SPEDAS training session - advanced course -

T. Hori, M. Teramoto, T.-F. Chang, Y. Tsugawa, M. Shoji, S. Kurita, Y. Miyoshi, N. Umemura, T. Segawa
(ERG Science Center, ISEE, Nagoya Univ.)

Goal of this training

- To get familiar with how to load, plot, and manipulate the particle data of the ERG satellite.
  - load and plot the particle flux data of LEP-e, LEP-i, MEP-e, MEP-i, HEP, and XEP.
  - Use the "part_products" libraries to make a plot of:
    - Energy-time spectrogram
    - Phi-/Theta-angle spectrogram
    - Pitch angle spectrogram
    - Gyro-phase spectrogram
    - velocity moments

Keep in mind upon the training

- This is a "hands-on" training for SPEDAS, not a time for e-mail check!
- Communicate with lecturers, tutors, and neighboring skilled users.
- Today's session might not be able to cover all topics in the handouts due to time limitation. It is 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.

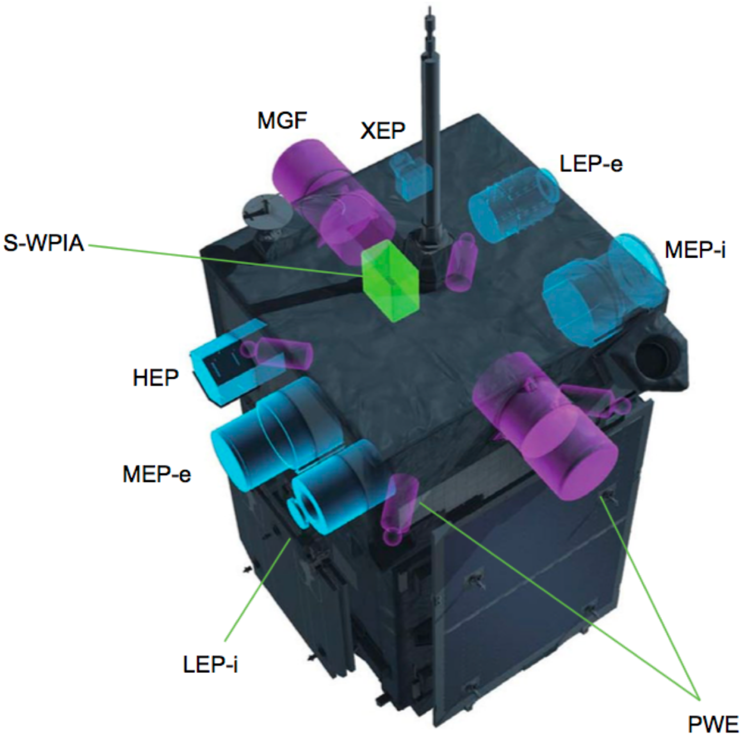
Keep in mind upon the training (cont'd)

- We are going to download fairly large data files today. For a smooth and successful training, please be sure to follow the rules listed below, when you are hooked up to the special WiFi network for the training:
  - Stop using/synchronizing online storages (Dropbox, Google drive, OneDrive, iCloud, etc.) to **save the band width of the network**.
  - Turn off Windows Update temporarily (Windows users).
  - Do not access online movie sites (Youtube, Niko-Niko-Douga(ニコニコ動画), etc.).

Brief introduction of Arase's particle data

About particle data obtained by Arase

- LEP-e (PI: S.-Y. Wang, ASIAA)
  - https://earth-planets-space.springeropen.com/articles/10.1186/s40623-017-0748-6
- LEP-i (PI: K. Asamura, JAXA/ISAS)
  - https://earth-planets-space.springeropen.com/articles/10.1186/s40623-018-0846-0
- MEP-e (PI: S. Kasahara, Univ. of Tokyo)
  - https://earth-planets-space.springeropen.com/articles/10.1186/s40623-018-0847-z
- MEP-i (PI: S. Yokota, Osaka Univ.)
  - https://earth-planets-space.springeropen.com/articles/10.1186/s40623-017-0754-8
- HEP (PI: T. Mitani, JAXA/ISAS)
  - https://earth-planets-space.springeropen.com/articles/10.1186/s40623-018-0853-1
- XEP (PI: N. Higashio, JAXA/RDD)
  - https://earth-planets-space.springeropen.com/articles/10.1186/s40623-018-0901-x

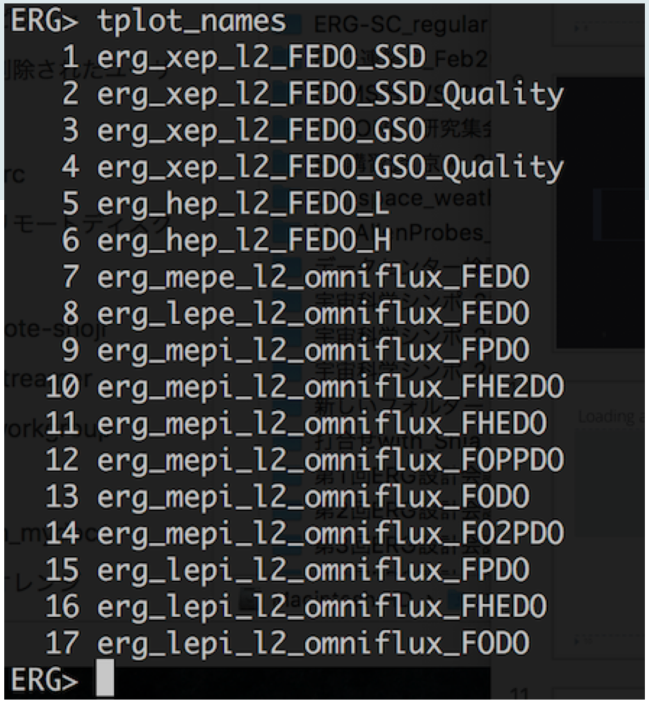


[Miyoshi+2018]

