

SPEEDAS training session – Beginner course –

by ERG Science Center (ERG-SC)

T. Hori¹⁾, S. Nakamura¹⁾, S. Imajo¹⁾, C. W. Jun¹⁾, M. Kitahara¹⁾,
M. Shoji¹⁾, S. Matsuda²⁾, M. Teramoto³⁾, K. Keika⁴⁾,
N. Umemura¹⁾, T. Segawa¹⁾ and Y. Miyoshi¹⁾

¹⁾ERG Science Center, ISEE, Nagoya Univ.

²⁾ Kyushu Institute of Technology, ³⁾ISAS/JAXA, ⁴⁾Univ. of Tokyo

Purpose of this course

- ▶ To get familiar with the SPEDAS scheme.
- ▶ To get you ready to use SPEDAS to analyze ERG data.

What will you be doing in this training?

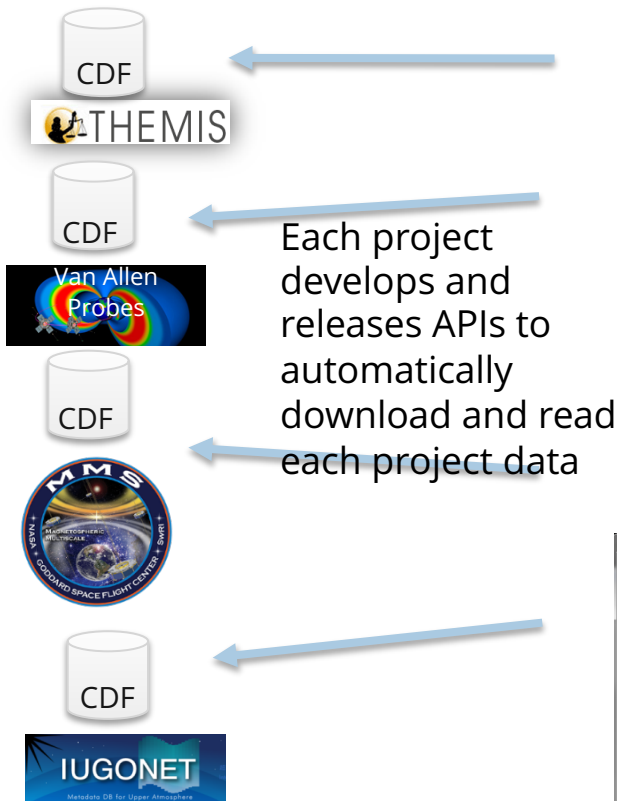
- ▶ Learn about the data model employed by SPEDAS.
- ▶ Load/manipulate/visualize ERG satellite and other ground data.

What's SPEDAS?

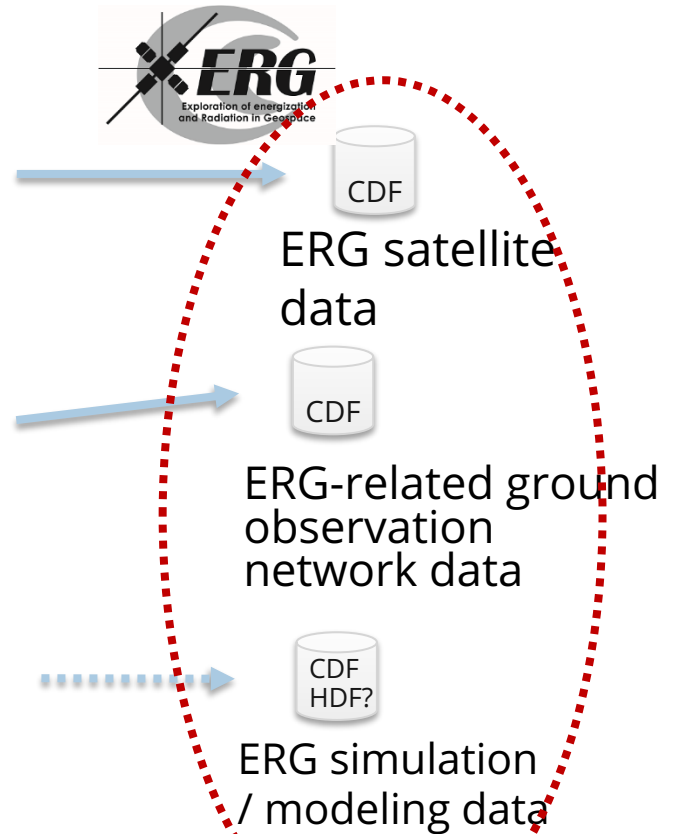
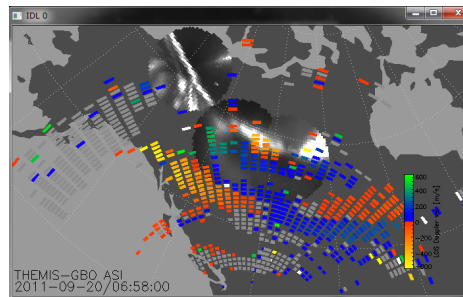
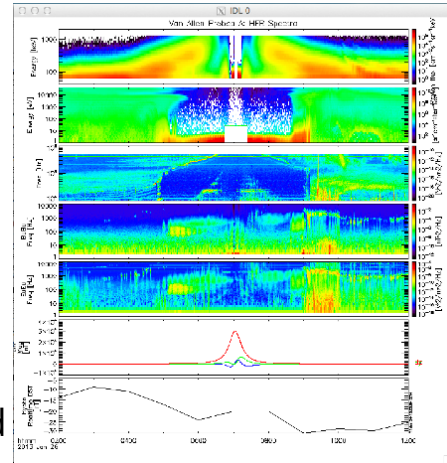
Space Physics Environment Data Analysis Software (SPEDAS)



Data repository



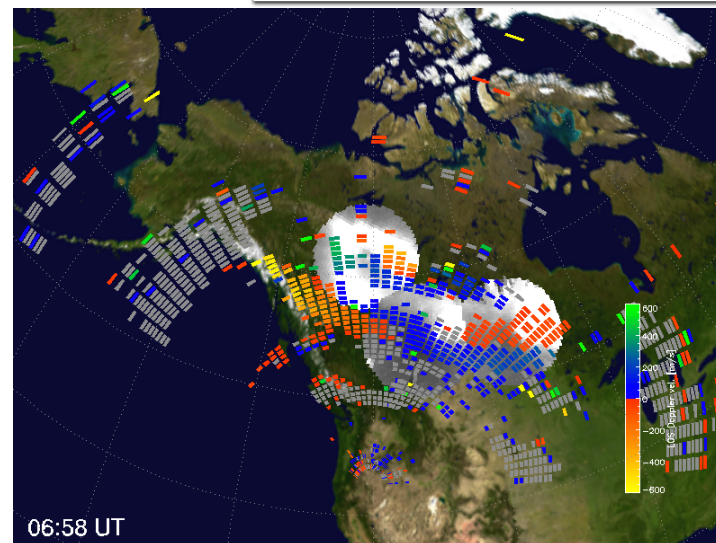
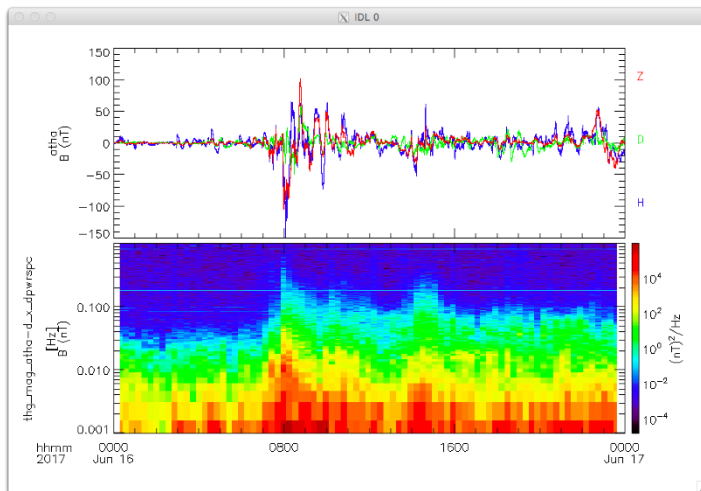
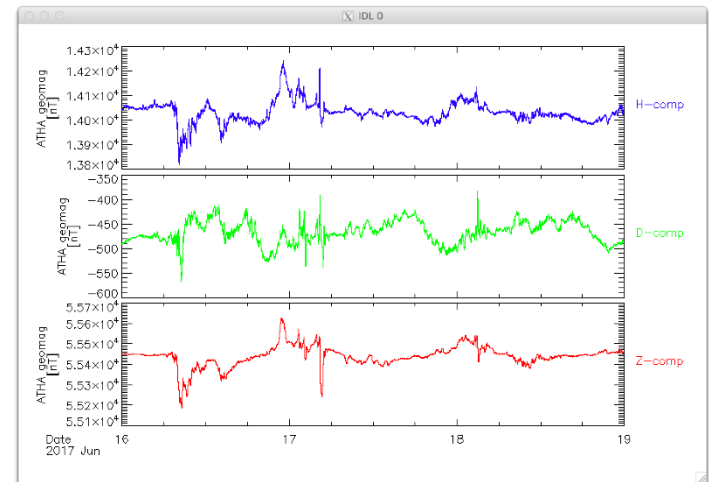
SPEDAS framework



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

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 delimiters (') 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.

```
% Syntax error.
```

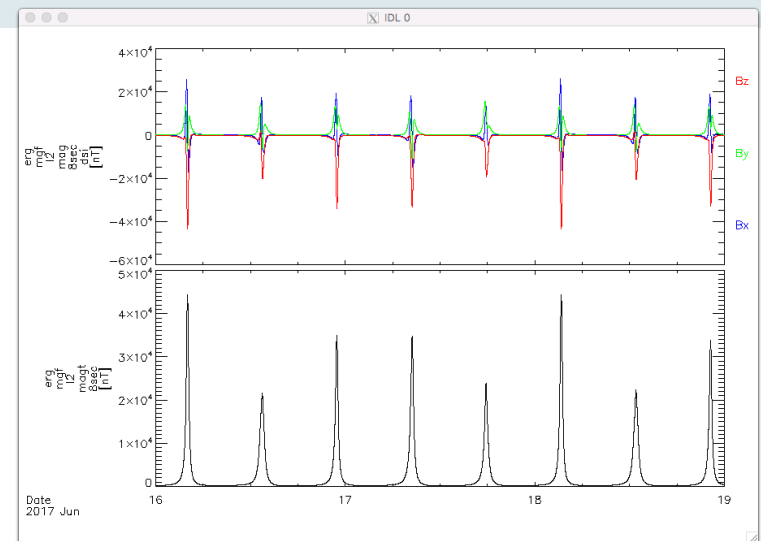
→ , ' () [] is missing or mismatched in most cases.

