IDL/SPEDAS tutorial featuring Arase data

...brought you by

Center for Heliospheric Science
ISEE, Nagoya Univ.

Goal of this tutorial

- We are going to learn how to load, plot, and manipulate ERG satellite data as well as other related observation data using IDL/SPEDAS.

Keep in mind upon the tutorial

- This is a "hands-on" tutorial for IDL/SPEDAS, not a time for e-mail check!
- Communicate with the lecturer, tutors, and neighboring skilled users.
- Today's session might not be able to cover all topics in the tutorial slides due to time limitation. It is highly recommended to go through the entire contents later by yourself.
- We will proceed rather slowly with intermediate-level users, but you can practice at your own pace.

0. Installation

To use IDL/SPEDAS for data analysis

One has to install the following software and packages to your PC or server to use IDL/SPEDAS:

- Interactive Data Language (IDL) *commercial software*
  - https://www.nv5geospatialsoftware.co.jp/Software-Technology/IDL
- The latest version of SPEDAS
  - http://spedas.org/wiki/index.php?title=Downloads_and_Installation

To use IDL/SPEDAS for data analysis (cont'd)

For Arase data:

- ERG plug-in tool for SPEDAS
  - https://ergsc.isee.nagoya-u.ac.jp/data_info/howto.shtml.en

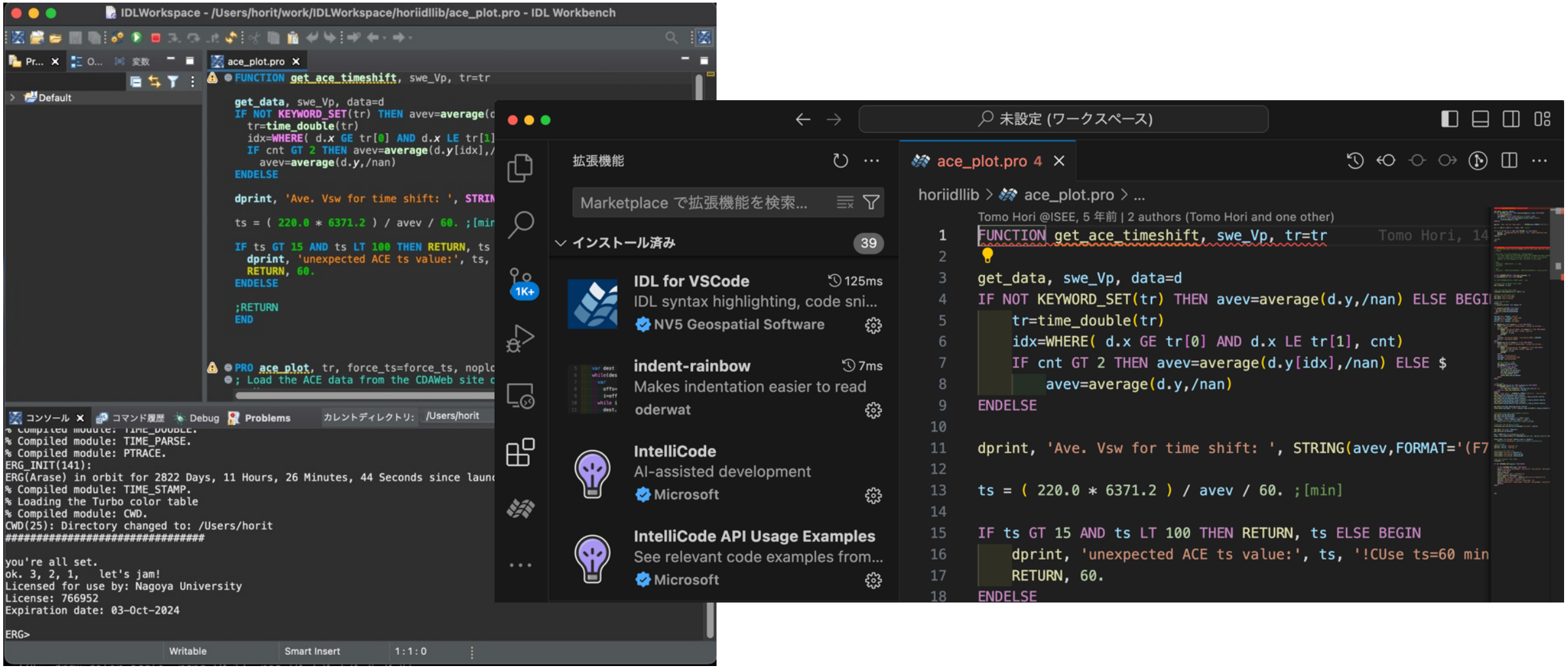
For Tsyganenko B-field models and field-line tracing

- IDL GEOPACK dynamic link module (DLM)
  - https://ampere.jhuapl.edu/tools/?page=infoIDLTab

For using SPICE kernels

- Icy DLM
  - https://naif.jpl.nasa.gov/naif/toolkit_IDL.html

How you can develop an IDL code for SPEDAS?



- IDL workbench, or some free software such as VScode + IDL for VScode plug-in

1. What's SPEDAS?

Space Physics Environment Data Analysis System (SPEDAS)

Data repository

CDF

THEMIS

CDF

Van Allen Probes

CDF

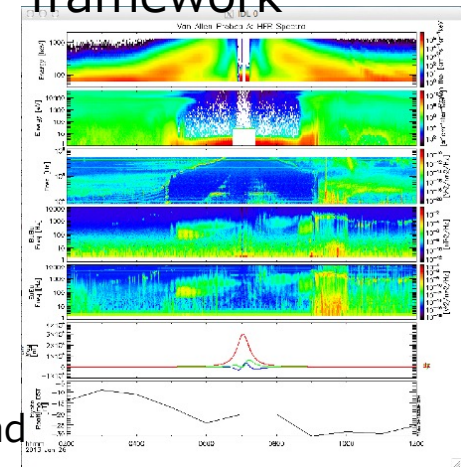
MMS

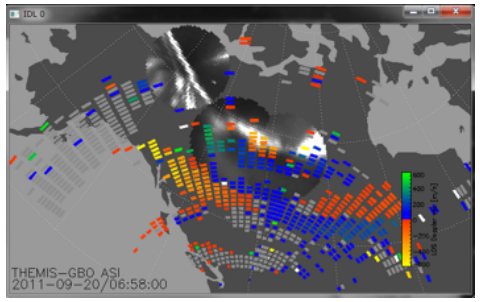
CDF

IUGONET

Each project develops and releases APIs to automatically download and read each project data

SPEDAS framework





ERG

CDF

ERG satellite data

CDF

ERG-related ground observation network data

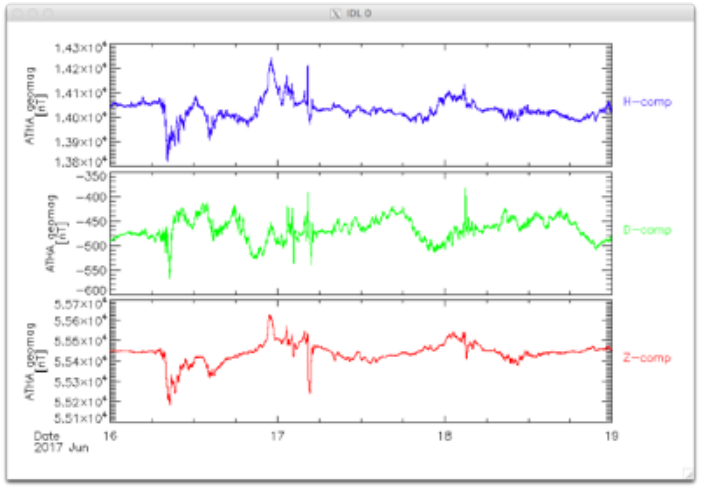
CDF HDF?

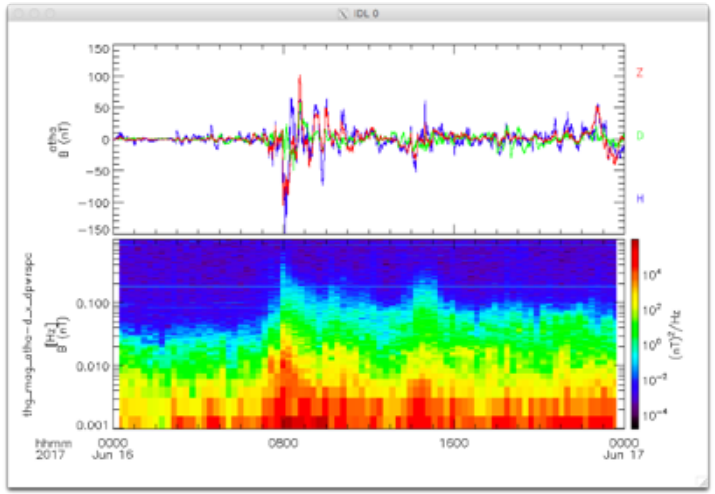
ERG simulation / modeling data

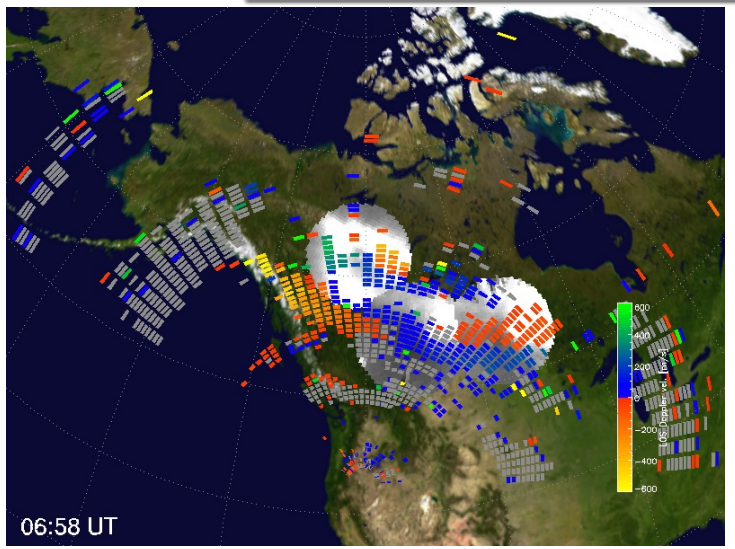
ERG-SC develops data archives and APIs to read the data from SPEDAS

What can you do with SPEDAS?

- loading Arase data and other related data
- time-series plots
- filtering of data
- frequency analysis
- mapping to the ground maps
- ...







Basics of SPEDAS: tplot and tplot variable

A few basics of IDL before entering SPEDAS...

- Insert a comma (,) between a command, its arguments, and keywords.

```
IDL> tplot , 1 , title='New plot'
```

- A string is expressed as a text sandwiched by delimeters (') or (").

```
IDL> print, 'This is a text.'
```

- An array is expressed as comma-separated elements that are bracketed.

```
IDL> arr1 = [ 2, 3, 4, 5 ]
IDL> string_arr1 = [ 'text1', 'text2', 'text3' ]
```

- A string + a string array gives an array of combined strings:

```
IDL> strs  = 'erg_mgf_' + [ 'bx', 'by', 'bz' ]
IDL> print, strs
    erg_mgf_bx    erg_mgf_by    erg_mgf_bz
```

A few basics of IDL before entering SPEDAS... (cont'd)

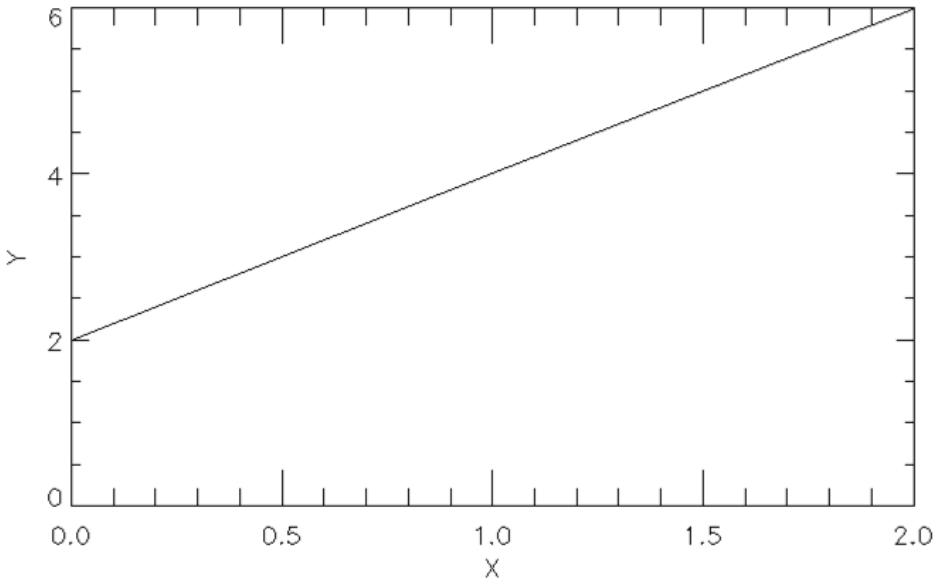
- Typical errors beginners often encounter:

  `% Attempt to call undefined procedure: '????'.`
  → command/routine name (????) is misspelled.

  `% Syntax error.`
  → , ' ( ) [ ] is missing or mismatched in most cases.

- "plot" command:

```
IDL> plot, [0, 1, 2], [2, 4, 6], xtitle='X', ytitle='Y'
```



- "append_array" command (from SPEDAS):

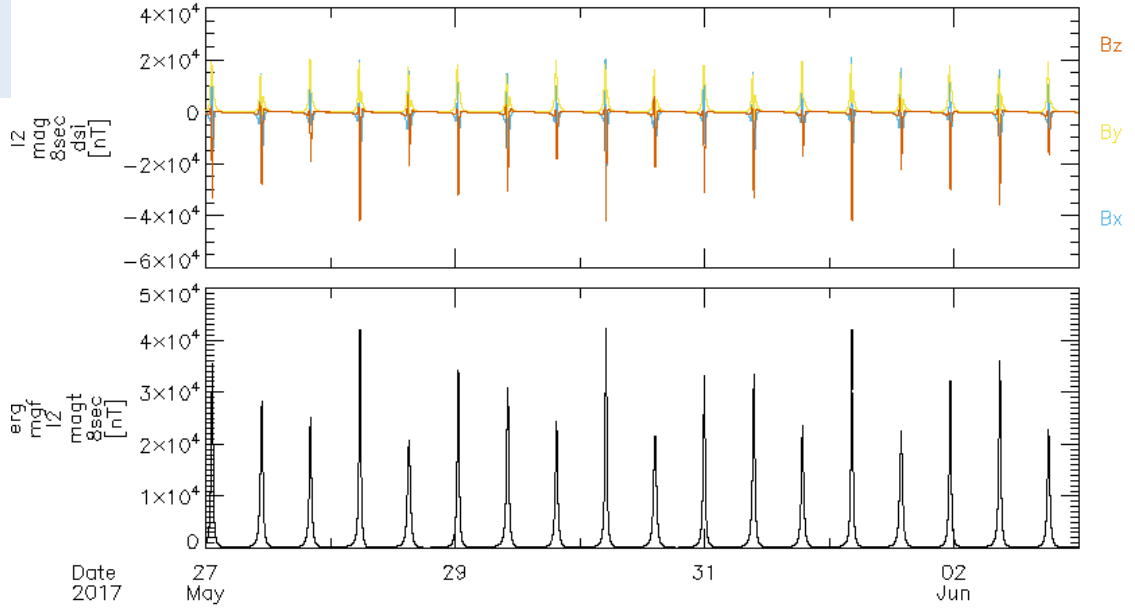
```
IDL> arr = [1, 2, 3]
IDL> append_array, arr, [4, 5, 6]  ;; combine the latter array with the former
IDL> print, arr
  1 2 3 4 5 6
```

- Use Up arrow key to reuse previously typed commands.
  You can edit them with Left/Right arrow, Backspace keys and execute!

How SPEDAS works?

One of the simplest procedures would be:

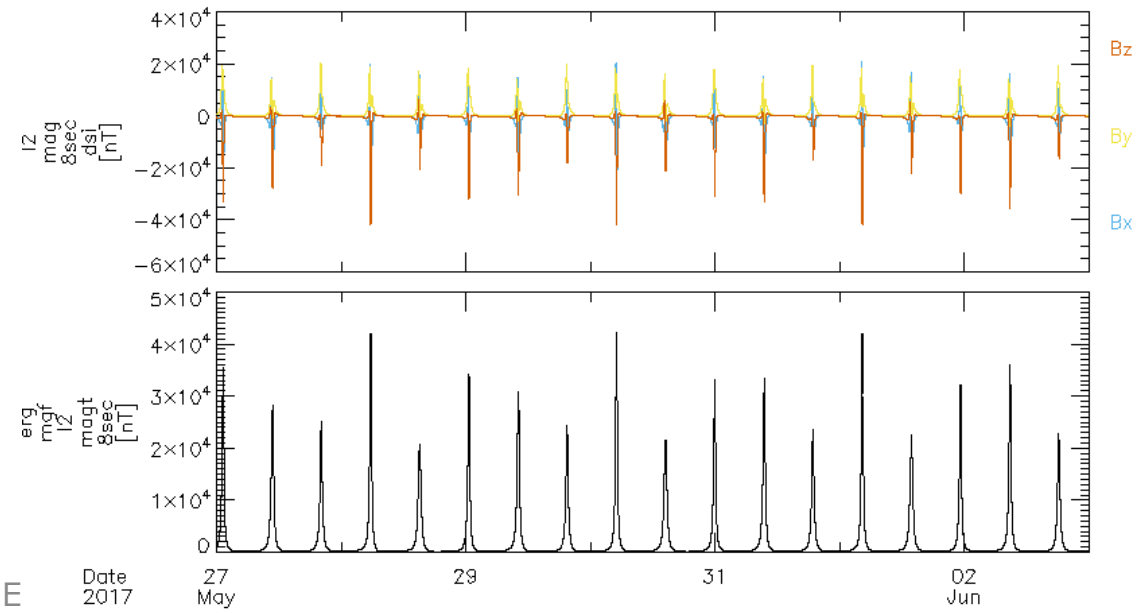
1. Run IDL
2. Initialize the SPEDAS environment on IDL
3. Set a date/time range for which data are loaded.
4. Load data
5. (Manipulate the loaded data)
6. Plot the data



How SPEDAS works? (cont'd)

In SPEDAS command lines,

```
prompt> idl
IDL> erg_init
ERG> timespan, '2017-05-27', 7, /day
ERG> erg_load_mgf
   (manipulate tplot variables)
ERG> initct, 1080, line_clrs=7  ;; colormap setting
ERG> tplot, ['erg_mgf_l2_mag_8sec_dsi', 'erg_mgf_l2_magt_8sec']
```



Set a date/time range

```
ERG> timespan, timestr, N, option
```

`timestr` : a string expressing a particular date/time in UTC in the format of 'yyyy-mm-dd/hh:mm:ss'

`N` : number of time length (Default: 1)

`option` : unit (/day, /hour, /min, /sec, Default: /day)

For 1 day from 2017-05-27/00:00:00 UTC

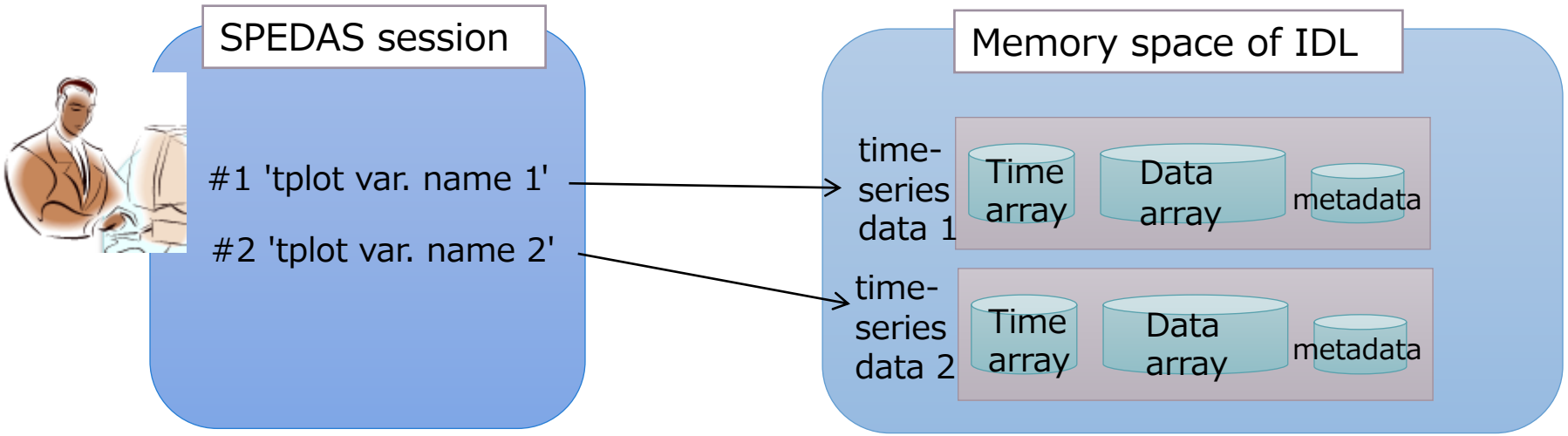
```
ERG> timespan, '2017-05-27'
```

For 90 min from 2017-05-29/03:25:30 UTC

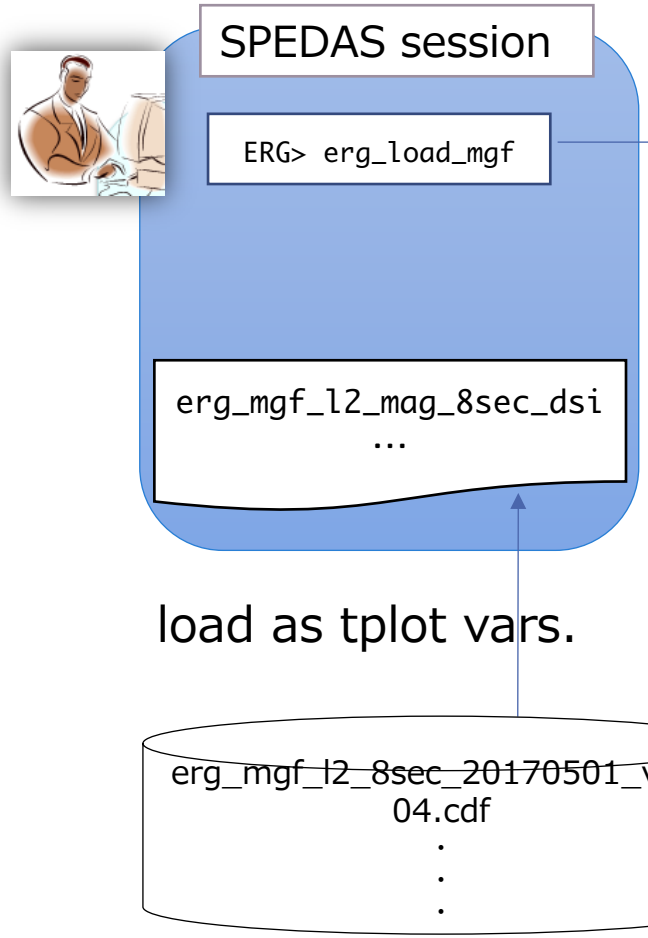
```
ERG> timespan, '2017-05-29/03:25:30', 90, /min
```

tplot variable as the primary data model

- 'erg_mgf_l2_mag_8sec_dsi' in prev. page is called ***tplot variable***.
- "Tplot variables" bind an indexed name-string to a data structure on IDL containing time-series data with metadata.



