**SPLAT**

**(Stereo-PLastic Analysis Tool)**

**Software Package for STEREO-PLASTIC**

**Lorna Ellis** ([Lorna.Ellis@unh.edu](mailto:Lorna.Ellis@unh.edu))

**Kristin Simunac** ([ksimunac@cisunix.unh.edu](mailto:ksimunac@cisunix.unh.edu))

**Lynn Kistler** ([Lynn.Kistler@unh.edu](mailto:Lynn.Kistler@unh.edu))

**Chris Mouikis** ([chris.mouikis@unh.edu](mailto:chris.mouikis@unh.edu))

Version 1.0.0 03/09/2007

Version 1.4 02/11/2021

**1. Introduction**

SPLAT is a software package for the data-analysis of the STEREO PLASTIC instrument. It is written in IDL and uses the Berkeley TPLOT graphics package (by Davin Larson <http://sprg.ssl.berkeley.edu/~davin/>). This package provides the data extraction and processing routines which are initiated by the use of crib-sheets. Once the data are loaded in the IDL memory, tplot functions are used to plot the data. The crib-sheets provided, are generic idl scripts that are used to load particular data products. This version of the software represents the current state of the PLASTIC software at UNH.

**2. Installation**

**2.1. System Requirements**

SPLAT has been developed in IDL6 in a Linux-Unix environment. SPLAT should work on any Unix and Unix based (MAC OSX) system. Porting to other platforms is not currently planned but is possible. Filenames & paths follow the UNIX naming convention.

**2.2. Folder Structure**

SPLAT is delivered as a zipped file (splat*.zip*). Unzip this file using an unzip utility or the following unix command:

*unzip splat.zip*

A folder, named *splat*, is installed in the current directory. Under this folder the following folders/files can be found:

* A ***README*** file with a quick-start description on how to get started with *splat*.
* A ***Doc*** folder that contains the splat manual and a set of html files that list all the procedures/functions and the relevant documentation as provided at the file header.
* A ***cal\_data*** folder that contains some calibration files. Others are maintained in STEREO’s SolarSoft library.
* The folder ***splat\_v1.4***. All the processing routines are in this folder. In principle, there is no need to edit any of the files in this folder. The processing routines are written in a fairly transparent way, with a lot of comments which should allow anybody with some IDL programming experience to read them. In case a bug is found or an improvement is needed we would like to be informed so that we can include the corrections/improvements in a new version. The splat\_v1.4 folder comprises of three subfolders: **splat\_core** that contains all the SPLAT data extraction and processing routines, **splat\_util** that contains utility procedures/functions usable with SPLAT, and **splat\_tplot** that contains all the *tplot* (graphics functionality) routines.
* The folder ***splat\_user.*** This folder is each user's personal folder and contains: the splat*.sh* start-up script, a list of crib-sheets for the creation of different products and a number of subfolders for the better organization of SPLAT related tasks. The crib-sheets are IDL scripts that call the main processing routines for the creation of a particular product (i.e. energy spectrum, moments and etc.) Once these products are created, they are stored in memory and can be plotted using the TPLOT routines.
* In the ***Public*** subfolder of the ***splat\_user*** folder, there are also routines that are used for the creation of some Level 2 and Level 3 data products. These routines are often hard-coded for use at UNH, but they show the algorithms and code used to create some of the Level 2 and Level 3 data products.

**2.3. Editing splat.sh start-up script**

Each user has to edit the *splat.sh* start-up script in the *splat\_user* folder. The *splat.sh* script is a c-shell script setting environment variables for all software and data paths. In the example script included, it is assumed that both *splat\_user* and *splat\_v1.4* directories are installed in the user's home directory under the directory *splat*. The user has to ensure that all software paths defined in this file are valid. Data paths need to be valid if they are to be used for the creation of a particular product. Here is a description of some of the environment variables that are set by the splat.sh script (only highlighted variables are the ones that need to be edited if the default setup is kept):

* **SOFT**: path of where the directory splat is installed. Here the user’s home directory is assumed.
* **SPLAT**: path to the *splat* directory.
* **SPLAT\_USER**: path to the *splat\_user* directory.
* **IDL\_PATH**: It requires that the IDL\_PATH environment variable is set (which is usually the case - typing the command: echo $IDL\_PATH in the unix command line should return the idl path) and it updates the IDL\_PATH to include the *splat* software.
* **STA\_PLA\_L1DATA**: path to plastic *CDFs* directory for spacecraft A.
* **STB\_PLA\_L1DATA**: path to plastic *CDFs* directory for spacecraft B.
* **PLACAL**: the path to the calibration files.
* **SPLAT\_TEMP**: path to a temporary directory. This directory is used for splat “internal” purposes and because of this, the splat\_temp path has to be valid. By default it points to the splat\_user/tmp directory.
* **OUT\_DIR:** Points to the directory where plots can be stored. This is not used extensively. Most of the time plots are saved locally.
* **ERROR\_LOG**: if set to ‘on’ it writes a separate error log file in the splat\_user/error\_log folder for every idl session. The default setting is *‘off’*
* **IDL\_STARTUP**: runs the splat\_startup script when idl is started.

