SPEDAS training session

T. Hori[1], C. W. Jun[1], S. Kurita[1], Y. Miyoshi[1], M. Shoji[1],
M. Teramoto[2], S. Matsuda[3],
N. Umemura[1], T. Segawa[1], T.-F. Chang[4]

[1] ERG Science Center, ISEE, Nagoya Univ., [2] Kyushu Institute of Technology, [3] ISAS/JAXA, [4] NCKU

Purpose of the training

▸ To get familiar with the SPEDAS scheme.

▸ To get you ready to use SPEDAS to analyze various satellite and ground observation data including ERG (Arase) data.

What will you be doing in this training?

- ▸ Learn about the data model employed by SPEDAS.
- ▸ Load/manipulate/visualize ERG satellite and other ground data.

Keep in mind

- ▸ This is a "hands-on" training for SPEDAS, not the time for checking e-mails or playing with a smartphone.

- ▸ Communicate with the lecturer, tutors, neighboring skilled users.

- ▸ The training session today does not cover everything in the handouts due to time limitation. It is highly recommended to practice the uncovered parts later by yourself.

- ▸ We will proceed slowly with the training to make sure that entry-level IDL/SPEDAS users can follow. You can go faster by yourself.

What's SPEDAS?

SPEDAS training session, ERG WS @NCU, Taiwan   Jan. 15, 2020

# Space Physics Environment Data Analysis Software (SPEDAS)

Data repository

SPEDAS framework

THEMIS

CDF

Van Allen Probes

CDF

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

CDF

MMS

CDF

IUGONET

THEMIS-GBO_ASI
2011-09-20/06:58:00

ERG satellite data

CDF

ERG-related ground observation network data

CDF

ERG simulation / modeling data

CDF HDF?

ERG-SC develops data archives and APIs to read the data from SPEDAS
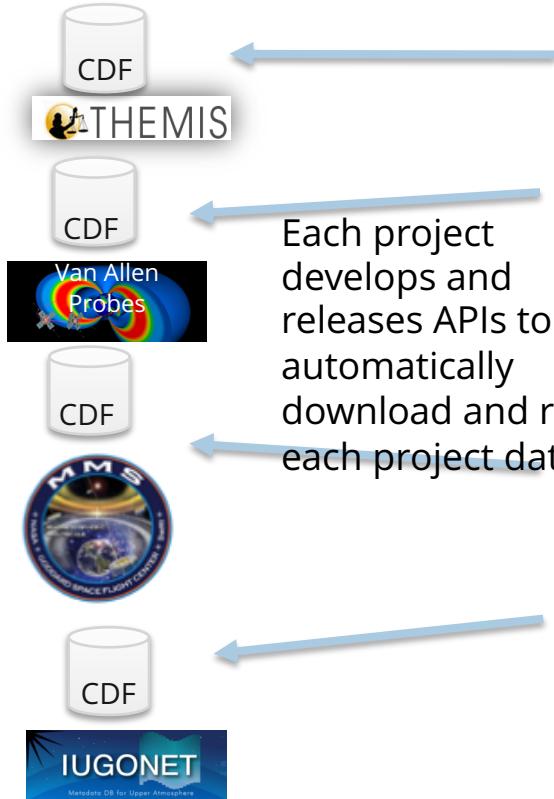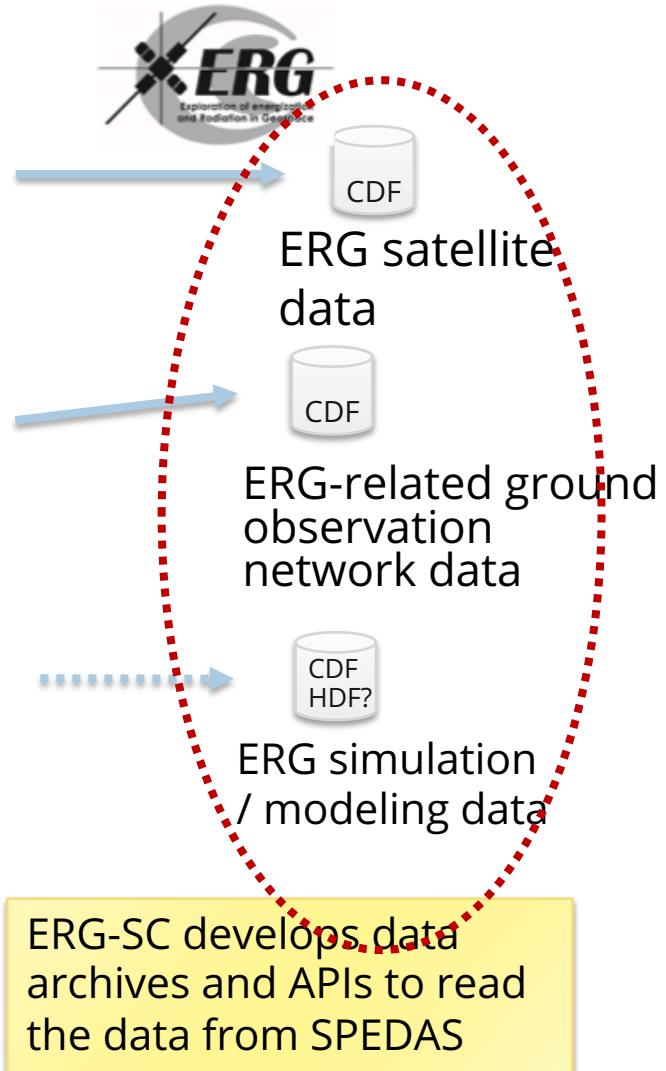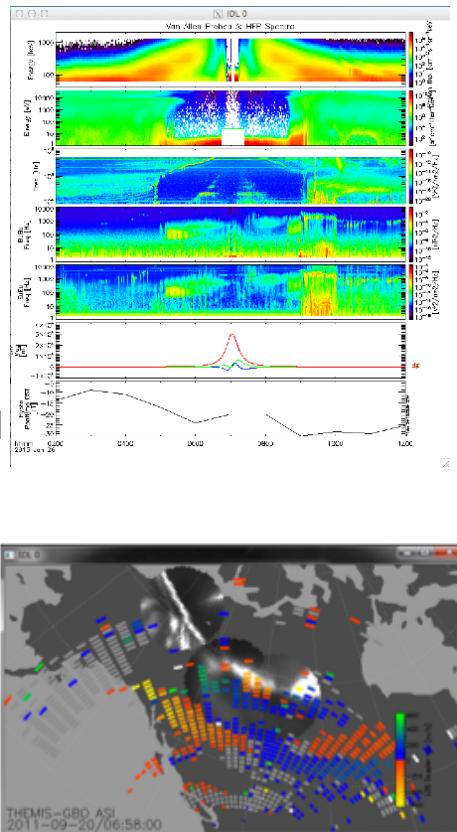
What can you do with SPEDAS?

- ▸ time-series plots
- ▸ filtering of data
- ▸ frequency analysis
- ▸ mapping to the ground maps
- ▸ ...
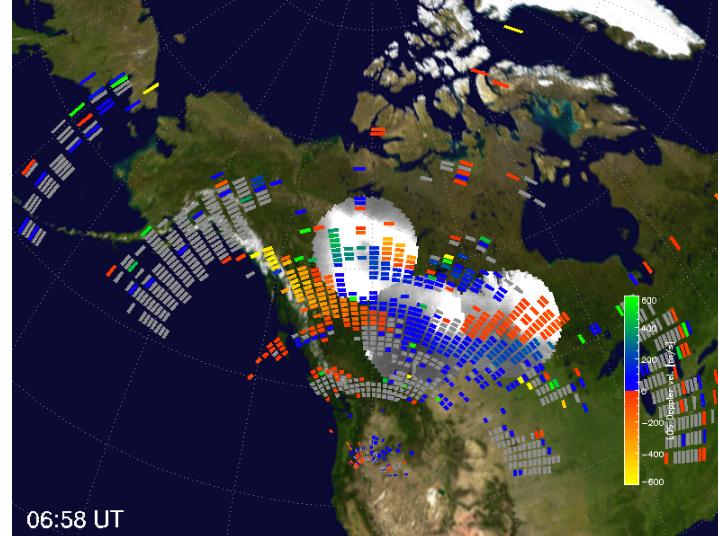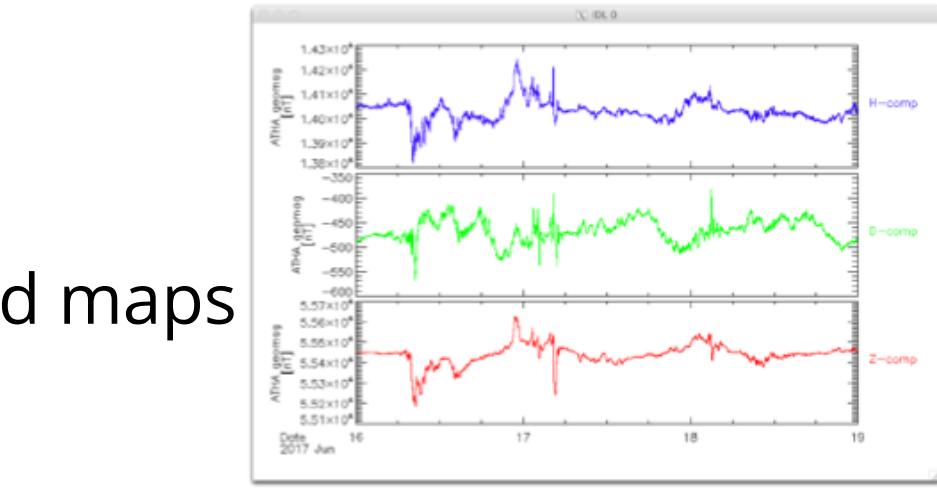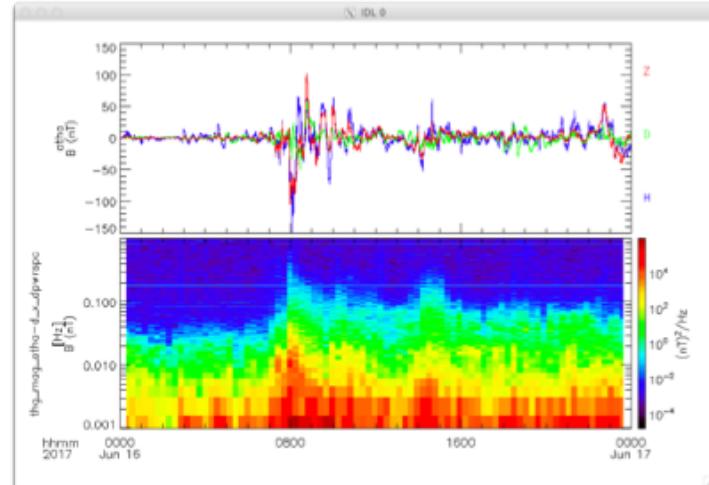
Basics of SPEDAS:
tplot and tplot variable