Ex: How the MGF load procedure works



Index of /data/ergsc/satellite/erg/mgf/l2/8sec

Name	Last modified	Size
Parent Directory		-
erg_mgf_l2_8sec_20170501_v01.01.cdf	2019-07-01 15:17	2.7M
erg_mgf_l2_8sec_20170501_v01.02.cdf	2019-11-06 18:44	2.7M
erg_mgf_l2_8sec_20170501_v03.03.cdf	2020-03-30 10:32	2.7M
erg_mgf_l2_8sec_20170501_v03.04.cdf	2020-06-19 22:40	2.7M
erg_mgf_l2_8sec_20170502_v01.01.cdf	2019-07-01 15:23	2.9M
erg_mgf_l2_8sec_20170502_v01.02.cdf	2019-11-06 18:49	2.9M
erg_mgf_l2_8sec_20170502_v03.03.cdf	2020-03-30 10:32	2.9M
erg_mgf_l2_8sec_20170502_v03.04.cdf	2020-06-19 22:45	2.9M
erg_mgf_l2_8sec_20170503_v01.01.cdf	2019-07-01 15:29	2.7M
erg_mgf_l2_8sec_20170503_v01.02.cdf	2019-11-06 18:53	2.7M
erg_mgf_l2_8sec_20170503_v03.03.cdf	2020-03-30 10:32	2.7M
erg_mgf_l2_8sec_20170503_v03.04.cdf	2020-06-19 22:51	2.7M
erg_mgf_l2_8sec_20170504_v01.01.cdf	2019-07-01 15:35	2.9M
erg_mgf_l2_8sec_20170504_v01.02.cdf	2019-11-06 18:58	2.9M
erg_mgf_l2_8sec_20170504_v03.03.cdf	2020-03-30 10:32	2.9M
erg_mgf_l2_8sec_20170504_v03.04.cdf	2020-06-19 22:56	2.9M

1 year latency for each ERG product

Working group password is required for the data within 1 year after observation.

Access the data structure in a tplot variable – get_data –

```
ERG> get_data, 'thg_mag_atha', data=data, dlimits=dlimits, lim=lim
          data: the data structure is stored
          dlimits: most of metadata are stored
          lim: some plot properties are stored


ERG> help, data.x , data.y
```

```
ERG> get_data, 'erg_mgf_l2_magt_8sec', data=data
ERG> help, data
** Structure <8f806478>, 2 tags, length=161968, data length=161968, refs=1:
   X               DOUBLE    Array[10123]
   Y               DOUBLE    Array[10123]
ERG>
```

"get_data" extracts the data structure of a tplot variable and saves in a structure "data" of IDL session in the above case, so that users can access them by referring to as "data.x" or "data.y", for example.

```
ERG> help, dlimits
** Structure <382fa08>, 10 tags, length=904, data length=890, refs=8:
   CDF             STRUCT    -> <Anonymous> Array[1]
   SPEC            BYTE         0
   LOG             BYTE         0
   COLORS          INT       Array[3]
   CONSTANT        FLOAT          0.00000
   LABELS          STRING    Array[3]
   LABFLAG         INT              1
   YSUBTITLE       STRING    'B (nT)'
   YTITLE          STRING    'atha'
   DATA_ATT        STRUCT    -> <Anonymous> Array[1]
```

The information on the original CDF data file (CDF) and metadata, and various plot properties are extracted into a structure "dlimits".

Create a new tplot variable – store_data

```
ERG> store_data, 'varname' , data = { x:timearr, y:datarr }
```

varname: name of a newly created tplot variable

timearr: 1-D array containing time values in SPEDAS time of time-series data

datarr: 1-D or 2-D array containing the data values of time-series data. The size of 1st dimension should be identical to that of timearr.

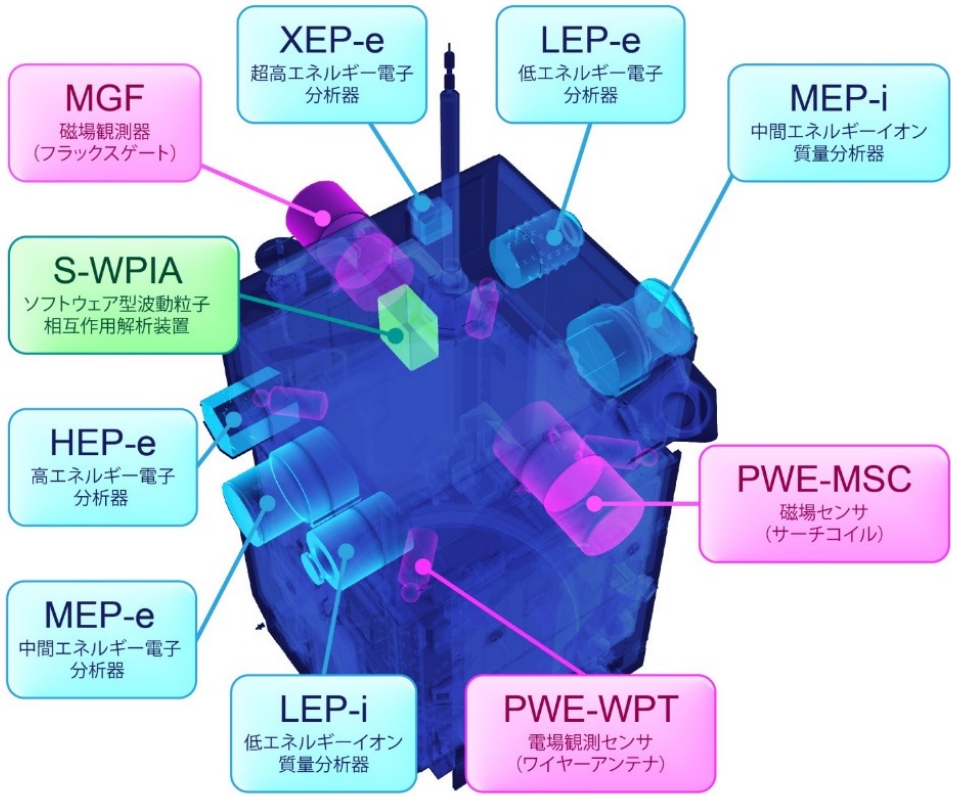
- SPEDAS time is the UNIX time in double-precision floating-point values. UNIX time is the elapsed second since 00:00 UTC on January 1, 1970.
- Usually we use `time_double()` function to calculate a SPEDAS time value from a time string such as '2017-06-16/12:30:00'.
- SPEDAS time values can easily be converted to time strings with `time_string()` function.

```
ERG> timestr='2017-06-16/12:30:00'
ERG> spedastime = time_double( timestr )
ERG> print, spedastime
   1.4976162e+09
ERG> print, time_string( spedastime )
2017-06-16/12:30:00
```

Please refer to the SPEDAS wiki at http://spedas.org/wiki/index.php?title=Time_handling for more details of the time handling in SPEDAS.

Data Products of Arase

- Overview of the ERG project
  - Miyoshi et al., EPS, 2018a
- ERG Science Center
  - Miyoshi et al., EPS, 2018b



- Orbit（predict/definitive, L2/L3）
- Attitude（L2）
- MGF（pre/L2）
  - Matsuoka et al., EPS, 2018
- PWE（pre/L2/L3）
  - Kasahara et al., EPS, 2018
  - Kumamoto et al., EPS, 2018
  - Ozaki et al., EPS, 2018
  - Matsuda et al., EPS, 2018
  - Kasaba et al., EPS, 2017
- XEP（L2/L3/L4）
- HEP (HEP-H/HEP-L, pre/ L2/L3/L4)
  - Mitani et al., EPS, 2018
- MEPi(pre/L2/L3/L4)
  - Yokota et al., EPS, 2017
- MEPe(pre/L2/L3/L4)
  - Kasahara et al., EPS, 2018
- LEPi(pre/L2/L3/L4)
  - Asamura et al., EPS, 2018
- LEPe(pre/L2/L3/L4)
  - Kazama et al., EPS, 2017

Listing tplot data & viewing the content

```
ERG> tplot_names
```

```
ERG> print_tinfo, 'erg_mgf_l2_mag_8sec_dsi'
```

```
ERG> tplot_names
   1 erg_mgf_l2_mag_8sec_dsi
   2 erg_mgf_l2_mag_8sec_gse
   3 erg_mgf_l2_mag_8sec_gsm
   4 erg_mgf_l2_mag_8sec_sm
   5 erg_mgf_l2_magt_8sec
   6 erg_mgf_l2_rmsd_8sec_dsi
   7 erg_mgf_l2_rmsd_8sec_gse
   8 erg_mgf_l2_rmsd_8sec_gsm
   9 erg_mgf_l2_rmsd_8sec_sm
  10 erg_mgf_l2_rmsd_8sec
  11 erg_mgf_l2_n_rmsd_8sec
  12 erg_mgf_l2_dyn_rng_8sec
  13 erg_mgf_l2_quality_8sec
  14 erg_mgf_l2_quality_8sec_gc
  15 erg_mgf_l2_igrf_8sec_dsi
  16 erg_mgf_l2_igrf_8sec_gse
  17 erg_mgf_l2_igrf_8sec_gsm
  18 erg_mgf_l2_igrf_8sec_sm
ERG>
```

All tplot variables are listed with unique index numbers

```
ERG> print_tinfo,'erg_mgf_l2_mag_8sec_dsi'
% Compiled module: PRINT_TINFO.
% Compiled module: IS_NUM.
*** Variable: erg_mgf_l2_mag_8sec_dsi
8 sec resolution B in DSI coordinates
** Structure <6f21748>, 2 tags, length=2421920, data length=2421920, refs=1:
   X               DOUBLE    Array[75685]
   Y               DOUBLE    Array[75685, 3]
Data format: [epoch_8sec, B in DSI]
% Compiled module: TAG_EXIST.
```

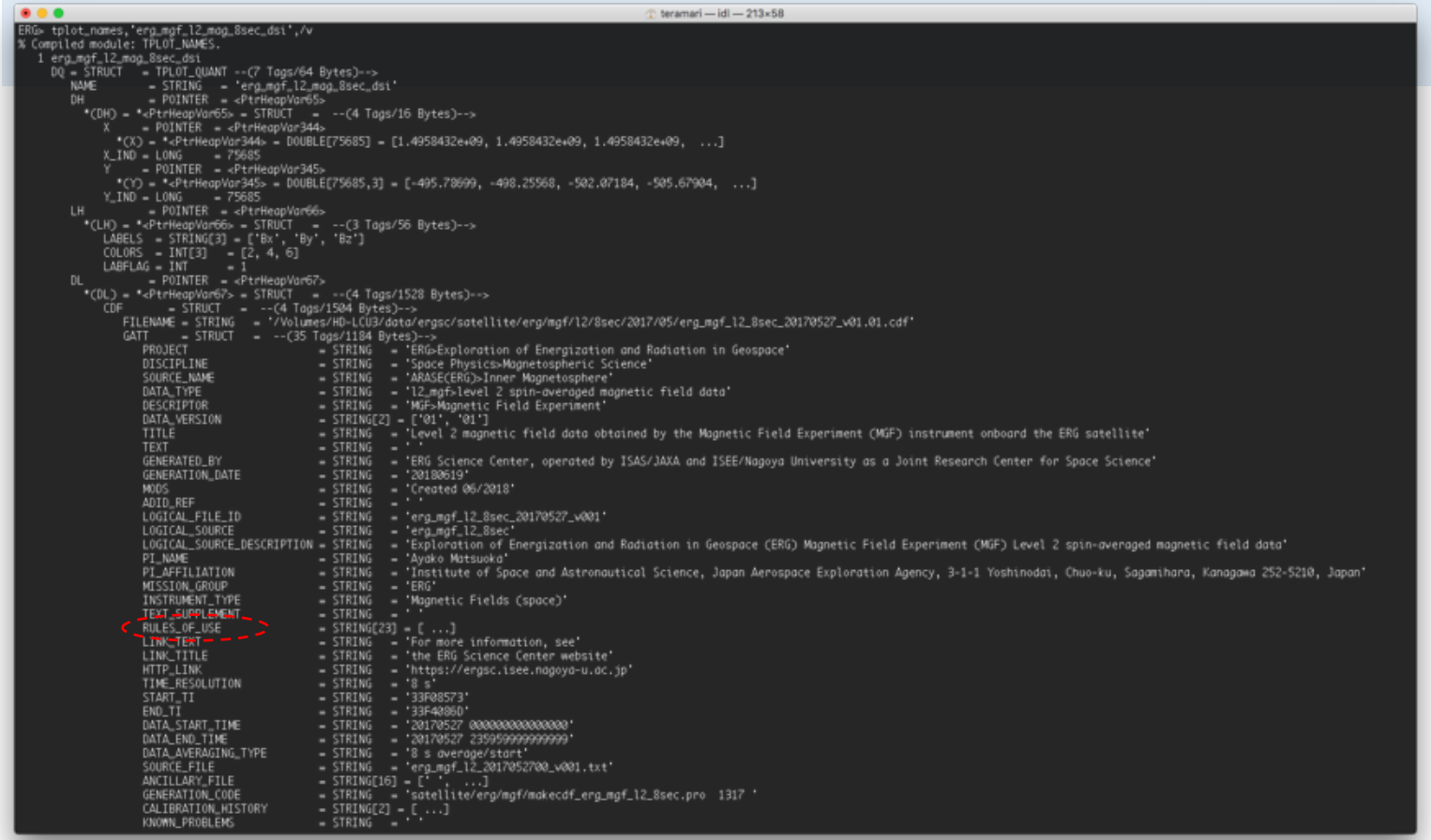
The actual data structure bound to tplot variable 'erg_mgf_l2_mag_8sec_dsi' is shown.

X: time array containing time labels in decimal UNIX time

Y: data array, in this case, a 2-D array of time x 3-components

Listing tplot data & viewing the content

```
ERG> tplot_names, 'erg_mgf_l2_mag_8sec_dsi', /verbose
ERG> tplot_names, 1, /v
```



Metadata (information on the data) are dumped.

RULES_OF_USE carries "**rules of the road**" in using the data.

Rules of the road (Statements on data policy)

PI: Ayako Matsuoka
Affiliation: Institute of Space and Astronautical Science, Japan Aerospace Exploration Agency, 3-1-1 Yoshinodai, Chuo-ku, Sagamihara, Kanagawa 252-5210, Japan

Rules of the Road for the users of the products from the ERG/Arase Project
1.Users of all level of scientific products from the ERG/Arase project should contact instrument PI team(s), Project Manager (PM),and Project Scientist (PS) before using the data for any presentation and publication. PI(s) / PM (Iku Shinohara, iku at stp.isas.jaxa.jp) / PS (Yoshizumi Miyoshi, miyoshi at isee.nagoya-u.ac.jp) may suggest potential coauthor(s) from the ERG/Arase project side for the presentation and publication. Some necessary articles suggested by the PI(s) should be cited.
2.Users should always use the latest version of data files in CDF provided from the ERG science center for their data analysis, presentation and publications. Redistribution of the data files is strictly prohibited.
3.Users should send presentation materials and papers including ERG/Arase data to instrument PI team(s), PM, and PS enough before presentation and paper submission, so that sufficient time is available for those responsible for the data to check if the data are properly processed/used and to get necessary comments back to the data users.
4. Publications that use ERG/Arase satellite data should cite the project overview paper (Miyoshi et al., Earth Planets Space, DOI:10.1186/s40623-018-0862-0, 2018) and include the following text in the paper acknowledgements: "Science data of the ERG (Arase) satellite were obtained from the ERG Science Center operated by ISAS/JAXA and ISEE/Nagoya University (https://ergsc.isee.nagoya-u.ac.jp/index.shtml.en)."
Rules of the Road of Level-2 data (MGF)
The MGF Level-2 data should be used based on a full understanding of the measurement limit. The data are produced to achieve the accuracy

- All data users should follow the Rules of the road.
- Users are encouraged to report any suspicious data to the PI. (erg_xxx_info @isee.nagoya-u.ac.jp)

Important tips for the color table

Two commands exist in SPEDAS to define a color table:

- `loadct2`
- `initct`

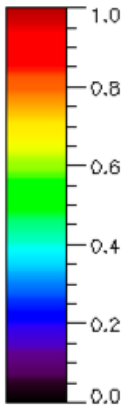
Color table setting with `loadct_sd`

`loadct_sd, num` (*num*: the number of color table)

accepted keywords:

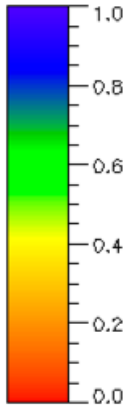
`center_hatched, hatched_width, hatched_color`

FAST-special (default)



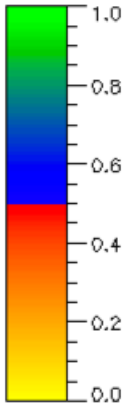
`loadct_sd, 42 or 43`

SD Cutlass



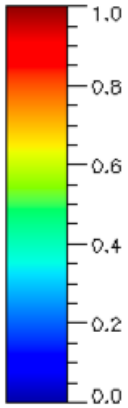
`loadct_sd, 44`

SD APL



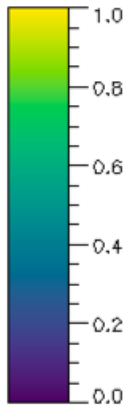
`loadct_sd, 45`

JET



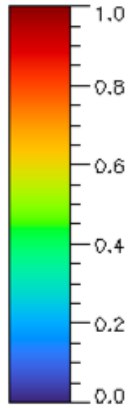
`loadct_sd, 46`

Viridis



`loadct_sd, 47`

Turbo



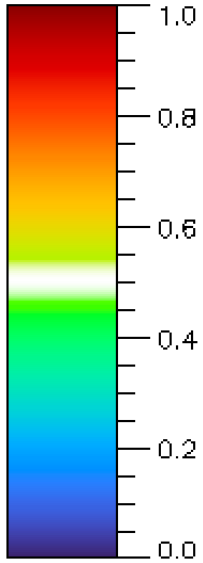
`loadct_sd, 48`

color-blind friendly

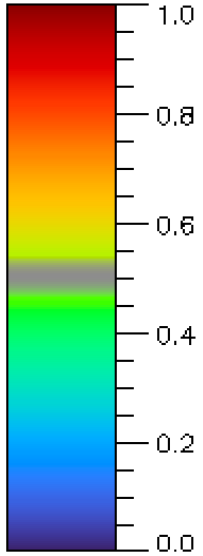
`loadct_sd` overwrites color tables of No. 43–48 with those on the left.