- ▶ 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:

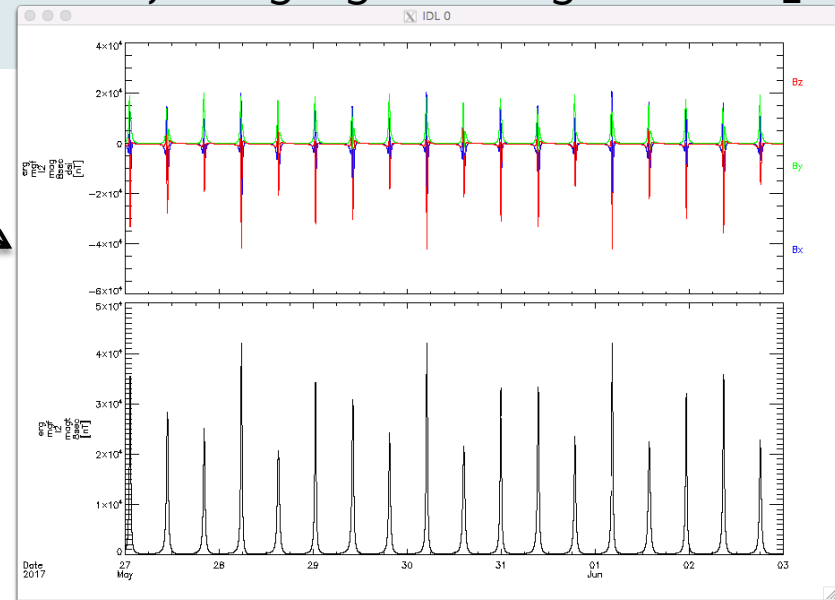
1. Run IDL
2. Initialize the SPEDAS environment on IDL
3. Set a date/time range for which data are loaded.
4. Load data
5. *(Manipulate the loaded data)*
6. Plot the data



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

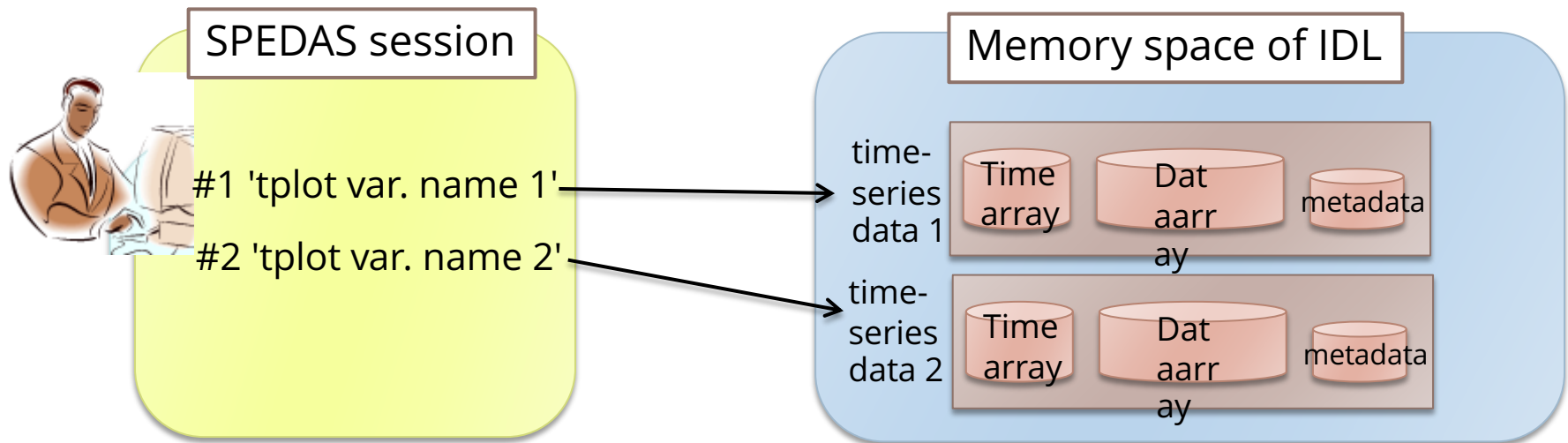
```
ERG> timespan, '2017-05-27'
```

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

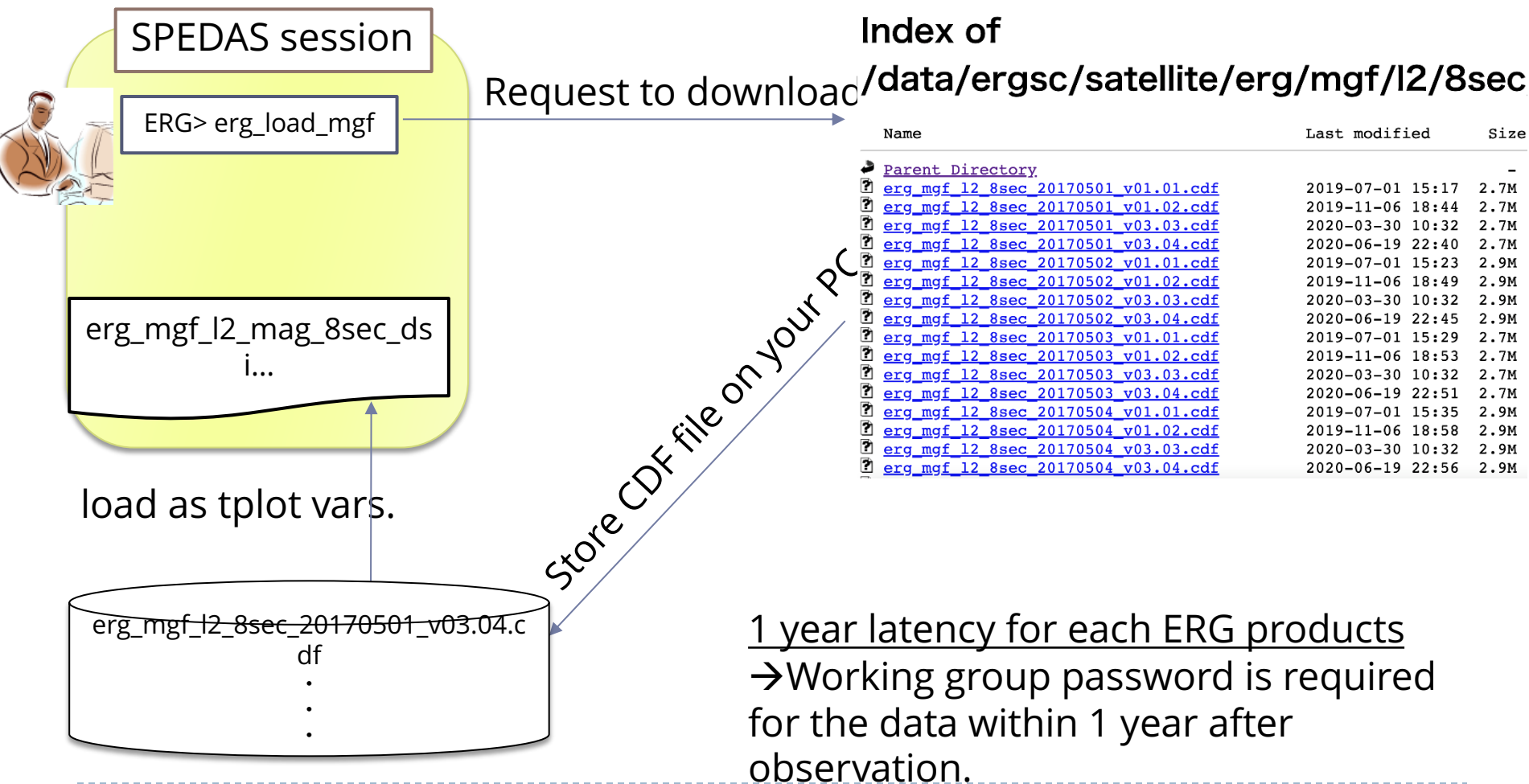
```
ERG> timespan, '2017-05-29/03:25:30', 90, /min
```

