>=>=>=>=>=>=>=>=>=>=>=>=>=>=>=>=>=>=>=>=>
Do you wish to constrain/limit any of these parameters before fitting (y/n)?  y
=>=>=>=>=>=>=>=>=>=>=>=>=>=>=>=>=>=>=>=>=>=>=>=>=>=>=>=>=>=>
-|-|-|-|-|-|-|-|-|-|-|-|-|-|-|-|-|-|-|-|-|-|-|-|-|-|-|-|-|-|-|-|-|-|-|-|-|-|-|-|
*** ONLY say yes to the following if there are no spurious peaks ***
[Type 'q' to quit at any time]
```

²⁷Written by Craig B. Markwardt (e-mail: craigmlheamail.gsfc.nasa.gov), where updated versions can be found at <http://cow.physics.wisc.edu/~craigm/idl/idl.html>.

```

-|-|-|-|-|-|-|-|-|-|-|-|-|-|-|-|-|-|-|-|-|-|-|-|-|-|-|-|-|-|-|-|-|-|-|-|-|-|-|-|-
=>=>=>=>=>=>=>=>=>=>=>=>=>=>=>=>=>=>=>=>=>=>=>=>=>=>=>
Do you wish to tie the density to the thermal speeds and peak (y/n)? y
=>=>=>=>=>=>=>=>=>=>=>=>=>=>=>=>=>=>=>=>=>=>=>=>=>=>=>=>=>=>=>

-|-|-|-|-|-|-|-|-|-|-|-|-|-|-|-|-|-|-|-|-|-|-|-|-|-|-|-|-|-|-|-|-|-|-|-|-|-|-|-|
*** Say no to the following unless the beam is not well isolated ***

[Type 'q' to quit at any time]
-|-|-|-|-|-|-|-|-|-|-|-|-|-|-|-|-|-|-|-|-|-|-|-|-|-|-|-|-|-|-|-|-|-|-|-|-|-|-|-|-
=>=>=>=>=>=>=>=>=>=>=>=>=>=>=>=>=>=>=>=>=>=>=>=>=>=>=>=>
Do you wish to limit the parallel drift speed (y/n)? n
=>=>=>=>=>=>=>=>=>=>=>=>=>=>=>=>=>=>=>=>=>=>=>=>=>=>=>=>=>=>=>

-|-|-|-|-|-|-|-|-|-|-|-|-|-|-|-|-|-|-|-|-|-|-|-|-|-|-|-|-|-|-|-|-|-|-|-|-|-|-|-|
*** Say no to the following unless the beam is not well isolated ***

[Type 'q' to quit at any time]
-|-|-|-|-|-|-|-|-|-|-|-|-|-|-|-|-|-|-|-|-|-|-|-|-|-|-|-|-|-|-|-|-|-|-|-|-|-|-|-
=>=>=>=>=>=>=>=>=>=>=>=>=>=>=>=>=>=>=>=>=>=>=>=>=>=>=>=>
Do you wish to limit the perpendicular drift speed (y/n)? n
=>=>=>=>=>=>=>=>=>=>=>=>=>=>=>=>=>=>=>=>=>=>=>=>=>=>=>=>=>=>

-|-|-|-|-|-|-|-|-|-|-|-|-|-|-|-|-|-|-|-|-|-|-|-|-|-|-|-|-|-|-|-|-|-|-|-|-|-|-|-|
*** Say no to the following unless the model lines are not ***
*** close to the actual data points in the cut plots ***

[Type 'q' to quit at any time]
-|-|-|-|-|-|-|-|-|-|-|-|-|-|-|-|-|-|-|-|-|-|-|-|-|-|-|-|-|-|-|-|-|-|-|-|-|-|-
=>=>=>=>=>=>=>=>=>=>=>=>=>=>=>=>=>=>=>=>=>=>=>=>=>=>=>=>
Do you wish to limit the parallel thermal speed (y/n)? n
=>=>=>=>=>=>=>=>=>=>=>=>=>=>=>=>=>=>=>=>=>=>=>=>=>=>=>=>=>=>

-|-|-|-|-|-|-|-|-|-|-|-|-|-|-|-|-|-|-|-|-|-|-|-|-|-|-|-|-|-|-|-|-|-|-|-|-|-|-|
*** Say no to the following unless the model lines are not ***
*** close to the actual data points in the cut plots ***

[Type 'q' to quit at any time]
-|-|-|-|-|-|-|-|-|-|-|-|-|-|-|-|-|-|-|-|-|-|-|-|-|-|-|-|-|-|-|-|-|-|-|-|-|-|-
=>=>=>=>=>=>=>=>=>=>=>=>=>=>=>=>=>=>=>=>=>=>=>=>=>=>=>=>
Do you wish to limit the perpendicular thermal speed (y/n)? n
=>=>=>=>=>=>=>=>=>=>=>=>=>=>=>=>=>=>=>=>=>=>=>=>=>=>=>=>=>
```

The next output is a re-plot of the model fit lines and two sets of fit results. The first set of fit results are output because the routine needs to use the parallel and perpendicular cut outputs to define the string used by PARINFO[0].TIED (i.e. to tie the density). The second set of fit results are the first real guess. Here are the two sets of results:

```
Beam Density [cm^(-3)]      =      0.42941755  +/-      0.0000000
Beam Para. Thermal Speed [km/s] =      171.70005  +/-      3.1233063e+08
Beam Perp. Thermal Speed [km/s] =      321.54174  +/-      6.0790646e+08
Beam Para. Drift Speed [km/s]   =     -672.73336  +/-      2.8878629e+08
Beam Perp. Drift Speed [km/s]   =      36.242618  +/-      5.4318718e+08
Beam Temp. Anisotropy [Tperp/Tpara] =      3.5069846
```

```
Model Fit Status          =      1
Number of Iterations      =      22
Degrees of Freedom        =      10196
Chi-Squared                =      1.8098177e-13
Reduced Chi-Squared       =      1.7750272e-17
```

```
Beam Density [cm^(-3)]      =      0.43788690  +/-      0.0000000
Beam Para. Thermal Speed [km/s] =      173.85046  +/-      3.1551299e+08
Beam Perp. Thermal Speed [km/s] =      326.87436  +/-      6.1702538e+08
Beam Para. Drift Speed [km/s]   =     -672.28271  +/-      2.9423215e+08
Beam Perp. Drift Speed [km/s]   =      36.146139  +/-      5.5591688e+08
Beam Temp. Anisotropy [Tperp/Tpara] =      3.5351677
```

```
Model Fit Status          =      1
Number of Iterations      =      22
Degrees of Freedom        =      10196
Chi-Squared                =      1.7817311e-13
Reduced Chi-Squared       =      1.7474805e-17
```

The following is the iterative process of using the limit and constraining options for **PARINFO**. Here is the output example:

```
-|-|-|-|-|-|-|-|-|-|-|-|-|-|-|-|-|-|-|-|-|-|-|-|-|-|-|-|-|-|-|-|-|-|-|
[Type 'q' to quit at any time]
```

```
The peak beam amplitude is A_b = 3.89E-09 [cm^(-3) km^(-3) s^(+3)].
You are currently using values > 1.0% of the peak beam amplitude.
```

```
Do the model fits suggest that low amplitude points have skewed the results?
-|-|-|-|-|-|-|-|-|-|-|-|-|-|-|-|-|-|-|-|-|-|-|-|-|-|-|-|-|-|-|-|
```

```
=>=>=>=>=>=>=>=>=>=>=>=>=>=>=>=>=>=>=>=>=>=>=>=>=>
Do you wish to try again with using a higher/lower percentage (y/n)? y
=>=>=>=>=>=>=>=>=>=>=>=>=>=>=>=>=>=>=>=>=>=>=>=>=>
```

```
=>=>=>=>=>=>=>=>=>=>=>=>=>=>=>=>=>=>=>=>=>=>=>=>  
Enter a percentage in decimal form (e.g. 1% = 0.01) less than 0.90: 15d-2  
=>=>=>=>=>=>=>=>=>=>=>=>=>=>=>=>=>=>=>=>=>=>=>=>=>  
  
-|-|-|-|-|-|-|-|-|-|-|-|-|-|-|-|-|-|-|-|-|-|-|-|-|-|-|-|-|-|-|-|-|-|-|-|-|  
[Type 'q' to quit at any time]  
-|-|-|-|-|-|-|-|-|-|-|-|-|-|-|-|-|-|-|-|-|-|-|-|-|-|-|-|-|-|-|-|-|  
  
=>=>=>=>=>=>=>=>=>=>=>=>=>=>=>=>=>=>=>=>=>=>=>=>=>  
Do you wish to change the TIED setting (y/n)? n  
=>=>=>=>=>=>=>=>=>=>=>=>=>=>=>=>=>=>=>=>=>=>=>=>=>=>=>  
  
-|-|-|-|-|-|-|-|-|-|-|-|-|-|-|-|-|-|-|-|-|-|-|-|-|-|-|-|-|-|-|-|-|-|-|  
[Type 'q' to quit at any time]  
-|-|-|-|-|-|-|-|-|-|-|-|-|-|-|-|-|-|-|-|-|-|-|-|-|-|-|-|-|-|-|  
  
=>=>=>=>=>=>=>=>=>=>=>=>=>=>=>=>=>=>=>=>=>=>=>=>=>=>  
Do you wish to change any of the input parameters, constraints, limits, etc. (y/n)? y  
=>=>=>=>=>=>=>=>=>=>=>=>=>=>=>=>=>=>=>=>=>=>=>=>=>=>=>=>
```

At this point we enter the routine `beam_fit_fit_prompts.pro`, which allows us to alter the tag values for **PARINFO**. Here is the output example:

```
Enter command (type '?' for help or 'q' to leave) :limit
```

```
Enter a value between 0-4 corresponding to the following  
definitions of the indices for PARAM:
```

```
0 = Number Density [cm^(-3)]  
1 = Parallel Thermal Speed [km/s]  
2 = Perpendicular Thermal Speed [km/s]  
3 = Parallel Drift Speed [km/s]  
4 = Perpendicular Drift Speed [km/s]  
5 = *** Not Used Here ***
```