**3. SPLAT Usage**

Once the start-up script *splat.sh* file is properly edited the user has to:

* execute the *splat.sh* using the following unix command: *source splat.sh* (the usage of the tcsh or csh is assumed). In each unix window (unix command line session) this shell should be executed only once
* if the user is running SPLAT in the SolarSoft environment, then the user must also source the SolarSoft startup script
* Start IDL
* From the IDL-prompt execute the SPLAT *crib-sheets*. For example in order to execute the crib-sheet for the creation of energy spectra, type in the IDL-prompt: *@pla\_en\_spec\_crib*. Available crib-sheets are described in the next section.

Once the *crib-sheet* is executed tplot commands can be used to display the data. Here is a very basic list of tplot commands. These commands should be executed in the idl command line after data products are loaded.

* **tplot\_names**: it lists the tplot variables that are loaded in the IDL memory. The name or the number of the tplot variable listed, should be used in order to refer to a particular variable.
* **tplot**: Assuming that four tplot variables are loaded, these can be plotted using: tplot, [1,2,3,4]. Different order of the numbers can be used in order to change the order that the variables are plotted.
* **ylim, zlim**: to adjust the y range of a tplot variable, type: ylim, 2, 1e1, 4e4, 1 where 2 is the tplot variable number, 1e1 is the minmum value, 4e4 the maximum value and 1 for log axis. For the z axis the zlim command should be used instead.
* **tlimit**: to zoom in a plot using the mouse. tlimit, 10, 12 should be used to specify the time interval between 10 and 12 UT. tlimit, 0, 0 should be used in order to get the maximum time interval available.
* **popen, pclose**: to create a ps file. First plot on the screen the plot using any combination of the above commands, then type: popen, 'filename.ps', then type: *tplot* and finally type: *pclose*

The *\_example\_crib* file, included in the *splat\_user* directory,

**3.1. SPLAT Crib-Sheets - DESCRIPTION OF CRIBS AND INPUT PARAMETERS**

Crib-sheets are sets of IDL-code calling SPLAT main-level plot routines. A set of example crib-sheets are contained in the *splat\_user* folder. A crib-sheet is executed by typing *@crib-sheet-name* at the IDL prompt. This will load all requested data into *tplot* variables and will produce a plot on screen.

**3.1.1. MOST COMMON INPUT PARAMETERS**

**time**: Start date/time string in tplot format ('*yyyy-mm-dd/hh:mm:ss'*)

*Example: time = ‘2007-01-20/00:00:00’*

**timespan**: Set the time span (duration). One of the following keywords has to be used: S*ECONDS, MINUTES, HOURS, DAYS (DEFAULT)*

*Example: timespan, time, 1, /DAYS*

**sat**: Satellite id ('A' or 'B'). An array of strings indicating the s/c

*Example: [‘A’, ‘B’]*

**moments**: Moment to be calculated. The following abbreviations should be used *'D': Density, 'V': Velocity, 'T': Temperature 'P': Pressure, 'J': Flux, 'E': Energy Flux.* The *sat*, *specie* and *moments* arrays *must have exactly the same number of element*

**energy**: Energy range (200 - 87000 eV) [*energy\_min, energy\_max*]

*Example: energy = [1000, 10000]*

**theta**: Deflection angle range (-25, 25 degrees) [theta\_min, theta\_max]

*Example: theta = [-25, 25]*

**phi**: Position angle range (-25, 25 degrees) [phi\_min, phi\_max]

*Example: theta = [-25, 25]*

**units**: Units options: *'Counts', 'c/s''*

*Example: units = ‘Counts’*

**3.1.2. pla\_en\_spec\_crib**

To plot PLASTIC energy spectra (quantity vs. energy vs. time)

**INPUT PARAMETERS**: *product, time, timespan, units*

Possible products are:

*# SOURCE ARRAY REDUCED ARRAY CLASSIFIER*

*01: SW-all : h\_alpha*

*02: SW-H(D) : h+peak*

*03: SW-alpha(D) : he++peak*

*04: SW-alpha(T) : he++trc*

*05: SW Z>2 : sw\_z>2\_h class:0-1*

*06: SW Z>2 : sw\_z>2\_l class:0-12*

*07: WIDE ANGLE : wap\_ssd\_tcr class:0-14*

*08: WIDE ANGLE(D) : wap\_ssd\_dcr class:0-6*

*09: WIDE ANGLE(D) : wap\_no\_ssd\_dcr class:0-6*

*10: SW PHA PRIORITY RATES : sw\_priority class:0-3*

*11: WAP PHA PRIORITY RATES : wap\_priority\_ssd class:0-1*

*12: WAP PHA PRIORITY RATES : wap\_priority\_no\_ssd class:0-1*

*PRODUCT INPUT EXAMPLES:*

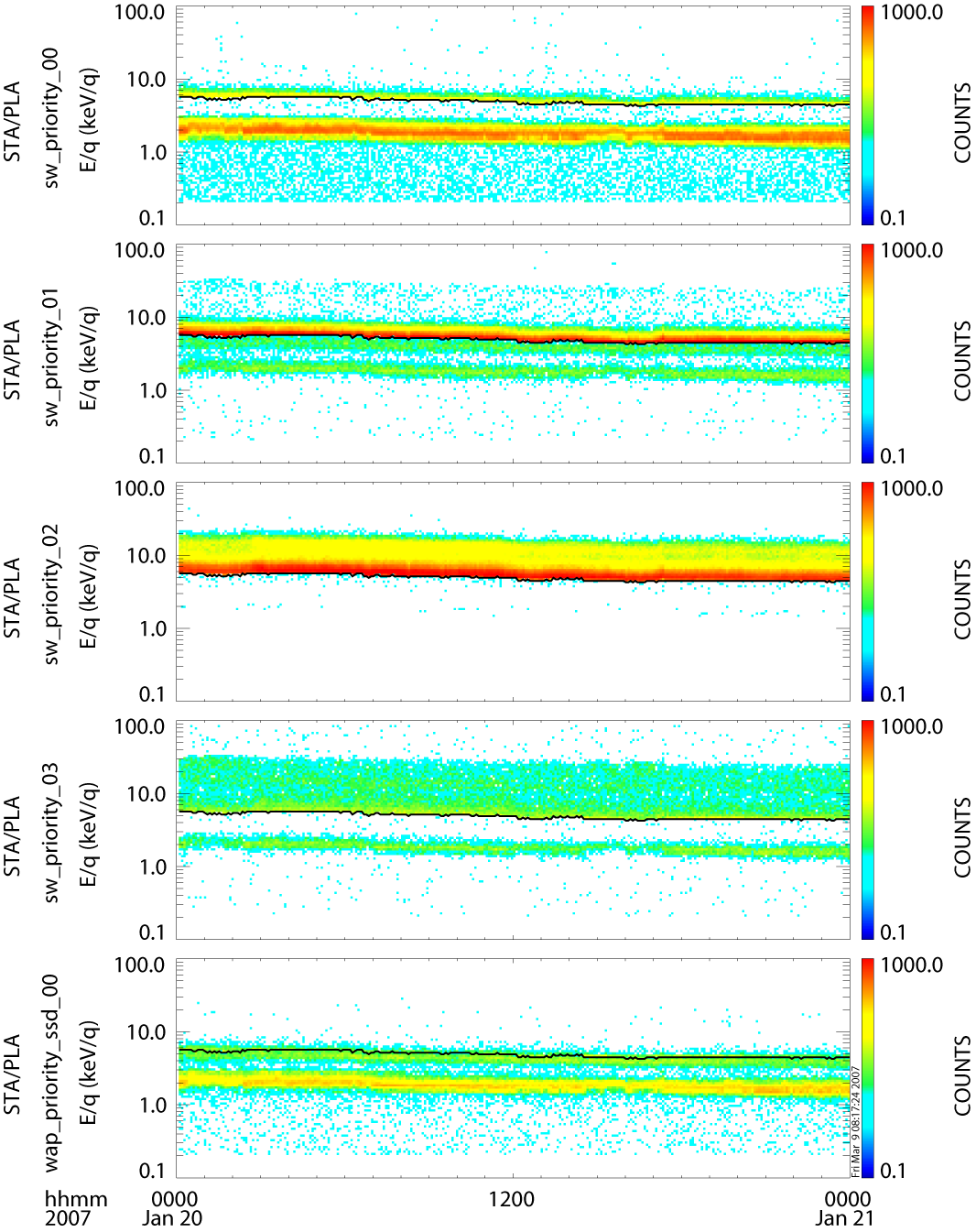
*prod = 'A1' - S/C A, product 1*

*prod = ‘B07/0-14’ - S/C B, product 7, classifiers 0 to 14*

*prod = 'B5/0-1' - S/C B, product 5, classifiers 0 & 1*

*prod = 'A10/0,2-3' - S/C B, product 10, classifiers 0 & 2 & 3*

**KEYWORDS**: *diagn* – it creates the corresponding s-channel plot (over-plotted line in following plots).



**3.1.3. pla\_en\_spec\_line\_crib**

To plot PLASTIC quantity vs. time line plots for a particular energy range.