tplot variable as the primary data model

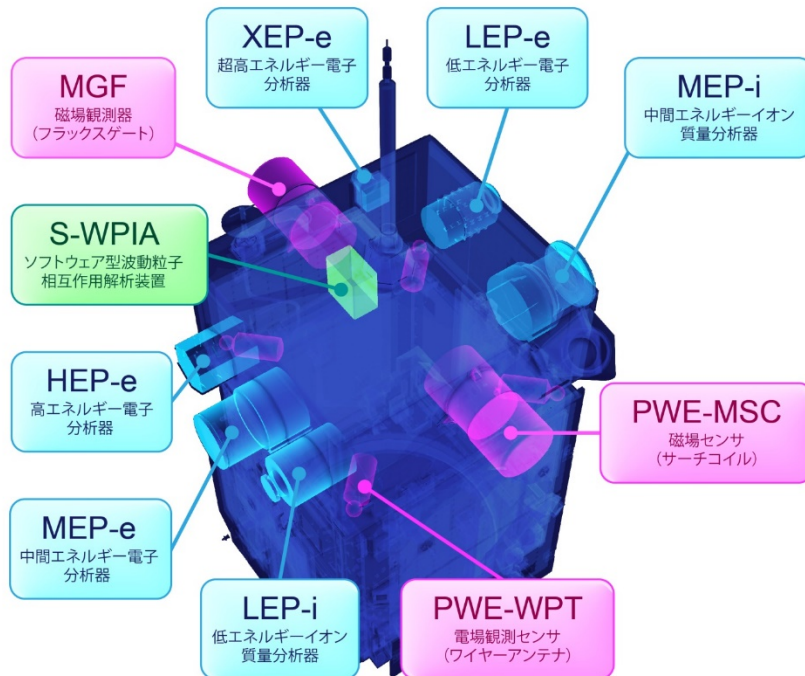
- ▶ 'erg_mgf_l2_mag_8sec_dsi' in prev. page is called *tplot variable*.
- ▶ "Tplot variables" bind an **indexed name-string** to a **data structure on IDL** containing time-series data with metadata.



Ex: How the MGF load procedure works



- 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>
```

```
[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.
```

All tplot variables are listed with unique index numbers

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
```

```
ERG> tplot_names,'erg_mgf_l2_mag_8sec_dsi',/v
% Compiled module: TPLOTT_NAMES.
1 erg_mgf_l2_mag_8sec_dsi
DQ = STRUCT = TPLOTT_QUANT --(7 Tags/64 Bytes)-->
  NAME = STRING = 'erg_mgf_l2_mag_8sec_dsi'
  BH = POINTER = <PtrHeapVar65>
  *(DH) = *-<PtrHeapVar65> = STRUCT --(4 Tags/16 Bytes)-->
    X = POINTER = <PtrHeapVar344>
    *(X) = *-<PtrHeapVar344> = DOUBLE[75685] = [1.4958432e+09, 1.4958432e+09, 1.4958432e+09, ...]
    X_IND = LONG = 75685
    Y = POINTER = <PtrHeapVar345>
    *(Y) = *-<PtrHeapVar345> = DOUBLE[75685,3] = [-495.78699, -498.25568, -502.07184, -505.67904, ...]
    Y_IND = LONG = 75685
  LH = POINTER = <PtrHeapVar66>
  *(LH) = *-<PtrHeapVar66> = STRUCT --(3 Tags/56 Bytes)-->
    LABELS = STRING[3] = ['Bx', 'By', 'Bz']
    COLORS = INT[3] = [2, 4, 6]
    LABFLAG = INT = 1
  DL = POINTER = <PtrHeapVar67>
  *(DL) = *-<PtrHeapVar67> = STRUCT --(4 Tags/1528 Bytes)-->
    CDF = STRUCT --(4 Tags/1504 Bytes)-->
      FILENAME = STRING = '/Volumes/H0-L0B3/data/ergsc/satellite/erg/mgf/l2/8sec/2017/05/erg_mgf_l2_8sec_20170527_v01_01.cdf'
      GATT = STRUCT --(35 Tags/1184 Bytes)-->
        PROJECT = STRING = 'ERG-Exploration of Energization and Radiation in Geospace'
        DISCIPLINE = STRING = 'Space Physics-Magnetospheric Science'
        SOURCE_NAME = STRING = 'MAGASC(ERG)-Inner Magnetosphere'
        DATA_TYPE = STRING = 'l2_mgf_level 2 spin-averaged magnetic field data'
        DESCRIPTOR = STRING = 'MGF-Magnetic Field Experiment'
        DATA_VERSION = STRING[2] = ['01', '01']
        TITLE = STRING = 'Level 2 magnetic field data obtained by the Magnetic Field Experiment (MGF) instrument onboard the ERG satellite'
        TEXT = STRING = ''
        GENERATED_BY = STRING = 'ERG Science Center, operated by ISAS/JAXA and ISEE/Nagoya University as a Joint Research Center for Space Science'
        GENERATION_DATE = STRING = '20180619'
        MDS = STRING = 'Created 06/2018'
        AID_REF = STRING = ''
        LOGICAL_FILE_ID = STRING = 'erg_mgf_l2_8sec_20170527_v001'
        LOGICAL_SOURCE = STRING = 'erg_mgf_l2_8sec'
        LOGICAL_SOURCE_DESCRIPTION = STRING = 'Exploration of Energization and Radiation in Geospace (ERG) Magnetic Field Experiment (MGF) Level 2 spin-averaged magnetic field data'
        PI_NAME = STRING = 'Ayako Matsuoka'
        PI_AFFILIATION = STRING = 'Institute of Space and Astronautical Science, Japan Aerospace Exploration Agency, 3-1-1 Yoshinodai, Chuo-ku, Sagamihara, Kanagawa 252-5210, Japan'
        MISSION_GROUP = STRING = 'ERG'
        INSTRUMENT_TYPE = STRING = 'Magnetic Fields (Space)'
        TEXT_SUPPLEMENT = STRING = ''
        RULES_OF_USE = STRING[23] = [ ... ]
        LINK_TEXT = STRING = 'For more information, see'
        LINK_TITLE = STRING = 'the ERG Science Center website'
        HTTP_LINK = STRING = 'https://ergsc.iasee.nagoya-u.ac.jp'
        TIME_RESOLUTION = STRING = '8 s'
        START_ITI = STRING = '33F08573'
        END_ITI = STRING = '33F40801'
        DATA_START_TIME = STRING = '20170527 00000000000000000000'
        DATA_END_TIME = STRING = '20170527 23595999999999999999'
        DATA_AVERAGING_TYPE = STRING = '8 s average/start'
        SOURCE_FILE = STRING = 'erg_mgf_l2_2017052700_v001.txt'
        ANCHILLARY_FILE = STRING = [' ', ...]
        GENERATION_CODE = STRING = 'satellite/erg/mgf/makecdf_erg_mgf_l2_8sec.pro 1317'
        CALIBRATION_HISTORY = STRING[2] = [ ... ]
        KNOWN_PROBLEMS = STRING = ''
```

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, Sagami-hara,
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> 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
```

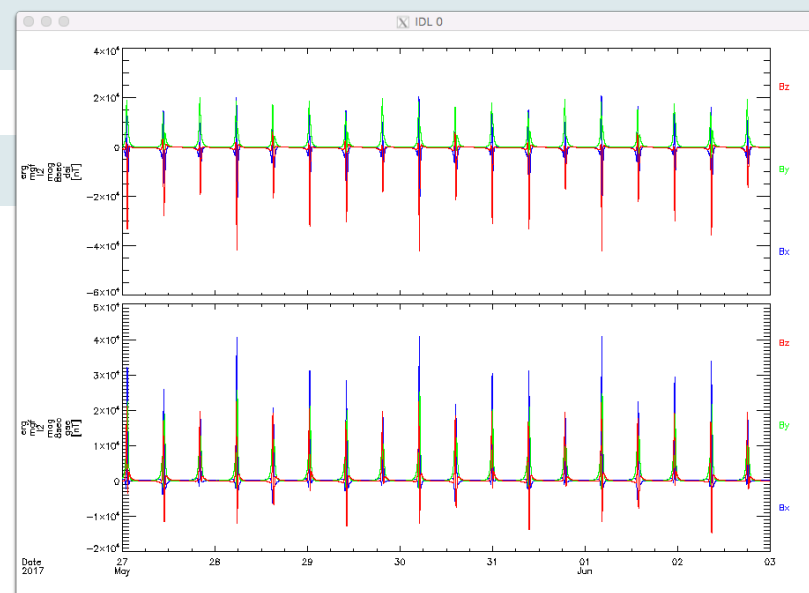
With an array of variable indices to combine multiple variable in a single plot window

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

Some wildcards can be used

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

Tplot accepts variables as arguments in various formats.

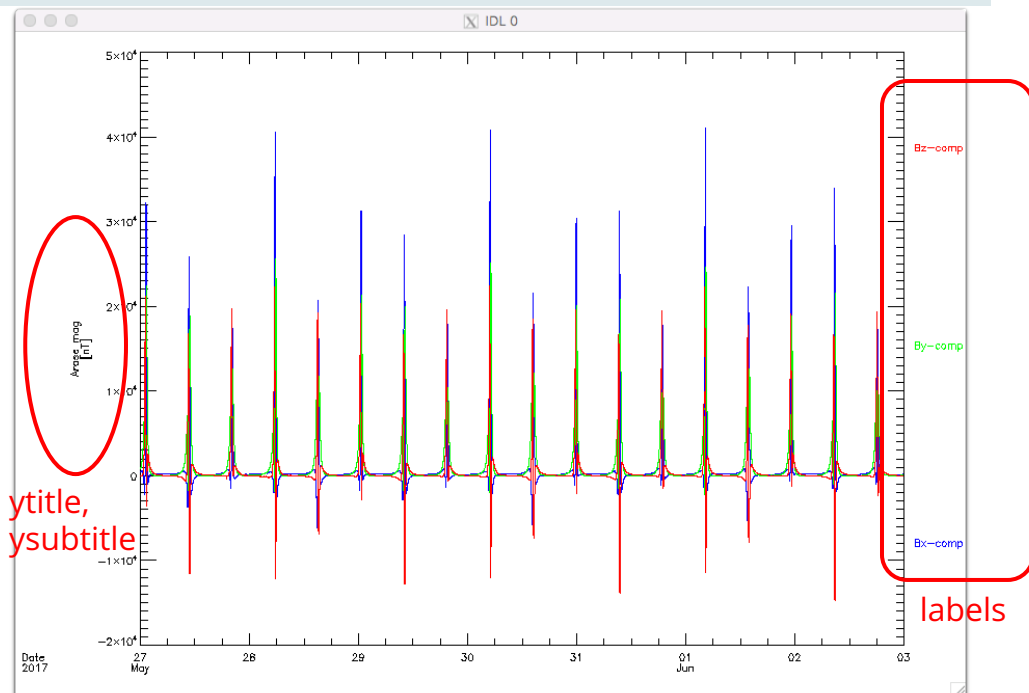


Decorate the plot panel for each tplot variable

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'
  