Omni-flux data

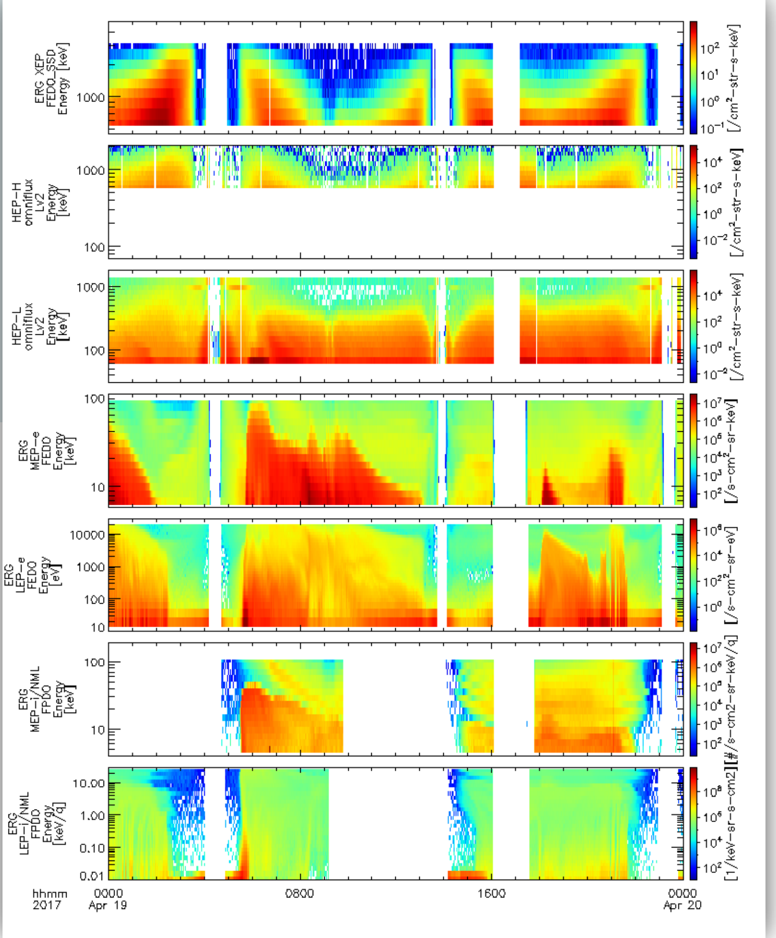
Loading and plotting omni-flux data (1)

```
;; 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
```



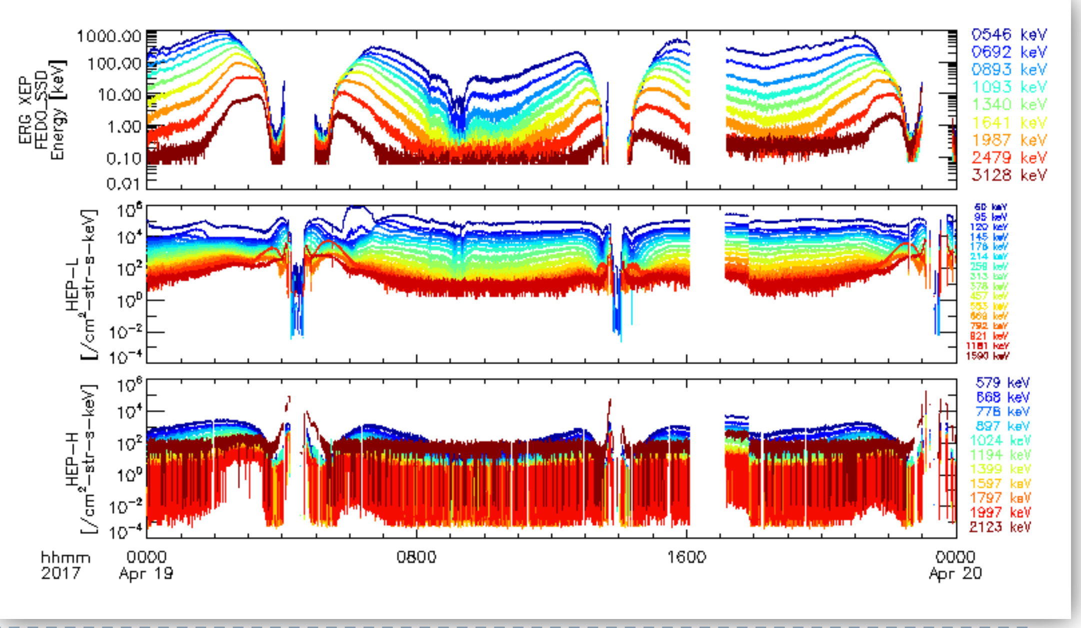
Loading and plotting omni-flux data (2)

```
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'     ]
```



Loading and plotting omni-flux data (3)

```
vn = 'erg_xep_l2_FEDO_SSD'
options, vn, spec=0 & ylim, vn, 0, 0, 1

tplot ;; Replot the previously plotted variables

erg_load_hep, datatype='omniflux', /lineplot
tplot, ['erg_xep_l2_FEDO_SSD', 'erg_hep_l2*line']
```



3-D flux data

Loading and plotting 3-D flux data (1)

```
;; ERG working group ID/password
uname = '???????????'  &  pass = '????????'

timespan, '2017-04-19'
;; erg_load_hep, datatype='3dflux', uname=uname, pass=pass
erg_load_mepe, datatype='3dflux'
;; erg_load_lepe, datatype='3dflux', uname=uname, pass=pass
;; erg_load_mepi_nml, datatype='3dflux'
;; erg_load_lepi_nml, datatype='3dflux', uname=uname, pass=pass
```

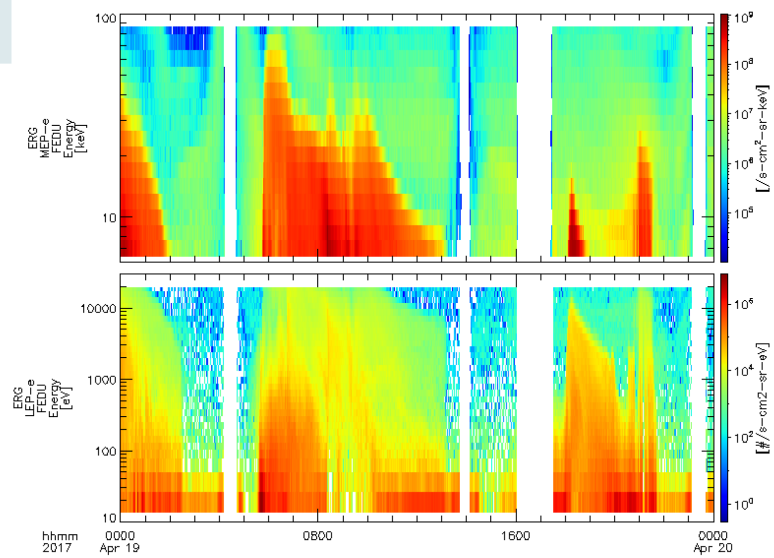
Setting the keyword, `datatype='3dflux'`, makes erg_load_??? load 3-D flux data, except for XEP whose full resolution data are not available as of this training.