Color table setting with loadct_sd

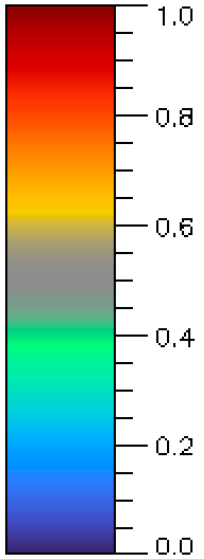
```
ERG> loadct_sd, 48, /center_hatched
ERG> loadct_sd, 48, /center_hatched, hatched_color=5
ERG> loadct_sd, 48, /center_hatched, hatched_color=5, hatched_width=60
```



/center_hatched insertss "white" around the middle of a color scale.



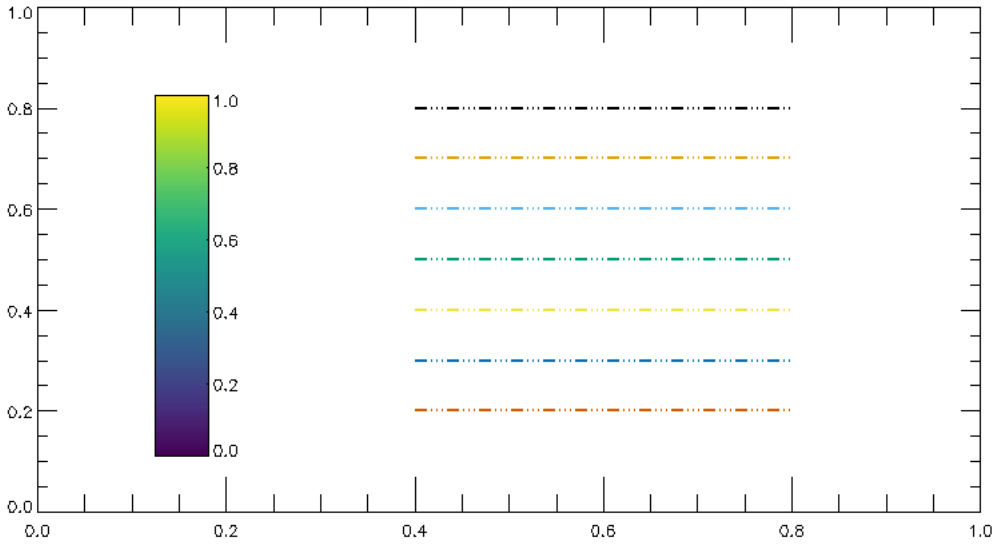
hatched_color replaces "the white" for hatching with an arbitrary color.

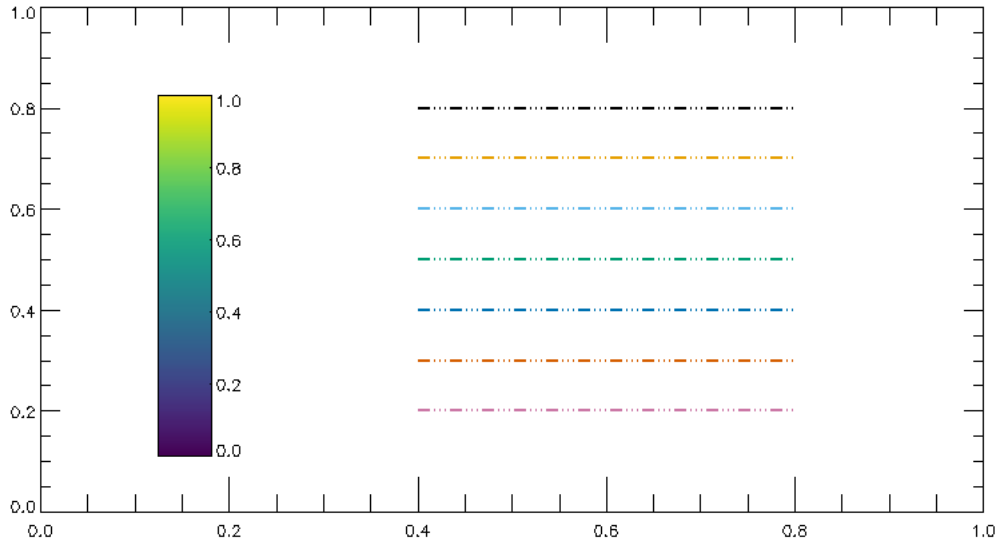


hatched_width can change the width of the hatched area. (default: hatched_width = 20)

More standardized way of setting the color table with initct

```
ERG> window
ERG> initct, 1080  ;; 1000-1074 are the same as 0-74 of loadct except 41-43, while 1075-1118 are the originals

ERG> initct, 1080, line_clrs=7
ERG> plot, [0],[0] & draw_color_scale, rang=[0,1], pos=[0.2,0.2,0.25,0.8]
ERG> for i=0, 6 do plots, [0.4,0.8],[0.8,0.8]-0.1*i, color=i, linestyle=4, thick=2

ERG> initct, 1080, line_clrs=8
ERG> plot, [0],[0] & draw_color_scale, rang=[0,1], pos=[0.2,0.2,0.25,0.8]
ERG> for i=0, 6 do plots, [0.4,0.8],[0.8,0.8]-0.1*i, color=i, linestyle=4, thick=2
```

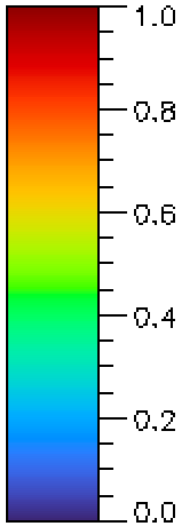




7 line colors are those optimized for color-blind individuals [Wong, Nature Methods, 2011].

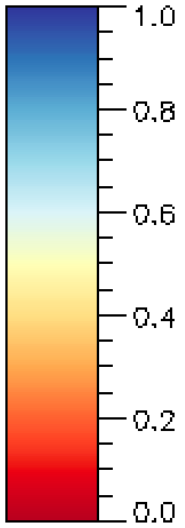
In this tutorial...

Either of the following 4 color tables are used in this training material:



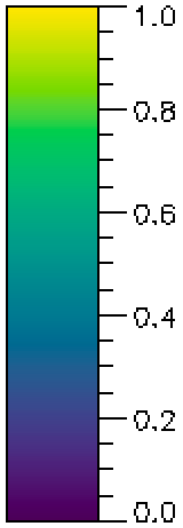
`loadct_sd, 48`

Turbo



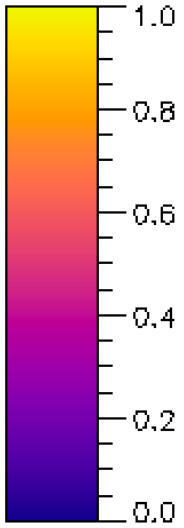
`initct, 1072`

CB-RdYlBu



`initct, 1080`

Viridis



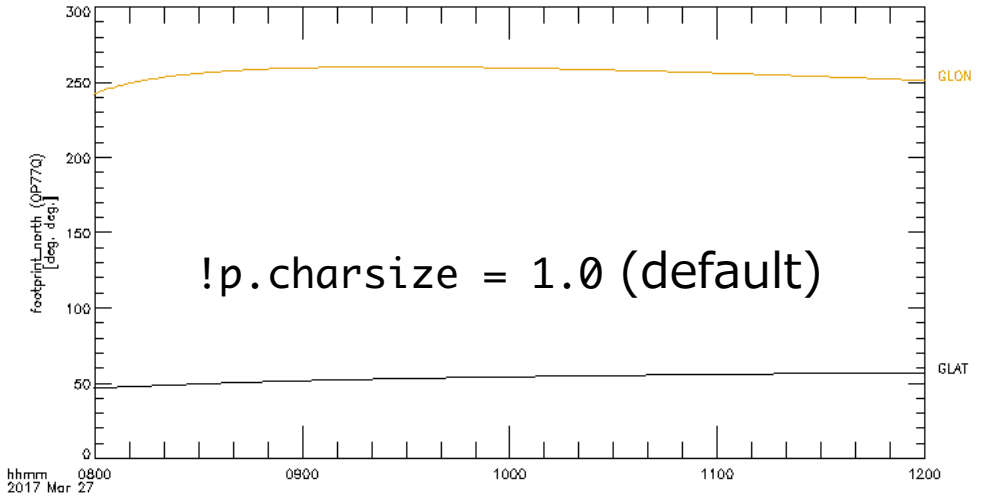
`initct, 1079`

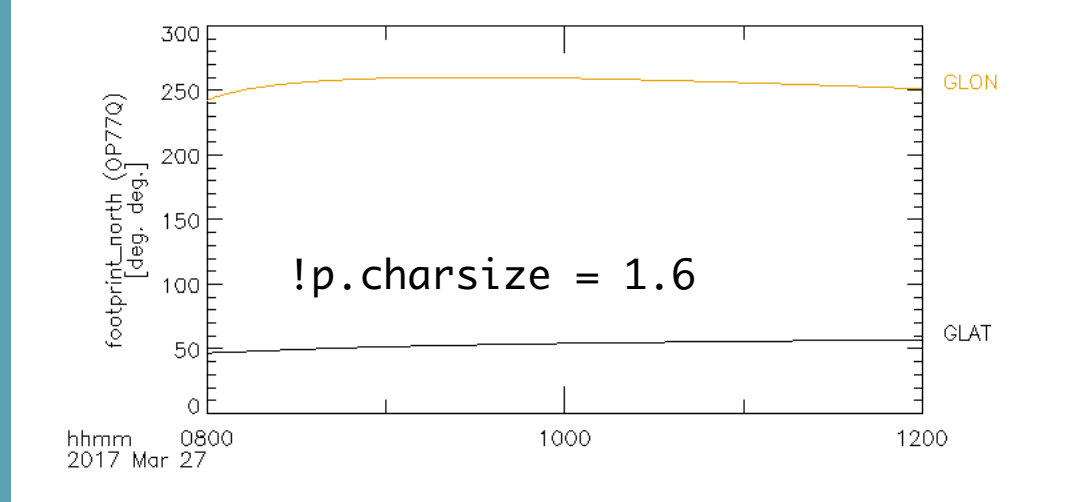
Plasma

Resources

- The color table list and how to use colormaps of the matplotlib in python:
  - IDL memo by Nishida-san, Kwasan Observatory, Kyoto Univ.
    https://www.kwasan.kyoto-u.ac.jp/~nishida/idl/defaultcolortable.html

Please pay attention to not only the color table but also the size of characters in the slides!





Manipulate and decorate plot panels

Plotting a tplot data by *tplot*

Tplot with tplot variable names (string)

```
ERG> tplot, 'erg_mgf_l2_mag_8sec_dsi'
ERG> tplot, [ 'erg_mgf_l2_mag_8sec_dsi' , 'erg_mgf_l2_mag_8sec_gse' ]
```

Tplot with the index number of a tplot variable

```
ERG> tplot, 1
```

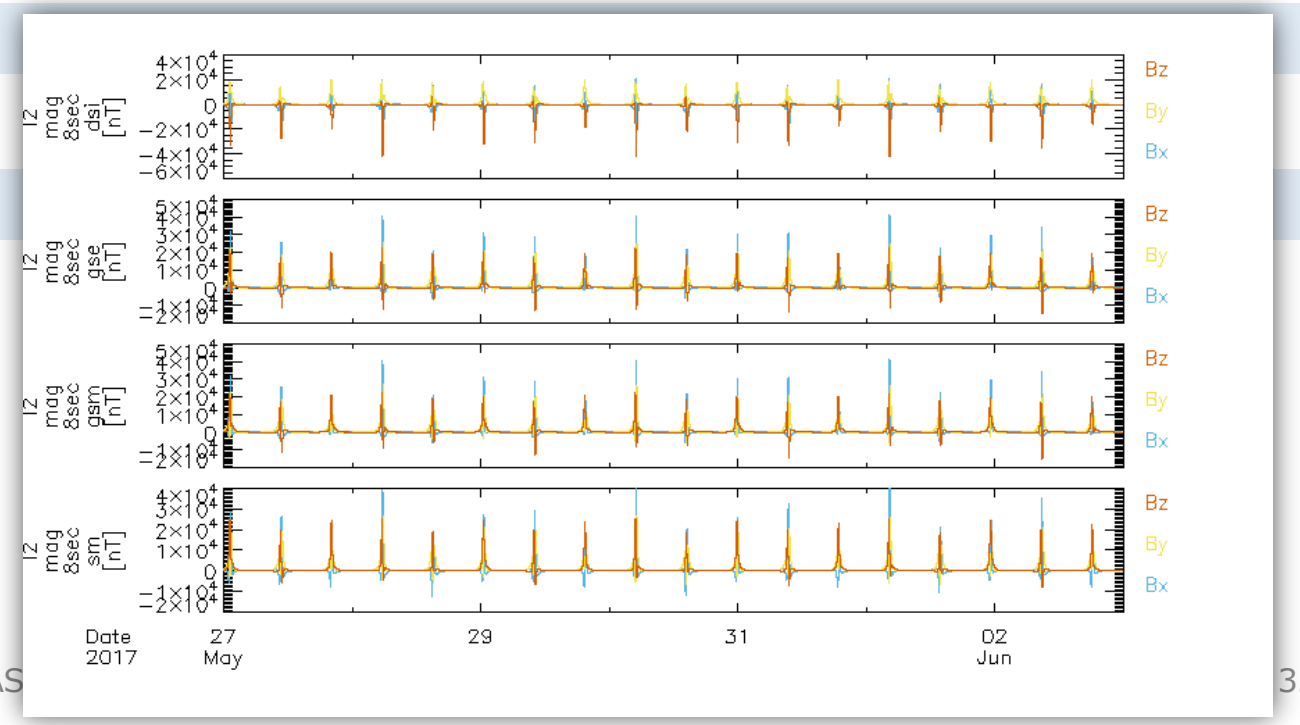
With an array of variable indices to combine multiple variable in a single plot window

```
ERG> tplot, [2,1]
```

Some wildcards can be used

```
ERG> tplot, 'erg_mgf_l2_mag_8sec_*'
```

The `tplot` command accepts variables as arguments in various formats.



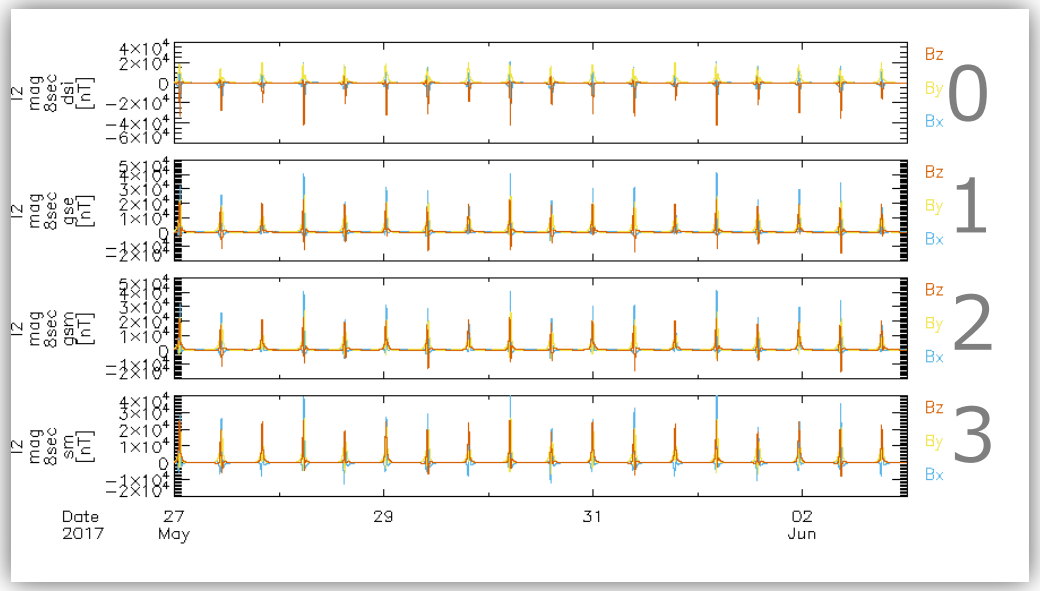
Useful manipulators for tplot: tplot_remove_panel, tnames_cp()

```
ERG> tplot, 'erg_mgf_l2_mag_8sec_*'

ERG> tplot_names, /current
 1 erg_mgf_l2_mag_8sec_dsi
 2 erg_mgf_l2_mag_8sec_gse
 3 erg_mgf_l2_mag_8sec_gsm
 4 erg_mgf_l2_mag_8sec_sm

ERG> tnames_cp( /tplot )
['erg_mgf_l2_mag_8sec_dsi','erg_mgf_l2_mag_8sec_gse','erg_mgf_l2_mag_8sec_gsm','erg_mgf_l2_mag_8sec_sm']

ERG> tplot_remove_panel, [1, 2]  ;; Remove 1st and 2nd panels and replot.
```

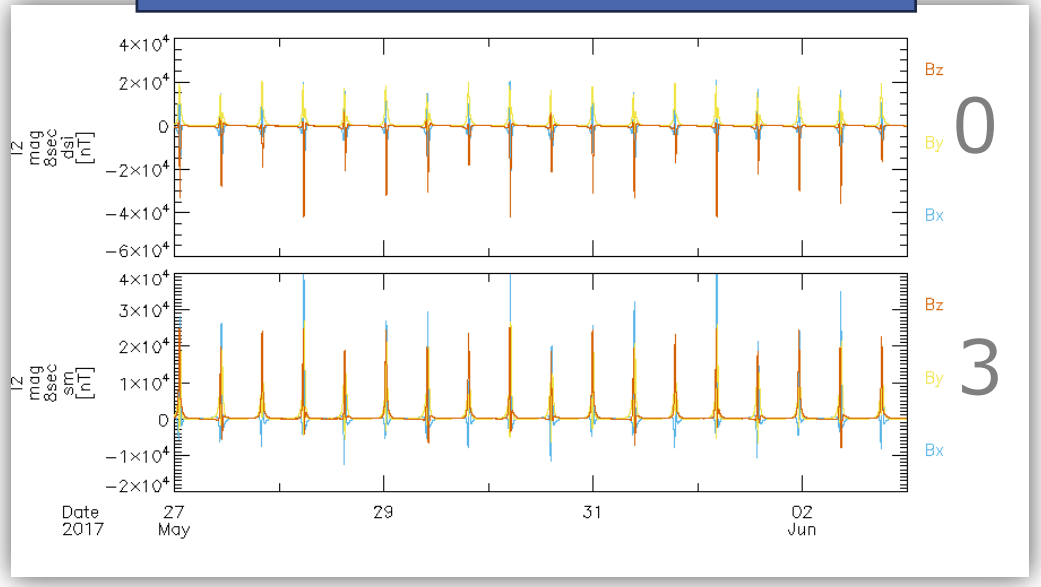
Then run `tplot, [1,4]`

Then copy & paste necessary vars and give them to the `tplot` command

Or use `tplot_remove_panel`



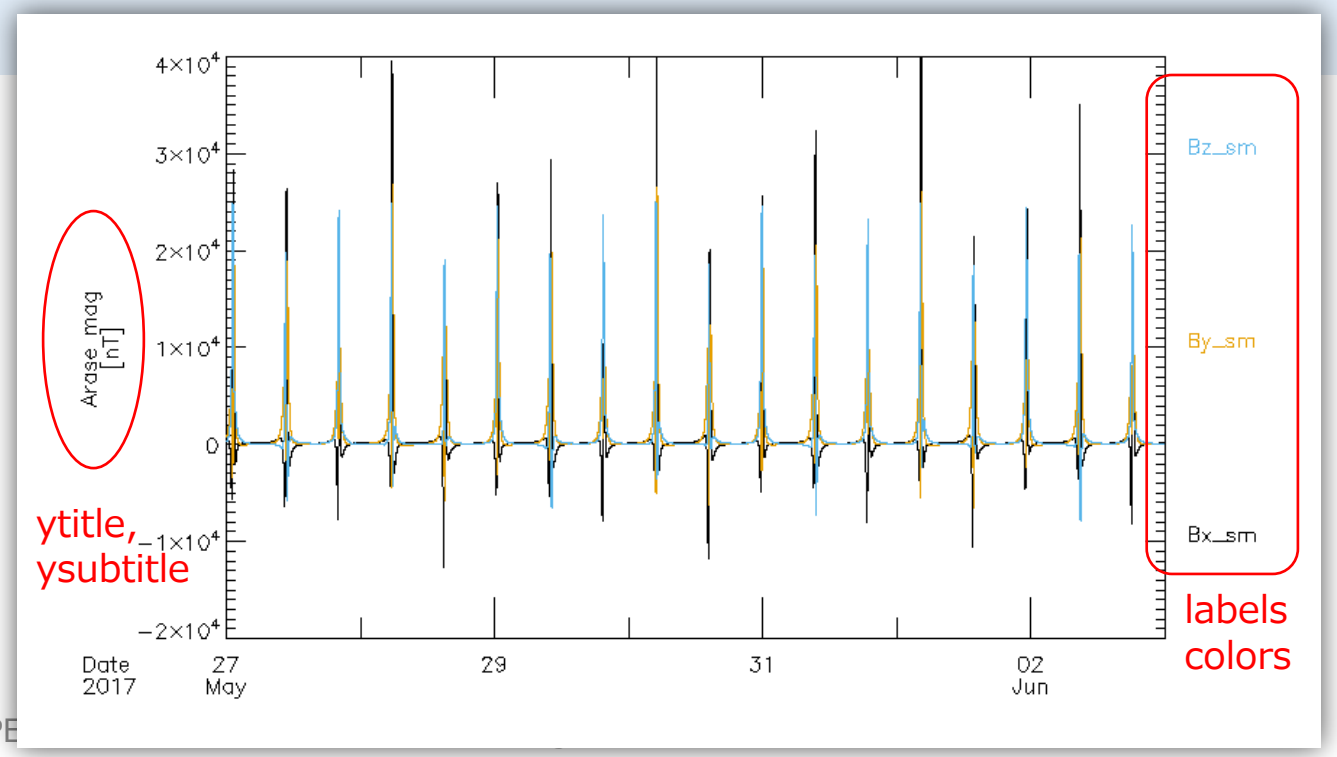
How to do this?