```



Separate a tplot variable with vector data

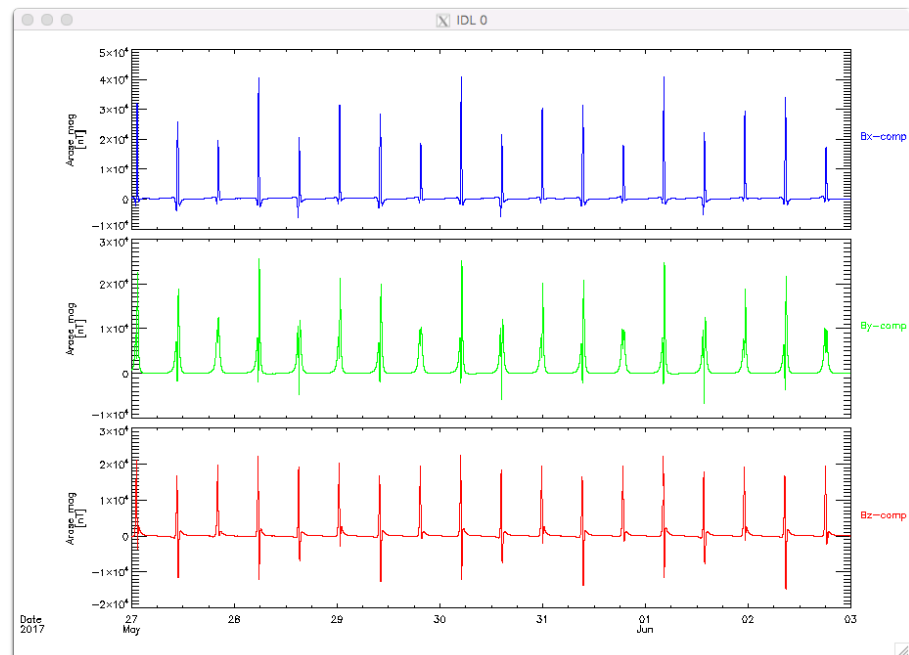
```

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

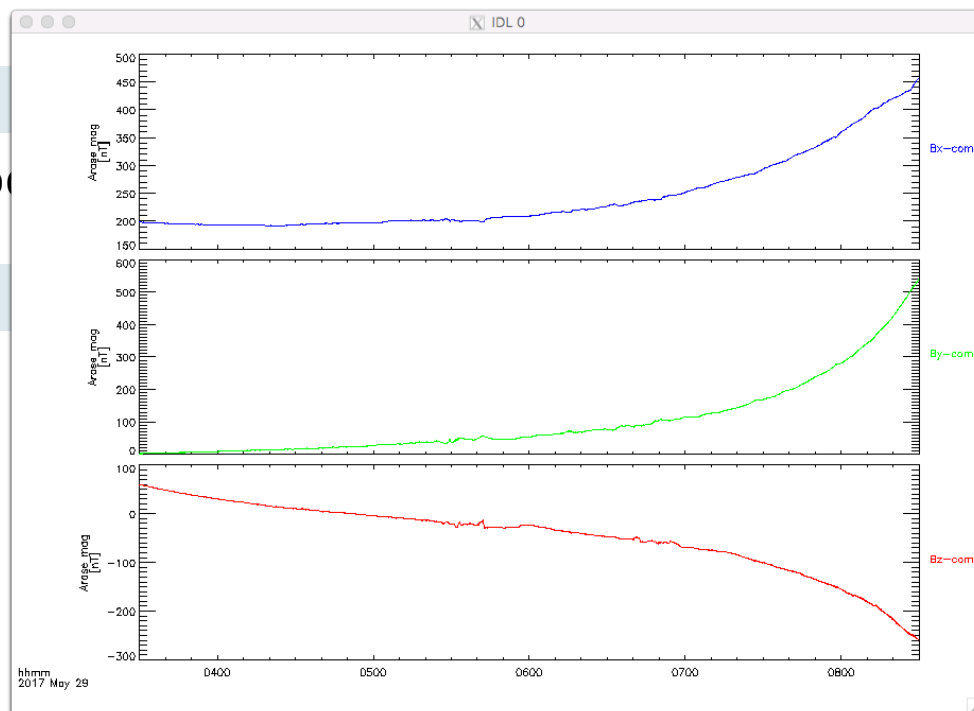
```
ERG> tlimit, '2017-05-29/03:30' , '2017-05-29/08:30'
```

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

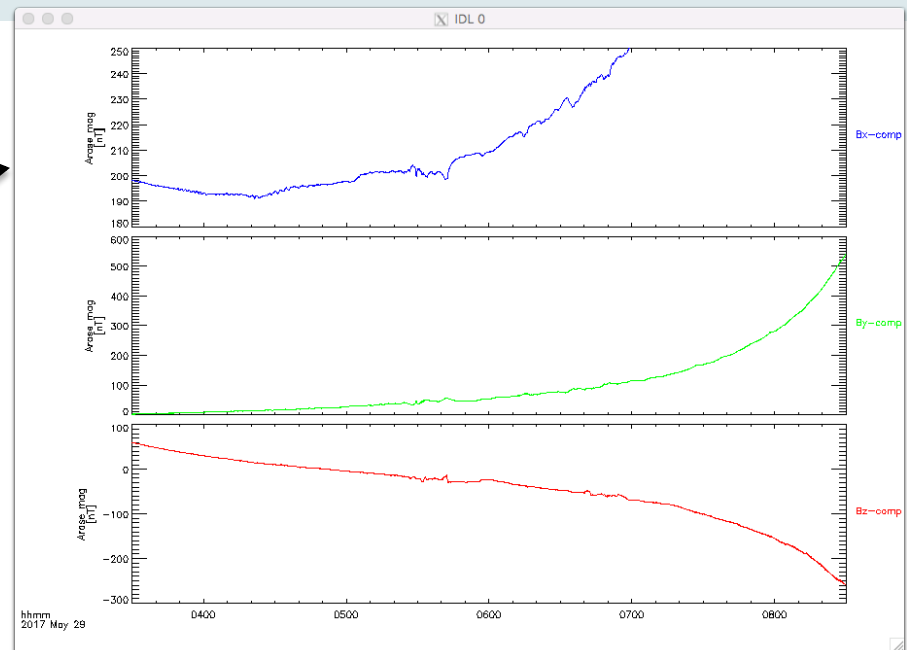
varname : variable name(s)

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

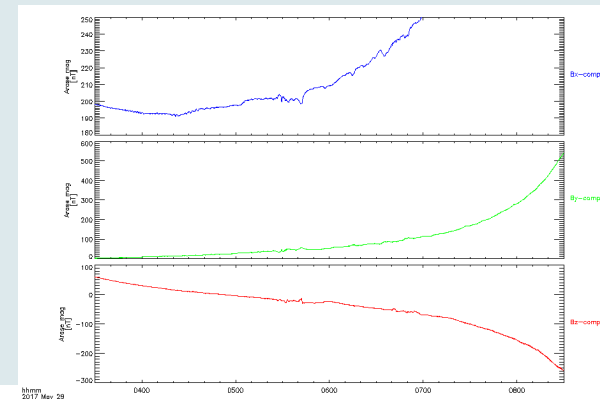
To a png file or postscript file