!! CAUTION !!

- Loading any two 3-D flux data of MEP-e, MEP-i, LEP-e, and LEP-i at once might use up the memory of your PC and make the PC unstable. If your PC has 8 GB memory or less, it is recommended to remove data pre-existing on memory by typing del_data, '*' before loading a new one.

Loading and plotting 3-D flux data (2)

```
;; In case you have loaded MEP-e and LEP-e:
tplot, [ 'erg_mepe_l2_3dflux_FEDU','erg_lepe_l2_3dflux_FEDU' ]

;; In case you have loaded MEP-e and HEP:
;; Please plot FEDO_? instead of FEDU_? for HEP, otherwise tplot stops
;; with an error.
tplot, [ 'erg_hep_l2_3dflux_FEDO*', 'erg_mepe_l2_3dflux_FEDU' ]
```

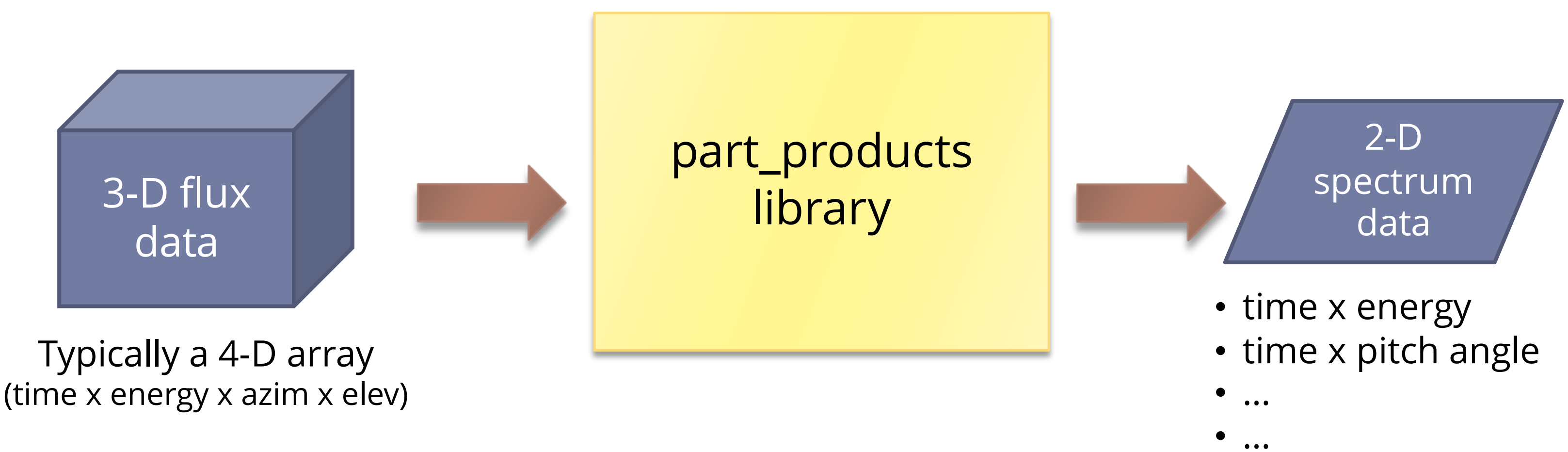


(Memo)

Play with part_products

What's "part_products"?

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



What's "part_products"?

- An experimental version of this library is available from ERG-SC for LEP-e, LEP-i, MEP-e, MEP-i, and HEP so far.
  - `erg_mep_part_products`: MEP-e and MEP-i
  - `erg_hep_part_products`: HEP
  - `erg_lepe_part_products`: LEP-e
  - `erg_lepi_part_products`: LEP-i

Typically a 4-D array
(time x energy x azim x elev)

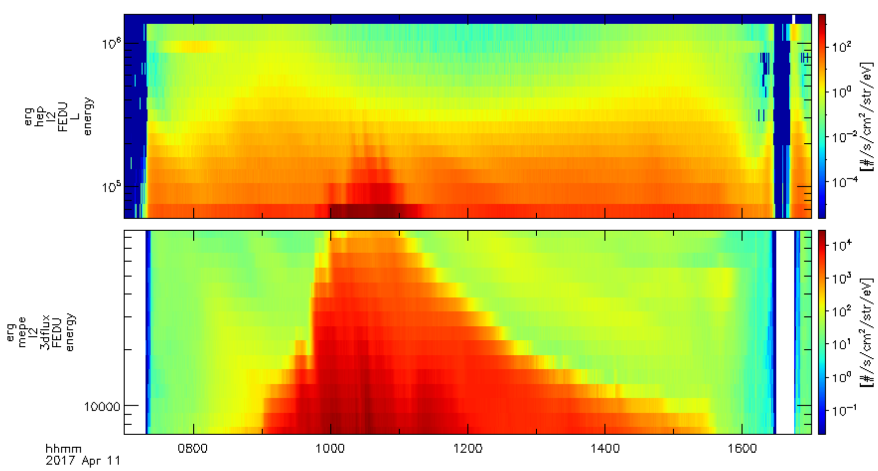
- time x energy
- time x phi/theta angle
- time x pitch angle
- ...