Decorate the plot panel for each tplot variable

```
options, varname, option1='…', option2='…', …
  varname: tplot variable name (wildcards accepted)
  option?: name of tplot variable attribute
```

```
ERG> options, 'erg_mgf_l2_mag_8sec_*', ytitle= 'Arase mag' , ysubtitle='[nT]'
ERG> options, 'erg_mgf_l2_mag_8sec_sm', labels=['Bx_sm', 'By_sm', 'Bz_sm' ]
ERG> options, 'erg_mgf_l2_mag_8sec_sm', colors=[ 0, 1, 2 ]
ERG> tplot, 'erg_mgf_l2_mag_8sec_sm'
```



Change the time range of a plot

Select a time period by mouse-clicks on the plot window

```
ERG> tlimit
```

Specify a time period explicitly

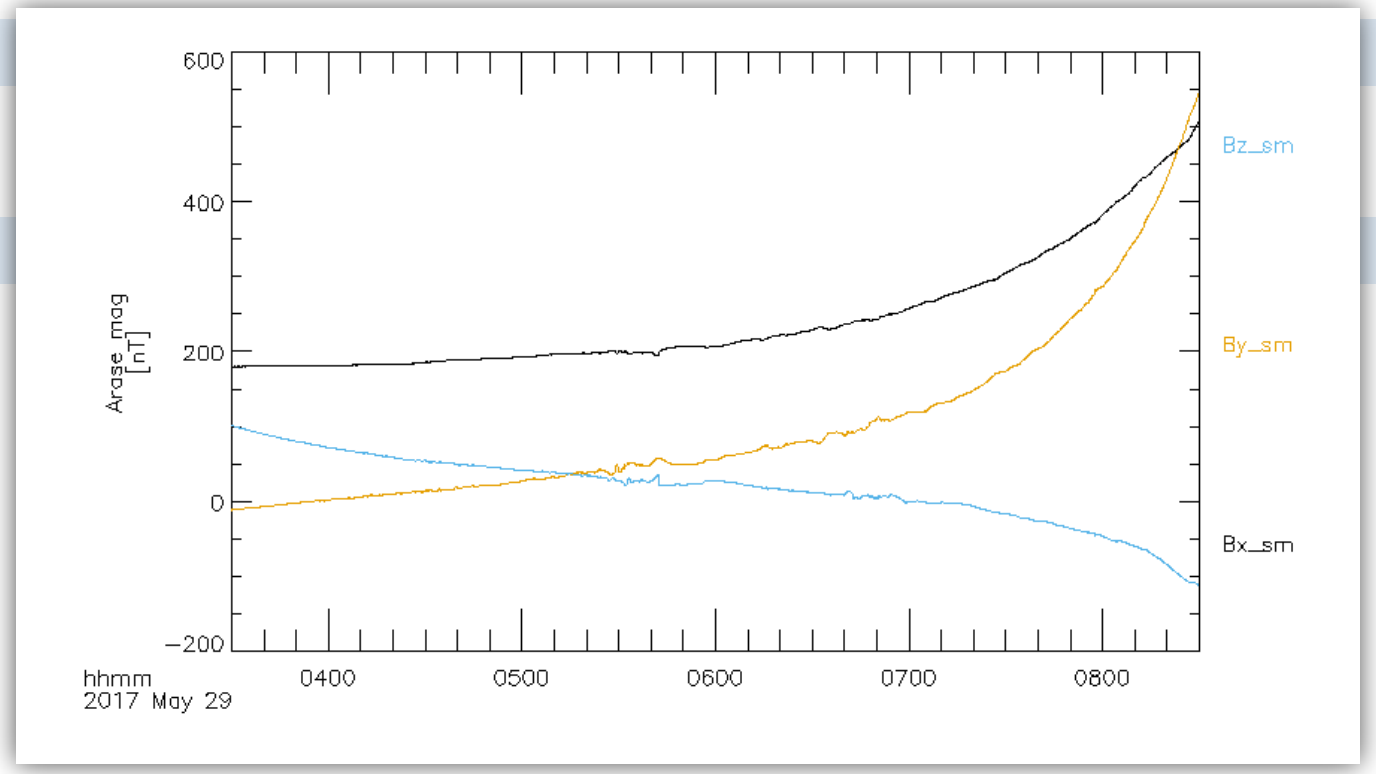
```
ERG> tlimit, '2017-05-29/03:30' , '2017-05-29/08:30'
```

Back to the last plot period

```
ERG> tlimit, /last
```

Restore the original plot period
that was set by `timespan`

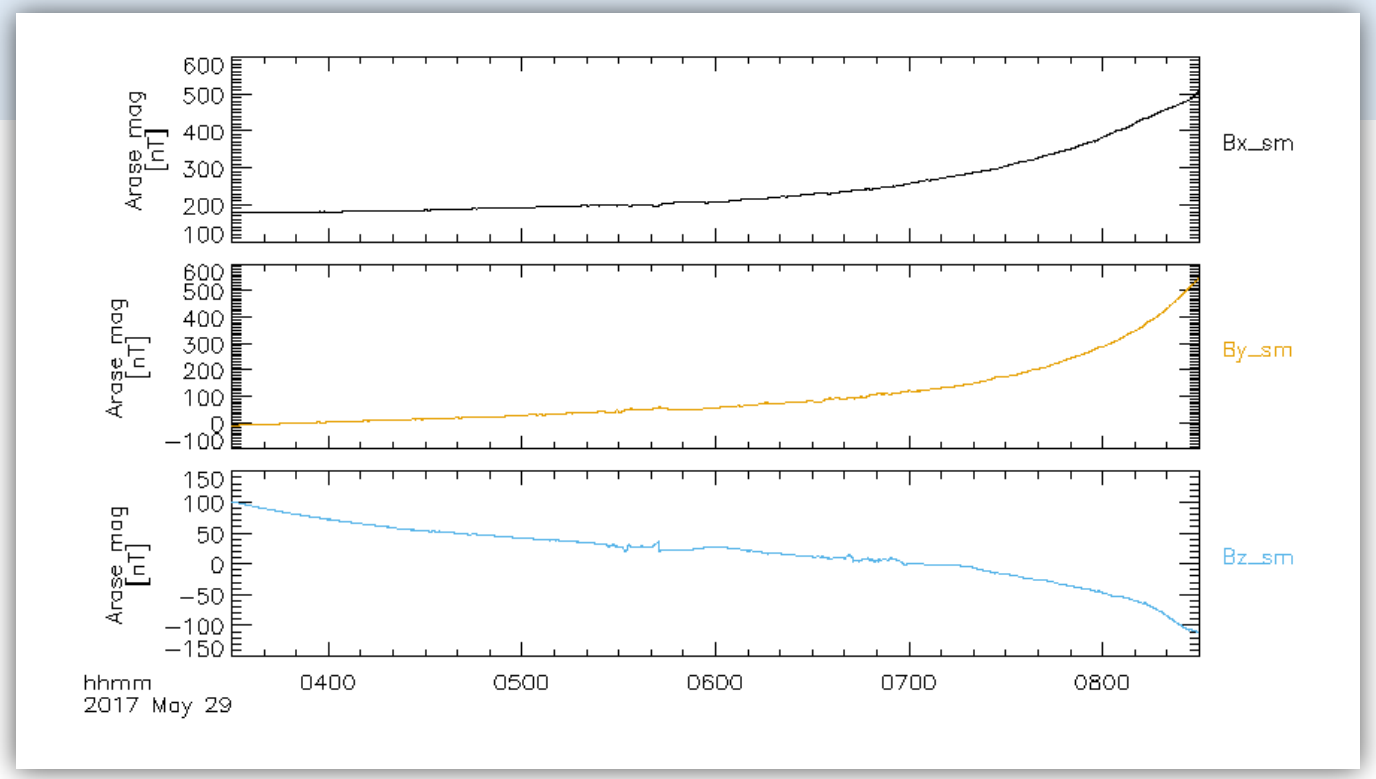
```
ERG> tlimit, /full
```



Separate a tplot variable with vector data

```
ERG> split_vec, 'erg_mgf_l2_mag_8sec_sm'
STORE_DATA(260): Creating tplot variable: 19 erg_mgf_l2_mag_8sec_sm_x
STORE_DATA(260): Creating tplot variable: 20 erg_mgf_l2_mag_8sec_sm_y
STORE_DATA(260): Creating tplot variable: 21 erg_mgf_l2_mag_8sec_sm_z


ERG> tplot, 'erg_mgf_l2_mag_8sec_sm_?'
```

`split_vec` takes a tplot variable with vector or array data to create new tplot variables containing each component of the vector/array data.

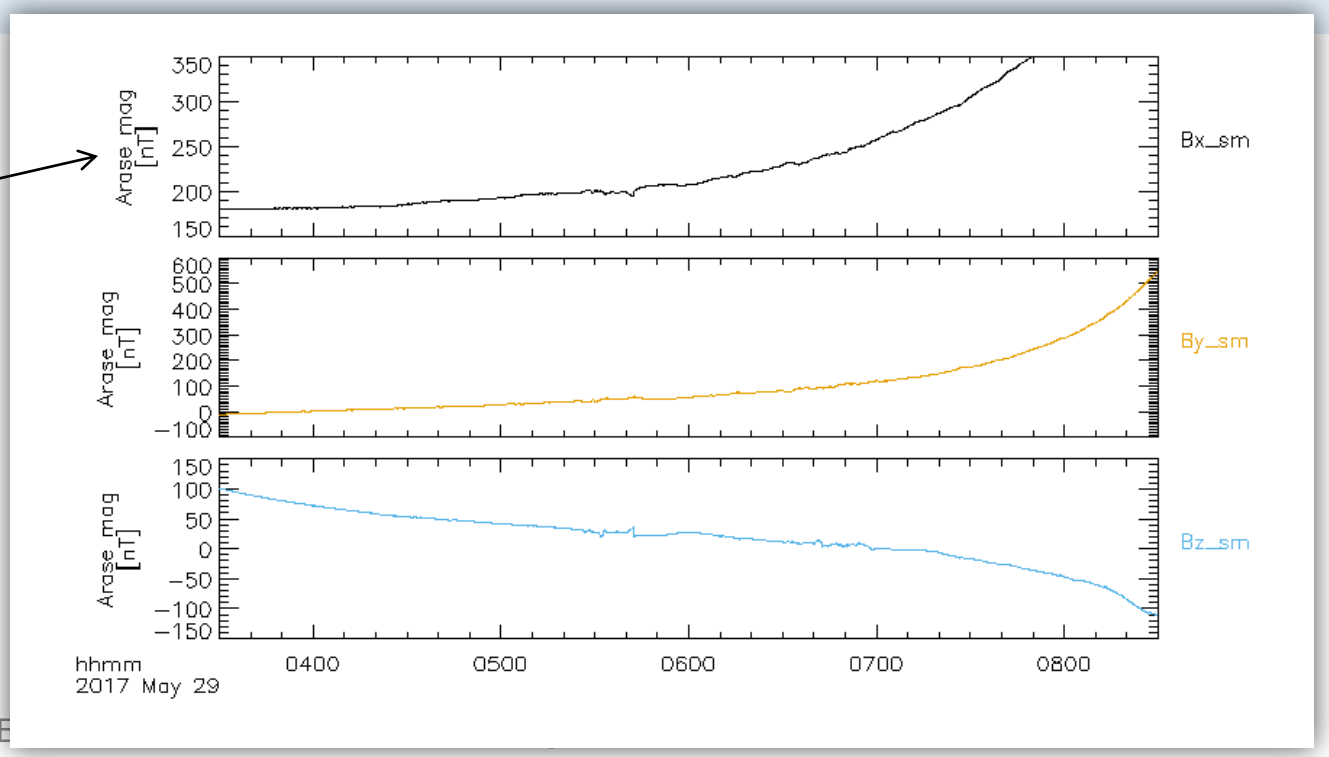


Change the vertical range of a plot

```
ylim, varname, ymin, ymax, logflag
   varname : variable name(s)
   ymin/ymax : lower/upper limit along vertical axis
                        set both to 0 (zero) for plotting with auto-scale
   logflag :  set 0 (zero) for plotting on a linear scale, or 1 for a log scale
```

```
ERG> ylim, 'erg_mgf_l2_mag_8sec_sm_x', 150,350,  0

ERG> tplot
```

Zoomed in a more limited range in the vertical scale.



Tips:
Putting 0 for both ymin and ymax sets the y range to auto-scale.

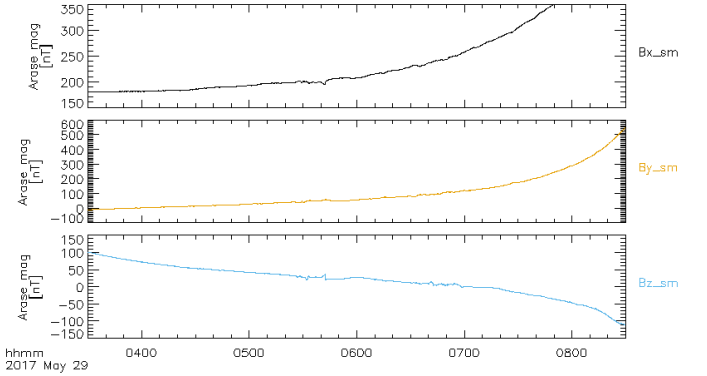
```
ERG> ylim, 'thg_mag_atha_x', 0, 0
```

Dump to png, postscript, and Ascii files

To a png file or postscript file.

```
ERG> cwd          ;Display the current directory
CWD(25): Directory changed to: /yyyy/xxxx
ERG> tplot, 'erg_mgf_l2_mag_8sec_sm_?'
ERG> makepng, 'erg_mgf_plot'      ;→ erg_mgf_plot.png


ERG> popen, 'erg_mgf_plot'
ERG> tplot          ;Redo the last plot
ERG> pclose       ; → erg_mgf_plot.ps
```

/yyyy/xxxx/erg_mgf_plot.png



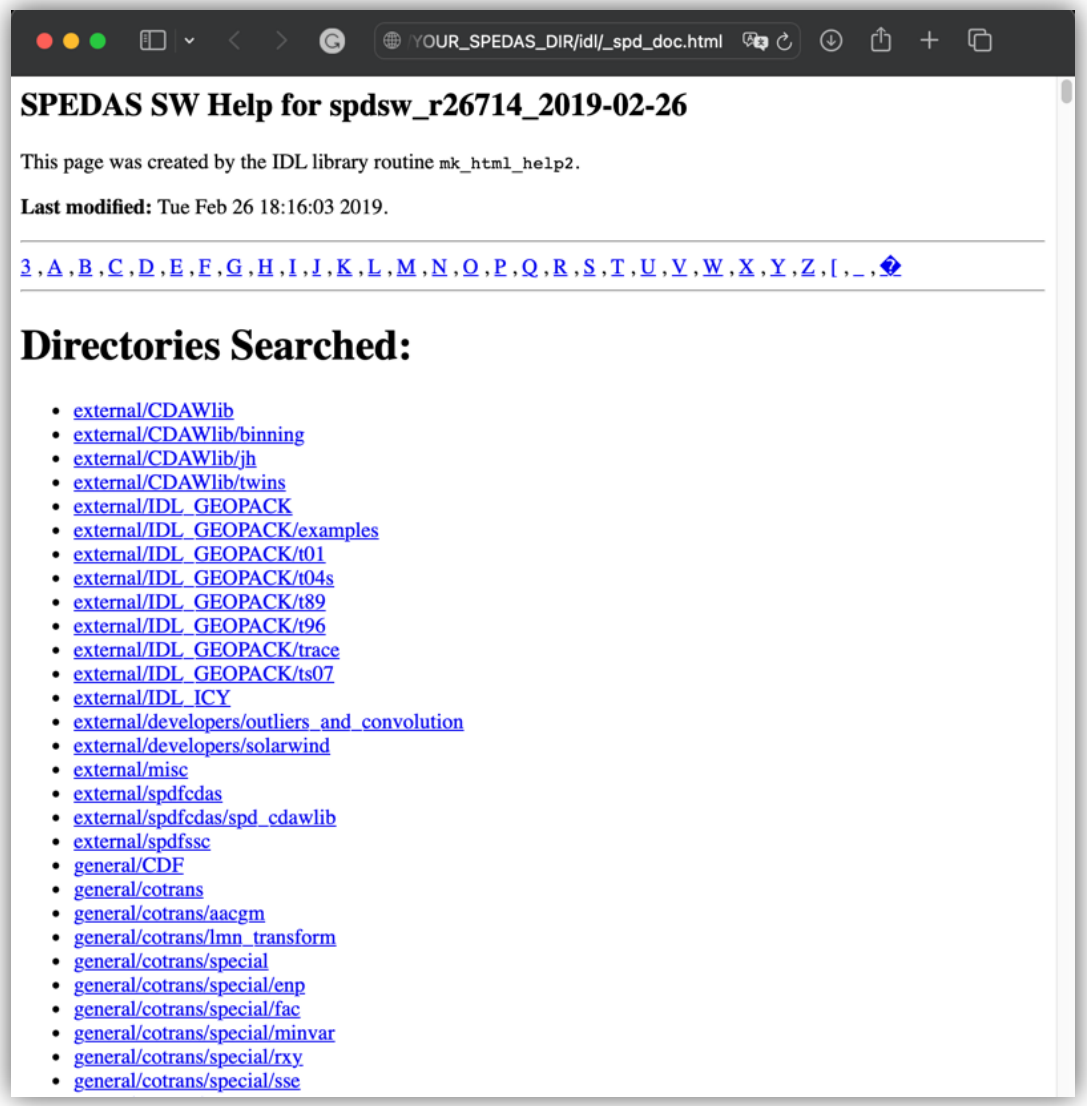
Dump the data content of a tplot variable to a Ascii file.

```
ERG> tplot_ascii, 'erg_mgf_l2_mag_8sec_gse'
                              ;--> erg_mgf_l2_mag_8sec_gse.txt
```

```
kawa3:~ teramari$ cat erg_mgf_l2_mag_8sec_gse.txt
2017-05-27/00:00:05.450         4.8450809e+02         7.9739140e+02        -6.6735940e+01
2017-05-27/00:00:13.435         4.8518178e+02         8.0074480e+02        -6.6972197e+01
2017-05-27/00:00:21.420         4.8546078e+02         8.0424135e+02        -6.9134306e+01
2017-05-27/00:00:29.421         4.8576370e+02         8.0771844e+02        -7.0975031e+01
2017-05-27/00:00:37.406         4.8738432e+02         8.1101906e+02        -6.7087595e+01
2017-05-27/00:00:45.406         4.8773360e+02         8.1453454e+02        -6.8856399e+01
2017-05-27/00:00:53.388         4.8791086e+02         8.1808762e+02        -7.1734257e+01
2017-05-27/00:01:01.388         4.8966219e+02         8.2141671e+02        -6.6654895e+01
2017-05-27/00:01:09.373         4.8989448e+02         8.2504219e+02        -6.8972403e+01
2017-05-27/00:01:17.358         4.9012297e+02         8.2864665e+02        -7.1513794e+01
2017-05-27/00:01:25.359         4.9170651e+02         8.3205503e+02        -6.7162655e+01
2017-05-27/00:01:33.344         4.9212651e+02         8.3568878e+02        -6.8484317e+01
2017-05-27/00:01:41.344         4.9235189e+02         8.3936319e+02        -7.1001025e+01
2017-05-27/00:01:49.326         4.9343653e+02         8.4292241e+02        -6.9100879e+01
2017-05-27/00:01:57.310         4.9437997e+02         8.4652938e+02        -6.8052480e+01
2017-05-27/00:02:05.311         4.9460349e+02         8.5029266e+02        -7.0758300e+01
2017-05-27/00:02:13.296         4.9510470e+02         8.5396844e+02        -7.1595251e+01
2017-05-27/00:02:21.297         4.9673170e+02         8.5753736e+02        -6.7421116e+01
2017-05-27/00:02:29.282         4.9693760e+02         8.6134959e+02        -7.0103764e+01
2017-05-27/00:02:37.267         4.9711743e+02         8.6515922e+02        -7.2860997e+01
2017-05-27/00:02:45.264         4.9883043e+02         8.6878014e+02        -6.8160833e+01
2017-05-27/00:02:53.248         4.9923304e+02         8.7263199e+02        -6.9972202e+01
2017-05-27/00:03:01.249         4.9946922e+02         8.7649747e+02        -7.2652833e+01
```

SPEDAS manual viewed with a web browser

Open /YOUR_SPEDAS_DIR/idl/_spd_doc.html with your web browser to view the automatically generated documents for SPEDAS routines.