```

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
    
```

/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
    
```

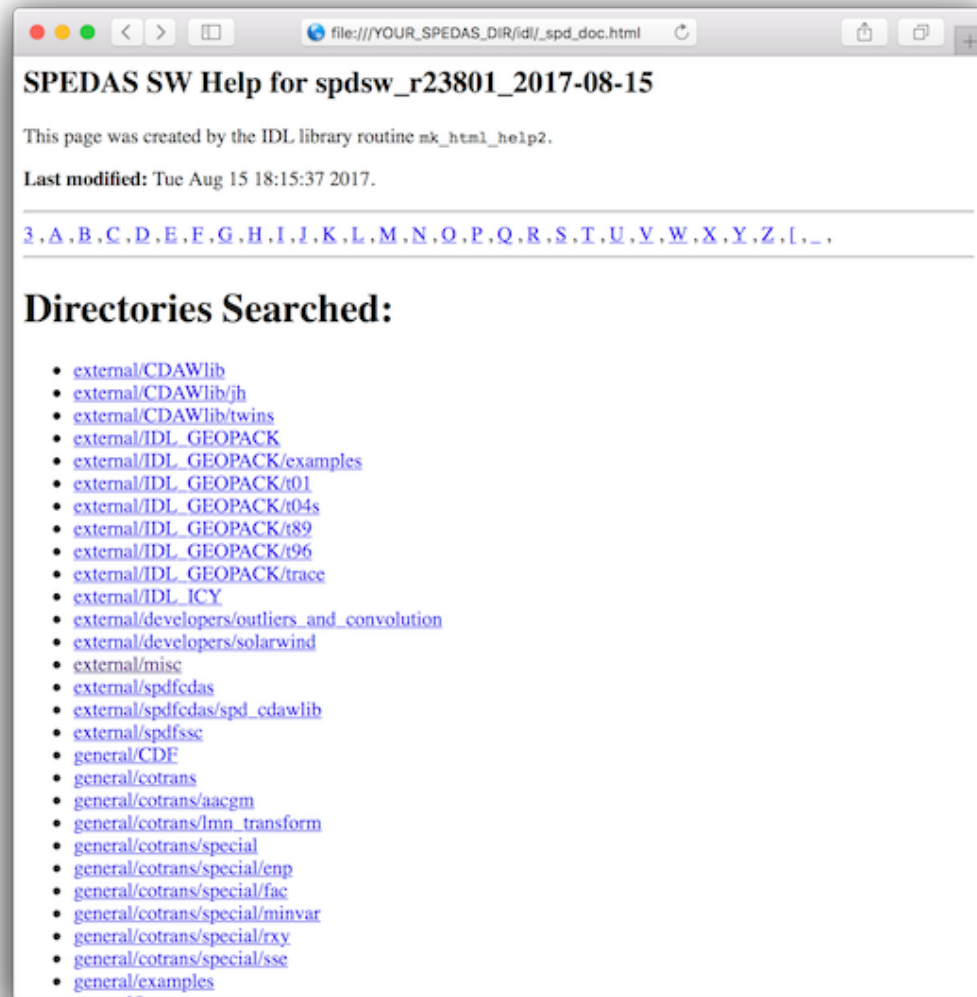
```

kawa3:~ teramari$ cat erg_mgf_l2_mag_8sec_gse.txt
2017-05-27/00:00:05.450      4.84508009e+02      7.9739140e+02      -6.6735940e+01
2017-05-27/00:00:13.435      4.8513178e+02      8.0074430e+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      4.9437937e+02      8.4652933e+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 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

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

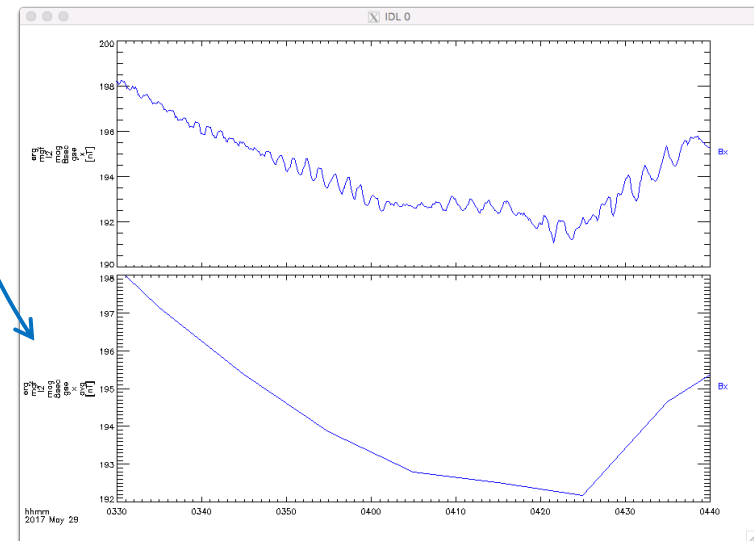
```

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.



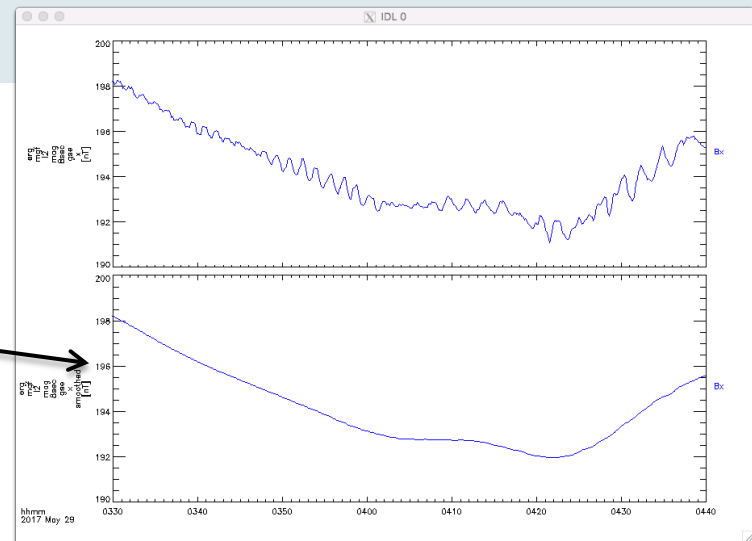
Smoothing data – tsmooth_in_time –

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
```

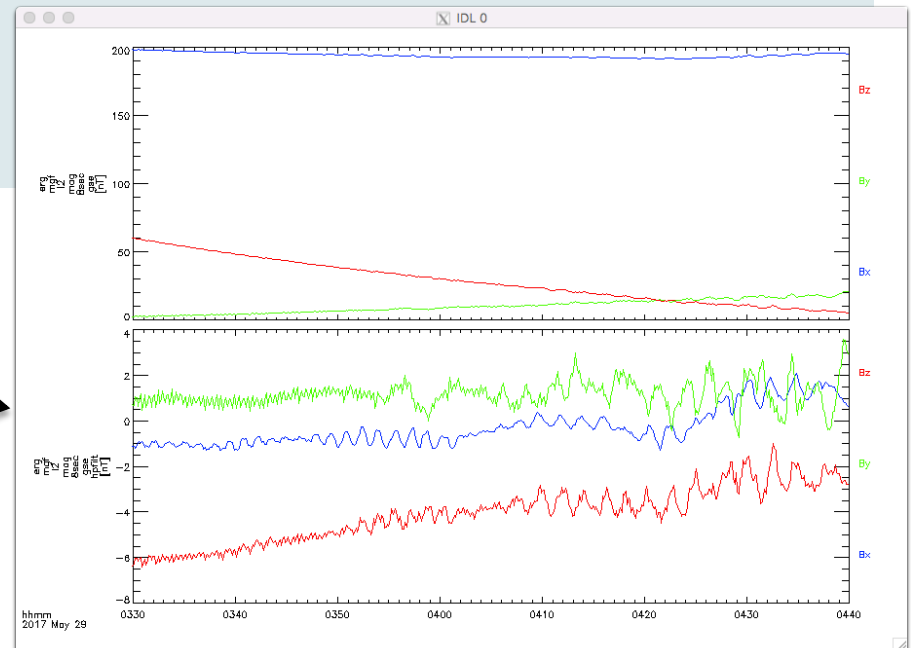
```
.. .. ..
 2  erg_mgf_l2_mag_8sec_gse
```

```
.. .. ..
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 uses any digital filtering** process such as FFT.



Basics of SPEDAS: Frequency analysis of tplot data

Dynamics spectra – tdpwrspc–

tdpwrspc, 'varname'

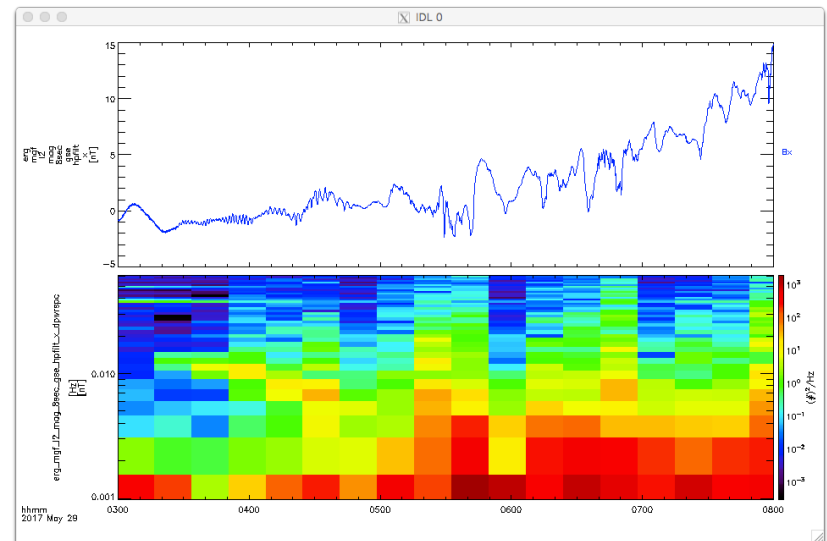
varname : tplot variable name(s)

```

ERG> tdpwrspc, 'erg_mgf_l2_mag_8sec_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 ]
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)

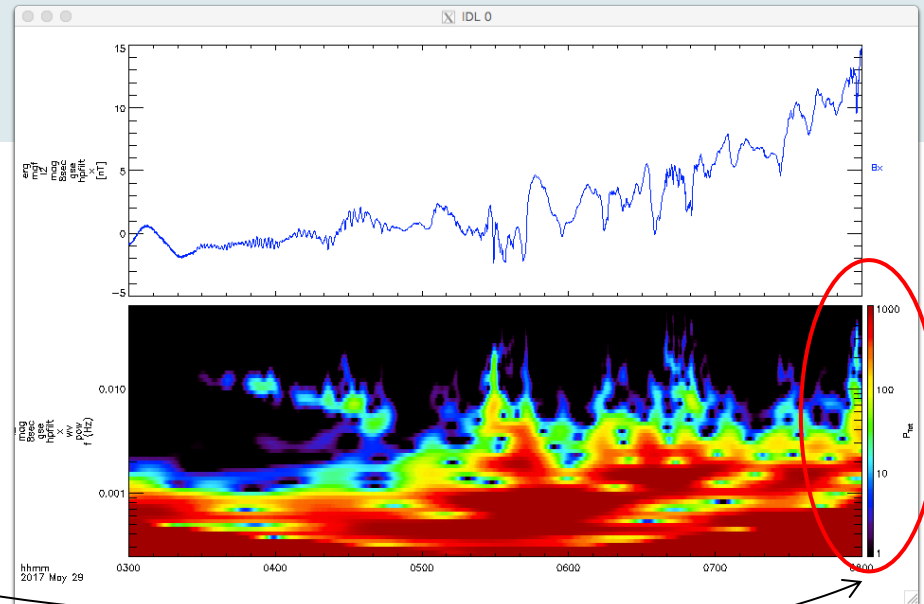
wav_data accepts data with **less than 32768 samples**. The number of data points is reduced as done with avg_data in this case.