Energy-time spectrogram

```
del_data, '*'
timespan, '2017-04-11/07:00',10, /hour
erg_load_hep, datatype='3dflux', uname=uname, pass=pass
erg_load_mepe, datatype='3dflux'

erg_hep_part_products, 'erg_hep_l2_FEDU_L'
erg_mep_part_products, 'erg_mepe_l2_3dflux_FEDU'

tplot, ['erg_hep_l2_FEDU_L_energy','erg_mepe_l2_3dflux_FEDU_energy']
```

Running part_productcts without any option gives omni-directional fluxes by averaging over all directions.

The default unit of energy and differential flux becomes eV and #/s/cm2/str/eV when flux data are processed with part_products.



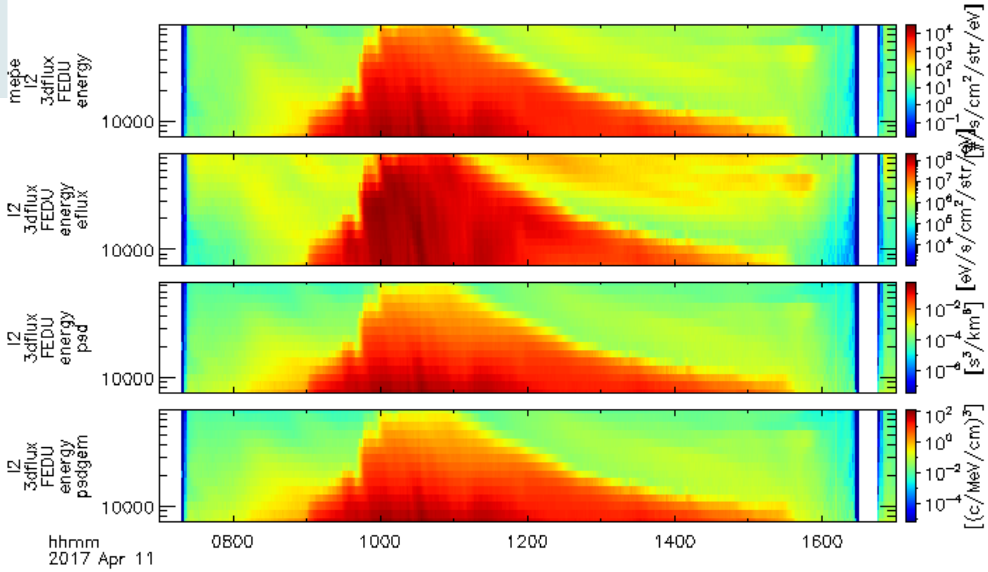
Energy-time spectrogram in different units

```
erg_mep_part_products, 'erg_mepe_l2_3dflux_FEDU', units='flux' ;; Default
erg_mep_part_products, 'erg_mepe_l2_3dflux_FEDU', units='eflux', suffix='_eflux'
erg_mep_part_products, 'erg_mepe_l2_3dflux_FEDU', units='df', suffix='_psd'
;; With keyword "relativistic" on, phase space density values are calculated
;; in (c/MeV/cm)³ considering the relativistic effect. But this conversion
;; is still experimental and needs to be further tested.
erg_mep_part_products, 'erg_mepe_l2_3dflux_FEDU', units='df', suffix='_psdgem', /rela

tplot, ['erg_mepe_l2_3dflux_FEDU_energy*']
```

Differential number flux

Differential energy flux

Phase space density

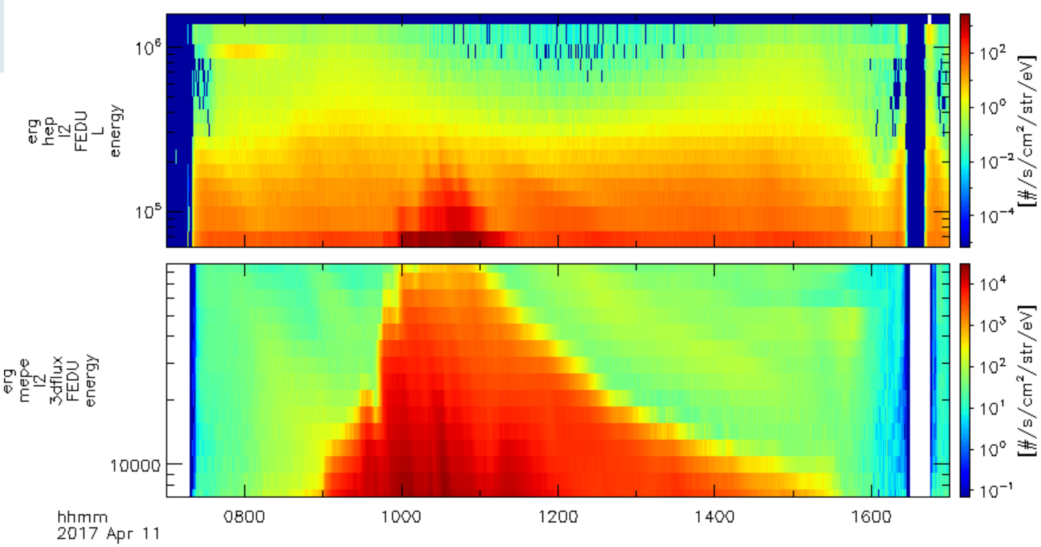


Energy-time spectrogram for limited directional channels

```
get_timespan, tr
erg_hep_part_products, 'erg_hep_l2_FEDU_L', theta=[-30.,30], trange=tr
erg_mep_part_products, 'erg_mepe_l2_3dflux_FEDU', phi=[0.,90], trange=tr

tplot, ['erg_hep_l2_FEDU_L_energy','erg_mepe_l2_3dflux_FEDU_energy']
```

Phi and theta should be given as a range of angle in the DSI coordinates.