SPEDAS training session, ERG WS @NCU, Taiwan   Jan. 15, 2020

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' ]
```

▸ Typical errors beginners often encounter:
```
% Attempt to call undefined procedure: '????'.
```
 → command/routine name (????) is misspelled in most cases.
```
% Syntax error.
```
  → , ' ( ) [ ] is missing or mismatched in most cases.

▸ **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
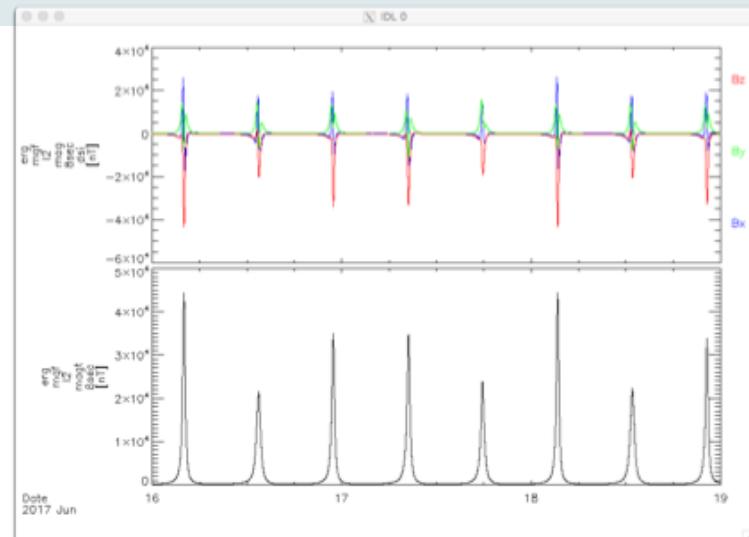
How SPEDAS works?

In SPEDAS command lines,

```
prompt> idl
IDL> erg_init
ERG> timespan, '2017-05-27', 07
ERG> erg_load_mgf
    (manipulate tplot variables)
ERG> tplot, ['erg_mgf_l2_mag_8sec_dsi', 'erg_mgf_l2_magt_8sec' ]
```
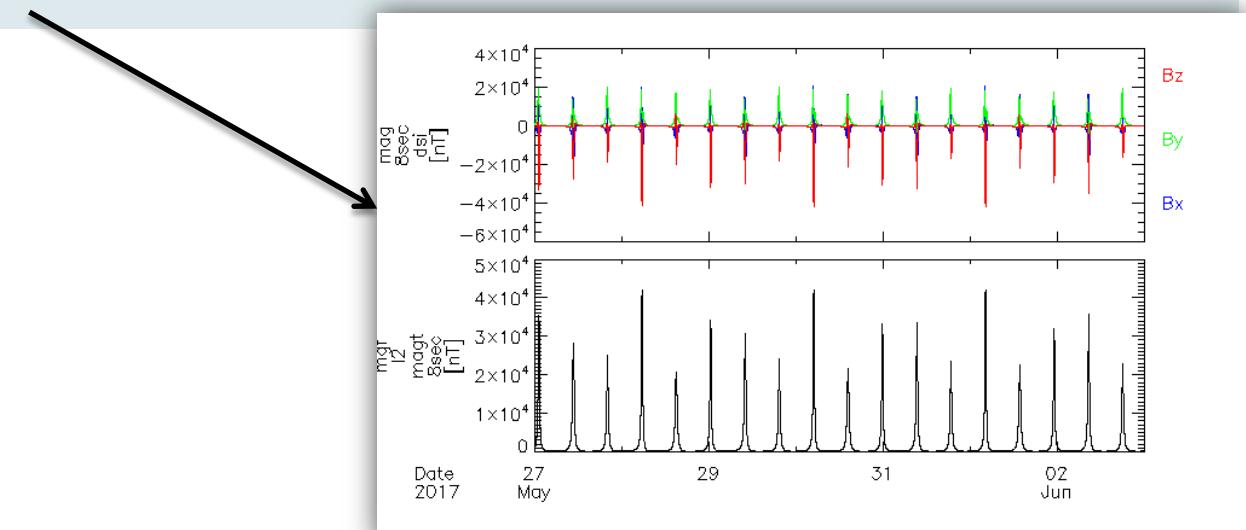
Set a date/time range

`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.
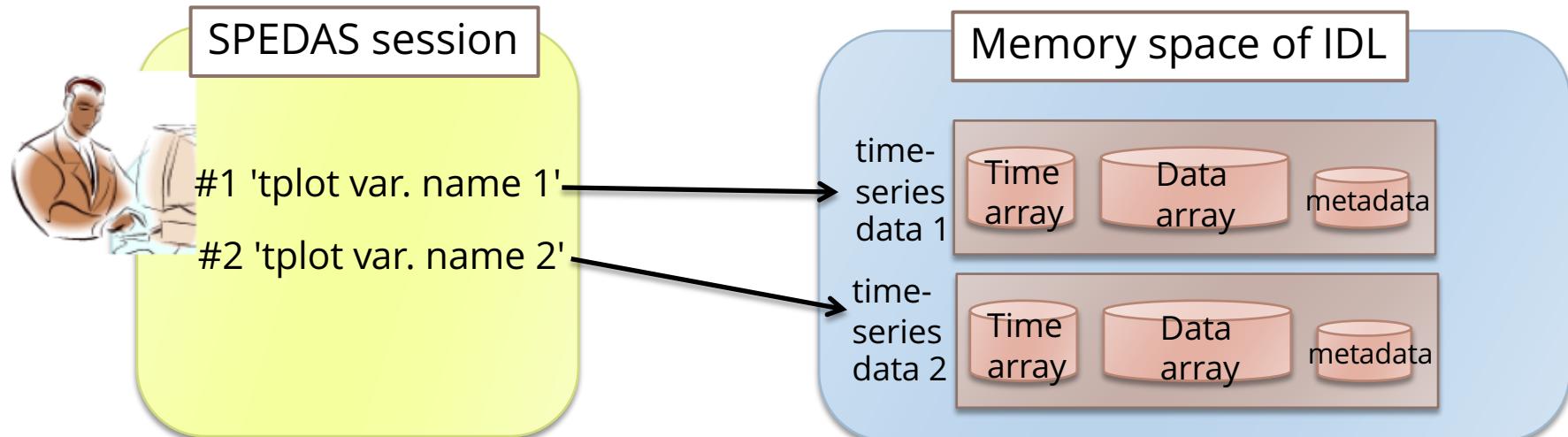
Ex: How the MGF load procedure works

Online repository: `https://ergsc.isee.nagoya-u.ac.jp/data/ergsc/`

Index of /data/ergsc/satellite/erg/mgf/l2/8sec/2017/05
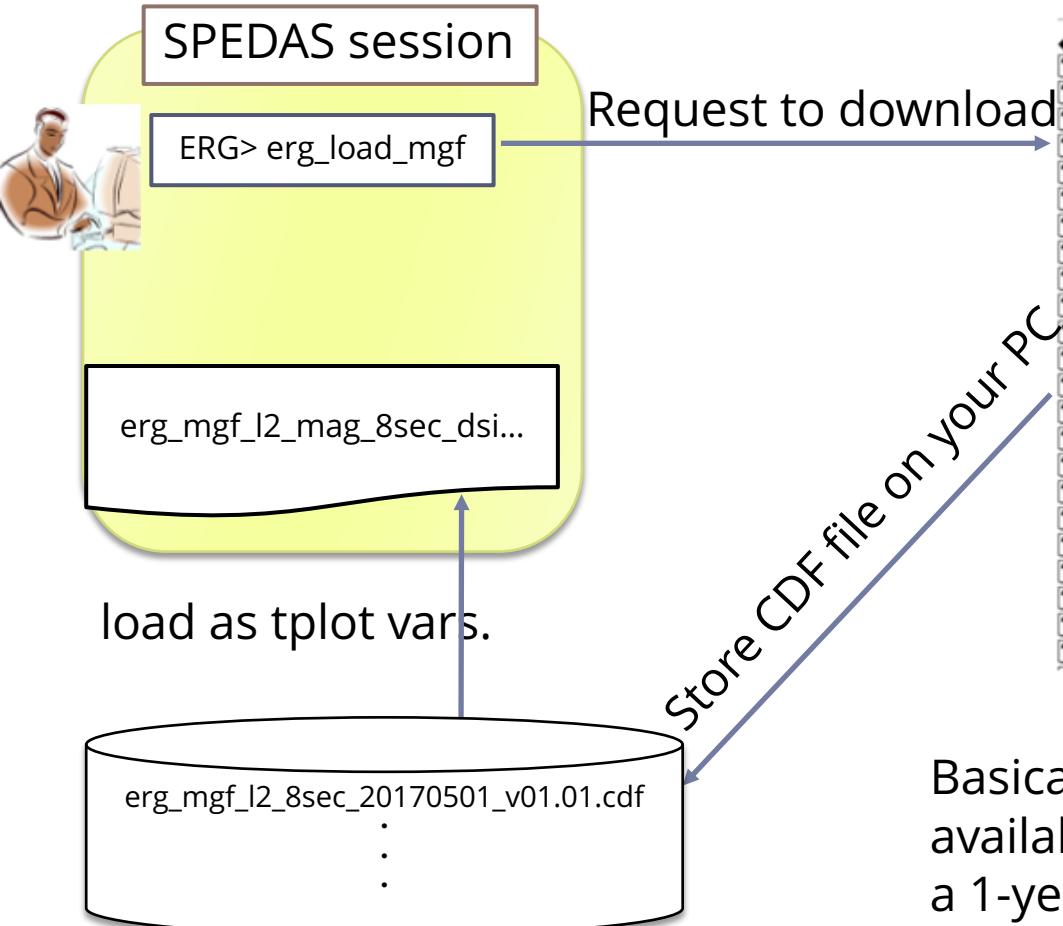
Name	Last modified	Size	Description
Parent Directory		-	
erg_mgf_l2_8sec_20170501_v01.01.cdf	2018-10-19 17:35	2.7M	
erg_mgf_l2_8sec_20170502_v01.01.cdf	2018-10-19 17:35	2.9M	
erg_mgf_l2_8sec_20170503_v01.01.cdf	2018-10-19 17:35	2.7M	
erg_mgf_l2_8sec_20170504_v01.01.cdf	2018-10-19 17:35	2.9M	
erg_mgf_l2_8sec_20170505_v01.01.cdf	2018-10-19 17:35	2.8M	
erg_mgf_l2_8sec_20170506_v01.01.cdf	2018-10-19 17:35	2.8M	
erg_mgf_l2_8sec_20170507_v01.01.cdf	2018-10-19 17:35	2.8M	
erg_mgf_l2_8sec_20170508_v01.01.cdf	2018-10-19 17:35	2.8M	
erg_mgf_l2_8sec_20170509_v01.01.cdf	2018-10-19 17:35	2.9M	
erg_mgf_l2_8sec_20170510_v01.01.cdf	2018-10-19 17:35	2.8M	
erg_mgf_l2_8sec_20170511_v01.01.cdf	2018-10-19 17:35	2.9M	
erg_mgf_l2_8sec_20170512_v01.01.cdf	2018-10-19 17:35	2.8M	
erg_mgf_l2_8sec_20170513_v01.01.cdf	2018-10-19 17:35	2.9M	
erg_mgf_l2_8sec_20170514_v01.01.cdf	2018-10-19 17:35	3.0M	
erg_mgf_l2_8sec_20170515_v01.01.cdf	2018-10-19 17:35	3.1M	
erg_mgf_l2_8sec_20170516_v01.01.cdf	2018-10-19 17:35	3.1M	
erg_mgf_l2_8sec_20170517_v01.01.cdf	2018-10-19 17:35	2.9M	
erg_mgf_l2_8sec_20170518_v01.01.cdf	2018-10-19 17:35	3.1M	
erg_mgf_l2_8sec_20170519_v01.01.cdf	2018-10-19 17:35	3.1M	
erg_mgf_l2_8sec_20170520_v01.01.cdf	2018-10-19 17:35	543K	
erg_mgf_l2_8sec_20170521_v01.01.cdf	2018-10-19 17:35	1.3M	
erg_mgf_l2_8sec_20170522_v01.01.cdf	2018-10-19 17:35	3.1M	
erg_mgf_l2_8sec_20170523_v01.01.cdf	2018-10-19 17:35	3.1M	

SPEDAS session

ERG> erg_load_mgf

Request to download

erg_mgf_l2_mag_8sec_dsi...

load as tplot vars.

Store CDF file on your PC

erg_mgf_l2_8sec_20170501_v01.01.cdf
.
.
.

Basically, ERG satellite data are made available to the science community with a 1-year latency

Data Products of Arase

- **Overview of the ERG project**
  - Miyoshi et al., EPS, 2018a
- **ERG Science Center**
  - Miyoshi et al., EPS, 2018b
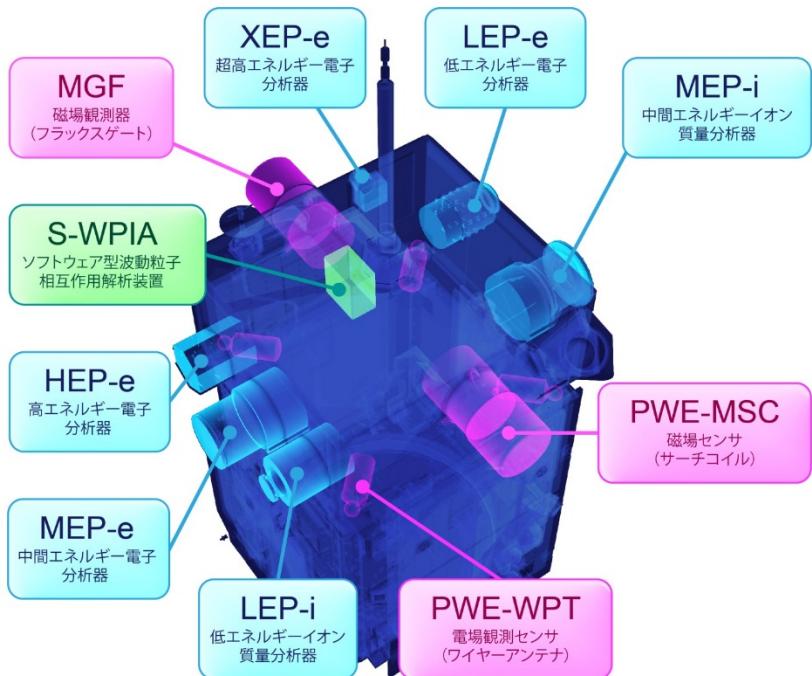
- Orbit（predict/definitive, L2/L3）
- Attitude（L2）
- MGF（L2）
  - Matsuoka et al., EPS, 2018
- PWE（L2/L3）
  - Kasahara et al., EPS, 2018
  - Kumamoto et al., EPS, 2018
  - Ozaki et al., EPS, 2018
  - Matsuda et al., EPS, 2018
  - Kasaba et al., EPS, 2017
- XEP（L2/L3/L4）
- HEP (HEP-H/HEP-L, L2/L3/L4)
  - Mitani et al., EPS, 2018
- MEPi(L2/L3/L4)
  - Yokota et al., EPS, 2017
- MEPe(L2/L3/L4)
  - Kasahara et al., EPS, 2018
- LEPi(L2/L3/L4)
  - Asamura et al., EPS, 2018
- LEPe(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