**INPUT PARAMETERS**: *product, time, timespan, units, energy*

The inputs are the same as the ones described for the energy spectra (3.1.2). An additional input is the energy range (3.1.1) over which the energy spectrum is integrated.

**KEYWORDS**: *diagn* – it is not used yet.

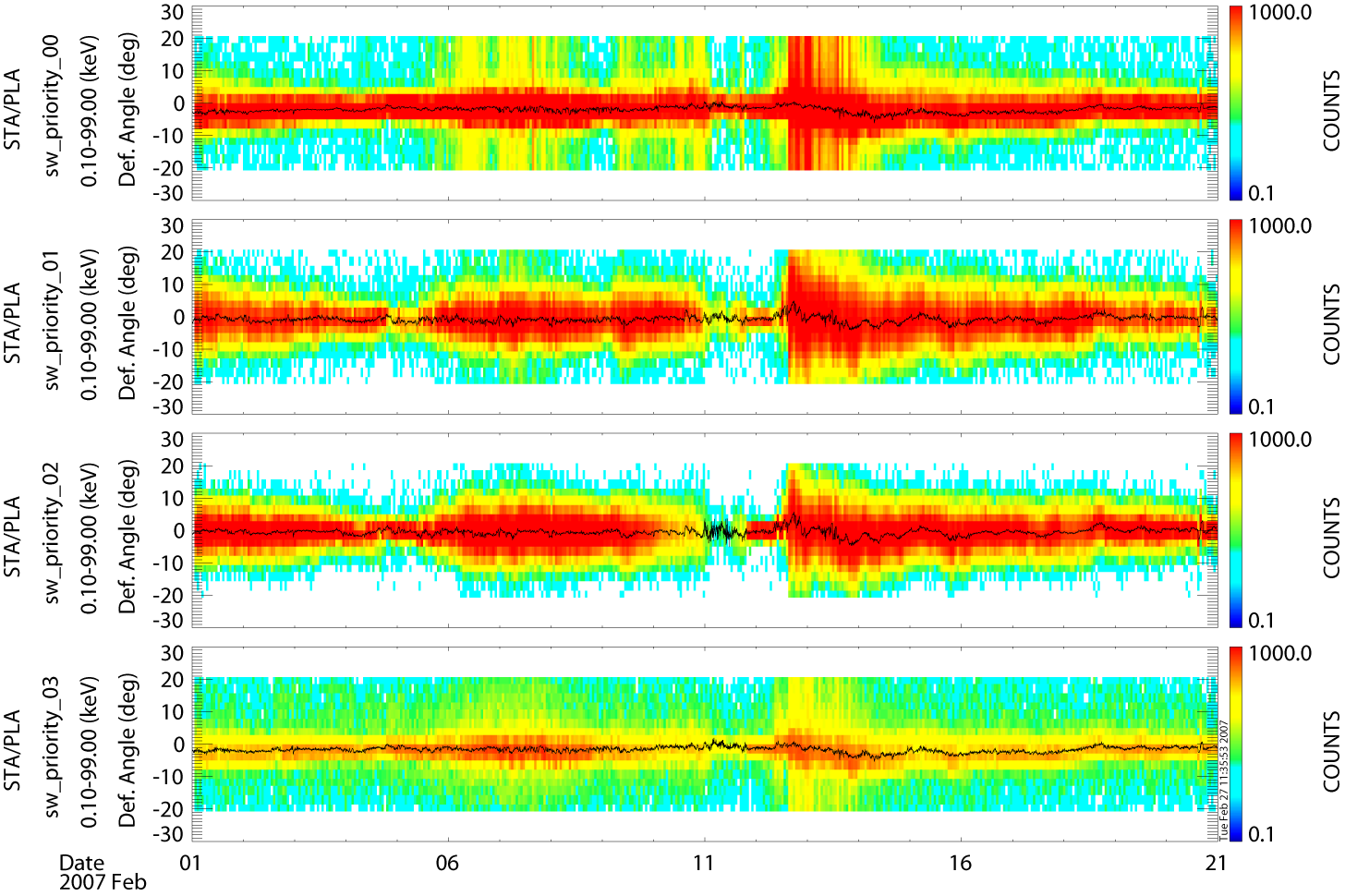
**3.1.4. pla\_def\_spec\_crib**

To plot PLASTIC deflection spectra (quantity vs. deflection angle vs. time)

**INPUT PARAMETERS**: *product, time, timespan, units, energy, phi*

The inputs are the same as the ones described for the energy spectra (3.1.2). An additional input is the energy range (3.1.1) over which the energy spectrum is integrated (and the phi range (3.1.1) which is not used yet).