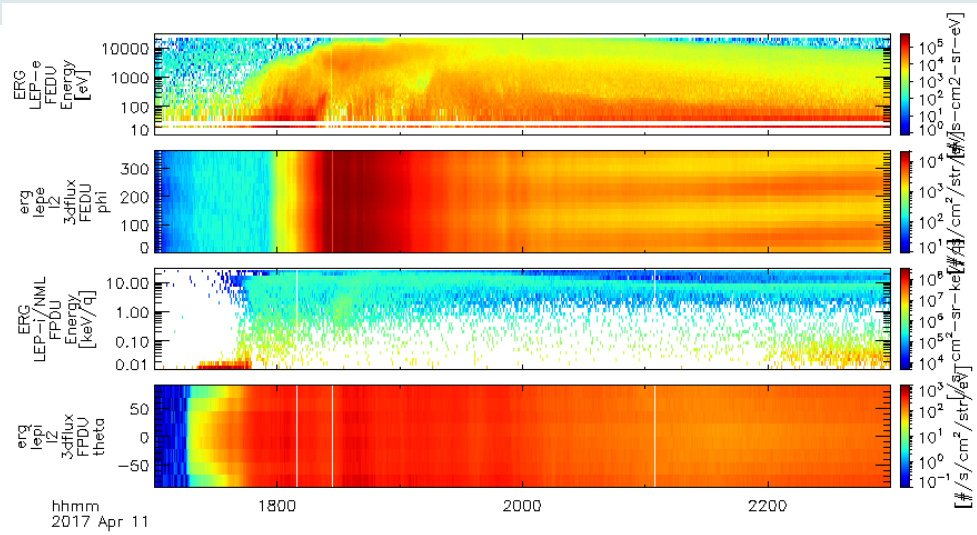
Keywords theta and phi can be set together to specify a limited area of solid angle in DSI.



Phi-/Theta-angle spectrogram

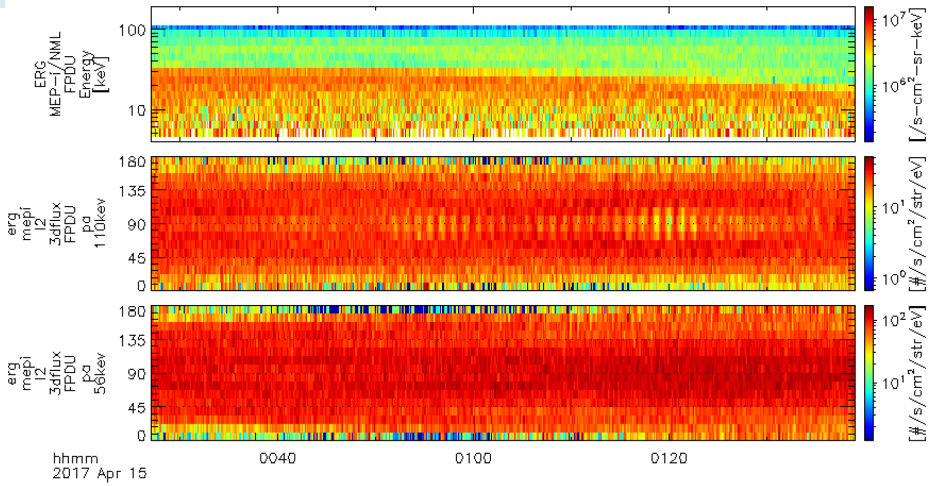
```
del_data, '*'
timespan, '2017-04-11/17:00', 6, /hour
erg_load_lepe, datatype='3dflux', /no_sort_enebin, uname=uname, pass=pass, varformat='FEDU'
erg_load_lepi_nml, datatype='3dflux', uname=uname, pass=pass, varformat='FPDU'

get_timespan, tr
erg_lepe_part_products, 'erg_lepe_l2_3dflux_FEDU', outputs='phi', trange=tr, energy=[3000.,10000.]
erg_lepi_part_products, 'erg_lepi_l2_3dflux_FPDU', outputs='theta', trange=tr, energy=[8000.,20000.]
zlim, 'erg_lepi_l2_3dflux_FPDU_theta', 1e-1, 1e+3, 1

tplot, ['erg_lepe_l2_3dflux_FEDU*', 'erg_lepi_l2_3dflux_FPDU*']
```

Keyword 'energy' specifies an energy range in eV for which particle flux data are averaged to deduce a phi-/theta-spectrogram.



Pitch-angle spectrogram (1)

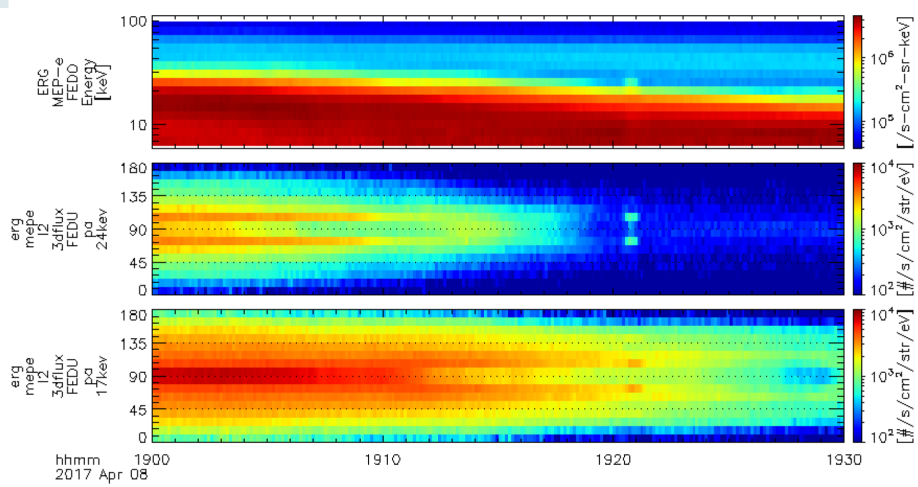
```
del_data, '*'
timespan, '2017-04-15/00:00', 2, /hour & get_timespan, tr
erg_load_mepi_nml, datatype='3dflux', varformat='FPDU'
erg_load_mgf
erg_load_orb
erg_mep_part_products, 'erg_mepi_l2_3dflux_FPDU', output='pa', pos='erg_orb_l2_pos_gse',
mag='erg_mgf_l2_mag_8sec_dsi', energy=[108000., 112000.], suffix='_110kev', trange=tr
erg_mep_part_products, 'erg_mepi_l2_3dflux_FPDU', output='pa', pos='erg_orb_l2_pos_gse',
mag='erg_mgf_l2_mag_8sec_dsi', energy=[54000., 58000.], suffix='_56kev', trange=tr


tplot, 'erg_mepi_l2_3dflux_FPDU*'
```