```

ERG> wav_data, 'erg_mgf_l2_mag_8sec_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', '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 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 TPLOTT** in `Your_SPEDAS_dir/idl/general/examples/`

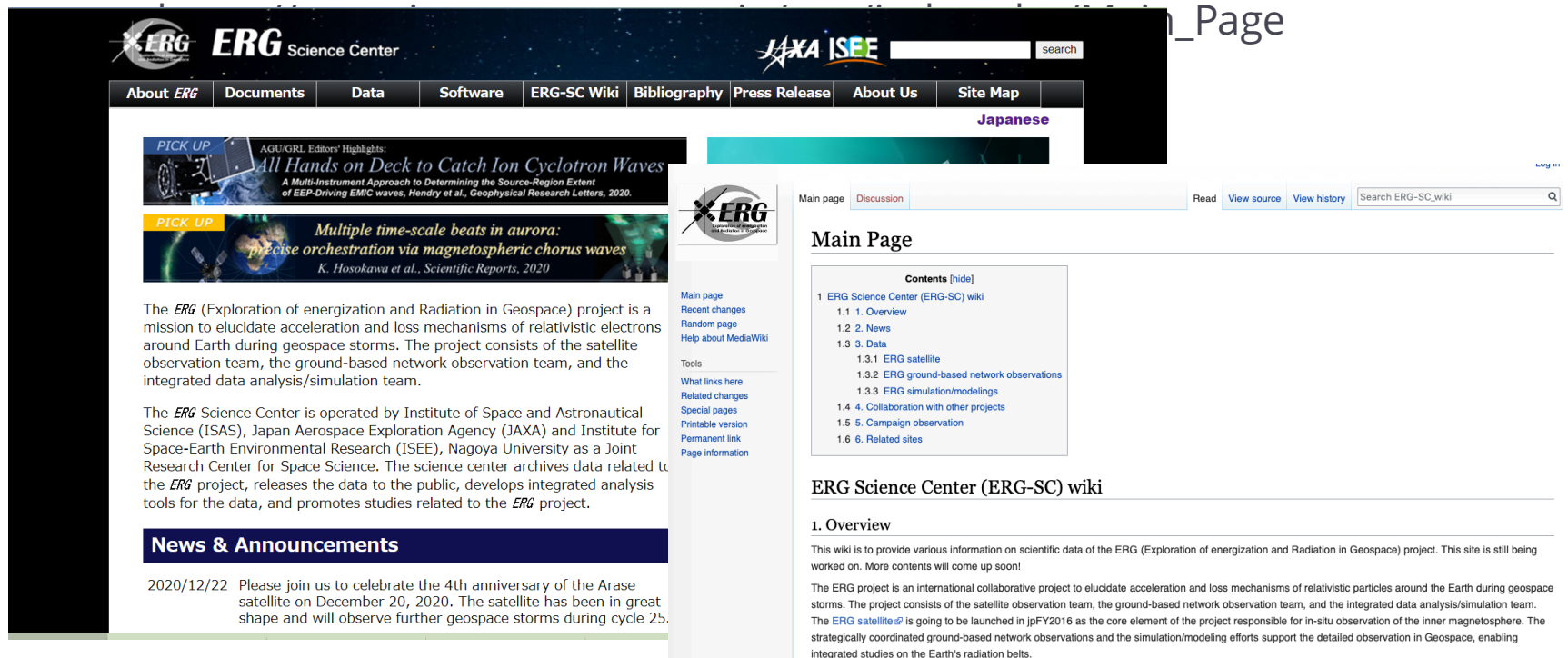
- ▶ `crib_tplot.pro` -- basic tplot intro
- ▶ `crib_tplot_annotation.pro` -- How to control annotations in tplot (labels, text, etc...)
- ▶ `crib_tplot_export_print.pro` -- How to export tplot data and tplot plots
- ▶ `crib_tplot_layout.pro` -- How to control tplot plot layouts
- ▶ `crib_tplot_range.pro` -- 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



The screenshot shows the ERG Science Center website and its corresponding MediaWiki page. The website header includes the ERG logo, 'ERG Science Center', and navigation tabs for 'About ERG', 'Documents', 'Data', 'Software', 'ERG-SC Wiki', 'Bibliography', 'Press Release', 'About Us', and 'Site Map'. A search bar for 'JAXA ISEE' is also present. The main content area features two 'PICK UP' articles: 'All Hands on Deck to Catch Ion Cyclotron Waves' and 'Multiple time-scale beats in aurora: precise orchestration via magnetospheric chorus waves'. A 'News & Announcements' section highlights the 4th anniversary of the Arase satellite launch on December 20, 2020.

The MediaWiki page, titled 'Main Page', includes a 'Contents' table of contents with links to '1. Overview', '2. News', '3. Data', '4. Collaboration with other projects', '5. Campaign observation', and '6. Related sites'. The '1. Overview' section states that the ERG project is an international collaborative project to study relativistic particles around Earth during geomagnetic storms. It mentions the ground-based network observation team and the ERG satellite launch in FY2016.

Sample cases of ERG data analysis

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



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

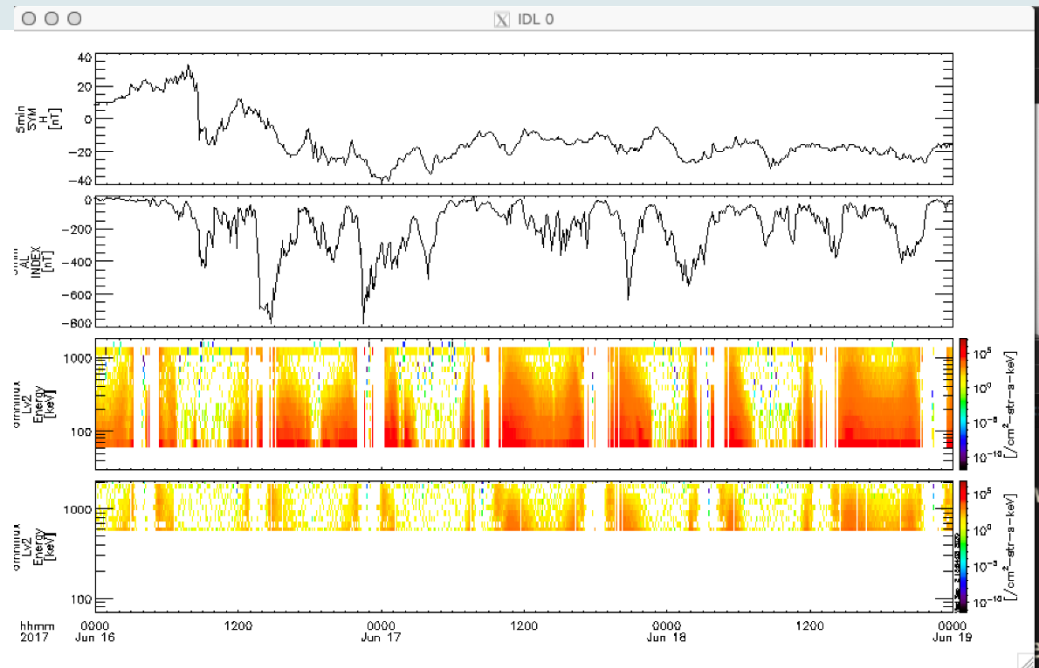
```
Load HEP data
```

```
ERG> erg_load_hep, datatype='omniflux'
```

```
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:

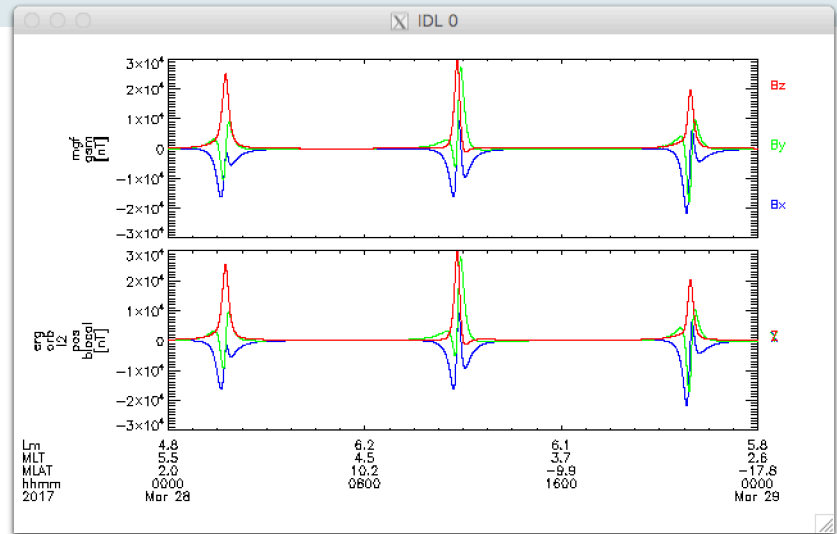
Coordinate transformation of MGF data

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