```
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)

Plotting a tplot data by *tplot*

Tplot with tplot variable names (string)

```
ERG> timespan, '2017-05-27', 07
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_mag_l2_mag_8sec_*'
```
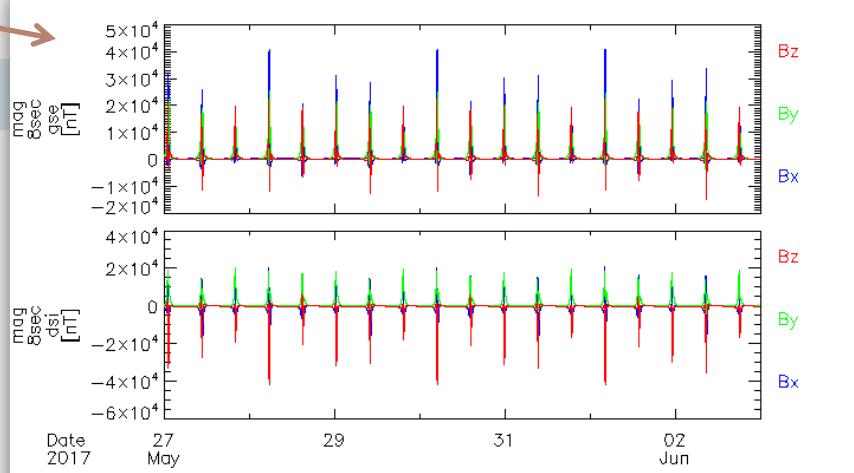
Tplot accepts variables as arguments in various formats.

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_gse', ytitle= 'Arase mag' , ysubtitle='[nT]'
ERG> options, 'erg_mgf_l2_mag_8sec_*', labels=['Bx-comp', 'By-comp', 'Bz-comp' ]
ERG> tplot, 'erg_mgf_l2_mag_8sec_gse'
```


ytitle, ysubtitle

labels
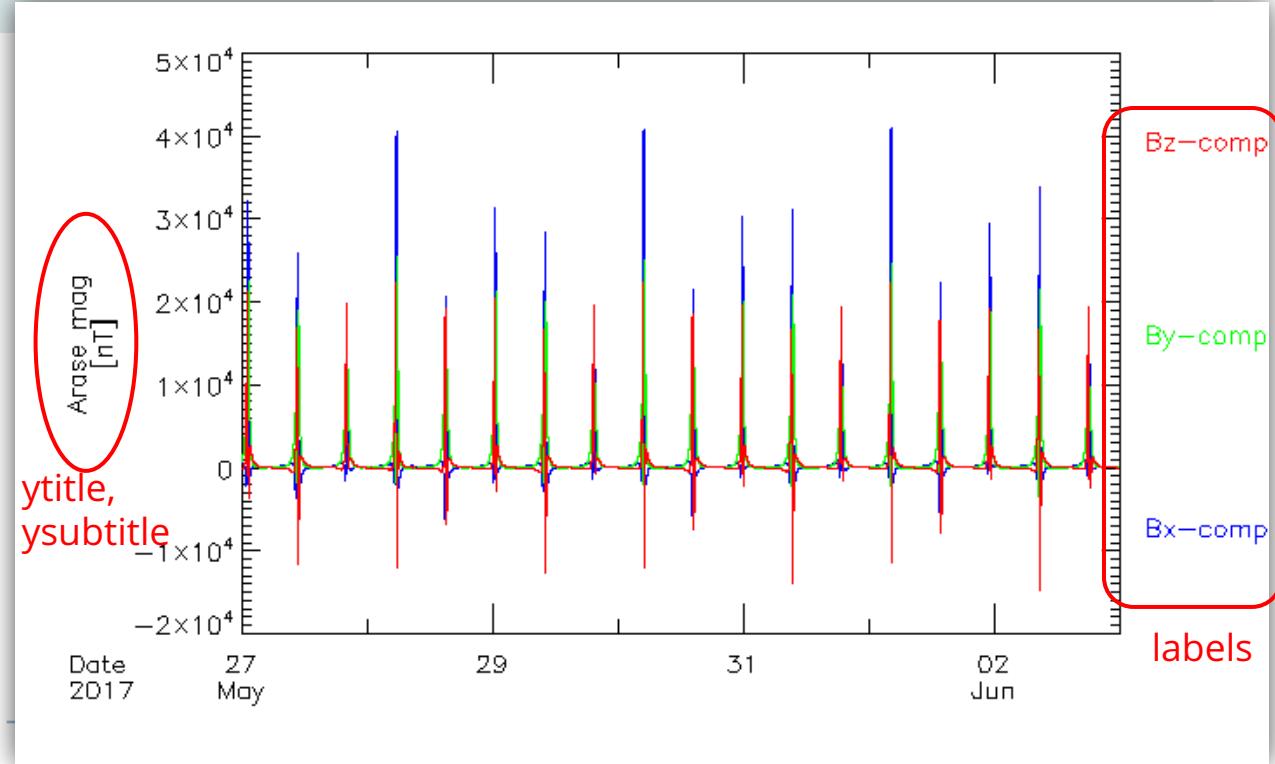
Separate a tplot variable with vector data