The flux modulations discussed by Yamamoto+, GRL, 2018

Pitch-angle spectrogram (2)

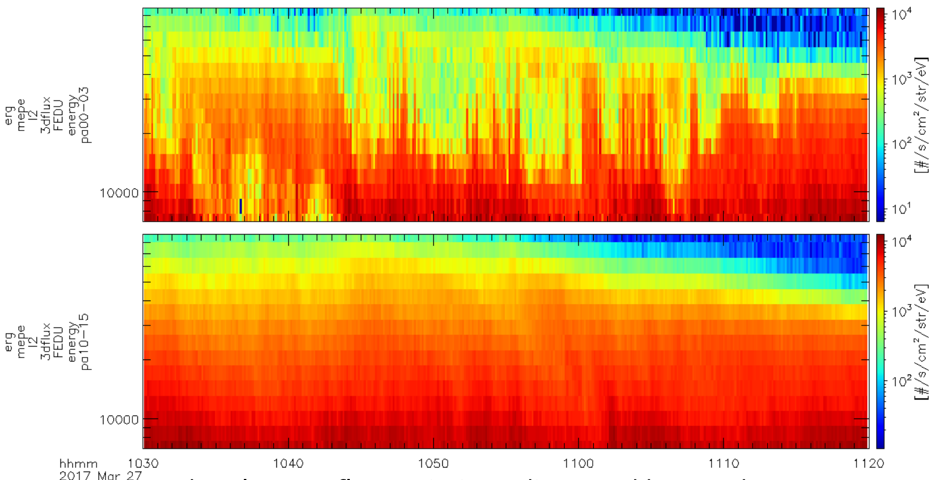
```
del_data, '*'
timespan, '2017-04-08/19:00', 30, /min & get_timespan, tr
erg_load_mepe, datatype='3dflux', varformat='FEDU
erg_load_mgf
erg_load_orb
erg_mep_part_products, 'erg_mepe_l2_3dflux_FEDU', output='pa', pos='erg_orb_l2_pos_gse',
mag='erg_mgf_l2_mag_8sec_dsi', energy=[16000., 18000.], suffix='_17kev', trange=tr
erg_mep_part_products, 'erg_mepe_l2_3dflux_FEDU', output='pa', pos='erg_orb_l2_pos_gse',
mag='erg_mgf_l2_mag_8sec_dsi', energy=[23000., 25000.], suffix='_24kev', trange=tr

zlim, 'erg_mepe_l2_3dflux_FEDU_pa*', 1e+2, 1e+4, 1
tplot, 'erg_mepe_l2_3dflux_'+['FEDO', 'FEDU_pa*']
```



The pitch-angle spectra discussed by Kurita+, GRL, 2018

Energy-time spectrogram for a limited pitch-angle range

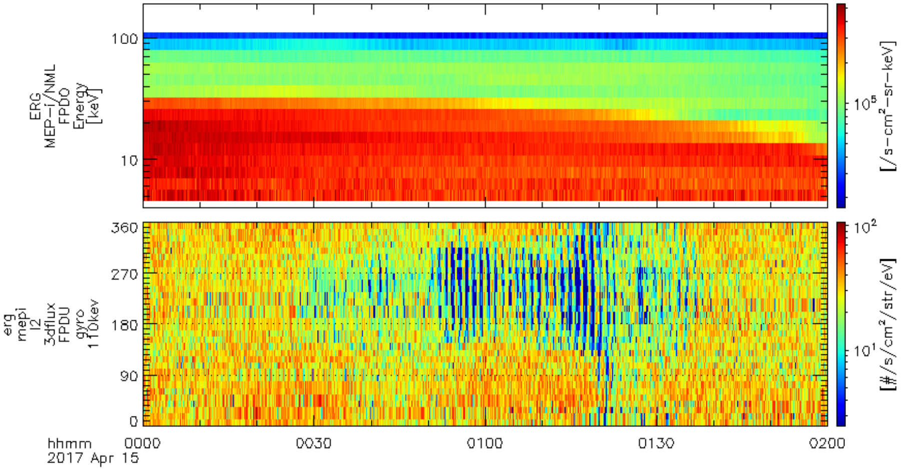
```
del_data, '*'
timespan, '2017-03-27/10:30', 50, /min
erg_load_mepe, datatype='3dflux'
erg_load_mgf & erg_load_orb


get_timespan, tr
erg_mep_part_products, 'erg_mepe_l2_3dflux_FEDU', pos='erg_orb_l2_pos_gse', mag='erg_mgf_l2_mag_8sec_dsi', output='energy', trange=tr, pitch=[0.,3.], suffix='_pa00-03', /no_regrid
erg_mep_part_products, 'erg_mepe_l2_3dflux_FEDU', pos='erg_orb_l2_pos_gse', mag='erg_mgf_l2_mag_8sec_dsi', output='energy', trange=tr, pitch=[10.,15.], suffix='_pa10-15', /no_regrid


tplot, 'erg_mepe_l2_3dflux_'+['FEDU_energy*']
```



The electron flux variations discussed by Kasahara+, Nature, 2018

Gyro-phase spectrogram

```
del_data, '*'
timespan, '2017-04-15/00:00', 2, /hour
erg_load_mepi_nml, datatype='3dflux', varformat='FPDU'
erg_load_mgf
erg_load_orb

get_timespan, tr
erg_mep_part_products, 'erg_mepi_l2_3dflux_FPDU', output='gyro',
pos='erg_orb_l2_pos_gse', mag='erg_mgf_l2_mag_8sec_dsi', energy=[108000., 112000.],
pitch=[85.,95.], trange=tr, suffix='_110kev'

tplot, 'erg_mepi_l2_3dflux_'+['FPDO', 'FPDU_gyro*']
```



The flux modulations discussed by Yamamoto+, GRL, 2018

Velocity moments of 3-D distribution functions

Moment calculation by part_products

- A part_products calls moments_3d(), an internal routine, to calculate various velocity moments from a 3-D distribution function.

- So far part_products for only MEP-e and MEP-i support the moment calculation.