```
=>=>=>=>=>=>=>=>=>=>=>=>=>=>=>=>=>=>=>=>=>=>=>  
Enter the index associated with the parameter defined above to impose limits: 3  
<==<=<=<=<=<=<=<=<=<=<=<=<=<=<=<=<=<=<=<=<=<=<=<=<=<=<
```

```
-|-|-|-|-|-|-|-|-|-|-|-|-|-|-|-|-|-|-|-|-|-|-|-|-|-|-|-|-|-|-|-|-|-|-|  
The current setting for PARINFO.VALUE = -682.446 [km/s]  
The current limit range is = 0.000-0.000 [km/s]
```

```
PARAM[3] is not currently limited...  
-|-|-|-|-|-|-|-|-|-|-|-|-|-|-|-|-|-|-|-|-|-|-|-|-|-|-|-|-|-|-|-|-|-|
```

```
=>=>=>=>=>=>=>=>=>=>=>=>=>=>=>=>=>=>=>=>=>=>=>=>  
Do you wish to impose limits on lower bound for V_BX (para. drift speed) (y/n)?  
<=<<=<=<=<=<=<=<=<=<=<=<=<=<=<=<=<=<=<=<=<=<=<=<=<=<
```

```
=>=>=>=>=>=>=>=>=>=>=>=>=>=>=>=>=>=>=>=>=>=>=>=>=>=>  
Do you wish to impose limits on upper bound for V_BX (para. drift speed) (y/n)?  
<=<<=<=<=<=<=<=<=<=<=<=<=<=<=<=<=<=<=<=<=<=<=<=<=<=<
```

```
=>=>=>=>=>=>=>=>=>=>=>=>=>=>=>=>=>=>=>=>=>=>=>=>=>=>=>  
Enter a lower bound for V_BX [km/s] (format = XXXX.xxx): -720d0  
<=<<=<=<=<=<=<=<=<=<=<=<=<=<=<=<=<=<=<=<=<=<=<=<=<=<=<
```

```
=>=>=>=>=>=>=>=>=>=>=>=>=>=>=>=>=>=>=>=>=>=>=>=>=>=>=>  
Enter a upper bound for V_BX [km/s] (format = XXXX.xxx): -690d0  
<=<<=<=<=<=<=<=<=<=<=<=<=<=<=<=<=<=<=<=<=<=<=<=<=<=<=<
```

```
Enter command (type '?' for help or 'q' to leave) :limit
```

```
Enter a value between 0-4 corresponding to the following  
definitions of the indices for PARAM:
```

```
0 = Number Density [cm^(-3)]  
1 = Parallel Thermal Speed [km/s]  
2 = Perpendicular Thermal Speed [km/s]  
3 = Parallel Drift Speed [km/s]  
4 = Perpendicular Drift Speed [km/s]  
5 = *** Not Used Here ***
```

```
=>=>=>=>=>=>=>=>=>=>=>=>=>=>=>=>=>=>=>=>=>=>=>=>=>  
Enter the index associated with the parameter defined above to impose limits: 4  
<=<<=<=<=<=<=<=<=<=<=<=<=<=<=<=<=<=<=<=<=<=<=<=<=<=<
```

```
-|-|-|-|-|-|-|-|-|-|-|-|-|-|-|-|-|-|-|-|-|-|-|-|-|-|-|-|-|-|-|-|-|-|-|-|-|-|  
The current setting for PARINFO.VALUE = -30.000 [km/s]  
The current limit range is = 0.000-0.000 [km/s]
```

```
PARAM[4] is not currently limited...  
-|-|-|-|-|-|-|-|-|-|-|-|-|-|-|-|-|-|-|-|-|-|-|-|-|-|-|-|-|-|-|-|-|-|-|-|
```