```
ERG> split_vec, 'erg_mgf_l2_mag_8sec_gse'
STORE_DATA(260): Creating tplot variable: 19 erg_mgf_l2_mag_8sec_gse_x
STORE_DATA(260): Creating tplot variable: 20 erg_mgf_l2_mag_8sec_gse_y
STORE_DATA(260): Creating tplot variable: 21 erg_mgf_l2_mag_8sec_gse_z

ERG> tplot, 'erg_mgf_l2_mag_8sec_gse_?'
```

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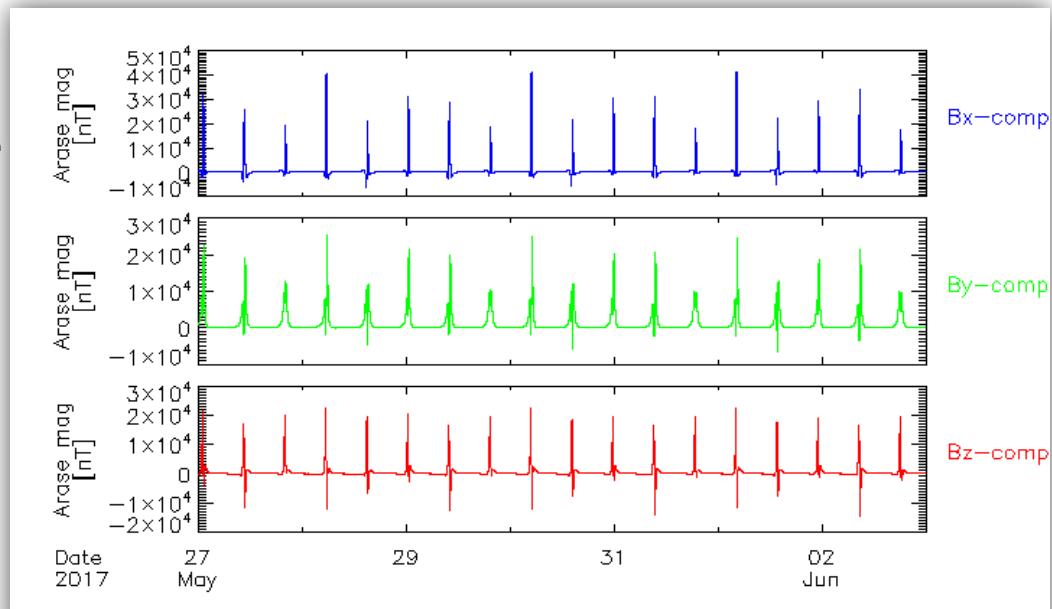
Change the time range of a plot

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
```
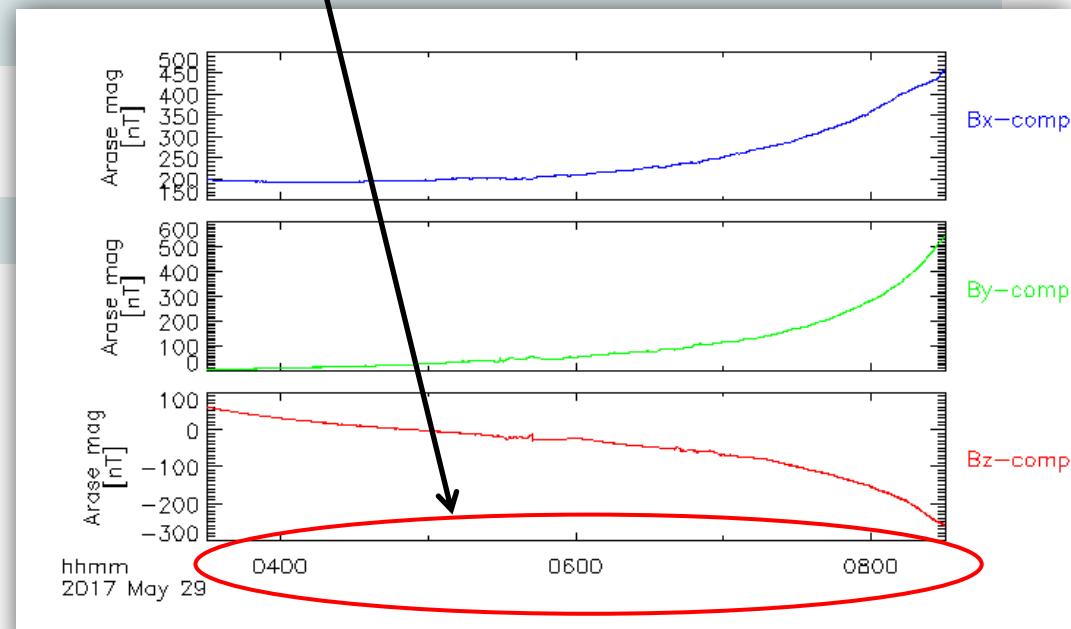
Change the vertical scale of a plot

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_gse_x', 180,250,  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
```
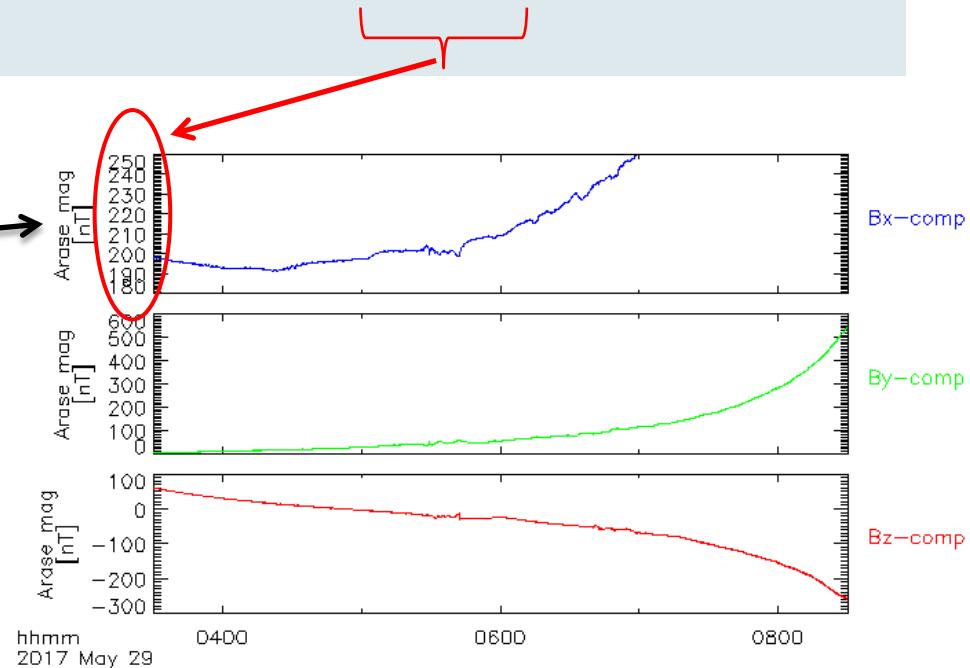
Dump to png, postscript, and Ascii files

To a png file or postscript file

/yyyy/xxxx/erg_mgf_plot.png