Moment calculation (1)

```
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
```

```
ERG> tplot_names, 'erg_mepi_l2_3dflux_FPDU_*'
  47 erg_mepi_l2_3dflux_FPDU_avgtemp
  48 erg_mepi_l2_3dflux_FPDU_density
  49 erg_mepi_l2_3dflux_FPDU_eflux
  50 erg_mepi_l2_3dflux_FPDU_flux
  51 erg_mepi_l2_3dflux_FPDU_mftens
  52 erg_mepi_l2_3dflux_FPDU_ptens
  53 erg_mepi_l2_3dflux_FPDU_sc_current
  54 erg_mepi_l2_3dflux_FPDU_velocity
  55 erg_mepi_l2_3dflux_FPDU_vthermal
  56 erg_mepi_l2_3dflux_FPDU_magf
  57 erg_mepi_l2_3dflux_FPDU_magt3
  58 erg_mepi_l2_3dflux_FPDU_t3
  59 erg_mepi_l2_3dflux_FPDU_sc_pot
  60 erg_mepi_l2_3dflux_FPDU_symm
  61 erg_mepi_l2_3dflux_FPDU_symm_theta
  62 erg_mepi_l2_3dflux_FPDU_symm_phi
  63 erg_mepi_l2_3dflux_FPDU_symm_ang
ERG>
```

Primary parameters calculated with the part_products:

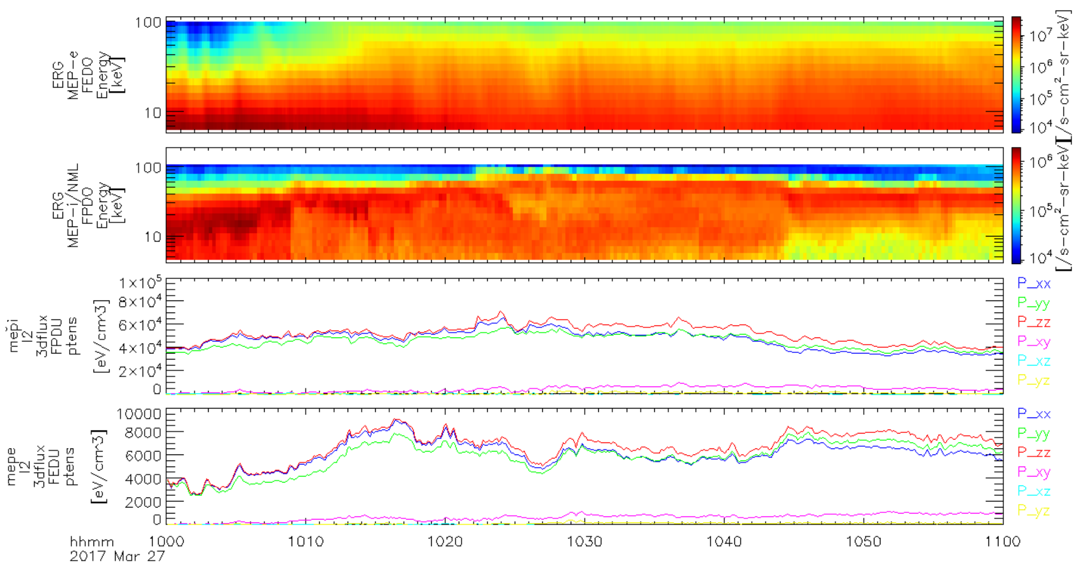
- density: number density
- avgtemp: scalar temperature (!)
- velocity: bulk velocity
- vthermal: thermal velocity
- mtens: momentum flux density tensor
- ptens: pressure tensor
- t3: temperature tensor (!)
- magt3: perpendicular/parallel temperature (!)
- flux: number flux
- eflux: energy flux

All vector and tensor quantities in DSI coordinates.

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

Moment calculation (2)

```
ylim, '*FPDU*ptens', 0, 1e+5, 0   ;; set y-ranges with linear scale
ylim, '*FEDU*ptens', 0, 1e+4, 0
tplot, ['erg_mep?_l2_3dflux_'+['F?DO','F?DU_ptens']]
```



Particle flux and pressure variations discussed by Hori+, GRL, 2018

(Memo)

Appendix:
Internal routines of part_products

An alternative way to deduce pitch-angle spectra rather manually

3-D data structure common to particle data that SPEDAS can handle

```
ERG> help, dist[100]
   PROJECT_NAME     STRING     'ERG'
   SPACECRAFT       LONG                  1
   DATA_NAME        STRING     'LEP-i Proton 3dflux'
   UNITS_NAME       STRING     'flux'
   UNITS_PROCEDURE  STRING     'erg_convert_flux_units'
   SPECIES          STRING     'proton'
   VALID            BYTE          1
   CHARGE           FLOAT          1.00000
   MASS             FLOAT        0.0104535
   TIME             DOUBLE        1.4959304e+09
   END_TIME         DOUBLE        1.4959304e+09
   DATA             FLOAT      Array[30, 16, 8]
   BINS             FLOAT      Array[30, 16, 8]
   ENERGY           FLOAT      Array[30, 16, 8]
   DENERGY          FLOAT      Array[30, 16, 8]
   NENERGY          LONG                 30
   NBINS            LONG                128
   PHI              FLOAT      Array[30, 16, 8]
   DPHI             FLOAT      Array[30, 16, 8]
   THETA            FLOAT      Array[30, 16, 8]
   DTHETA           FLOAT      Array[30, 16, 8]
ERG>
```

An example for LEP-i 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 30 ene. ch x 16 spin sector x 8 sensors.

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.

erg_???_get_dist(): Put 3-D flux data in a 3-D data structure

```
timespan, '2017-05-28'
erg_load_lepi_nml, datatype='3dflux', varformat='FPDU'

dist = erg_lepi_get_dist( 'erg_lepi_l2_3dflux_FPDU', /structure )

help, dist
help, dist[0]
```

Each get_dist() should be used for the 3-D flux data of each instrument, by providing a tplot variable of 3-D flux data as the argument.