```
=>=>=>=>=>=>=>=>=>=>=>=>=>=>=>=>=>=>=>=>=>=>=>=>=>=>  
Do you wish to impose limits on lower bound for V_BY (perp. drift speed) (y/n)?  
<=<<=<=<=<=<=<=<=<=<=<=<=<=<=<=<=<=<=<=<=<=<=<=<
```

Enter a upper bound for V_BY [km/s] (format = XXXX.xxx): **1d1**

Enter command (type '?' for help or 'q' to leave) :q

Note that I went through a couple of iterations of trying different limits for the parallel and perpendicular drift speeds and the percentage of the peak phase space density prior to arriving at these results:

Beam Density [cm ⁻³]	=	0.23143641	+/-	0.0000000
Beam Para. Thermal Speed [km/s]	=	141.44332	+/-	2.6556252e+08
Beam Perp. Thermal Speed [km/s]	=	259.24014	+/-	5.0144619e+08
Beam Para. Drift Speed [km/s]	=	-690.00000	+/-	-0.0000000
Beam Perp. Drift Speed [km/s]	=	10.000000	+/-	-0.0000000
Beam Temp. Anisotropy [Tperp/Tpara]	=	3.3592290		

Model	=	1
Fit	=	14
Status	=	10196
Number of Iterations	=	2.0089488e-13
Degrees of Freedom	=	1.9703304e-17
Chi-Squared	=	
Reduced Chi-Squared	=	

Beam Density [cm ⁻³]	=	0.23121946	+/-	0.0000000
Beam Para. Thermal Speed [km/s]	=	142.31730	+/-	2.6779102e+08
Beam Perp. Thermal Speed [km/s]	=	261.17159	+/-	5.0619395e+08
Beam Para. Drift Speed [km/s]	=	-690.00000	+/-	-0.0000000
Beam Perp. Drift Speed [km/s]	=	10.000000	+/-	-0.0000000
Beam Temp. Anisotropy [Tperp/Tpara]	=	3.3677237		

Model	Fit	Status	=	1
Number of Iterations	=	16		
Degrees of Freedom	=	10196		
Chi-Squared	=	2.0004703e-13		
Reduced Chi-Squared	=	1.9620148e-17		

After you are satisfied with your fit results, input answers to the prompts that result in no changes and eventually you will exit the program and the routines will produce Figure 7. Depending

on how well you isolate the beam, the fewer the spurious peaks near the beam, and the ellipticity of the beam peak contours, the number of constraints/limits will you need to impose will change.

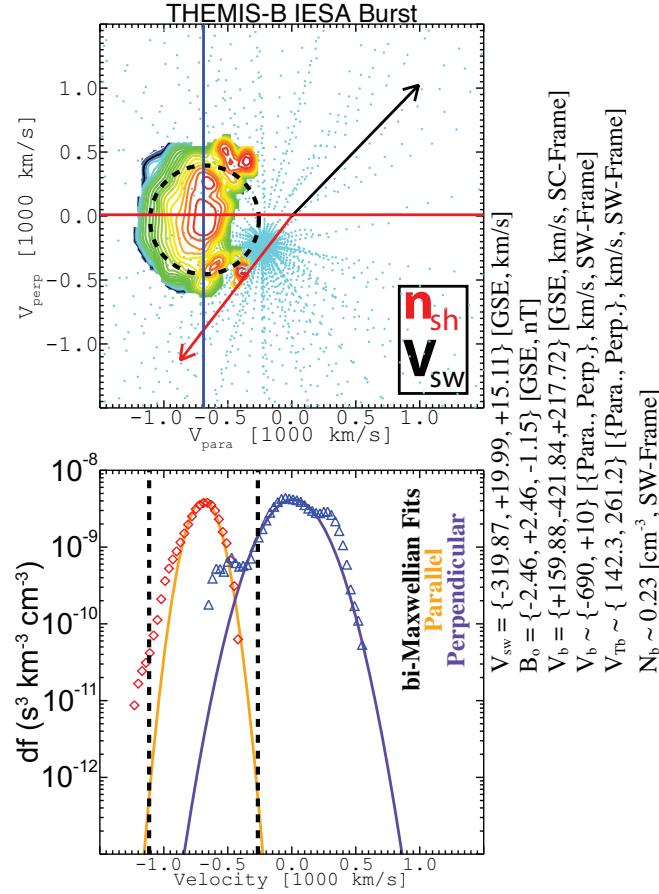


Figure 7: The figure is same DF as that shown in Figure 6a, except here we included the model fit cut plots and results.

6 Future Changes/Additions

The following list shows things I hope to change, barring time, in the future:

1. allow masks of the `core` and `beam` to be elliptical instead of limited to circular
2. allow user to choose whether they wish to use a bi-Maxwellian versus a bi-kappa distribution
3. allow crosshairs to rotate by a user specified angle in cases where `beam` feature semi-major axis is not aligned with a plot axis
4. when using the cursor to find the *correct* bulk flow velocity, allow user to use all three planes simultaneously to assure correct offsets

A Particle Data Structures in IDL

Full 3-dimensional particle distributions from the THEMIS ESA (or Wind 3DP) instruments come as data structures in the TDAS (Wind 3DP) software [McFadden *et al.*, 2008a]. The list of structure tags includes (but is not limited to), in no particular order:

1. **PROJECT_NAME** \equiv scalar [string] *e.g.* 'THEMIS-B'
2. **SPACECRAFT** \equiv scalar [string] *e.g.* 'b'
3. **DATA_NAME** \equiv scalar [string] *e.g.* 'IESA 3D burst'²⁸
4. **UNITS_NAME** \equiv scalar [string] *e.g.* 'counts'²⁹
5. **UNITS PROCEDURE** \equiv scalar [string] (*e.g.* 'thm_convert_esa_units') that tells `conv_units.pro` which IDL routine to use to convert the data units
6. **TIME** \equiv scalar [double] defining the Unix³⁰ time associated with start of data sample (might be a slight delay from the sun pulse timestamp)
7. **END_TIME** \equiv scalar [double] defining the Unix time associated with end of data sample
8. **DELTA_T** \equiv scalar [double] defining total duration of IDL structure [= END_TIME - TIME]
9. **INTEG_T** \equiv scalar [double] defining the average time needed for the 1024 counter readouts per spin (s) [= (END_TIME - TIME)/1024]
10. **NENERGY** \equiv scalar [integer] defining the number of energy bins
11. **NBINS** \equiv scalar [integer] defining the number of solid angle bins