**KEYWORDS**: *diagn* – It creates the corresponding average deflection plot (over-plotted line in following plots)



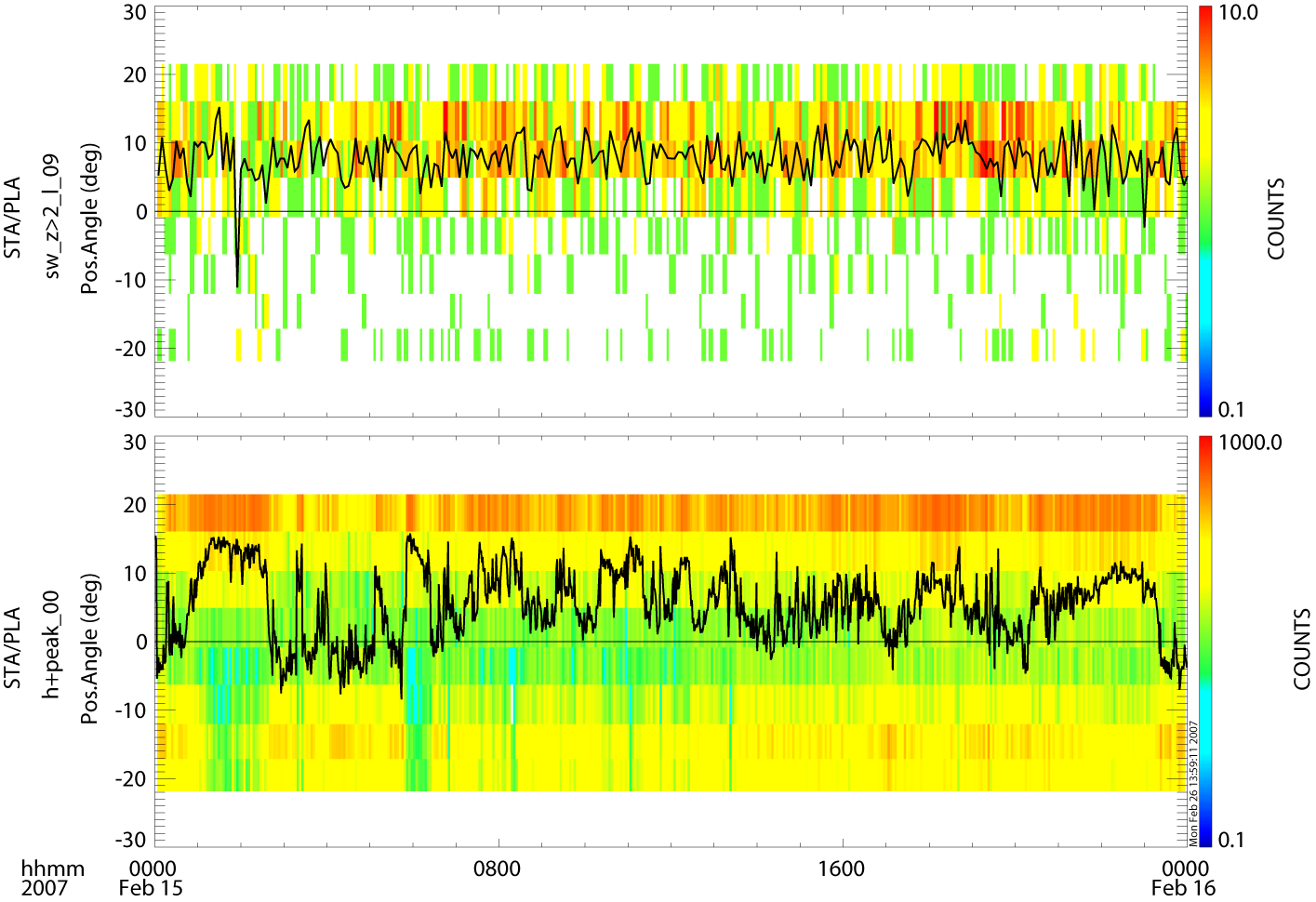
**3.1.5. pla\_pos\_spec\_crib**

To plot PLASTIC position spectra (quantity vs. deflection angle vs. time)

**INPUT PARAMETERS**: *product, time, timespan, units, energy, theta*

The inputs are the same as the ones described for the energy spectra (3.1.2). An additional input is the energy range (3.1.1) over which the energy spectrum is integrated (and the theta range (3.1.1) which is not used yet).