Basics of SPEDAS: Manipulate tplot variables and apply filters to them

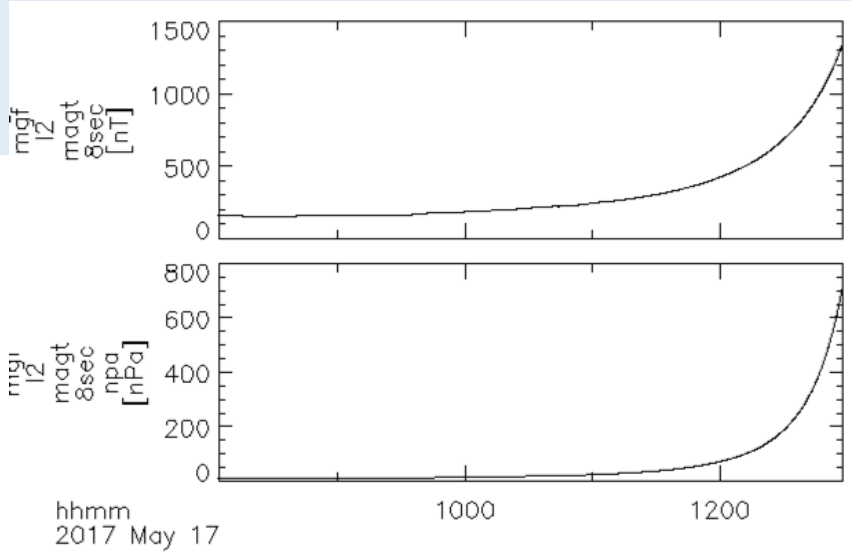
"calc" to make calculations using tplot variables

```
calc, ' … … … … … …'
```

where '… … ' is a formula with tplot variables

```
ERG> timespan, '2017-05-29'
ERG> erg_load_mgf
ERG> calc, ' "erg_mgf_l2_magt_8sec_npa" = "erg_mgf_l2_magt_8sec"^2 /2/(4*!pi*1e-7) * 1e-9' ;; to in nPa
ERG> options, 'erg_mgf_l2_magt_8sec_npa', ysubtitle='[nPa]'
ERG> tplot, ['erg_mgf_l2_magt_8sec*']
```

In this example, the magnetic pressure ($|B|^2/2\mu$) is calculated from the B-field magnitude obtained by the MGF instrument.



"tinterpol_mxn" to interpolate a tplot var for another tplot var

```
IDL> erg_load_pwe_hfa, level='l3'     ;; load the e- density data deduced from UHR measurements
IDL> tinterpol_mxn, 'erg_pwe_hfa_l3_1min_ne_mgf', 'erg_mgf_l2_magt_8sec_npa'
IDL> get_data, 'erg_pwe_hfa_l3_1min_ne_mgf_interp', t_ne, dat_ne
IDL> get_data, 'erg_mgf_l2_magt_8sec_npa', t_mag, dat_mag
IDL> help, t_ne, t_mag, dat_ne, dat_mag
T_NE            DOUBLE    = Array[10123]
T_MAG           DOUBLE    = Array[10123]
DAT_NE          DOUBLE    = Array[10123]
DAT_MAG         DOUBLE    = Array[10123]
```

The number of time frame of the HFA Ne data (1-min values) is quite different from that of the MGF data (8-s values).

By applying tinterpol_mxn, you can get an interpolated Ne data having time frames common to the MGF data!

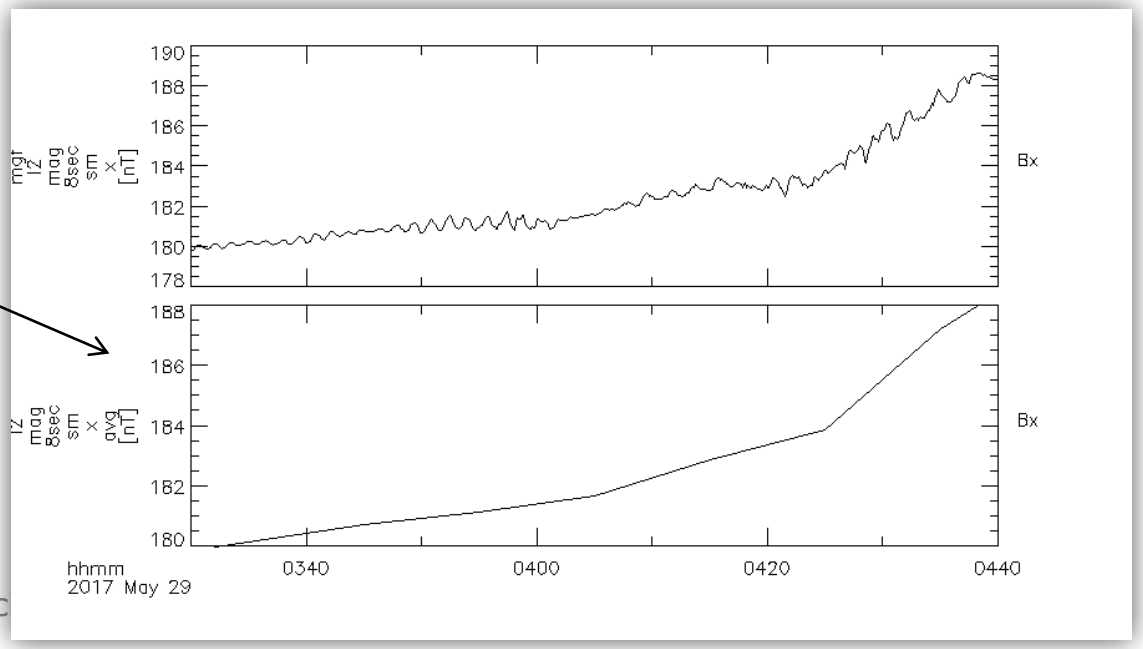
boxcar-average data– avg_data –

```
avg_data, 'varname', timebin
     varname: tplot variable names or index numbers
     timebin : a time window in sec with which the boxcar-averaging is applied to the data
```

```
ERG> del_data, '*'
ERG> timespan, '2017-05-29-03:30',70,/min
ERG> erg_load_mgf  & options, 'erg_mgf_l2_mag_8sec_sm', colors=[0, 1, 2]
ERG> split_vec, 'erg_mgf_l2_mag_8sec_sm'
ERG> avg_data, 'erg_mgf_l2_mag_8sec_sm_x' , 600.
ERG> tplot, ['erg_mgf_l2_mag_8sec_sm_x', 'erg_mgf_l2_mag_8sec_sm_x_avg' ]
```

Remove all tplot variables and reload the data

The data boxcar-averaged with a time bin of 600 second



As a result, the number of data points is reduced to every 600 s.

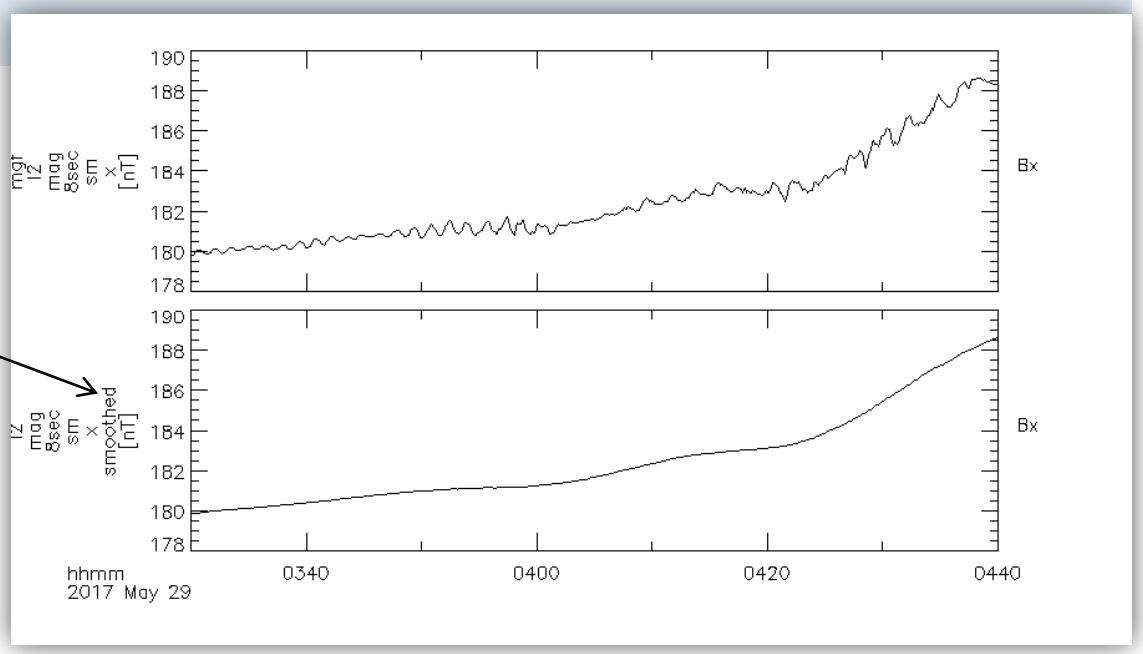
Smoothing data – tsmooth_in_time –

```
tsmooth_in_time, 'varname', timebin
   varname : tplot variable name(s)
   timebin : time window in second for running average
```

```
ERG> tsmooth_in_time, 'erg_mgf_l2_mag_8sec_sm_x', 600.
ERG> tplot_names

     … … … …
      19 erg_mgf_l2_mag_8sec_sm_x
     … … … …
      23 erg_mgf_l2_mag_8sec_sm_x_smoothed
ERG> tplot, [ 19 , 23 ]
```

The data is running-averaged with a time window of 600 second. We can use this as a rough low-pass filter.

Note that the number of data points is conserved.



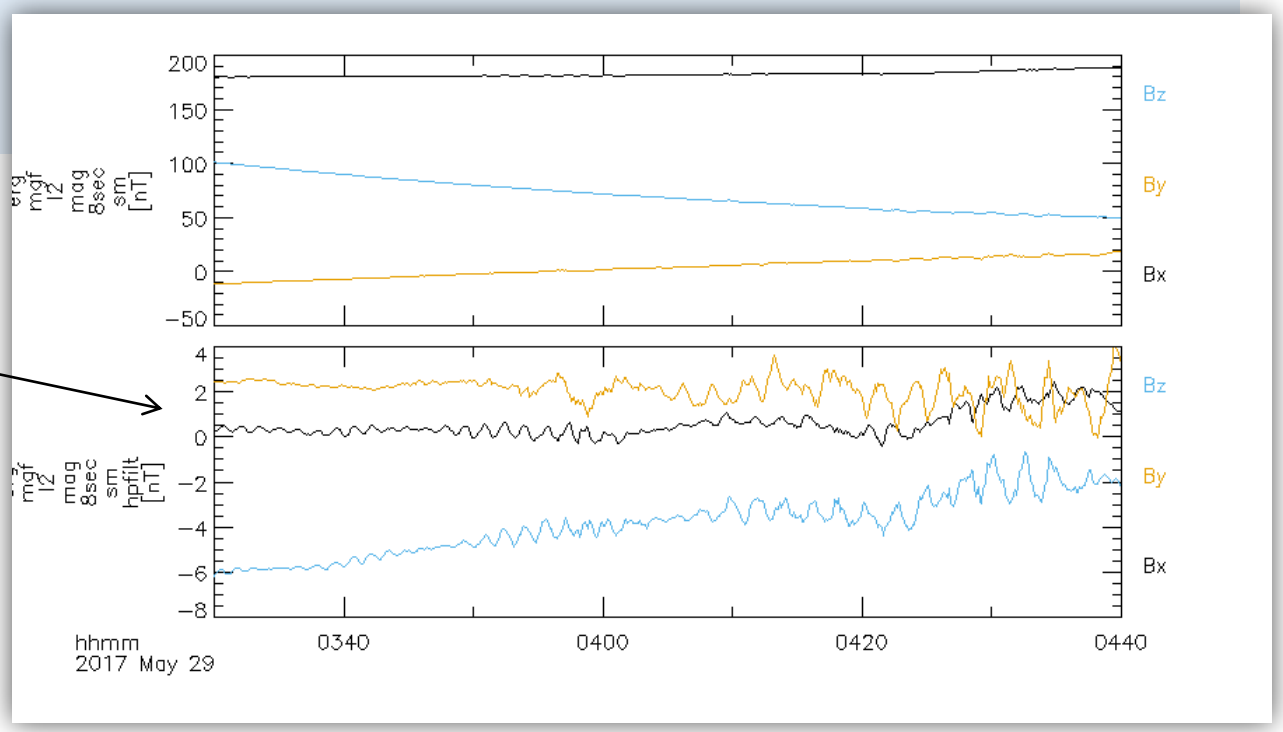
High-pass filter in time – thigh_pass_filter –

```
thigh_pass_filter, 'varname', timebin
   varname : tplot variable name(s)
   timebin : time window in second for running average
```

```
ERG> thigh_pass_filter, 'erg_mgf_l2_mag_8sec_sm', 600.
ERG> tplot_names
  ... ... ... ...
   4 erg_mgf_l2_mag_8sec_sm
  ... ... ... ...
  24 erg_mgf_l2_mag_8sec_sm_hpfilt
ERG> tplot, [ 4 , 24 ]
```

Time variations with periods shorter than 600 sec are shown.

Actually this command just subtracts the low-pass-filtered values derived with tsmooth_in_time from the original data, not uses any digital filtering process such as FFT.



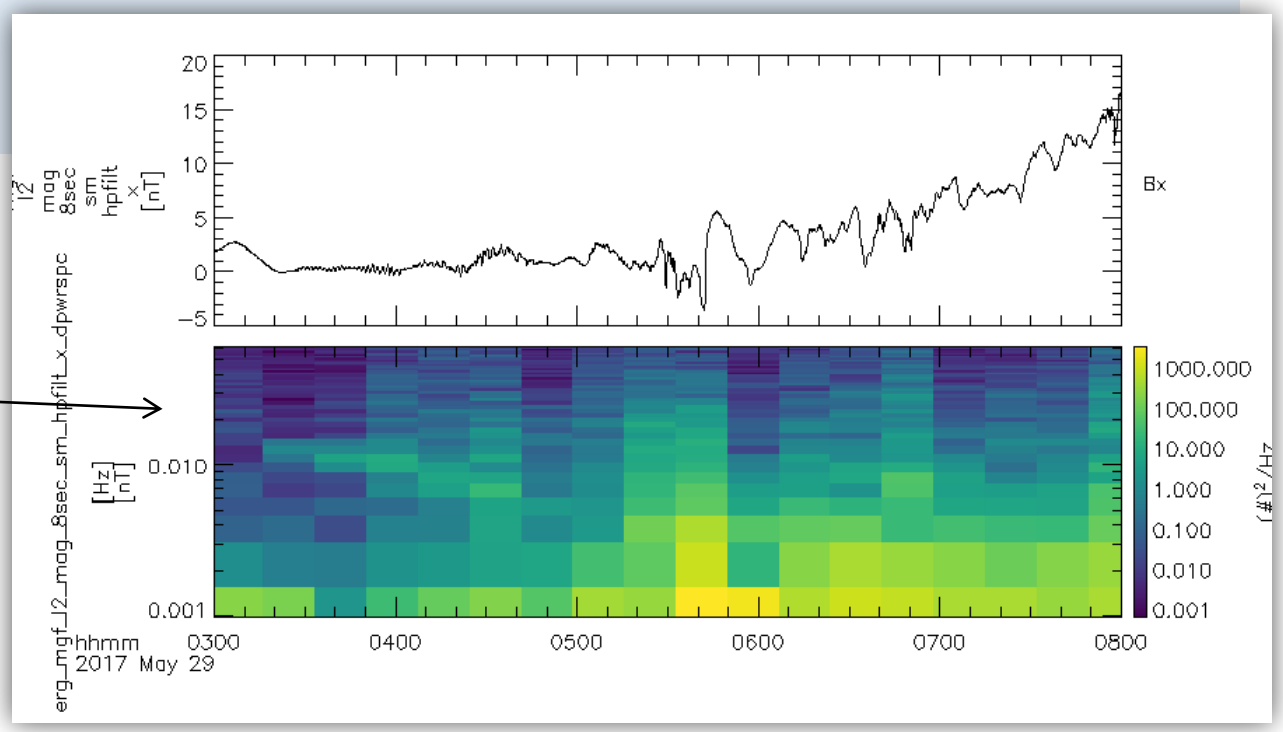
Basics of SPEDAS: Frequency analysis of tplot data

Dynamics spectra – tdpwrspc–

tdpwrspc, 'varname'
varname : tplot variable name(s)

```
ERG> tdpwrspc, 'erg_mgf_l2_mag_8sec_sm_hpfilt'
ERG> tplot_names

    … … … …
     25 erg_mgf_l2_mag_8sec_sm_hpfilt_x
    … … … …
     28 erg_mgf_l2_mag_8sec_sm_hpfilt_x_dpwrspc
ERG> tplot, [ 25, 28 ]
ERG> tlimit,'2017-05-29/03:00','2017-05-29/08:00'
```

FFT with the hanning window is applied to derive dynamic frequency spectra of the data.



Wavelet analysis – wav_data –

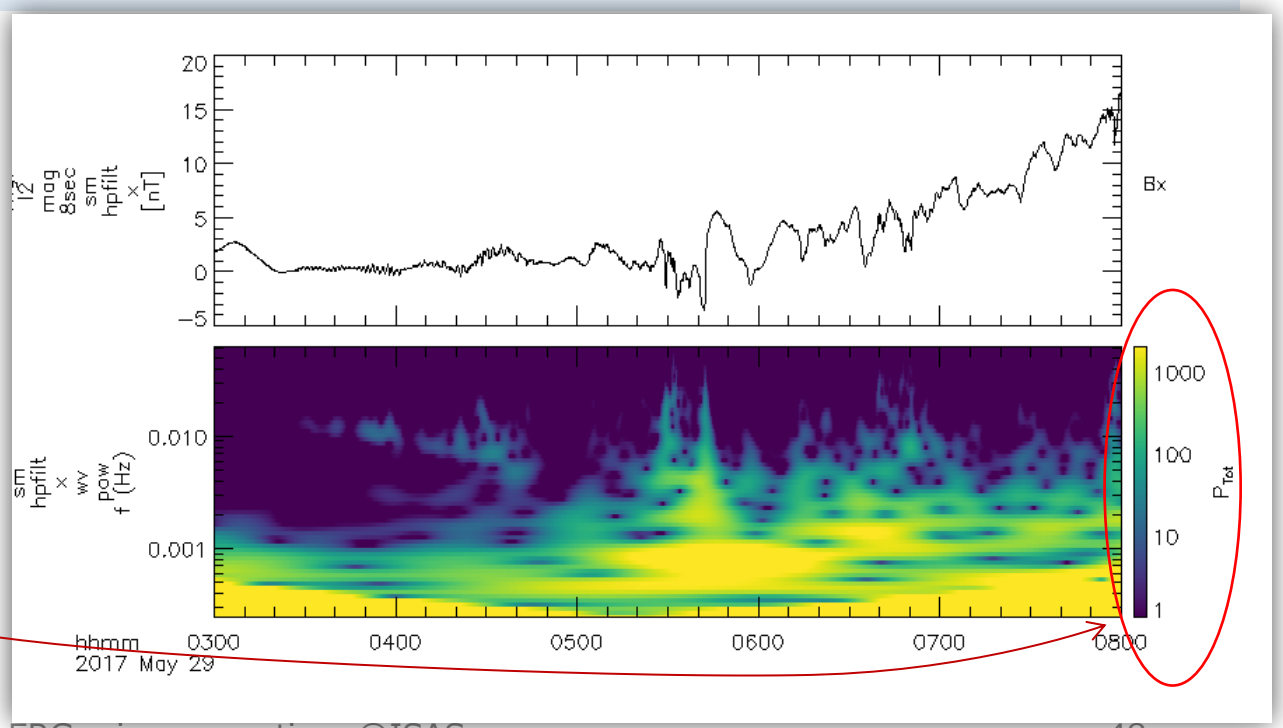
wav_data, 'varname'
varname : tplot variable name(s)

wav_data accepts data with less than 32768 samples. The number of data points is reduced as done with avg_data in this case.

```
ERG> wav_data, 'erg_mgf_l2_mag_8sec_sm_hpfilt_x'
STORE_DATA(260): Creating tplot variable: 31 erg_mgf_l2_mag_8sec_sm_hpfilt_x_wv_pow
ERG> zlim, 31, 1, 2000, 1
ERG> tplot, ['erg_mgf_l2_mag_8sec_sm_hpfilt_x','erg_mgf_l2_mag_8sec_sm_hpfilt_x_wv_pow']
```

Wavelet analysis is applied to derive dynamic spectra of the data.

zlim is similar to "ylim" command, but set the lower/upper limit of the color scale for a spectrum-type plot.