12. **DT_ARR** \equiv [NENERGY,NBINS]-element array [float] of anode accumulation times [unitless] per bin \Rightarrow accumulation time [s] of any given bin = (INTEG_T * DT_ARR)
13. **DATA** \equiv [NENERGY,NBINS]-element array [float] defining the data point values for each energy/angle bin [units depend on the value of UNITS_NAME]
14. **ENERGY** \equiv [NENERGY,NBINS]-element array [float] of average energy bin values [eV]
15. **DENERGY** \equiv [NENERGY,NBINS]-element array [float] defining the energy range [eV] of each value of ENERGY
16. **PHI** \equiv [NENERGY,NBINS]-element array [float] defining the average azimuthal angle³¹ [deg] for each bin
17. **DPHI** \equiv [NENERGY,NBINS]-element array [float] defining the angular range(uncertainty) [deg] for each value of PHI
18. **THETA** \equiv [NENERGY,NBINS]-element array [float] defining the average poloidal angle³² [deg] for each bin

²⁸see `dat_themis_esa_str_names.pro` for more possibilities

²⁹see `thm_convert_esa_units_lbwiii.pro` for descriptions and more possibilities

³⁰seconds since January 1, 1970

³¹ $+95^\circ \lesssim \theta \lesssim +450^\circ$ in DSL coordinates for THEMIS ESA and GSE coordinates for Wind/3DP, where $+180^\circ$ is roughly in the sun direction

³² $-90^\circ \leq \theta \leq +90^\circ$, where 0° is roughly in the spin plane

19. **DTHETA** \equiv [NENERGY,NBINS]-element array [float] defining the angular range(uncertainty)³³ [deg] for each value of DTHETA
20. **EFF** \equiv [NENERGY,NBINS]-element array [double] defining the efficiency correction [unitless] to the geometry factor accounting for dead time corrections
21. **GEOM_FACTOR** \equiv scalar [float] defining the total geometry factor of the detector [cm^2sr]
22. **GF** \equiv [NENERGY,NBINS]-element array [float] defining the relative geometric factor per bin \Rightarrow the geometry factor of each bin is = (GEOM_FACTOR * GF * EFF)
23. **DEAD** \equiv scalar [float] defining the detector dead time [$\sim 170 \pm 10 \text{ ns}$] of the Amptek A121 preamplifier³⁴
24. **CHARGE** \equiv scalar [float] defining the sign of the particle charge being measured
25. **MASS** \equiv scalar [float] defining the mass [$(\text{eV}/c)^2$ with c in km/s] of the particles being measured
26. **MAGF** \equiv [3]-element array [float] defining the average magnetic field vector [nT] for the duration of the distribution (coordinate system depends on user preference but should match the basis defining PHI and THETA to be meaningful and useful)
27. **VELOCITY** \equiv [3]-element array [double] defining the average bulk flow velocity [km/s] for the duration of the distribution (coordinate system issue similar to MAGF)
28. **SC_POT** \equiv scalar [float] defining the estimate of the spacecraft potential (eV)

B Unit Conversions

To convert between different units³⁵, a few quantities must be calculated first. Let us assume we start with the units of counts. Let us assume we have a particle distribution IDL data structure called *dat* (see Section A for structure tag definitions), then we can define the following quantities:

1. E \equiv particle kinetic energy (eV) [associated with *dat.ENERGY*]
2. N_E \equiv number of energy bins [associated with *dat.NENERGY*]
3. N_A \equiv number of solid angle bins [associated with *dat.NBINS*]
4. δt \equiv sample/accumulation time (s) [associated with *dat[0].INTEG_T[0]*dat.DT_ARR*]
5. gf \equiv differential geometry factor for each data point [associated with *dat.GF * dat.GEOM_FACTOR * dat.EFF*]
6. M_s \equiv particle mass of species s ($(\text{eV}/c)^2$ with c in km/s) [associated with *dat.MASS*]
7. τ \equiv dead time [associated with *dat.DEAD*]
8. $f(E,\Omega)$ \equiv the data [associated with *dat.DATA* in counts], where Ω is the solid angle
9. $g(E,\Omega)$ \equiv the data in new user specified units

³³this is limited primarily by the anodes being used in a particular mode

³⁴see, for example, *Paschmann and Daly* [1998] for explanation of dead times

³⁵see, for example, [thm_convert_esa_units_lbwiii.pro](#)

10. $\delta g \equiv$ estimated uncertainty in $g(E,\Omega)$