**KEYWORDS**: *diagn* – It creates the corresponding average position plot (over-plotted line in following plots)



**3.1.6. pla\_obmom\_crib**

To plot PLASTIC on-board moments (quantity vs. time)

**INPUT PARAMETERS**: *sat\_moment, time, timespan*

Possible moment products:

D: Density

VX: Velocity Vx component

VY: Velocity Vy component

VZ: Velocity Vz component

VXYZ: All three velocity components

VT: Velocity Vtotal component

TXX: Temperature xx

TYY: Temperature yy

TZZ: Temperature zz

TXXYYZZ: All three temperature components

PXX: Pressure xx

PYY: Pressure yy

PZZ: Pressure zz

PXXYYZZ: All three pressure components

**sat\_moment input examples:**

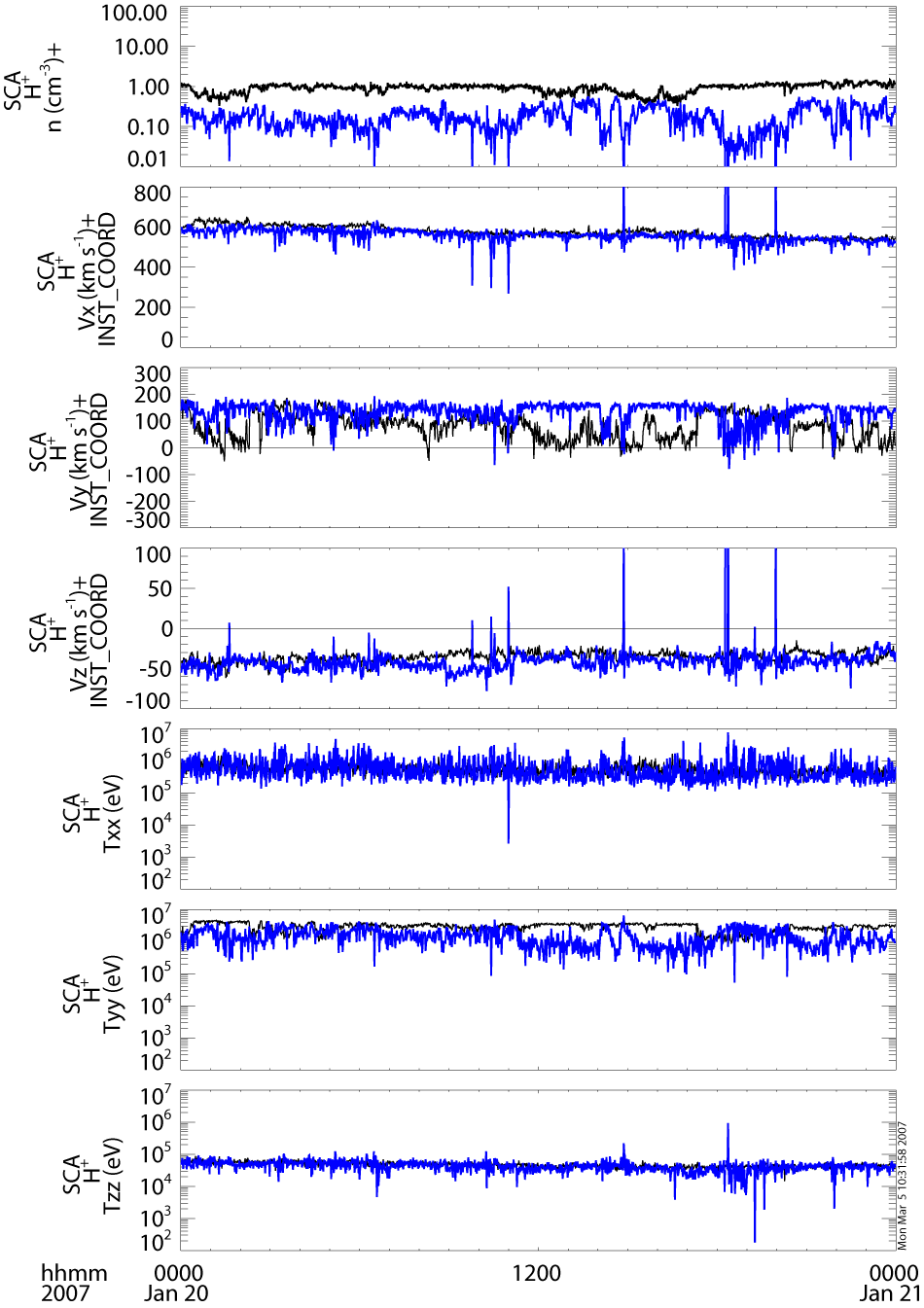
sat\_moment = [‘A\_D’, ‘B-D’]

sat\_moment = [‘A\_D’, ‘A\_VX’, ‘A\_VY’, ‘A\_TXXYYZZ’]

Note: For this version of the software one input per spacecraft will load all moments. Therefore the first example: sat\_moment = [‘A\_D’, ‘B-D’] will load all moments for both spacecraft. After the crib-sheet is executed using the *tplot\_names* command will show the list of all the moments.