```
ERG> timespan, '2017-03-28' & erg_load_orb & erg_load_mgf
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'
```

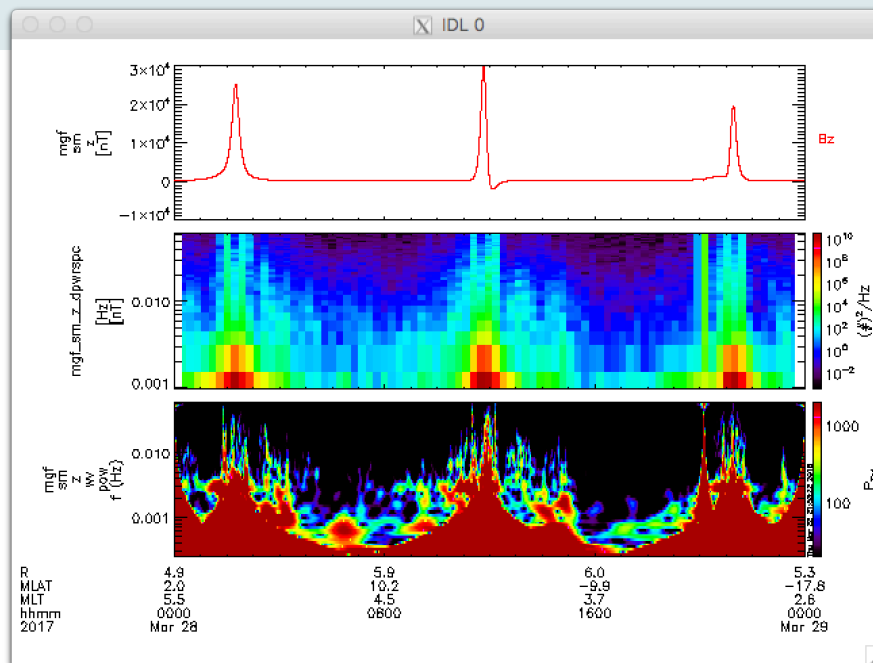
Dynamic spectra by FFT with the Hanning window

```
ERG> wav_data, 'mgf_sm_z'
```

Dynamic spectra by a wavelet analysis

```
ERG> tplot, 'mgf_sm_z*'
```

Use **tlimit** to zoom in a period between perigee points!

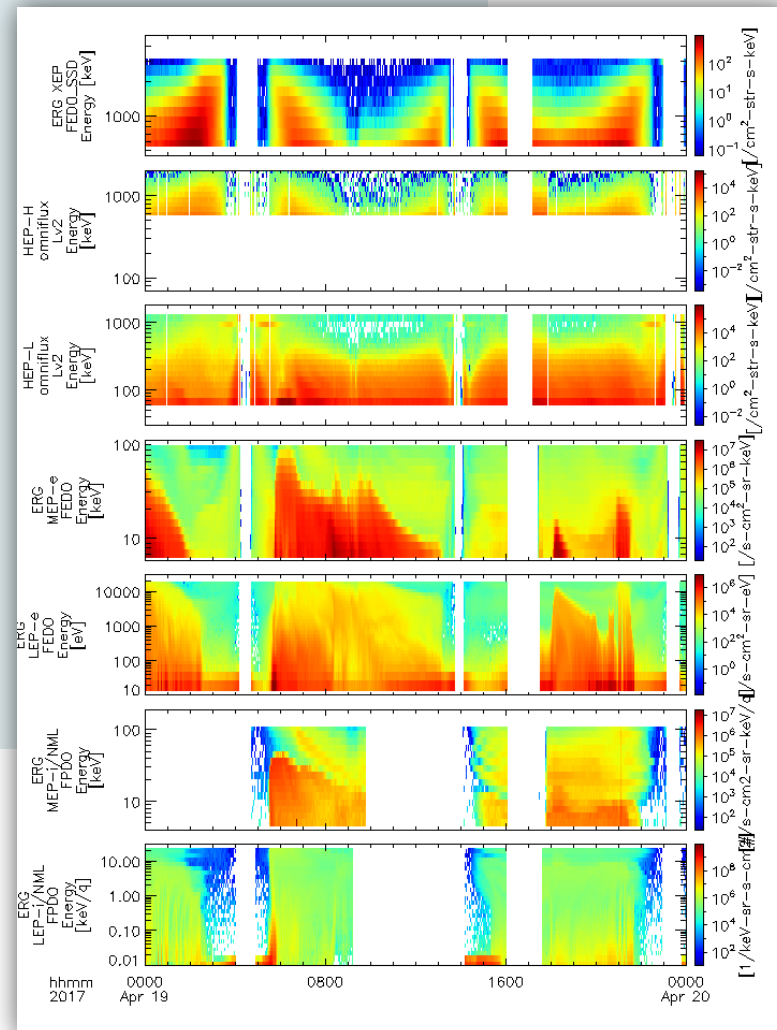


Onboard instrument data: Level-2 particle omni-flux data



```
;; Set the time span
timespan, '2017-04-19'
;; 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'
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

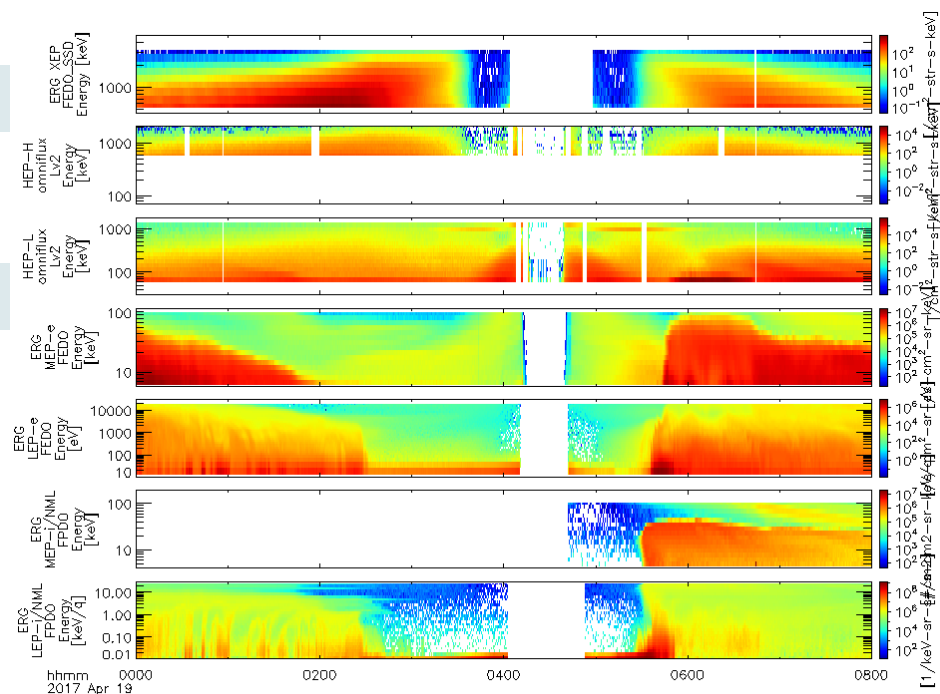
```
ERG> tlimit, '2017-04-19/00:00' , '2017-04-19/08:00'
```

Back to the last plot period

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

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

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



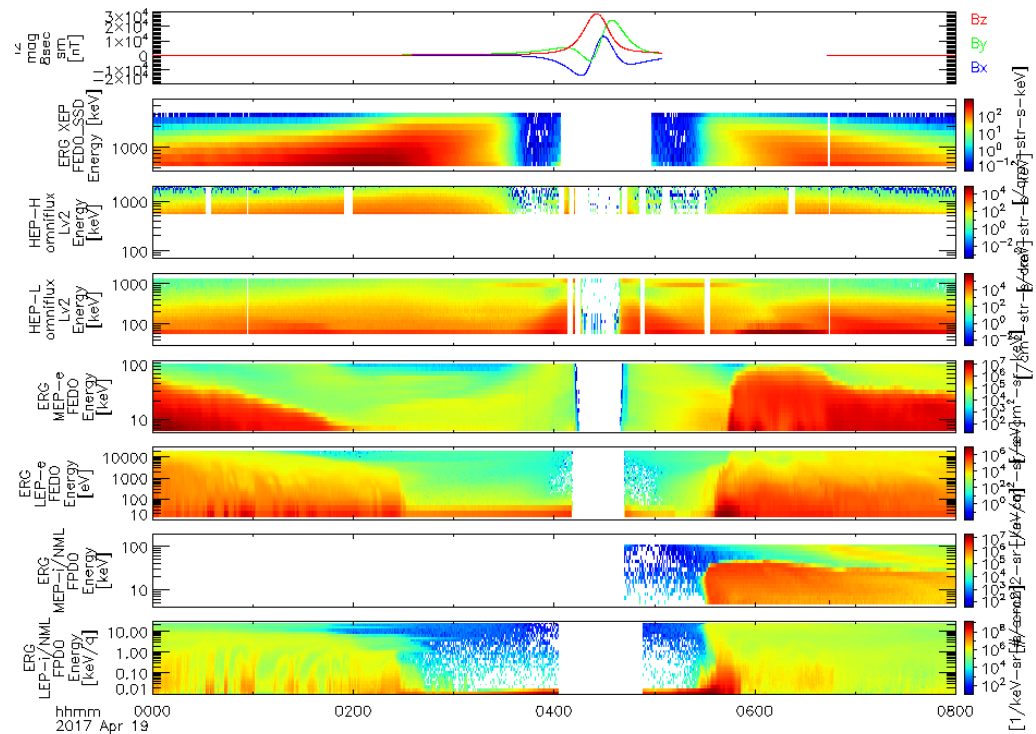
Onboard instrument data: Add MGF data with /add keyword