To correct for the dead time, we define:

$$\delta t_c \equiv \frac{\tau f(E, \Omega)}{\delta t} \quad (1)$$

The scale factors used to convert from counts to any of the following are:

$$Counts : scale = 1.0 \quad (2a)$$

$$rate : scale = (\delta t)^{-1} \quad (2b)$$

$$crate : scale = (\delta t)^{-1} \quad (2c)$$

$$eflux : scale = (\delta t * gf)^{-1} \quad (2d)$$

$$flux : scale = (\delta t * gf * E)^{-1} \quad (2e)$$

$$df : scale = (\delta t * gf * E)^{-1} * \left(\frac{mass^2}{2.0 \times 10^5} \right) \quad (2f)$$

where the final result in new units is given by:

$$g(E, \Omega) = scale * \left(\frac{f(E, \Omega)}{\delta t_c} \right) . \quad (3)$$

The uncertainty in $g(E, \Omega)$ is given by:

$$\delta g = scale * \left(\frac{f(E, \Omega)}{\delta t_c} \right)^{1/2} . \quad (4)$$

C Bi-Maxwellian Distribution Functions

In general, for uncorrelated velocity variables, we can write:

$$f(V_x, V_y, V_z) = f(V_x) f(V_y) f(V_z) . \quad (5)$$

Note that a generalized Gaussian probability density function is given by:

$$f(x) = \frac{A_o}{\sqrt{2\pi\sigma^2}} e^{-\frac{(x - x_o)^2}{2\sigma^2}} \quad (6)$$

where x_o is the displacement of the peak from $x = 0$, A_o is a normalization amplitude, and σ^2 is the variance (defined in Equation 22). For this distribution, one can find the Full Width at Half Maximum (FWHM) = $2\sqrt{2\ln 2}\sigma$, or the width of the distribution at half its peak value. In terms of physical parameters, FWHM = $2\sqrt{\ln 2} V_{T_s}$, where V_{T_s} is the *most probable speed*, which we will use as the thermal speed. The most probable speed is given by:

$$V_{T_s} = \sqrt{\frac{2k_B T_s}{m_s}} \quad (7)$$

where k_B is Boltzmann's constant, T_s is the temperature, m_s is the mass, and s is the particle species. Note that all thermal speeds discussed herein will be defined by Equation 7. Now if we

change Equation 6 by letting $2\sigma^2 \rightarrow V_{T_s}^2$, $x_o \rightarrow v_o$ ³⁶, and $A_o \rightarrow n_o$ ³⁷, then a one-dimensional Maxwell-Boltzmann velocity distribution, or Maxwellian, is given by:

$$f_s(v) = \frac{n_o}{\sqrt{\pi} V_{T_s}} e^{-\left(\frac{v - v_o}{V_{T_s}}\right)^2}. \quad (8)$$

Now we can take Equation 5 and let $V_x \rightarrow V_\perp \cos \phi$, $V_y \rightarrow V_\perp \sin \phi$, and $V_z \rightarrow V_\parallel$, where ϕ is the phase angle of the velocity. If we assume azimuthal symmetry³⁸, then the distribution is said to be *gyrotropic*. This leads to $V_{T\perp,x} = V_{T\perp,y} \equiv V_{T\perp}$, which gives us the general form of a bi-Maxwellian give by:

$$f(V_\parallel, V_\perp) = \frac{n_o}{\pi^{3/2} V_{T\perp}^2 V_{T\parallel}} e^{-\left[\left(\frac{V_\parallel - V_{o\parallel}}{V_{T\parallel}}\right)^2 + \left(\frac{V_\perp - V_{o\perp}}{V_{T\perp}}\right)^2\right]} \quad (9)$$

where the subscripts $\perp(\parallel)$ are the perpendicular(parallel) directions with respect to a quasi-static background magnetic field.

D Fluid Moment Definitions

Let us assume we have a function, $f_s(\mathbf{x}, \mathbf{v}, t)$, which defines the number of particles of species s in the following way:

$$dN = f_s(\mathbf{x}, \mathbf{v}, t) d^3x d^3v \quad (10)$$

which tells us that $f_s(\mathbf{x}, \mathbf{v}, t)$ is the particle distribution function of species s that defines a probability density in phase space. We can define moments of the distribution function as expectation values of any dynamical function, $g(\mathbf{x}, \mathbf{v})$, as:

$$\langle g(\mathbf{x}, \mathbf{v}) \rangle = \frac{1}{N} \int d^3x d^3v g(\mathbf{x}, \mathbf{v}) f(\mathbf{x}, \mathbf{v}, t) \quad (11)$$

where $\langle \rangle$ is the average, which can mean ensemble average, arithmetic mean, etc.

If we define a set of fluid moments with similar format to that of Equation 28, which act as averages, then we have:

$$\text{number density: } n_s = \int d^3v f_s(\mathbf{x}, \mathbf{v}, t) \quad (12a)$$

$$\text{average velocity: } \mathbf{U}_s = \frac{1}{n_s} \int d^3v \mathbf{v} f_s(\mathbf{x}, \mathbf{v}, t) \quad (12b)$$