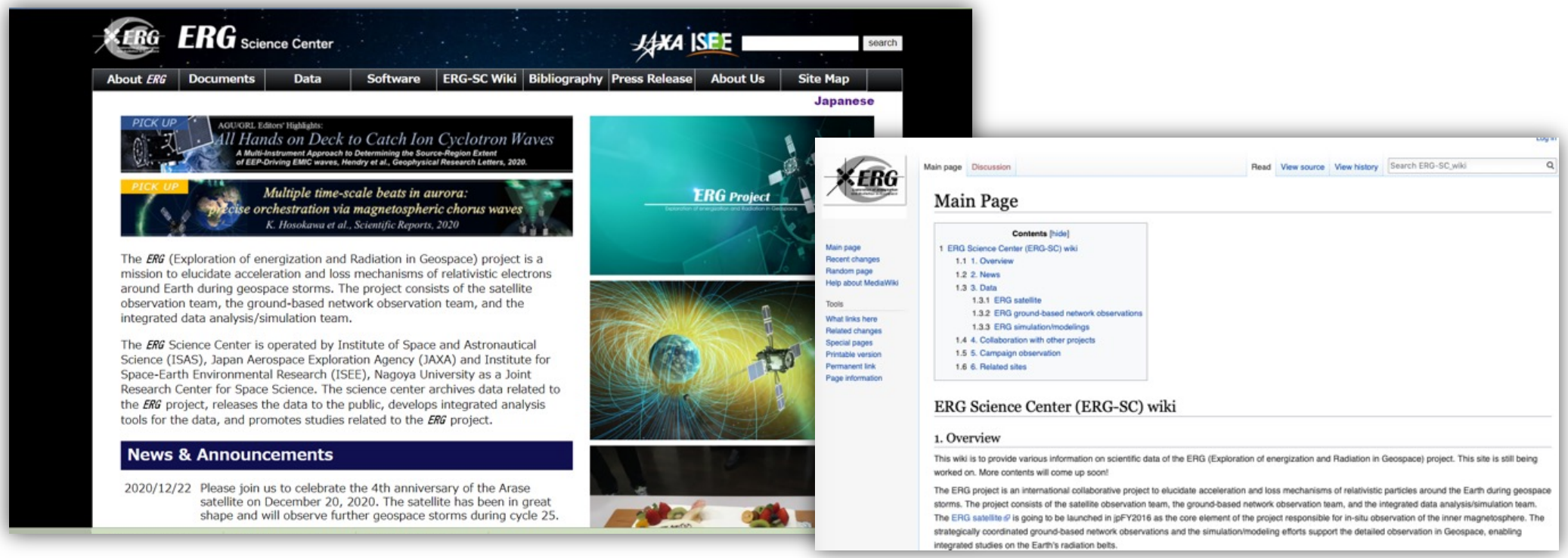


Other information sources for SPEDAS

- SPEDAS wiki
  - http://spedas.org/wiki/index.php?title=Main_Page
    - User's guide, Plug-in developer's guide, tips and tricks, The list of available crib sheets, …
- Change log of the source repository for the bleeding edge of SPEDAS
  - http://spedas.org/changelog/
- Crib sheets for TPLOT in Your_SPEDAS_dir/idl/general/examples/
  - **crib_tplot.pro** -- basic tplot intro
  - **crib_tplot_annotation.pro** -- How to control annotations in tplot (labels, text, etc...)
  - **crib_tplot_export_print.pro** -- How to export tplot data and tplot plots
  - **crib_tplot_layout.pro** -- How to control tplot plot layouts
  - **crib_tplot_range.pro** -- How to control the range and scaling of tplot plots
  - **crib_tplot_ticks.pro** -- How to control tplot plot ticks. (location, size, etc...)
- SPEDAS-J wiki (for Japanese SPEDAS user)
  - https://github.com/spedas-j/member_contrib/wiki

Other information sources for Arase

- ERG-SC web page (incl. ERG plug-in and data download pages.)
  - https://ergsc.isee.nagoya-u.ac.jp
- ERG-SC wiki
  - https://ergsc.isee.nagoya-u.ac.jp/mw/index.php/Main_Page



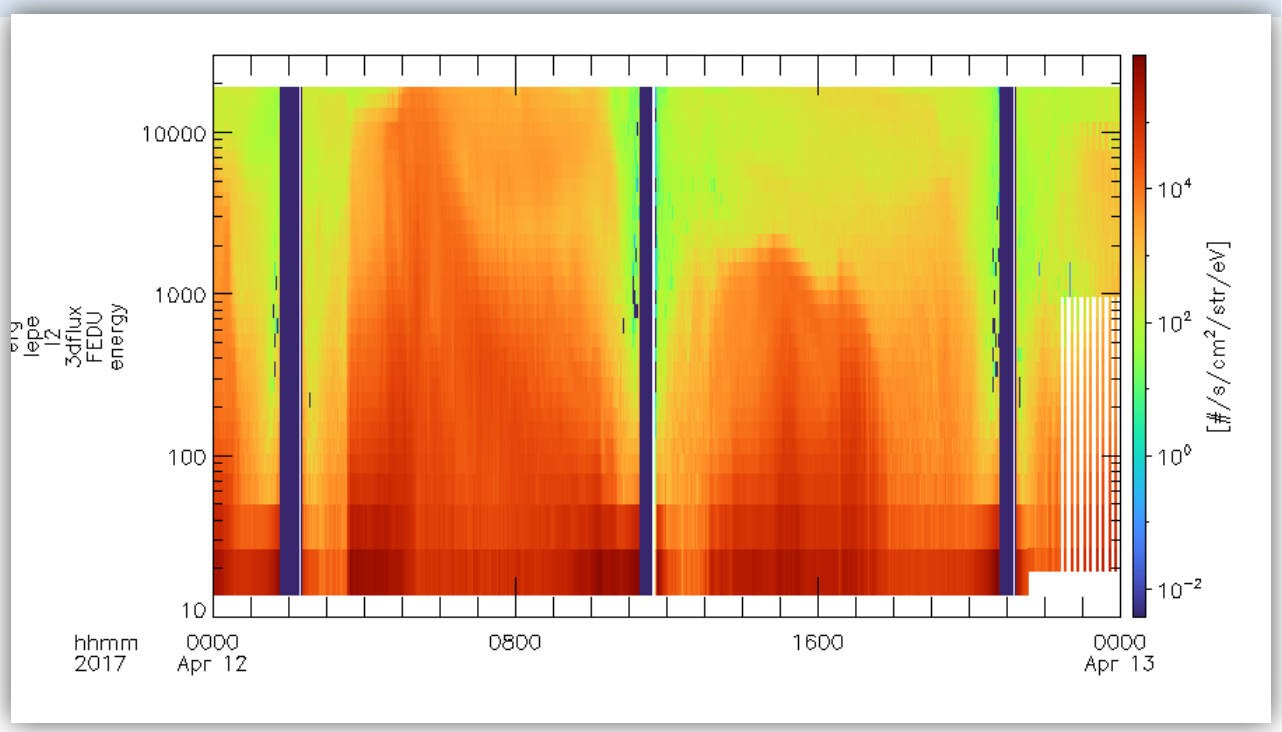
Quick review of the usages of the part_products library

What's "part_products"?

- A set of generic routines bundled to SPEDAS to make tplot variables for various types of spectrum plot.

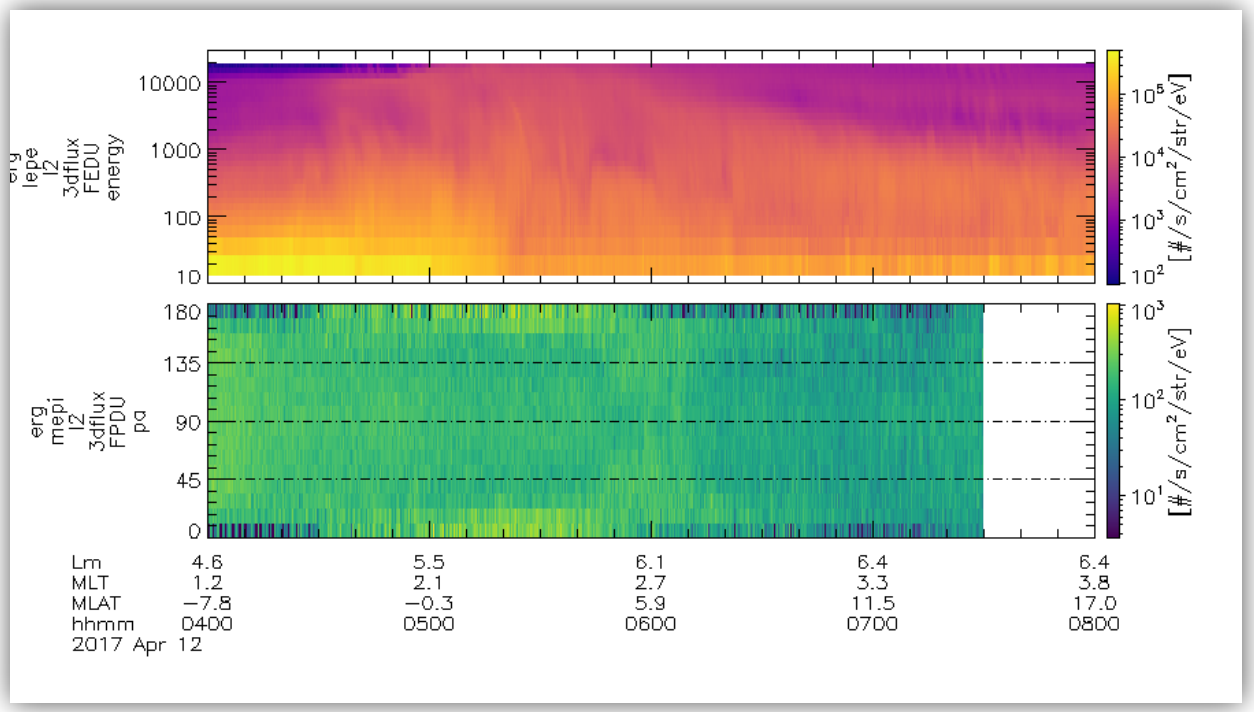
Basic usage of part_products: Energy spectra of average flux

```
timespan, '2017-04-12'
erg_load_lepe, datatype='3dflux', uname=uname, pass=pass, /no_sort_enebin

erg_lep_part_products, 'erg_lepe_l2_3dflux_FEDU', output='energy'

loadct_sd, 48  ;; using the color table Turbo
tplot, 'erg_lepe_l2_3dflux_FEDU_energy'
```



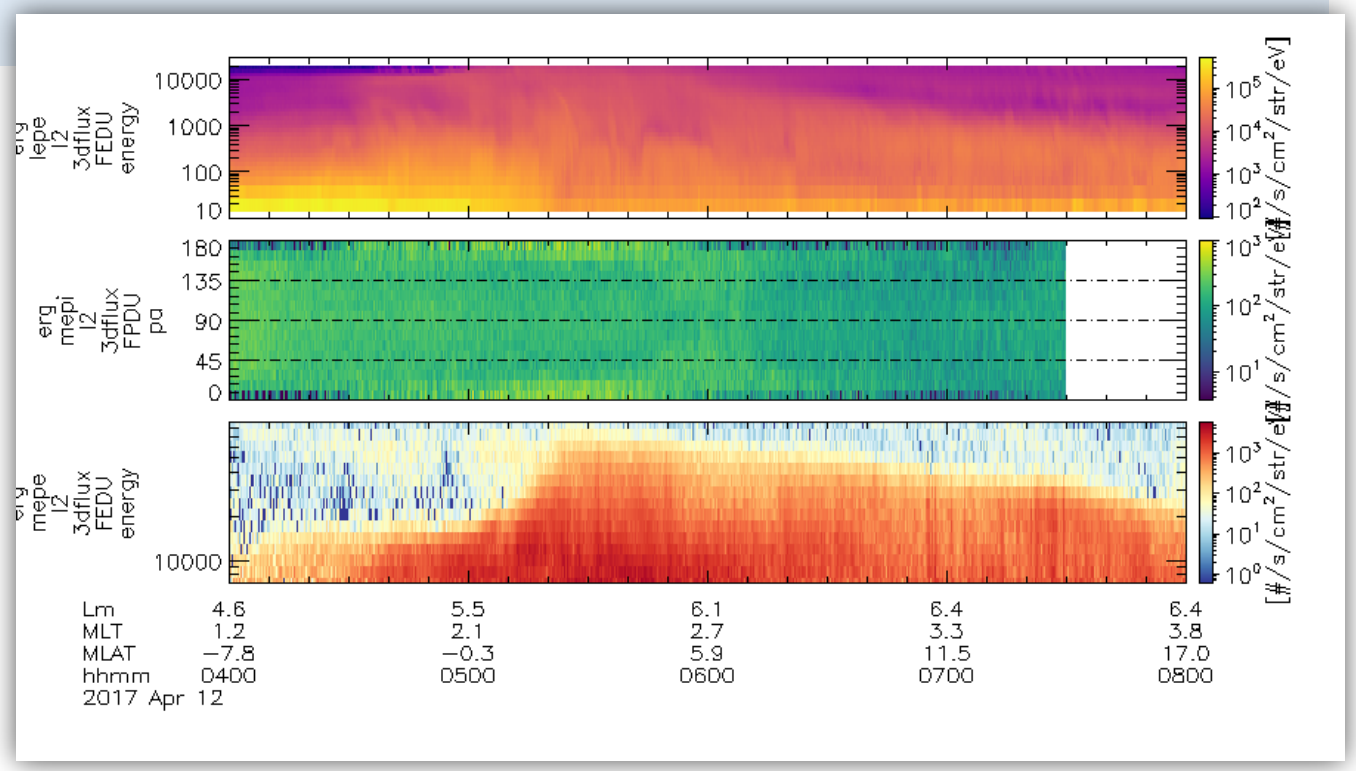
Basic usage of part_products: Pitch angle spectra

```
timespan, '2017-04-12/04:00',4, /hour & get_timespan, tr
erg_load_mgf & set_erg_var_label
magvn = 'erg_mgf_l2_mag_8sec_dsi' & posvn = 'erg_orb_l2_pos_gse'

erg_load_mepi_nml, datatype='3dflux'

erg_mep_part_products, 'erg_mepi_l2_3dflux_FPDU', output='pa', $
          energy=[8000., 13000.], trange=tr, mag=magvn, pos=posvn, /no_ang_weighting

options, 'erg_lepe_l2_3dflux_FEDU_energy', 'color_table', 1079  ;; Different color tables are used for variables.
options, 'erg_mepi_l2_3dflux_FPDU_pa', 'color_table', 1080      ;; 1079: Plasma, 1080: Viridis

tplot, ['erg_lepe_l2_3dflux_FEDU_energy','erg_mepi_l2_3dflux_FPDU_pa']
```



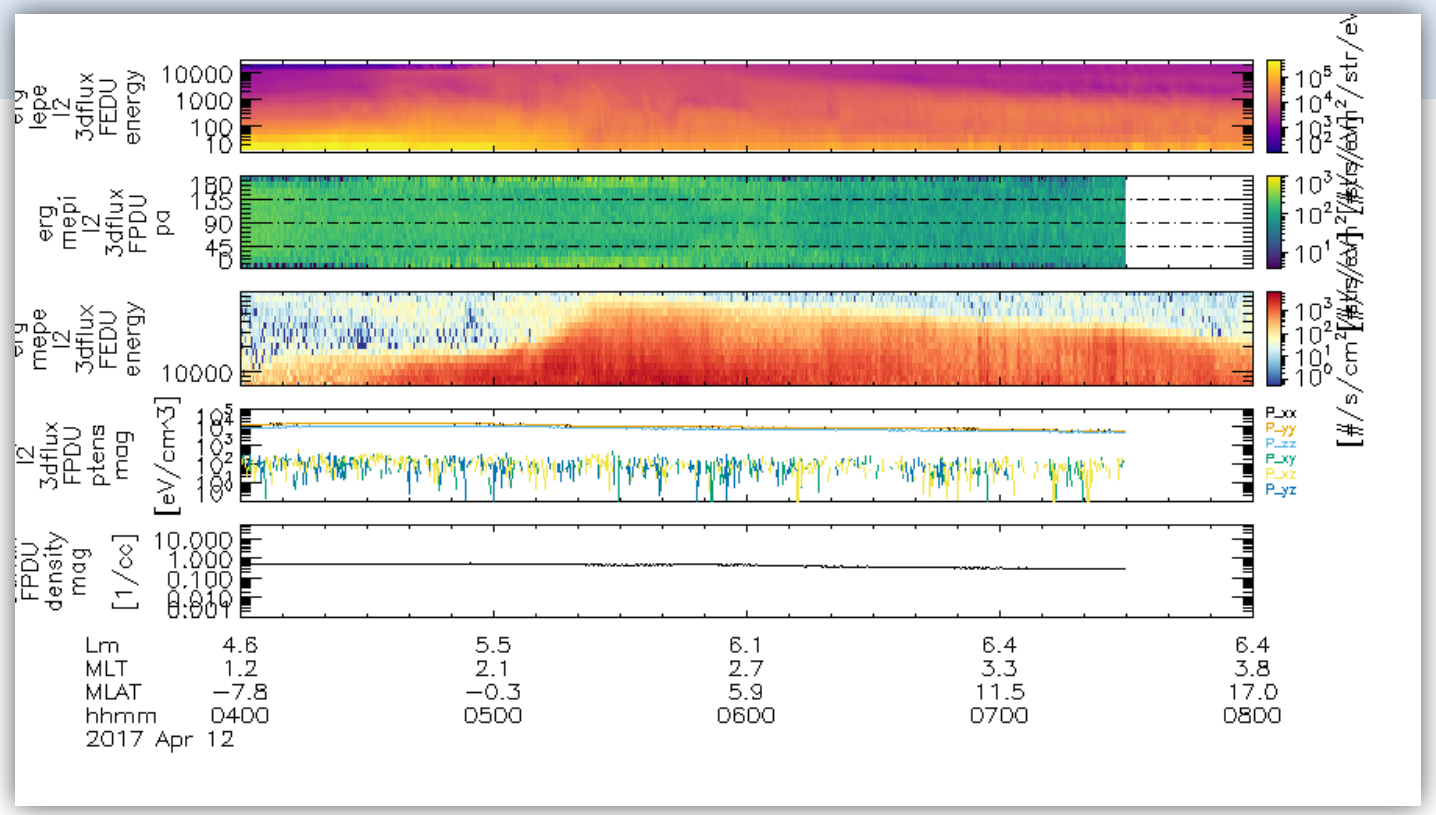
Basic usage of part_products: Energy spectra for a limited PA range

```
erg_load_mepe, level='l2', datatype='3dflux'

erg_mep_part_products, 'erg_mepe_l2_3dflux_FEDU', output='energy', $
   trange=tr, mag=magvn, pos=posvn, pitch=[0, 7], /no_ang_weighting

options, 'erg_mepe_l2_3dflux_FEDU_energy', color_table=1072, reverse_color_table=1

tplot, ['erg_lepe_l2_3dflux_FEDU_energy','erg_mepi_l2_3dflux_FPDU_pa',$
        'erg_mepe_l2_3dflux_FEDU_energy']
```