**KEYWORDS:** INST\_COORD: moments are calculated in instrument coordinates

In the following figure on-board moments from STA (black) and STB (blue) are plotted.



**3.1.7. pla\_mon\_crib**

To plot PLASTIC monitor rates (quantity vs. time)

**INPUT PARAMETERS**: *sat, sat\_mon, e\_step, time, timespan, units, no\_eng,*

Possible *sat\_mon* products:

‘s\_valid’

‘s\_e\_not\_req’

‘s\_e\_req’

‘s\_no\_pos’

‘s\_mult\_pos’

‘s\_no\_e’

‘s\_mult\_e’

‘ra\_sat\_a’

‘ra\_sat\_b’

‘ra\_sat\_both’

‘ssd\_sw’

‘ssd\_st’

‘sf0’

‘sfr0’

‘stp0’

‘ra\_trig’

‘pos1\_0’

‘pos1\_1’

‘w\_no\_pos’

‘w\_mult\_pos’

‘w\_valid’

‘sf2’

‘sfr2’

‘stp2’

‘pos2\_0’

‘pos2\_1’

‘pos2\_2’

‘pos2\_3’

‘pos3\_0’

‘pos3\_1’

‘pos3\_2’

‘pos3\_3’

‘stop0’ 🡪 sfr0/sf0

‘start0’ 🡪 sfr0/stp0

‘stop2’ 🡪 sfr2/sf2

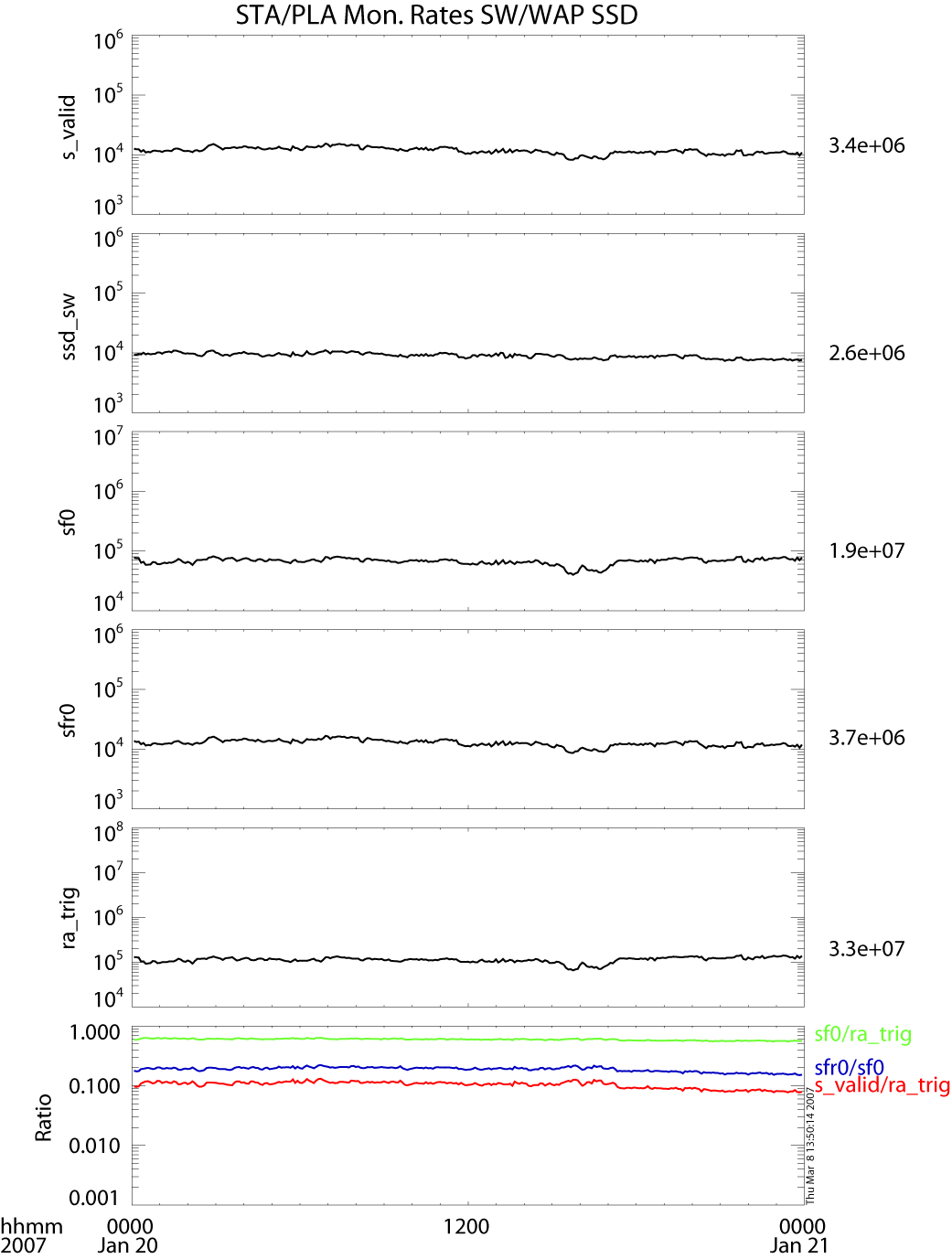
‘start2’ 🡪 sfr2/stp2

**sat\_mon input examples:**

sat\_moment = [‘s\_valid’, ‘ssd\_sw’, ‘sf0’, ‘sfr0’, ‘ra\_trig’] (following plot)

Possible *e\_step* values:

are from 0 to 31. Each e\_step value corresponds to 4 energy bins (from the 0-127 energy bins). e\_step = 32 will return the whole energy range.



**3.1.8. pla\_mon\_full\_crib**

To plot PLASTIC monitor rates for full energy range (quantity vs. time)

Same input parameters as above. The e\_step takes values from 0 to 127.

**3.1.9. pla\_hk\_crib**

To plot house keeping parameters (parametrer vs. time)

**INPUT PARAMETERS**: sat, sat\_hk

The list of the available housekeeping parameters is included in the crib-sheet

**3.1.10. pla\_dig\_hk\_crib**

To plot house keeping parameters (parametrer vs. time)

**INPUT PARAMETERS**: sat, sat\_dig\_hk

The list of the available housekeeping parameters is included in the crib-sheet

**3.1.11. pla\_sc\_hk\_crib**

To plot house keeping parameters (parametrer vs. time)