```
ERG>  cwd        ;Display the current directory
CWD(25): Directory changed to: /yyyy/xxxx
ERG> tplot, 'erg_mgf_l2_mag_8sec_gse_?'
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
```
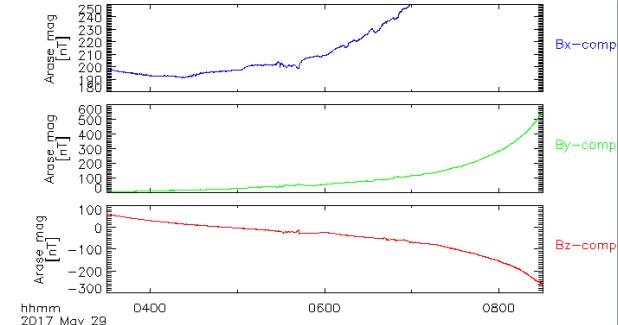

Dump the data content of a tplot variable to a Ascii file

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

SPEDAS manual viewed by web browsers

▸ Open

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

Basics of SPEDAS:
Various filtering routines for tplot data

SPEDAS training session, ERG WS @NCU, Taiwan   Jan. 15, 2020

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
ERG> split_vec, 'erg_mgf_l2_mag_8sec_gse'
ERG> avg_data, 'erg_mgf_l2_mag_8sec_gse_x' , 600.
ERG> tplot, ['erg_mgf_l2_mag_8sec_gse_x', 'erg_mgf_l2_mag_8sec_gse_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.
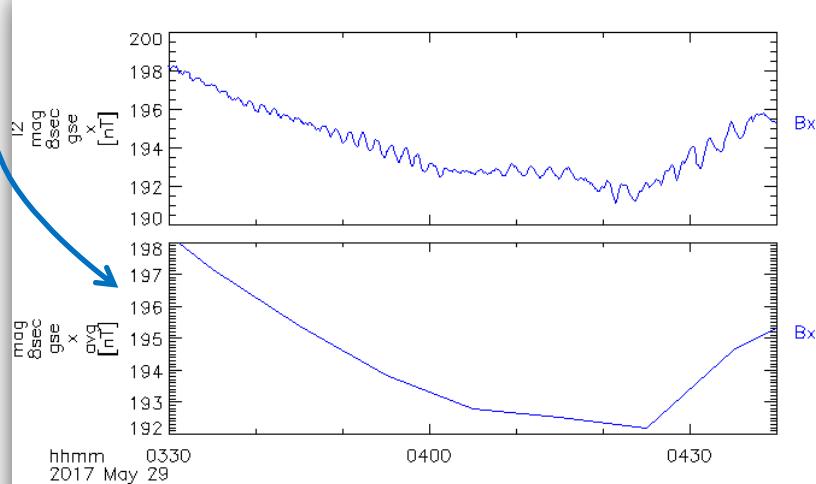
Smoothing data – tsmooth_in_time –

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_gse_x', 600.
ERG> tplot_names

    … … … …
    19 erg_mgf_l2_mag_8sec_gse_x
    … … … …
    23 erg_mgf_l2_mag_8sec_gse_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.
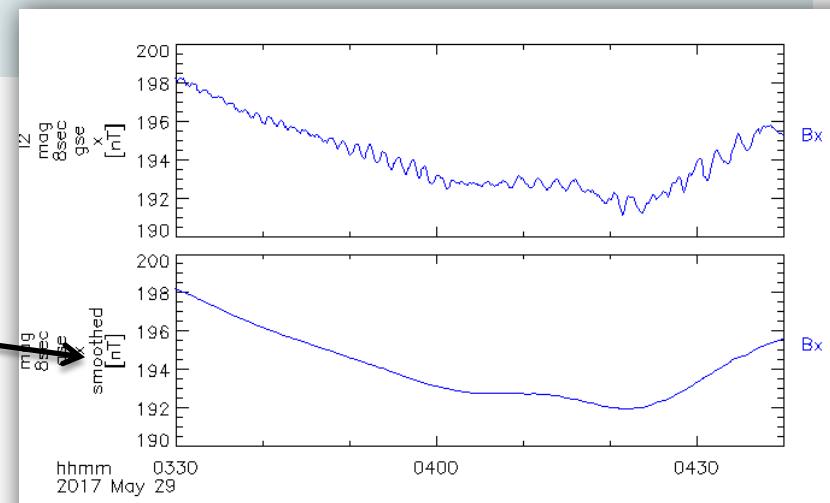
High-pass filter in time – thigh_pass_filter –

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_gse', 600.
ERG> tplot_names

  … … … …
  2 erg_mgf_l2_mag_8sec_gse
  … … … …
  24 erg_mgf_l2_mag_8sec_gse_hpfilt
ERG> tplot, [ 2 , 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 using any digital filtering process such as FFT.
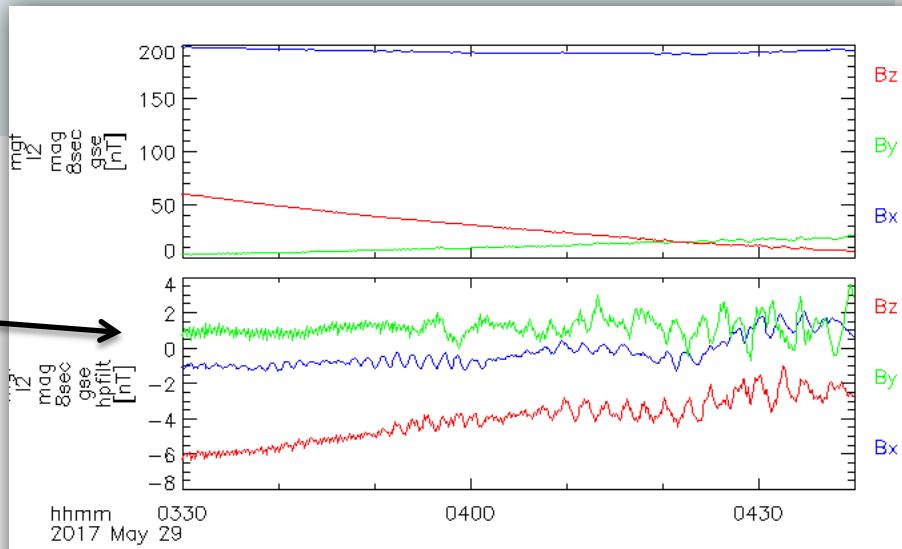
Basics of SPEDAS:
Frequency analysis of tplot data

SPEDAS training session, ERG WS @NCU, Taiwan   Jan. 15, 2020

Dynamics spectra – tdpwrspc–

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

```
ERG>  timespan,'2017-05-29-03:00',300,/min
ERG>  tdpwrspc, 'erg_mgf_l2_mag_8sec_gse_hpfilt'
ERG>  tplot_names

        … … … …
        25 erg_mgf_l2_mag_8sec_gse_hpfilt_x
        … … … …
        28 erg_mgf_l2_mag_8sec_gse_hpfilt_ _x_dpwrspc
ERG>  tplot, [ 25, 28 ]
```

FFT with the hanning window
is applied to derive dynamic
frequency spectra of the data.
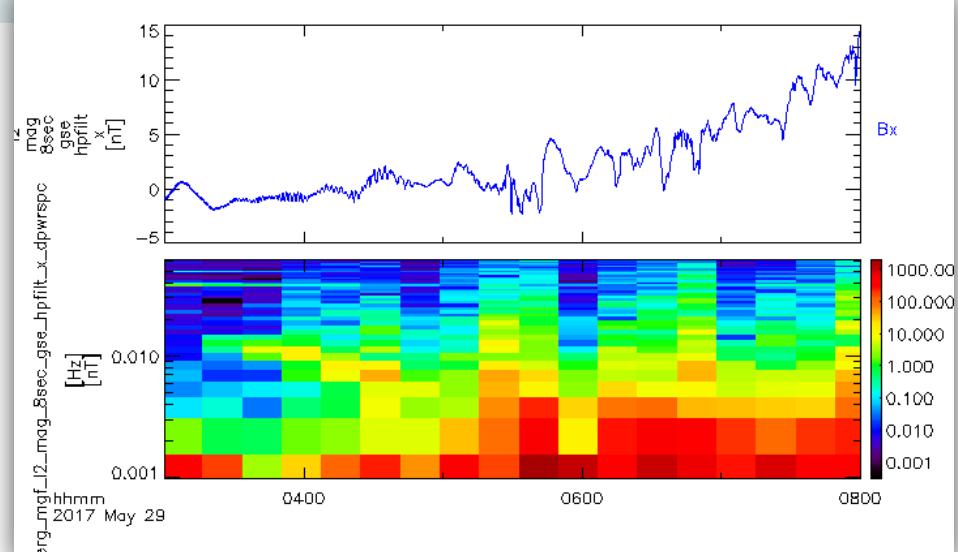
Wavelet analysis – wav_data –

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_gse_hpfilt_x'
STORE_DATA(260): Creating tplot variable: 31 erg_mgf_l2_mag_8sec_gse_hpfilt_x_wv_pow
ERG> zlim, 31, 1, 1000, 1
ERG> tplot, 'erg_mgf_l2_mag_8sec_gse_hpfilt_x' + [ '', '_wv_pow' ]
```

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

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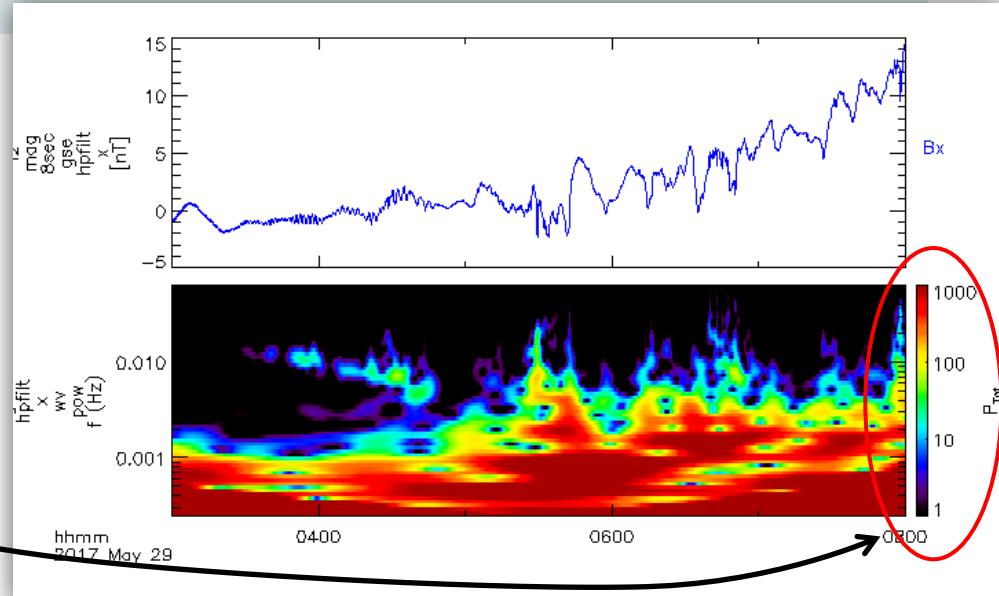
Other information sources for SPEDAS

- **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

Sample cases of ERG data analysis

SPEDAS training session, ERG WS @NCU, Taiwan   Jan. 15, 2020

Data analysis:
3-day plot of HEP data and geomag. indices

```
ERG> timespan, '2017-06-16', 3, /day
```

Load HEP data

```
ERG> erg_load_hep, datatype='omniflux'
ERG> zlim, 'erg_hep_l2_FEDO_*', 1e+0, 1e+4, 1
```

Load other index data

```
ERG> omni_hro_load, /res5min
ERG> tplot,[ 'OMNI_HRO_5min_SYM_H' , 'OMNI_HRO_5min_AL_INDEX' , 'erg_hep_l2_FEDO_*' ]
```
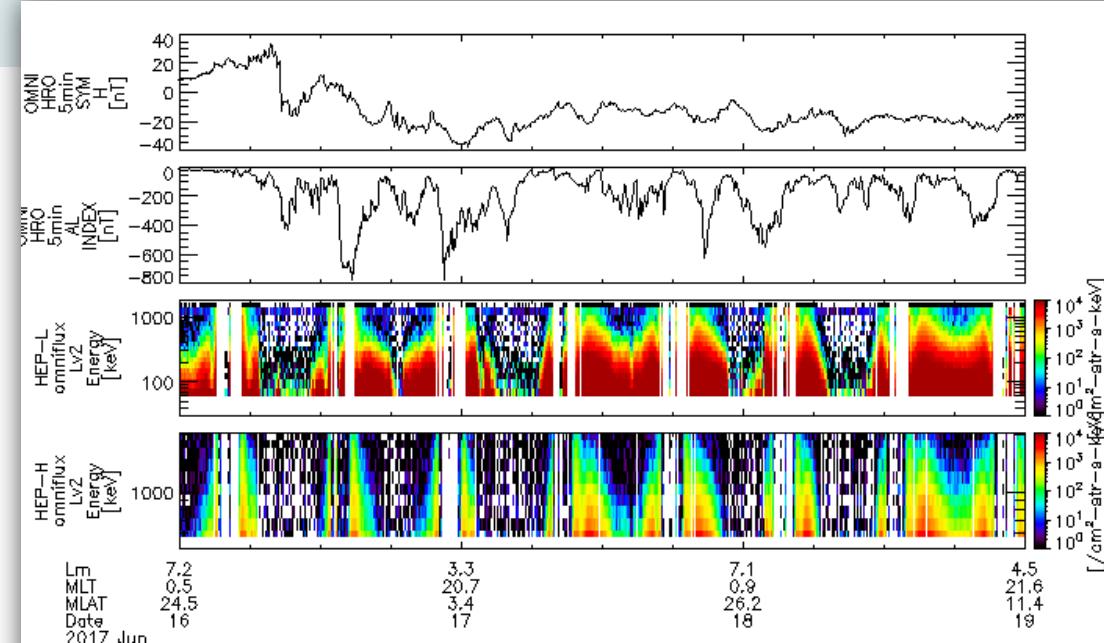
Onboard instrument data:
Coordinate transformation of MGF data