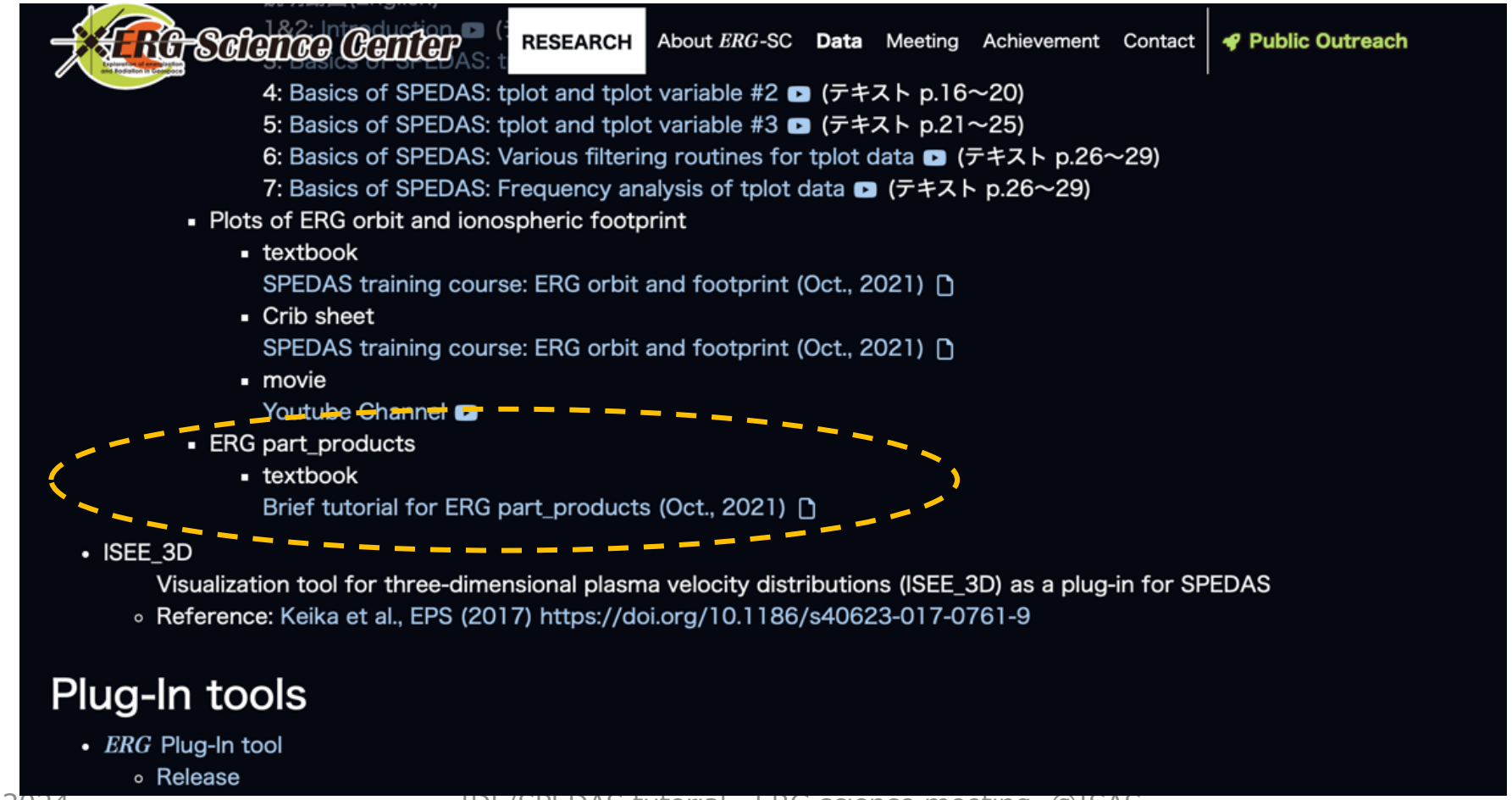
Basic usage of part_products: Velocity moments

```
erg_mep_part_products, 'erg_mepi_l2_3dflux_FPDU', output='fac_moments', $
            trange=tr, mag=magvn, pos=posvn, /no_ang_weighting

options, 'erg_mepi_l2_3dflux_FPDU_ptens_mag', colors=[0,1,2,3,4,5] ;; Set line colors

vns = tnames( /tplot )  ;; Obtain a list of the previously plotted variables

tplot, [ vns, 'erg_mepi_l2_3dflux_FPDU_'+['ptens_mag', 'density_mag'] ]
```



Resources

- A bit more (still brief!) tutorial for the part_products library is available from the ERG-SC website at:
  https://ergsc.isee.nagoya-u.ac.jp/data_info/howto.shtml.en



Arase Orbit and ionospheric footprint

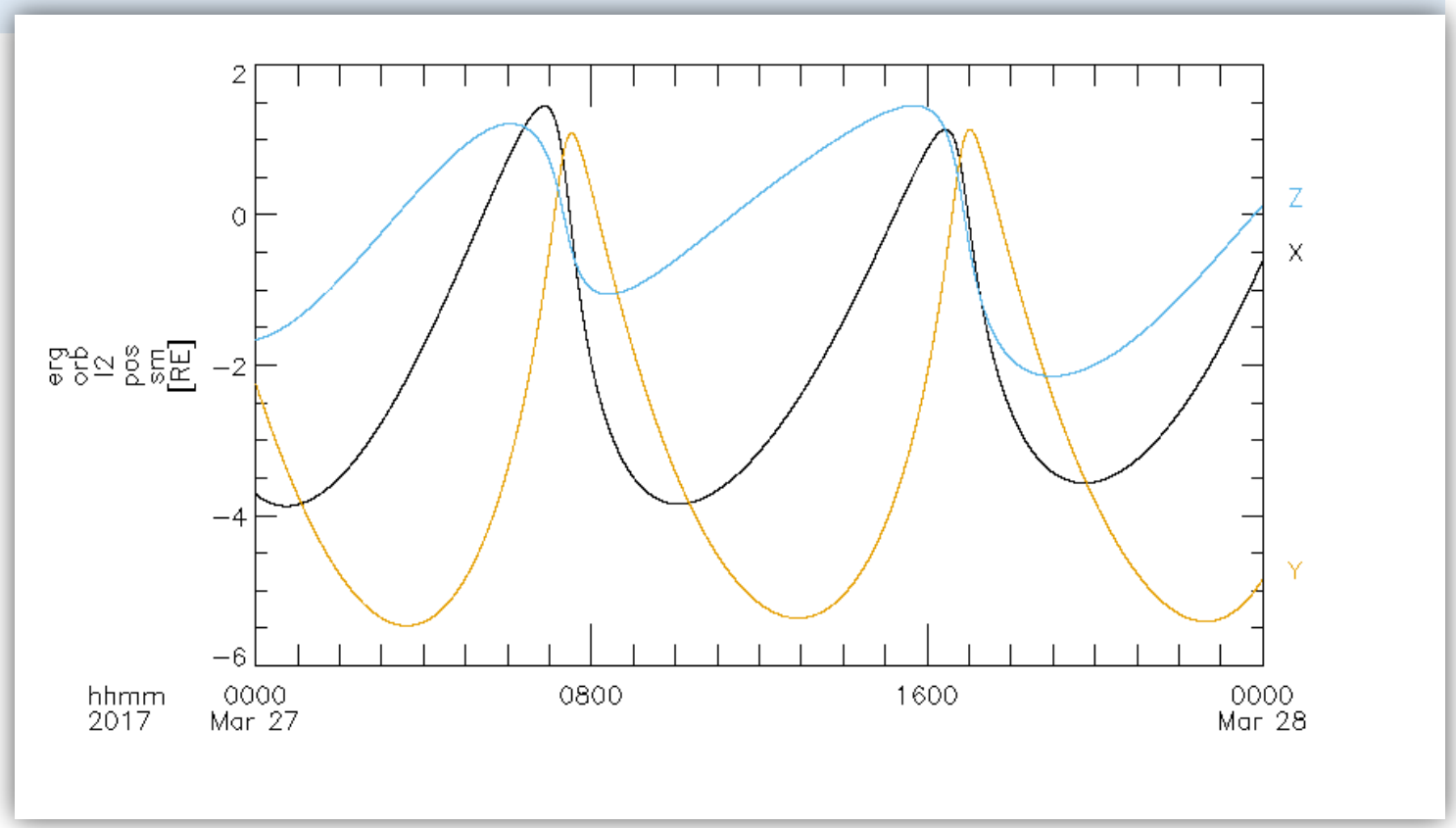
Arase orbit datasets

Scientific data				@ ERG-SC website
Orbit Data				
Predicted Orbit	Short-term	CDF file	wiki	
	Mid-term	CDF file		
	Long-term	CDF file		
Definitive Orbit	Lv.2 definitive orbit data	CDF file	wiki	DOI: 10.34515/DATA.ERG-12000 (Lv.2 definitive orbit data)
	Lv.3 definitive orbit data	CDF file		DOI: 10.34515/DATA.ERG-12001 (Lv.3 definitive orbit data)

Level (model)	Data type	Data-load routine
Level 2 (IGRF)	Definitive	`erg_load_orb`
	Predicted (spre, mpre, lpre)	`erg_load_orb_predict, datatype='spre or mpre or lpre'`
Level 3 (IGRF + external fields)	OP77Q model	`erg_load_orb_l3, model='op or t89 or ts04'`
	T89 model using Kp	
	TS05 (TS04) model using IMF and Dst	

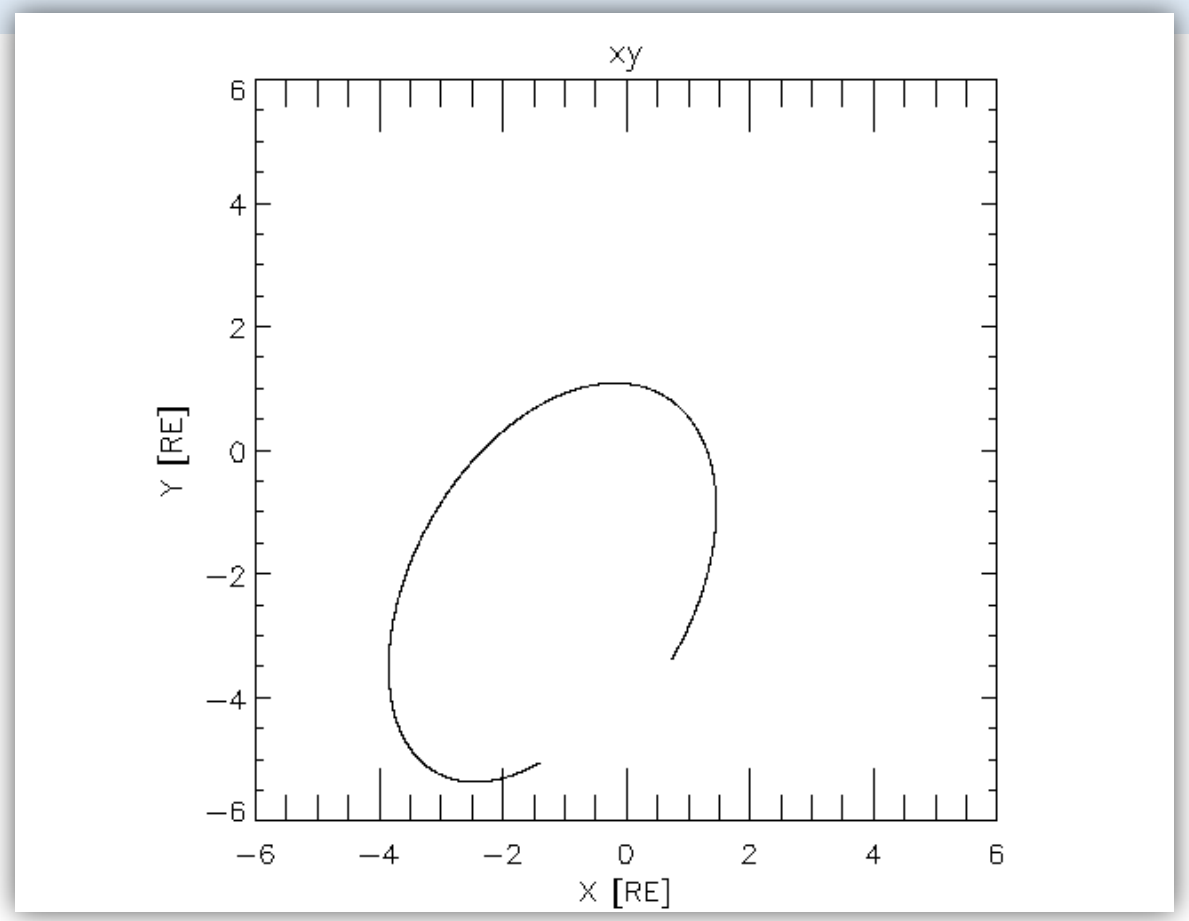
Arase orbit data with tplot

```
timespan, '2017-03-27/00:00', 1, /day  ;; Set to 2017-03-27 00:00-24:00 UT
erg_load_orb     ;; Download and load the orbit data
options, 'erg_orb_l2_pos_sm', colors=[0,1,2], labels=['X','Y','Z'] ;; colors and labels
tplot,'erg_orb_l2_pos_sm'   ;; Plot the data
```



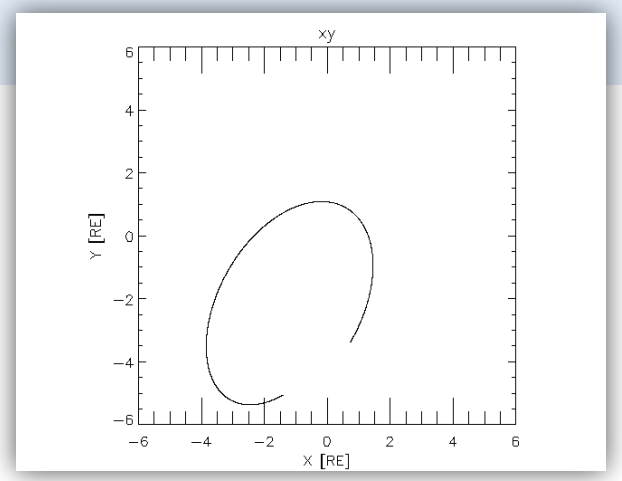
Arase orbit data with `tplotxy`

```
timespan, '2017-03-27/06:00', 8, /hour   ;; 2017-03-27 06:00-14:00 UT
tplotxy, 'erg_orb_l2_pos_sm', xrange=[-6, 6], yrange=[-6, 6]   ;; Plot the data
```



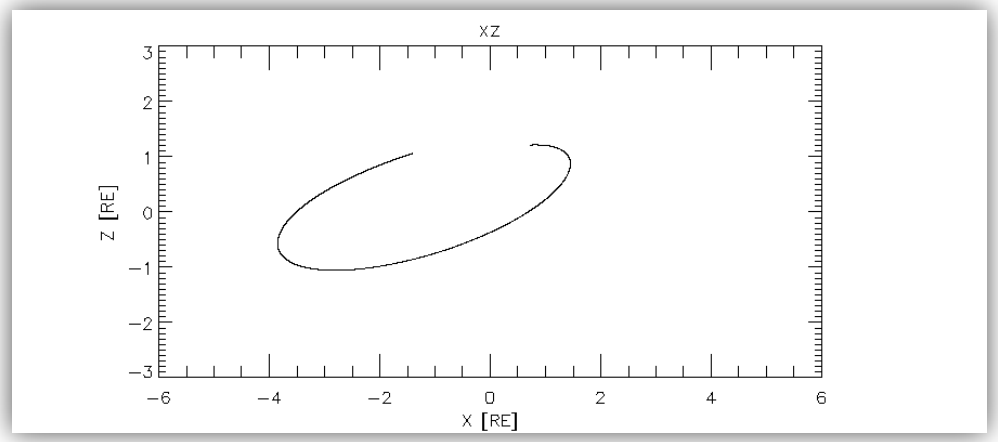
Arase orbit data with tplotxy (cont'd)

```
tplotxy, 'erg_orb_l2_pos_sm', xrange=[-6, 6], yrange=[-6, 6], versus='xy'   ;; on the SM X-Y plane
tplotxy, 'erg_orb_l2_pos_sm', xrange=[-6, 6], yrange=[-6, 6], versus='xz'   ;; on the SM X-Z plane
```



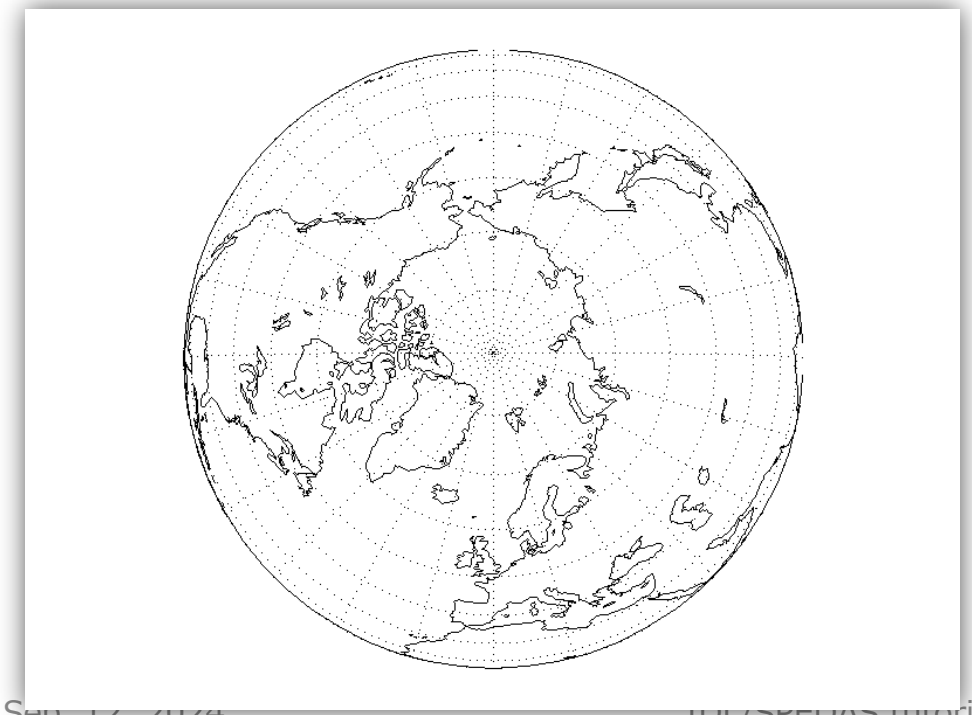
tplotxy accepts many options to modify the plot:

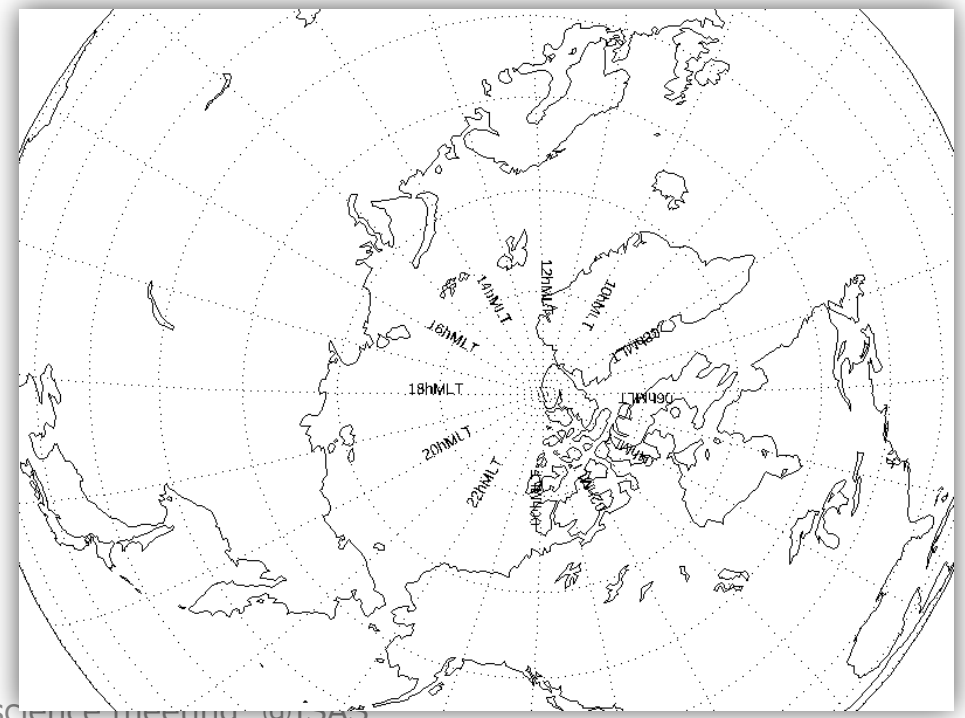
```
tplotxy, tplot_variable_name  $ ; (e.g.,  'position_sm' )
, versus='xy' ; choose axis (e.g., xy or xz)
, multi= 'n_col n_raw'  ; make multi plots ( '# of colum # of raw' )
, over = over ; overplot on the current window
, add = add ; add plot
, additional plot options ; (e.g., title, xrange, color, linestyle, etc.)
```



Arase footprint with the map2d library

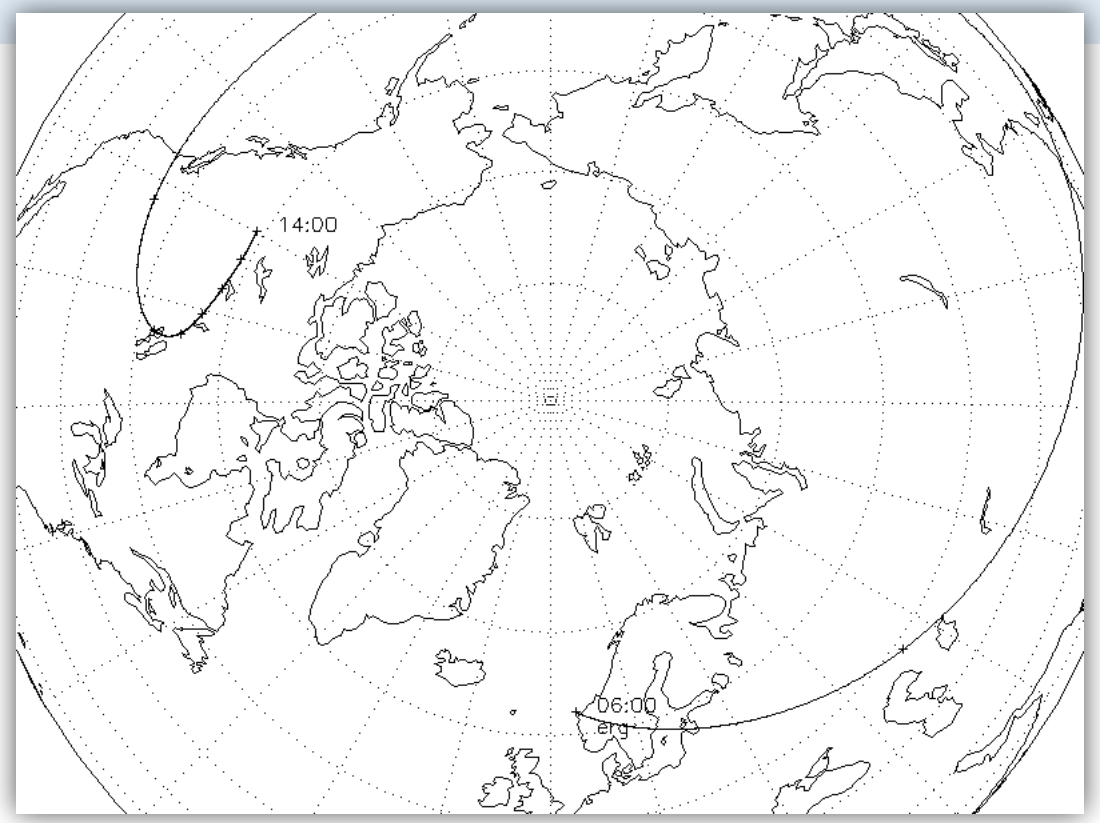
```
map2d_init & map2d_coord, 'geo'
map2d_set, /erase & overlay_map_coast


map2d_coord, 'aacgm'
map2d_time, 1000  ;; Set to draw the grid and world map at 10:00 UT on 2017-03-27
map2d_set, /erase & overlay_map_coast
```