**INPUT PARAMETERS**: sat, sat\_sc\_hk

The list of the available housekeeping parameters is included in the crib-sheet

**3.2 Level 2 and Level 3 Processing Programs**

In the ***Public*** subfolder of the ***splat\_user*** folder, there are also routines that are used for the creation of some Level 2 and Level 3 data products. These routines are often hard-coded for use at UNH, but they show the algorithms and code used to create some of the Level 2 and Level 3 data products. The following table lists the programs and a sample filename of the products they create.

|  |  |
| --- | --- |
| Sample Filename | Program |
| STA\_L2\_PLA\_1DMax\_1min\_20080101\_V09.cdf | create\_ra\_mom\_cdfs.pro |
| STA\_L2\_PLA\_1DMax\_10min\_20080101\_V09.cdf | create\_ra\_mom\_cdfs.pro |
| STA\_L2\_PLA\_1DMax\_1hr\_20080101\_V09.cdf | create\_ra\_mom\_cdfs.pro |
| STA\_L2\_PLA\_OBMom\_1min\_20080101\_V02.cdf | create\_ob\_mom\_cdfs.pro |
| STA\_L2\_PLA\_OBMom\_10min\_20080101\_V02.cdf | create\_ob\_mom\_cdfs.pro |
| STA\_L2\_PLA\_OBMom\_1hr\_20080101\_V02.cdf | create\_ob\_mom\_cdfs.pro |
| STA\_L2\_PLA\_Alpha\_RA\_1DMax\_10min\_200801\_V02.cdf | create\_alpha\_ra\_10min\_cdfs\_monthly.pro  and splat\_core/calc\_alpha\_bulk.pro |
| STA\_L2\_PLA\_Alpha\_RA\_1DMax\_1hr\_200801\_V02.cdf | create\_alpha\_ra\_1hr\_cdfs\_yearly.pro  and splat\_core/calc\_alpha\_bulk.pro |
| Suprathermal\_Event\_Analysis\_STA.txt | suprathermal\_event\_total\_mod.pro |

**4. Trouble Shooting**

If program execution stops with an error message, examine all the output message lines. In addition, an error log file is kept under splat\_user/error\_log. Quite often the error is related to missing data files or products. In some cases you might resume execution by returning from a subroutine: Type RETURN or RETURN,0 for that purpose, otherwise RETALL will bring you back to the main level, from where you can restart with corrected parameters.