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

```
ERG> erg_load_mgf
```

```
ERG> tplot, 'erg_mgf_l12_mag_8sec_sm' , /add
```



Orbit data:

Set time range and load ERG orbit data

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

```
ERG> del_data, '*'
```

```
ERG> timespan, '2017-03-28/00:00:00', 3, /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	Mcllwain'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

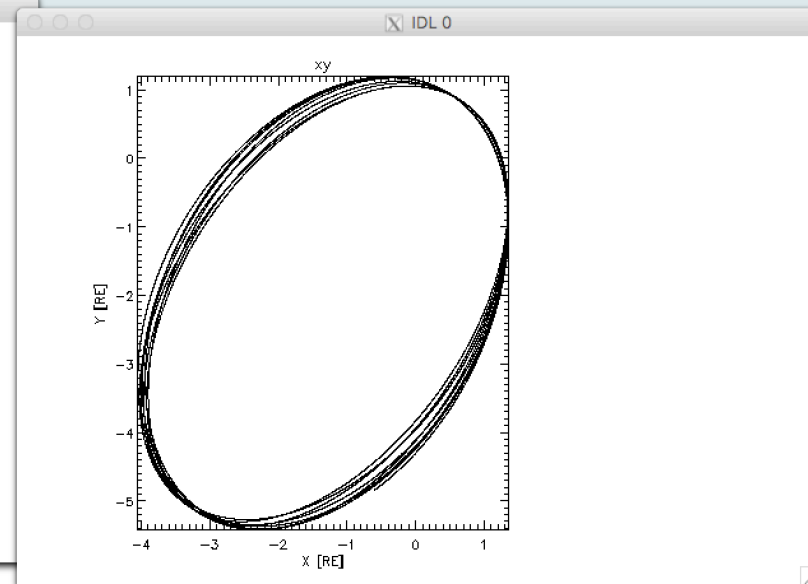
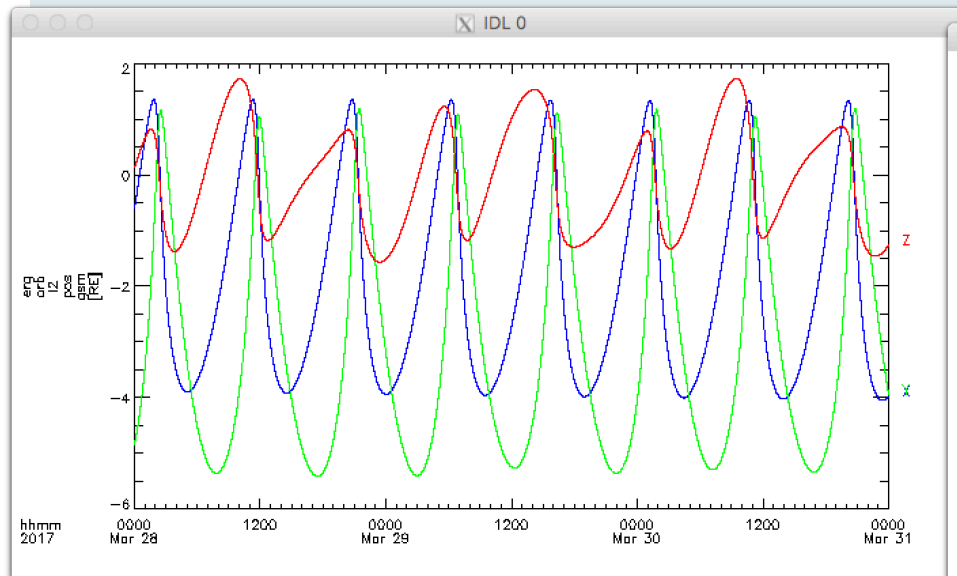


Plot orbit time series data

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

Plot orbit data in the X-Y plane.

```
ERG> tplotxy, 'erg_orb_l2_pos_gsm'
```

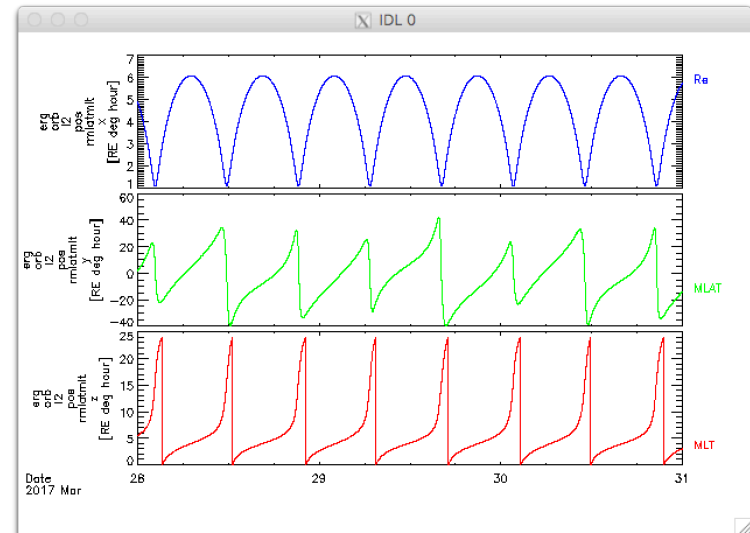


Separate a tplot variable with vector data: *split_vec*



```
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.



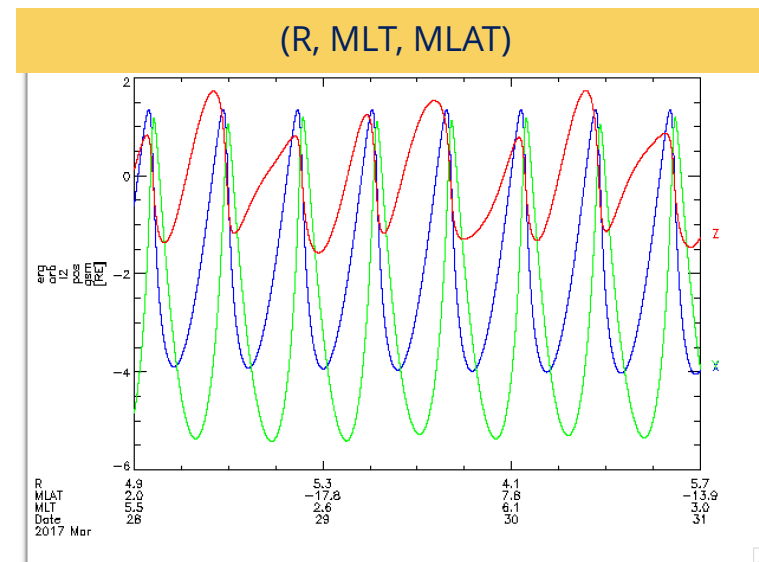
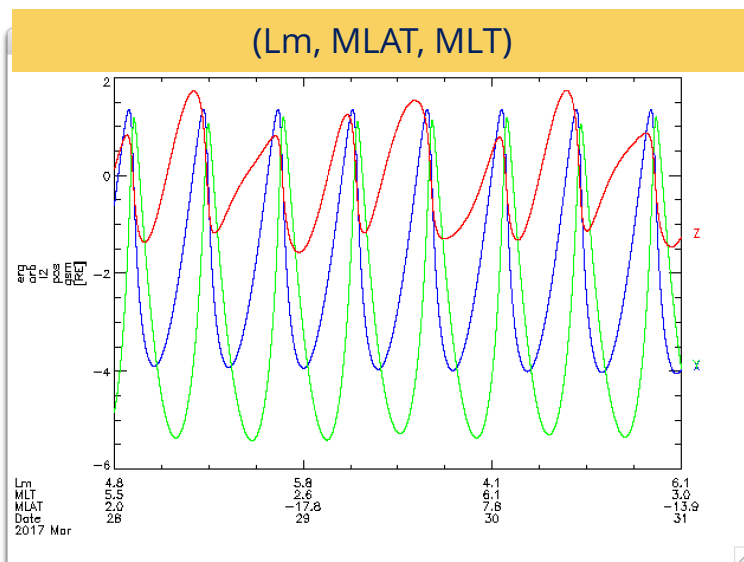
Orbit data: Insert orbit values below a time-series plot

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

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

```
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 ;just type "tplot" to replot the previous panels
```

An example to add
new labels
(R,MLAT,MLT)



Appendix 1:
Access the data structure in a tplot variable and
create a new tplot variable (get_data, store_data)

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, d
X          DOUBLE   Array[172800]
Y          FLOAT    Array[172800, 3]
V          LONG     Array[3]
```

"**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] クチャ
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.

Appendix 2: Various options for SPEDAS use

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.