Coordinate transformation (SGA, SGI, DSI, J2000)
From J2000 to the geophysical coordinates -------> using "cotrans"

```
ERG> timespan, '2017-03-28' & erg_load_mgf & set_erg_var_label
ERG> erg_cotrans, 'erg_mgf_l2_mag_8sec_dsi', 'mgf_j2000',
     in_coord='dsi', out_coord='j2000'
```

Coordinate transformation (SM<-->J2000)

```
ERG> spd_cotrans, 'mgf_j2000', 'mgf_gsm', in_coord='j2000', out_coord='gsm'
ERG> tplot, ['mgf_gsm', 'erg_orb_l2_pos_blocal']
```
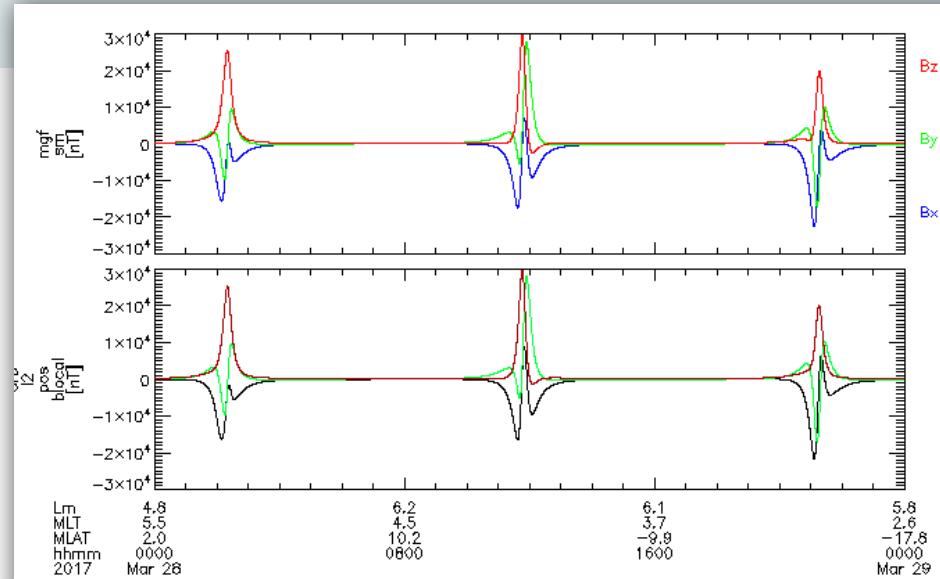
Data analysis:
Frequency analysis of MGF data

> *Please prepare a tplot variable "mgf_sm" containing the magnetic field vectors in SM coordinates for March 28* (hint: See the prev. page!)

```
ERG> timespan, '2017-03-28', 1, /day
ERG> tdpwrspc, 'mgf_sm'
ERG> wav_data, 'mgf_sm_z'
ERG> tplot,'mgf_sm_z*'
```

Dynamic spectra by FFT with the Hanning window

Dynamic spectra by a wavelet analysis
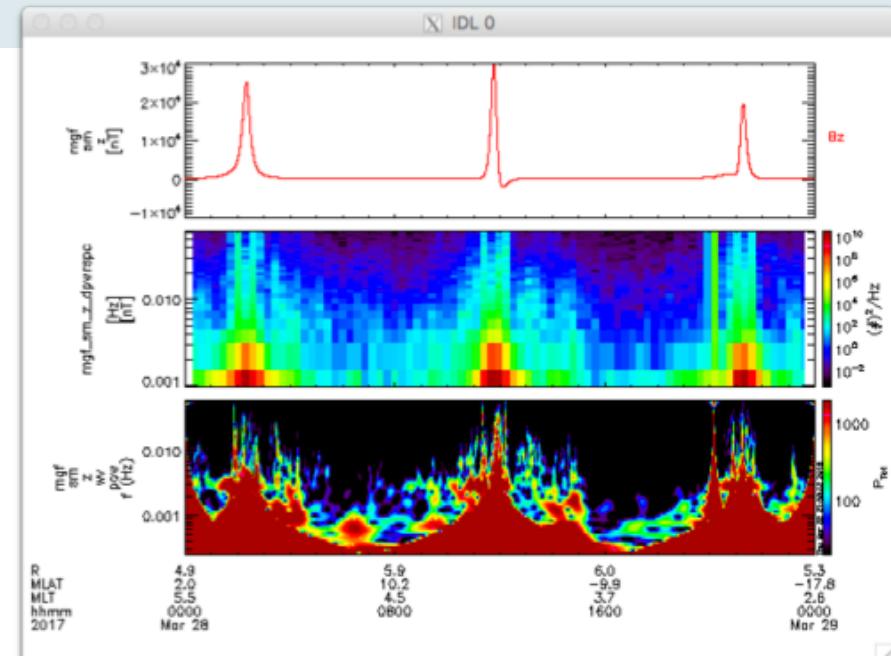
Use `tlimit` to zoom in a period between perigee points!

Onboard instrument data:
Level-2 particle omni-flux data

```
;; Set the time span
timespan, '2017-04-05'
;; Load data
erg_load_xep, datatype='omniflux'
erg_load_hep, datatype='omniflux'
erg_load_mepe, datatype='omniflux'
erg_load_lepe, datatype='omniflux'
erg_load_mepi_nml, datatype='omniflux'
erg_load_lepi_nml, datatype='omniflux'
set_erg_var_label
tplot_names

tplot,[ 'erg_xep_l2_FEDO_SSD', $
  'erg_hep_l2_FEDO_H','erg_hep_l2_FEDO_L', $
  'erg_mepe_l2_omniflux_FEDO', $
  'erg_lepe_l2_omniflux_FEDO', $
  'erg_mepi_l2_omniflux_FPDO', $
  'erg_lepi_l2_omniflux_FPDO'      ]
```
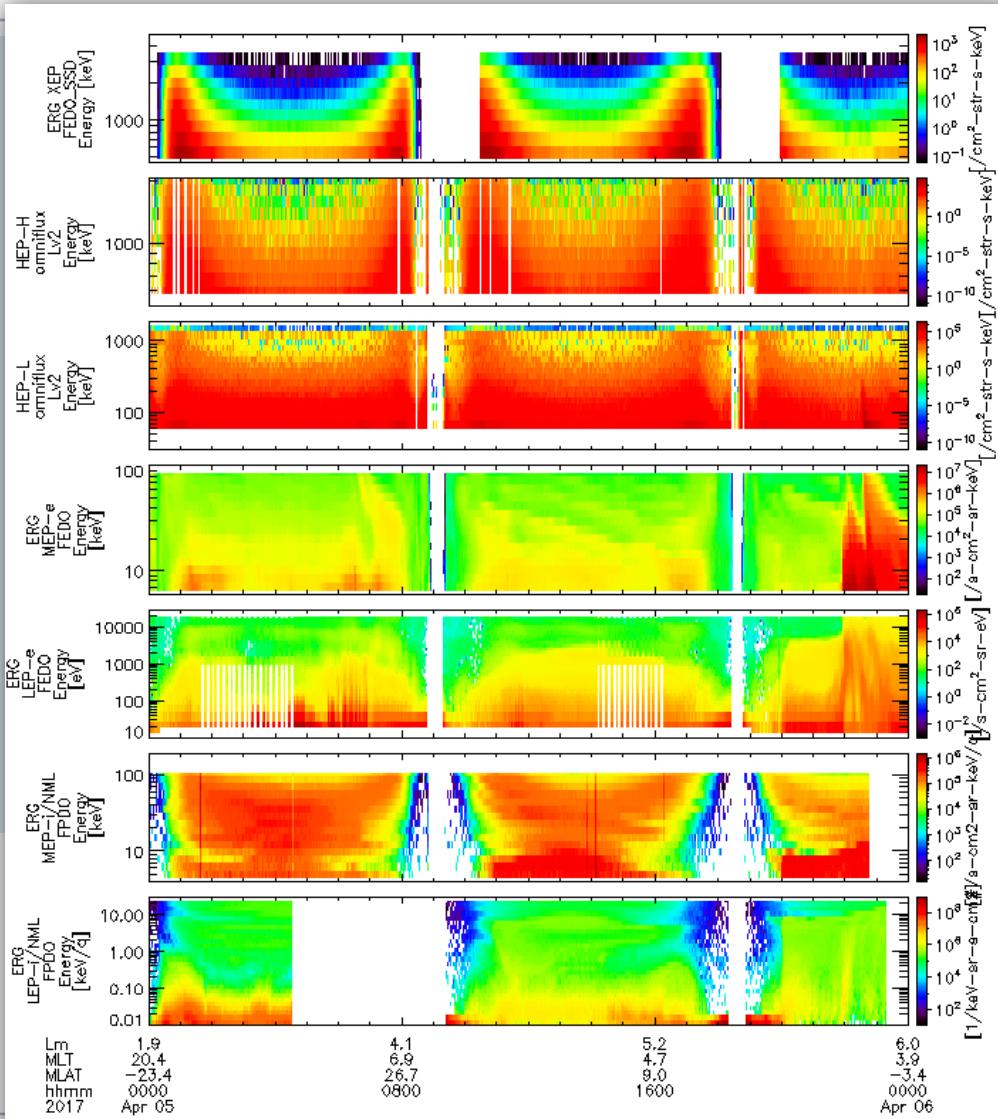
Change the time range of a plot: *tlimit*