Arase footprint with the map2d library (cont'd)

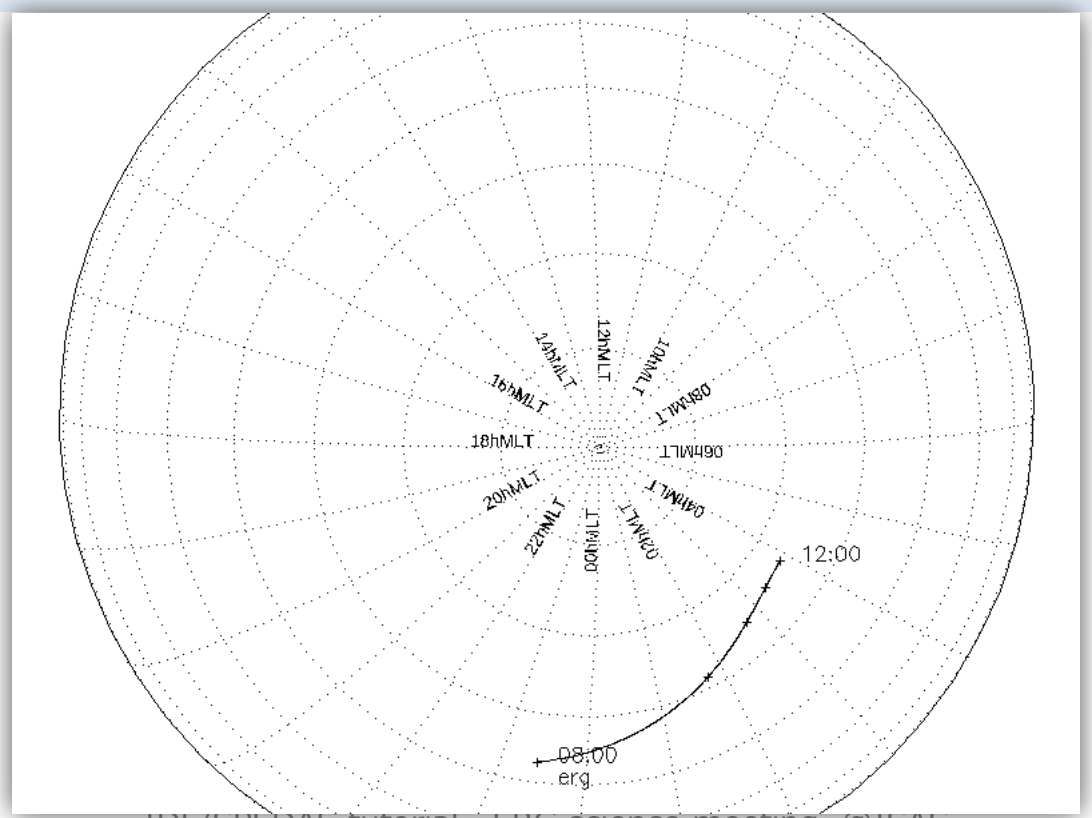
```
split_vec, 'erg_orb_l2_pos_iono_north' ;; Split the footprint position based on IGRF into GEO lat. and lon.
map2d_coord, 'geo'
map2d_set, /erase & overlay_map_coast
overlay_map_sc_ifoot, 'erg_orb_l2_pos_iono_north_0', 'erg_orb_l2_pos_iono_north_1'
```



The footprint trajectory of Arase is drawn on the world map in geographical coords.

Arase footprint with the map2d library (cont'd)

```
map2d_coord, 'aacgm'  ;; Switch the coordinate system to AACGM
timespan, '2017-03-27/08:00', 4, /hour  ;; Set the time range to 08:00-12:00 UT on 2017-03-27
map2d_time, 1000  ;; Draw the map at 10:00 UT
map2d_set, /erase, /mlt, scale=60e+6
overlay_map_sc_ifoot, 'erg_orb_l2_pos_iono_north_0', 'erg_orb_l2_pos_iono_north_1'
```

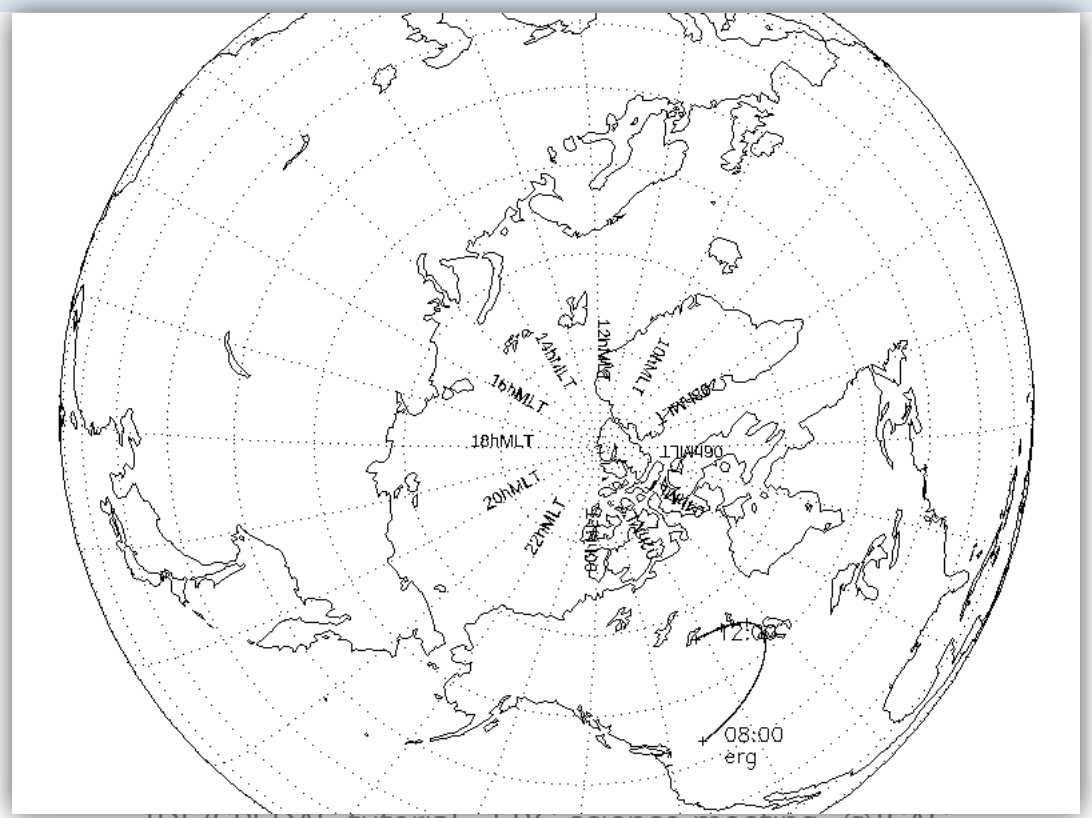


The footprint trajectory of Arase is drawn on the Mlat-MLT grid in Altitude-Adjusted Corrected Geomagnetic (AACGM) coords.

Arase footprint with the map2d library (cont'd)

```
map2d_coord, 'aacgm' & map2d_time, 1000  ;; Draw the map at 10:00 UT
map2d_set, /erase, /mlt, scale=60e+6
overlay_map_coast
overlay_map_sc_ifoot, 'erg_orb_l2_pos_iono_north_0', 'erg_orb_l2_pos_iono_north_1', plottime=!map2d.time
```

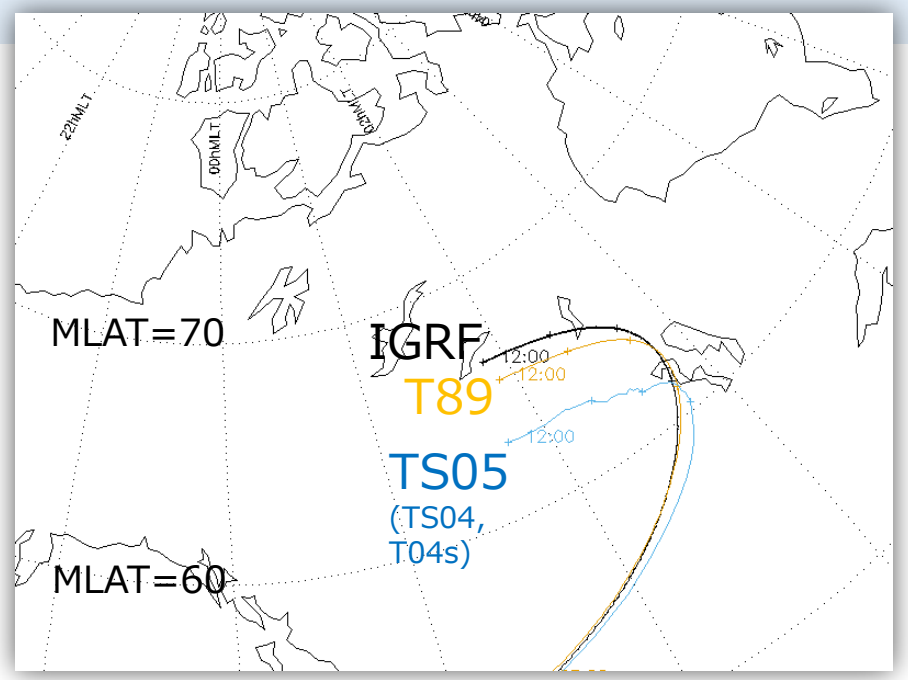
Why is the footprint trajectory different from the prev. one?



The footprint trajectory of Arase is drawn on the world map in Altitude-Adjusted Corrected Geomagnetic (AACGM) coords at 10:00 UT.

Arase footprint with the map2d library (cont'd)

```
erg_load_orb_l3, model='t89' & erg_load_orb_l3, model='ts04'

split_vec, 'erg_orb_l3_pos_iono_north_*'

map2d_time, 1000  ;; Draw the map at 10:00 UT

map2d_set, /erase, /mlt, scale=20e+6, glatc=60., glonc=250.

overlay_map_coast

overlay_map_sc_ifoot, 'erg_orb_l2_pos_iono_north_0', 'erg_orb_l2_pos_iono_north_1', $
                       plottime=!map2d.time, trace_color=0
overlay_map_sc_ifoot, 'erg_orb_l2_pos_iono_north_t89_0', 'erg_orb_l2_pos_iono_north_t89_1', $
                       plottime=!map2d.time, trace_color=1
overlay_map_sc_ifoot, 'erg_orb_l2_pos_iono_north_TS04_0', 'erg_orb_l2_pos_iono_north_TS04_1', $
                       plottime=!map2d.time, trace_color=2
```

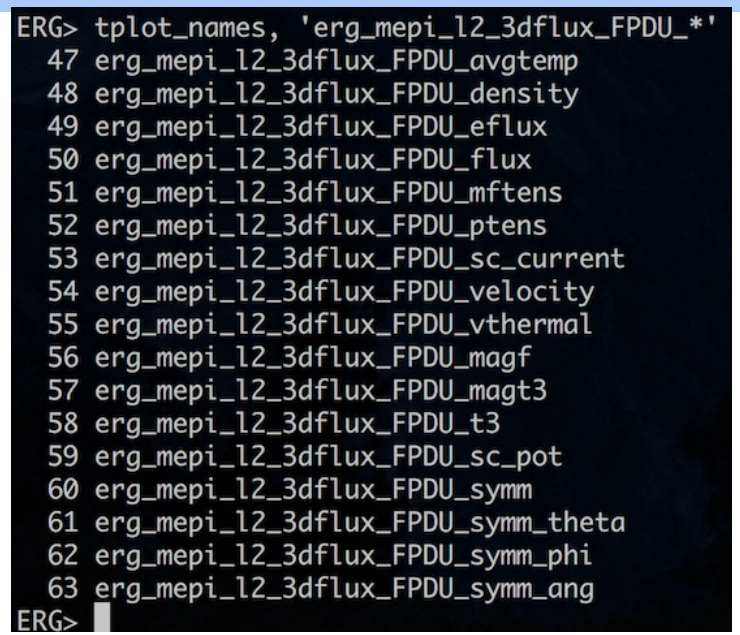


The footprint trajectories of Arase mapped with different B-field models.

Appendix

Appendix A-1:
List of velocity moments by part_products

```
del_data, '*'

timespan, '2017-03-27/10:00', 1, /hour  & get_timespan, tr

erg_load_mepe, datatype='3dflux', varformat='FEDU'

erg_load_mepi_nml, datatype='3dflux', varformat='FPDU'

erg_load_mgf & erg_load_orb

erg_mep_part_products, 'erg_mepi_l2_3dflux_FPDU', pos='erg_orb_l2_pos_gse', mag='erg_mgf_l2_mag_8sec_dsi',
output='moments', trange=tr

erg_mep_part_products, 'erg_mepe_l2_3dflux_FEDU', pos='erg_orb_l2_pos_gse', mag='erg_mgf_l2_mag_8sec_dsi',
output='moments', trange=tr
```



Primary parameters calculated with the part_products:

- density: number density
- avgtemp: scalar temperature (!)
- velocity: bulk velocity
- vthermal: thermal velocity
- mtens: momentum flux density tensor
- ptens: pressure tensor
- t3: temperature tensor (!)
- magt3: perpendicular/parallel temperature (!)
- flux: number flux
- eflux: energy flux

All vector and tensor quantities in DSI coordinates.

(!) Note that these are NOT a temperature defined as a width of Maxwellian distribution.

Appendix A-2:
3-D data structure common to particle data that SPEDAS can handle

```
ERG> help, dists[0]
** Structure <18a6808>, 21 tags, length=196736, data length=196725, refs=2:
   PROJECT_NAME    STRING    'ERG'
   SPACECRAFT      LONG                 1
   DATA_NAME       STRING    'LEP-e Electron 3dflux'
   UNITS_NAME      STRING    'flux'
   UNITS_PROCEDURE STRING    'erg_convert_flux_units'
   SPECIES         STRING    'e'
   VALID           BYTE         1
   CHARGE          FLOAT          -1.00000
   MASS            FLOAT       5.68566e-06
   TIME            DOUBLE       1.4920128e+09
   END_TIME        DOUBLE       1.4920128e+09
   DATA            FLOAT     Array[32, 16, 12]
   BINS            FLOAT     Array[32, 16, 12]
   ENERGY          FLOAT     Array[32, 16, 12]
   DENERGY         FLOAT     Array[32, 16, 12]
   NENERGY         LONG                32
   NBINS           LONG               192
   PHI             FLOAT     Array[32, 16, 12]
   DPHI            FLOAT     Array[32, 16, 12]
   THETA           FLOAT     Array[32, 16, 12]
   DTHETA          FLOAT     Array[32, 16, 12]
ERG>
```

An example for LEP-e 3-D flux data:

dist is an array of structures each of which contains a set of data for each spin.

"DATA" holds the flux data as a 3-D array of 32 ene. ch x 16 spin sector x 12 sensors for this case.

ENERGY and DENERGY are the central energies and energy ranges of the energy channels.

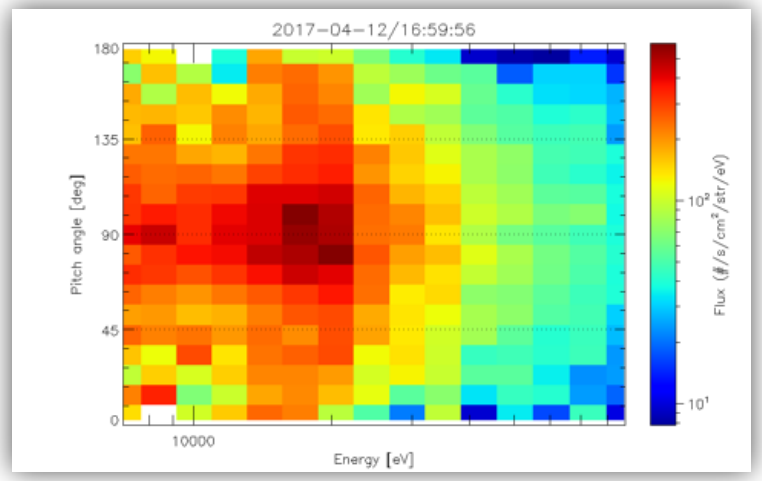
PHI, DPHI, THETA, and DTHETA have phi/theta angles of particle-going directions and angular widths measured by directional channels of a particle instrument in the DSI coordinate system.

Appendix A-3:
Some other options available for erg_part_en_pa_spec_plot

```
erg_part_en_pa_spec_plot, dist $
  , time=time $       ; a time or time range for plotting
  , units=units $     ; physical unit 'flux','eflux','df_km','df_cm'
  , with_contour=with_contour $  ; to overlay contour lines
  , zrange=zrange $   ; explicitly set the range for the color scale
  , npabin=npabin $   ; number of pitch angle bins (default: 19)
  , rslt=rslt $       ; to obtain data arrays which have been plotted
  , noplot=noplot     ; set to suppress replotting
```

You can use this for other particle data.
For example:

```
timespan, '2017-04-12/16:00', 2, /hour
get_timespan, tr
erg_load_mepe, datatype='3dflux'
dists = erg_mepe_get_dist( $
          'erg_mepe_l2_3dflux_FEDU', trange=tr)
erg_part_en_pa_spec_plot, dists
```



Exercise

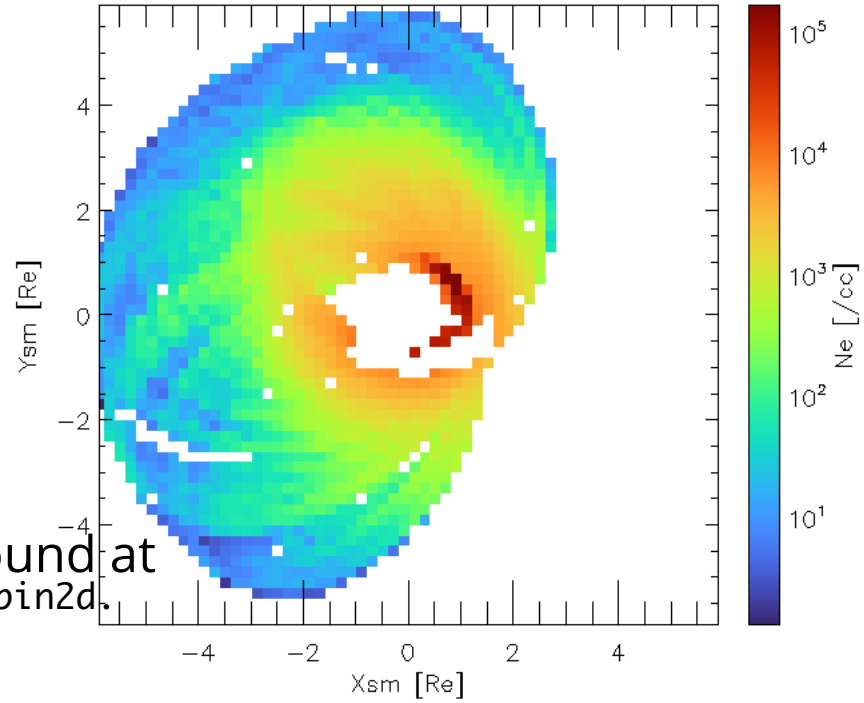
Problem 1:

Create a 2-D map of the electron density averaged in 0.5 x 0.5 Re bins on the SM X-Y plane by analyzing all PWE/HFA Lv.3 e- density data for 2017-04-01 to 2017-10-08 (200 days).

Routines to be used:

- `erg_load_pwe_hfa, level='l3'`
- `erg_load_orb`
- `bin2d`
- `plotxyz`

Information about the bin2d and plotxyz commands can be found at
`https://github.com/spedas-j/member_contrib/wiki/spedas_useful_command#bin2d`.



A solution for the exercise