$$\text{kinetic energy density: } W_s = \frac{m_s}{2} \int d^3v v^2 f_s(\mathbf{x}, \mathbf{v}, t) \quad (12c)$$

$$\text{pressure tensor (dyadic): } \overleftrightarrow{\mathbb{P}}_s = m_s \int d^3v \mathbf{w} \mathbf{w} f_s(\mathbf{x}, \mathbf{v}, t) \quad (12d)$$

$$\text{heat flux tensor (triadic): } Q_{l,m,n} = m_s \int d^3v \mathbf{w}_l \mathbf{w}_m \mathbf{w}_n f_s(\mathbf{x}, \mathbf{v}, t) \quad (12e)$$

³⁶drift speed

³⁷particle number density

³⁸ $\partial f / \partial \phi = 0$

where $\mathbf{w} = (\mathbf{v} - \mathbf{U}_s)$ and the pressure tensor can be written as:

$$\overleftrightarrow{\mathbb{P}}_s = \begin{bmatrix} P_{xx} & P_{xy} & P_{xz} \\ P_{yx} & P_{yy} & P_{yz} \\ P_{zx} & P_{zy} & P_{zz} \end{bmatrix} \quad (13)$$

which can be reduced to a symmetric tensor in most cases with the only off-diagonal elements being P_{xy} , P_{xz} , and P_{yz} . In a magnetized plasma, the magnetic field direction can often organize the collective particle motion so that the pressure tensor is reduced to a diagonal tensor³⁹. In general, one can separate the pressure tensor into a diagonal part and the off-diagonal part, which is usually called the stress tensor. The general diagonal elements of the pressure tensor are:

$$\overleftrightarrow{\mathbb{P}}_s = \begin{bmatrix} P_{\perp,1} & 0 & 0 \\ 0 & P_{\perp,2} & 0 \\ 0 & 0 & P_{\parallel} \end{bmatrix} \quad (14)$$

where a gyrotropic assumption will result in $P_{\perp,1} = P_{\perp,2}$. Thus, a gyrotropic plasma will have:

$$P_{\perp,s} = n_s k_B T_{\perp,s} \quad (15a)$$

$$P_{\parallel,s} = n_s k_B T_{\parallel,s} \quad (15b)$$

and a non-gyrotropic plasma will have:

$$T_{\perp,s} = \frac{1}{2n_s k_B} (P_{\perp,1,s} + P_{\perp,2,s}) \quad (16a)$$

$$T_{\parallel,s} = \frac{1}{n_s k_B} P_{\parallel,s} . \quad (16b)$$

Therefore, if we have the following relationships:

$$\text{gyrotropic: } V_{Ts} = \sqrt{\frac{1}{2} (V_{Ts,\perp}^2 + V_{Ts,\parallel}^2)} \quad (17a)$$

$$\text{non-gyrotropic: } V_{Ts} = \sqrt{\frac{2}{3m_s} \text{Tr} \left[\frac{\overleftrightarrow{\mathbb{P}}_s}{n_s k_B} \right]} \quad (17b)$$

$$= \sqrt{\frac{2k_B \langle T_s \rangle}{m_s}} \quad (17c)$$

where we have used $\text{Tr}[\cdot]$ as the trace and defined:

$$\langle T_s \rangle = \frac{1}{3} \text{Tr} \left[\frac{\overleftrightarrow{\mathbb{P}}_s}{n_s k_B} \right] . \quad (18)$$

The average temperature of particle species s shown in Equation 18 is the one most often used when calculating temperatures from electrostatic plasma analyzers [e.g. *Curtis et al.*, 1989]. The temperature is physically a measure of the average kinetic energy density of particle species s , and

³⁹Off-diagonal terms arise from non-gyrotropic features e.g. gyrating [e.g. *Meziane et al.*, 1997] or gyrophase-bunched [e.g. *Gurgiolo et al.*, 1981] ion distributions.

can be represented as:

$$T_{\perp,s} = \frac{1}{2} (T_{\perp,1,s} + T_{\perp,2,s}) \quad (19a)$$

$$\langle T_s \rangle = \frac{1}{3} (T_{\perp,1,s} + T_{\perp,2,s} + T_{\parallel,s}) \quad (19b)$$

therefore, if we already have $V_{T_s,\perp}$ and $V_{T_s,\parallel}$ and we assume $T_{\perp,1} \neq T_{\perp,2}$ (*i.e.* non-gyrotropic)⁴⁰, then we have:

$$V_{T_s} = \sqrt{\frac{1}{3} (V_{T_s,\perp,1}^2 + V_{T_s,\perp,2}^2 + V_{T_s,\parallel}^2)} \quad (20a)$$

$$= \sqrt{\frac{2V_{T_s,\perp}^2}{3} + \frac{V_{T_s,\parallel}^2}{3}} \quad (20b)$$

$$\neq \sqrt{\frac{1}{2} (V_{T_s,\perp}^2 + V_{T_s,\parallel}^2)} \quad (20c)$$

E Statistics Definitions

We define \bar{x} as the arithmetic mean of a set of $\{x_1, x_2, \dots, x_N\}$, given by:

$$\bar{x} = \frac{1}{N} \sum_{i=1}^N x_i \quad (21)$$

and this allows us to define the variance of $\{x_1, x_2, \dots, x_N\}$ as:

$$\sigma^2 = \frac{1}{N-1} \sum_{i=1}^N (x_i - \bar{x})^2 \quad (22)$$

where $N - 1$ = number of degrees of freedom and σ is the standard deviation. Sometimes \bar{x} is written as $\langle x \rangle$. If there exists a probability density function, $P(x)$, then for discrete (Equation 23a) or continuous (Equation 23b) values we have:

$$\langle x \rangle = \sum_{i=1}^N P(x_i) f(x_i) \quad (23a)$$

$$= \int dx P(x) f(x) \quad (23b)$$

$$\equiv \mu_x . \quad (23c)$$

⁴⁰In most cases, it is assumed that the electrons are gyrotropic and ions are non-gyrotropic. Physically, this is due to the relatively long sample period (≥ 3 s) of current particle detectors compared to Ω_{ce}^{-1} for electrons, which causes the resulting measured distribution to appear *smeared out* in phase space. Most non-gyrotropic features are lost due to the relatively long sample periods. For ions, however, Ω_{cp}^{-1} can be $\sim 1-10$ s (for $B_o \sim 1-10$ nT). Therefore, non-gyrotropic features (*e.g.* gyrophase bunching) can often be observed in ion distributions.

Note that $\langle \rangle$ represents the arithmetic mean. Similarly, the variance can be expressed as:

$$\sigma^2 = \sum_{i=1}^N P(x_i) (x_i - \mu)^2 \quad (24a)$$

$$= \int dx P(x) (x - \mu)^2 \quad (24b)$$

where we can write a variance operator as follows:

$$var(x) \equiv \langle x^2 \rangle - \langle x \rangle^2 \quad (25)$$

and define the covariance as:

$$cov(x, y) \equiv \langle (x - \mu_x)(y - \mu_y) \rangle \quad (26)$$

which can also be written as:

$$\mathcal{V}_{xy} = \sigma_{xx}\sigma_{yy} + \sigma_{xy}\sigma_{yy} \quad (27)$$

where we have defined $\sigma_{jk} \equiv cov(x_j, x_k)$ and $\sigma_{jj} = \sigma_j^2$.

The moments of the distribution can be defined as $\mu^n \equiv \langle (x - \langle x \rangle)^n \rangle$, which for a general Gaussian distribution takes the form:

$$\mu^n = \frac{1}{\sqrt{2\pi\sigma^2}} \int_{-\infty}^{\infty} dx (x - \mu)^n e^{-\frac{(x - \mu)^2}{2\sigma^2}} \quad (28)$$

References

- Angelopoulos, V. (2008), The THEMIS Mission, *Space Sci. Rev.*, *141*, 5–34, doi:10.1007/s11214-008-9336-1.
- Curtis, D. W., C. W. Carlson, R. P. Lin, G. Paschmann, and H. Reme (1989), On-board data analysis techniques for space plasma particle instruments, *Rev. Sci. Instr.*, *60*, 372–380, doi:10.1063/1.1140441.
- Gurgiolo, C., G. K. Parks, B. H. Mauk, K. A. Anderson, R. P. Lin, H. Reme, and C. S. Lin (1981), Non-E x B ordered ion beams upstream of the earth's bow shock, *J. Geophys. Res.*, *86*, 4415–4424, doi:10.1029/JA086iA06p04415.
- Lin, R. P., et al. (1995), A Three-Dimensional Plasma and Energetic Particle Investigation for the Wind Spacecraft, *Space Sci. Rev.*, *71*, 125–153, doi:10.1007/BF00751328.
- McFadden, J. P., C. W. Carlson, D. Larson, M. Ludlam, R. Abiad, B. Elliott, P. Turin, M. Markwordt, and V. Angelopoulos (2008a), The THEMIS ESA Plasma Instrument and In-flight Calibration, *Space Sci. Rev.*, *141*, 277–302, doi:10.1007/s11214-008-9440-2.
- McFadden, J. P., C. W. Carlson, D. Larson, J. Bonnell, F. Mozer, V. Angelopoulos, K.-H. Glassmeier, and U. Auster (2008b), THEMIS ESA First Science Results and Performance Issues, *Space Sci. Rev.*, *141*, 477–508, doi:10.1007/s11214-008-9433-1.
- Meziane, K., et al. (1997), Wind observation of gyrating-like ion distributions and low frequency waves upstream from the earth's bow shock, *Adv. Space Res.*, *20*, 703–706, doi:10.1016/S0273-1177(97)00459-6.
- Paschmann, G., and P. W. Daly (1998), Analysis Methods for Multi-Spacecraft Data. ISSI Scientific Reports Series SR-001, ESA/ISSI, Vol. 1. ISBN 1608-280X, 1998, *ISSI Sci. Rep. Ser.*, *1*.
- Wilson III, L. B., C. A. Cattell, P. J. Kellogg, K. Goetz, K. Kersten, J. C. Kasper, A. Szabo, and K. Meziane (2009), Low-frequency whistler waves and shocklets observed at quasi-perpendicular interplanetary shocks, *J. Geophys. Res.*, *114*, 10,106–+, doi:10.1029/2009JA014376.
- Wilson III, L. B., C. A. Cattell, P. J. Kellogg, K. Goetz, K. Kersten, J. C. Kasper, A. Szabo, and M. Wilber (2010), Large-amplitude electrostatic waves observed at a supercritical interplanetary shock, *J. Geophys. Res.*, *115*, 12,104–+, doi:10.1029/2010JA015332.
- Wilson III, L. B., et al. (2012), Observations of Electromagnetic Whistler Precursors at Supercritical Interplanetary Shocks, *Geophys. Res. Lett.*, *39*, L08,109–+, doi:10.1029/2012GL051581.