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

```
ERG> tlimit
```

Specify a time period explicitly

```
ERG> tlimit, '2017-04-05/19:00' , '2017-04-06/00:00'
```

Back to the last plot period

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

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

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


SPEDAS training session, ERG WS @NCU, Taiwan   Jan. 15, 2020
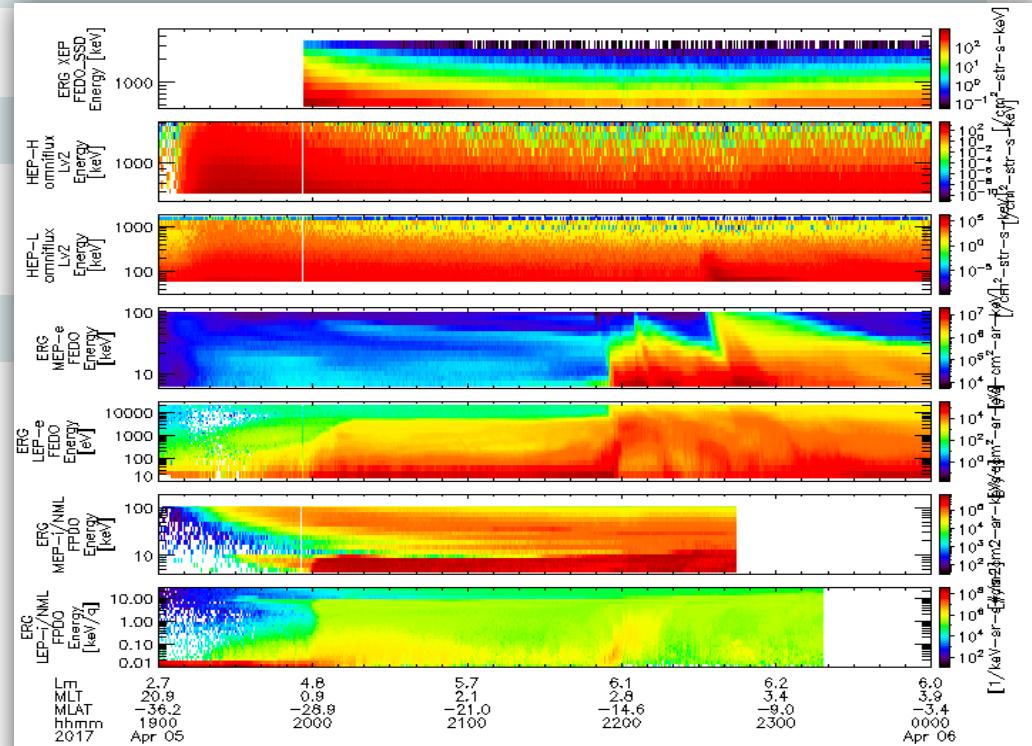
Onboard instrument data:
Add MGF data with /add keyword

Load MGF data and add it to the pre-existing particle plot

ERG> **erg_load_mgf**

ERG> **tplot**, **'erg_mgf_l2_mag_8sec_sm'** , **/add**

SPEDAS training session, ERG WS @NCU, Taiwan   Jan. 15, 2020
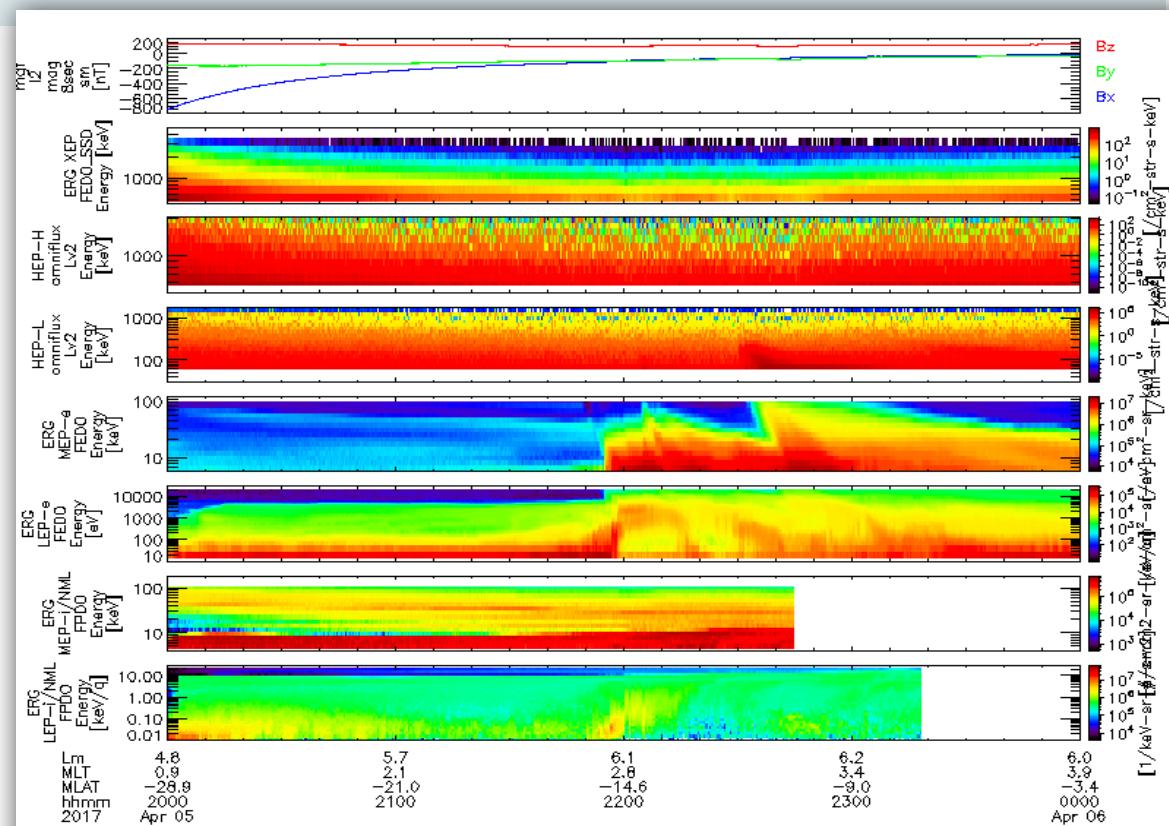
Orbit data:
Set time range and load ERG orbit data

Setup the time range ('YYYY-MM-DD/hh:mm:ss')

```
ERG> timespan,'2017-04-05/00:00:00',1,/day
```

Load orbit data

```
ERG> erg_load_orb
ERG> tplot_names
```

(*) Using the IGRF model

```
ERG> tplot_names
  1 erg_orb_l2_pos_gse
  2 erg_orb_l2_pos_gsm
  3 erg_orb_l2_pos_sm
  4 erg_orb_l2_pos_rmlatmlt
  5 erg_orb_l2_pos_eq
  6 erg_orb_l2_pos_iono_north
  7 erg_orb_l2_pos_iono_south
  8 erg_orb_l2_pos_blocal
  9 erg_orb_l2_pos_blocal_mag
 10 erg_orb_l2_pos_beq
 11 erg_orb_l2_pos_beq_mag
 12 erg_orb_l2_pos_Lm
 13 erg_orb_l2_vel_gse
 14 erg_orb_l2_vel_gsm
 15 erg_orb_l2_vel_sm
 16 erg_orb_l2_spn_num
 17 erg_orb_l2_man_prep_flag
 18 erg_orb_l2_man_on_flag
 19 erg_orb_l2_eclipse_flag
ERG>
```

("erg_orb_l2" means ERG Level-2 orbit data)

|	
pos_gse/gsm/sm	s/c position [Re] in GSE, GSM, SM coordinates
pos_rmlatmlt	Radial distance [Re], magnetic latitude [deg], local time [hr] of s/c position
pos_eq	s/c position mapped to the magnetic equator
pos_iono_north/south	Geographic latitude and longitude [deg] of s/c footprints at 100 km altitude in the northern/southern hemisphere
pos_blocal / blocal_mag	model B-field vector (blocal) and B-field strength (blocal_mag) [nT] at s/c position
pos_beq / beq_mag	model B-field vector (beq) and B-field strength (beq_mag) [nT] at s/c position mapped to the magnetic equator
pos_Lm	McIlwain's L-parameter of s/c position for pitch angles of 90, 60, and 30 deg
vel_gse/gsm/sm	s/c orbital velocity [km/s] in GSE, GSM, and SM
man_prep/man_on/eclipse_flag	flag for maneuver preparation (man_prep), maneuver on/off (man _on), and solar eclipse (eclipse)

Orbit data:
Definitive orbit as a time series plot: *tplot, tplotxy*

Plot orbit time series data

ERG> **tplot, 'erg_orb_l2_pos_gsm'**

Plot orbit data in the X-Y plane.

ERG> **tplotxy, 'erg_orb_l2_pos_gsm'**
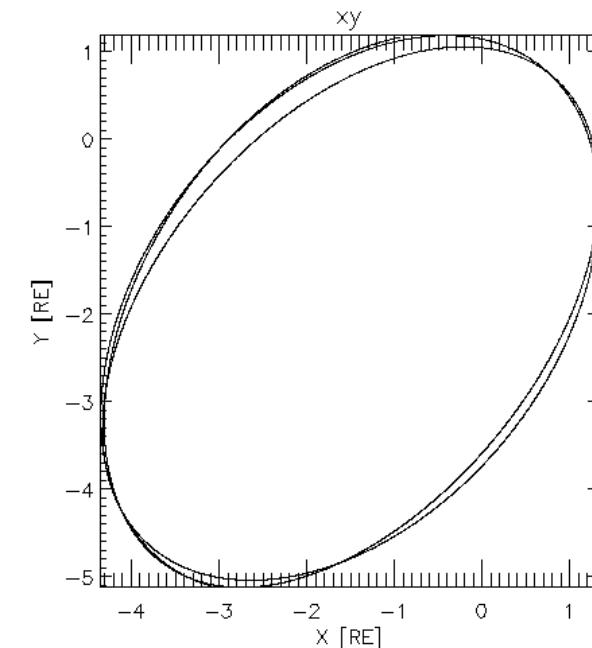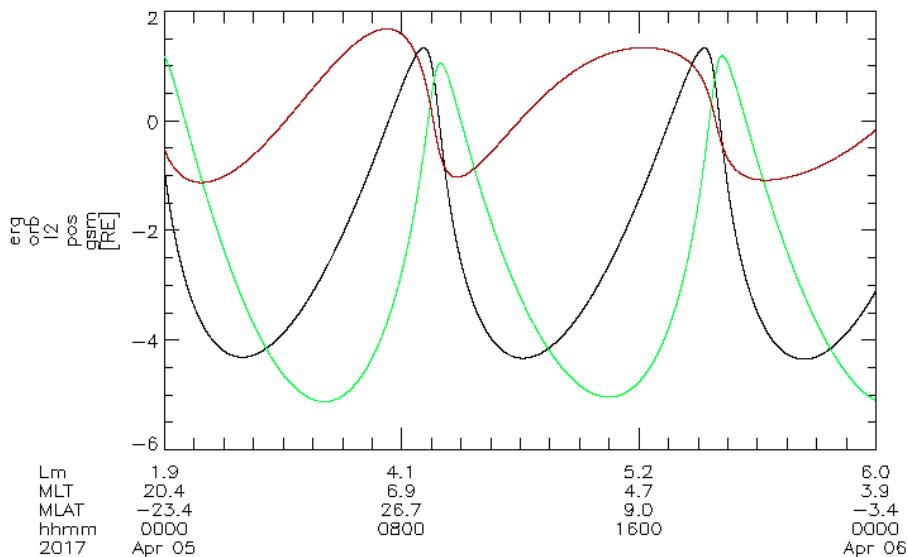
Separate a tplot variable with vector data: *split_vec*

```
ERG> split_vec, 'erg_orb_l2_pos_rmlatmlt'
STORE_DATA(264): Creating tplot variable: ...
STORE_DATA(264): Creating tplot variable: ...
STORE_DATA(264): Creating tplot variable: ...

ERG> tplot, 'erg_orb_l2_pos_rmlatmlt_?'
```

split_vec takes a tplot variable with vector or array data to create new tplot variables containing each component of the vector/array data.
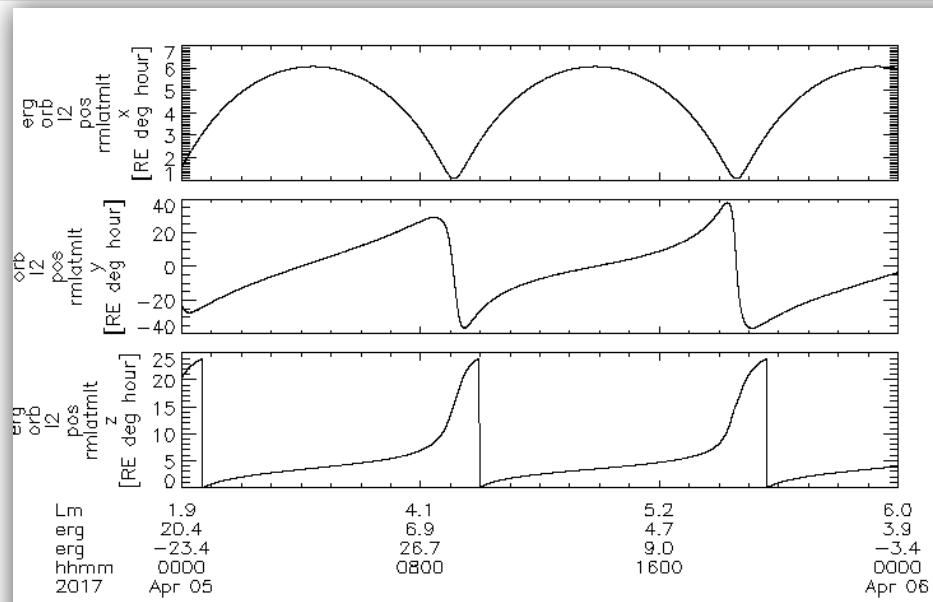
Orbit data:
Insert orbit values below a time-series plot

```
ERG> set_erg_var_label
ERG> tplot, 'erg_orb_l2_pos_gsm'

ERG> split_vec, 'erg_orb_l2_pos_rmlatmlt'
ERG> options, 'erg_orb_l2_pos_rmlatmlt_x', ytitle='R'
ERG> options, 'erg_orb_l2_pos_rmlatmlt_y', ytitle='MLAT'
ERG> options, 'erg_orb_l2_pos_rmlatmlt_z', ytitle='MLT'
ERG> tplot_options, var_label=['erg_orb_l2_pos_rmlatmlt_z',
'erg_orb_l2_pos_rmlatmlt_y', 'erg_orb_l2_pos_rmlatmlt_x']
ERG> tplot
```

Using subroutine to add labels
(Lm,MLT,MLAT) with time

An example to add
new labels
(R,MLAT,MLT)
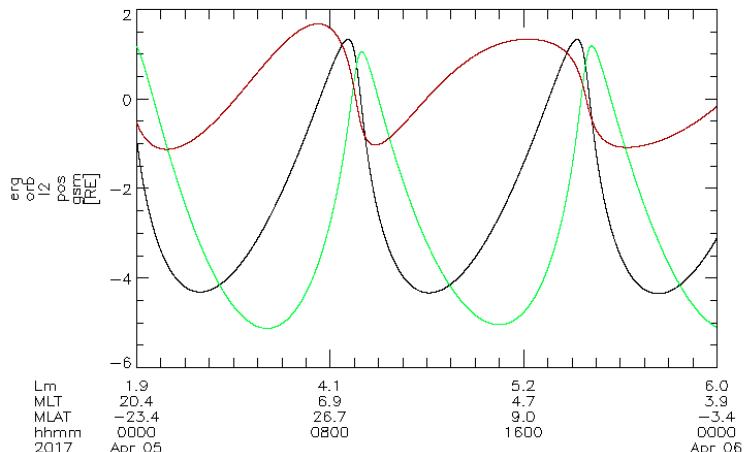
(Lm, MLAT, MLT)	(R, MLT, MLAT)
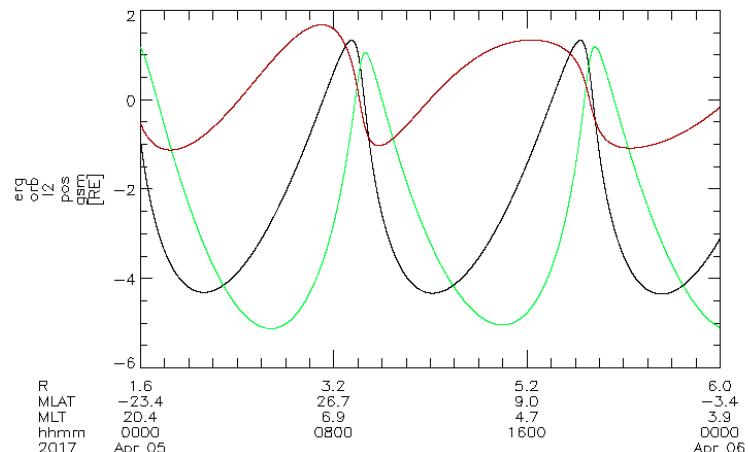
Appendix 1:
Access the data structure in a tplot variable and
create a new tplot variable (get_data, store_data)

SPEDAS training session, ERG WS @NCU, Taiwan   Jan. 15, 2020

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> help, data
** Structure <272b308>, 3 tags, length=3456016,
    X              DOUBLE    Array[172800]
    Y              FLOAT     Array[172800, 3]
    V              LONG      Array[3]
```

```
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]
```

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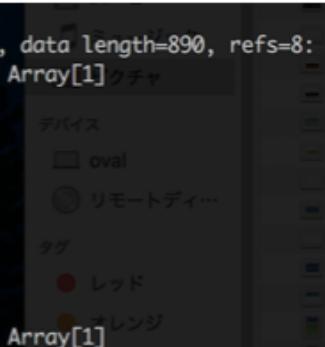
The information on the original CDF data file (CDF) and metadata, and various plot properties are extracted into a structure "dlimits".

Time handling in SPEDAS

```
ERG> print, data.x[0]
1.4914224e+09

ERG> print, time_string( data.x[0] )
2017-04-05/20:00:00

ERG> ts = time_struct( data.x[0] )
ERG> help, ts
** Structure TIME_STRUCTR, 14 tags, length=48, data length=42:
   YEAR            INT           2017
   MONTH           INT              4
   DATE            INT              5
   HOUR            INT             20
   MIN             INT              0
   SEC             INT              0
   FSEC            DOUBLE          0.0000000
   DAYNUM          LONG         736423
   DOY             INT             95
   DOW             INT              2
   SOD             DOUBLE      72000.000
   DST             INT              0
   TZONE           INT              0
   TDIFF           INT              0
```

- Date/time data in a tplot variable are expressed in **double-precision decimal Unix time** (the count of seconds elapsed from 00:00:00, Jan. 1, 1970 in UTC.

- `time_string()` converts a decimal Unix time to a string of human-readable date and time.

- `time_struct()` returns a structure containing the breakdown of a date/time for a given Unix time.

Both functions can take an array as input, returning an array of string or structure, respectively.

An inverse conversion (date/time string or structure → Unix time) can be made with `time_double()` function. Detailed information on these commands are available in the SPEDAS wiki at:
http://spedas.org/wiki/index.php?title=Time_handling

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.

▸ To create a new tplot variable from your own data, one has to prepare a data array and a time array which corresond to timearr and datarr in the above sample command, respectively.

▸ The time array and the data array must have a 1st dimension of the same size.

  ▸ e.g., time: double [1000], data: float [1000, 32]

▸ The time array can be generated from time values of your own data with time_double() function.

Appendix 2:
Various options for SPEDAS use

SPEDAS training session, ERG WS @NCU, Taiwan   Jan. 15, 2020

Options for SPEDAS use

- **Using SPEDAS source code**
  - All SPEDAS IDL source code including both the command-line tools and graphical user interface (GUI) tools.
  - To use it, you need to have IDL installed with a proper license.
  - The latest plug-ins can be used by installing by yourself.
  - Users can develop, edit, and run their own scripts.
- **Using SPEDAS save file**
  - binary files working with the IDL virtual machine (IDL-VM) environment
  - IDL-VM can be downloaded for free from Exelis/Harris Geospatial.
  - No command-line tools: only GUI is available.
  - only plug-ins implemented to the save file are available for use. Some latest plug-ins may not be included yet.
- **Using SPEDAS executable files**
  - A package bundling IDL-VM and the SPEDAS save file, basically equivalent to the above.

- **We use the source code version in this training session.**