- For LEP-i: `erg_lepi_get_dist()`
- For LEP-e: `erg_lepe_get_dist()`
- For MEP-e: `erg_mepe_gest_dist()`
- For MEP-i: `erg_mepi_get_dist()`
- For HEP: `erg_hep_get_dist()`

erg_pgs_make_fac, spd_pgs_do_fac: Transformation to the field-aligned coordinates (FAC)

```
;; Prepare the MGF and orbit data
erg_load_mgf & set_erg_var_label
;; Make transformation matrices for the FAC
erg_pgs_make_fac, dist.time, 'erg_mgf_l2_mag_8sec_dsi', 'erg_orb_l2_pos_gse', $
                  fac_output=fac_mat, fac_type='mphism'

dist_fac = dist ;; Make a copy of the dist structure

;; Transform phi/theta values in the dist structure to those in FAC for each time frame
for i = 0L, n_elements(dist.time)-1 do begin
  spd_pgs_do_fac, dist[i], reform( fac_mat[i, *, *], [3,3] ), $
                  output=dist_tmp,   error=error
  dist_fac[i] = dist_tmp
endfor
```

Note that:

- Both the magnetic field data in DSI and the orbit data in GSE should be given to `erg_pgs_make_fac`. They are automatically interpolated in time to match time frames of particle data given as a 1-D array in SPEDAS time unit (`dist.time` in the above case).
- The transformation matrix is made for the particle time frames, as a 3-D array of time x 3 x 3 (`fac_mat` in the above case).
- `spd_pgs_do_fac` changes only phi and theta arrays in a particle data structure.

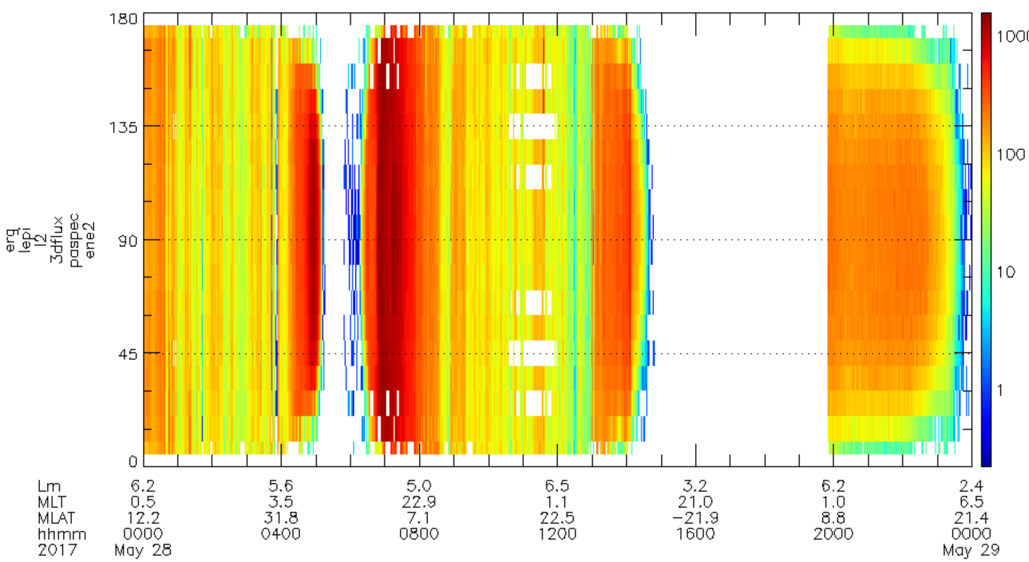
Binning and averaging flux data in FAC to deduce pitch-angle spectra

```
dist_fac.theta = 90. - dist_fac.theta  ;; colat. in FAC = pitch angle

;; Prepare data arrays for a selected energy channel
enech = 2 ;; ch02 --> 19.2 keV
dat_arr = reform( dist_fac.data[ enech, *, * ] )
pa_arr = reform( dist_fac.theta[ enech, *, * ] )
ntimes = n_elements(dist_fac.time)
dim = dimen( dist_fac[0].data )
t_arr =  rebin( reform( dist_fac.time, [1,1,ntimes] ), [dim[1:*],ntimes] )

;; Use a generic routine "bin2d" to calculate average fluxes for the time x pitch-angle bins
id = where( finite(dat_arr) and finite(pa_arr) ) ;;To exclude NaN and Inf from the averaging with bin2d
bin2d, t_arr[id], pa_arr[id], dat_arr[id], xrange=minmax(t_arr), yrange=[0.,180.], binum=[ntimes,18], $
       xc=time_c, yc=pa_c, ave=aveflux, binhist=binnm, /double
```

- `Bin2d` calculates an average flux for each time x pitch-angle bin. No smoothing or interpolation is applied, unlike how data are averaged by part_products.
- The original version of `bin2d.pro` does not accept /double keyword. Please download and **use `bin2d.pro` and `bin1d.pro` of the SPEDAS-j tools**, which are available from the SPEDAS-j website at: `https://github.com/spedas-j/member_contrib/tree/master/misc/bin12d`

Binning and averaging flux data in FAC to deduce pitch-angle spectra (cont'd)

```
;; Put the resultant arrays in a tplot varirable
vname = 'erg_lepi_paspec_ene02'
store_data, vname, data={ x:time_c, y:aveflux, v:pa_c }
;; Set some plot properties
options, vname, spec=1, constant=[45,90,135], ytickinterval=45., yminor=3
ylim, vname, 0, 180, 0
zlim, vname, 0, 0, 1 ;; auto-scale in log

;; Plot!
tplot, vname
```



One of the pitch-angle spectra shown in Asamura+, EPS, 2018 is reproduced!

Definition of the FA coordinate system"s" used by part_products

- In the field-aligned (FA) coordinate systems, the Z-axis is always in the local magnetic field direction. An X-axis or Y-axis should be defined separately to form a right-handed system. The following options for the Y-axis are available in the part_products library, which is usually given by keyword "`fac_type`" to the `erg_pgs_make_fac` routine.
- `'xgse'`
  - Y-axis: the vector product of Z-axis and the Xgse direction ($e_y = e_z \times e_{x\_gse}$)
  - X-axis: $e_y \times e_z$
- `'(m)phigeo'`
  - Y-axis: the vector product of Z-axis and the phi direction (roughly eastward) in the geographical (GEO) coordinate system at a satellite location. `mphigeo` uses the negative phi direction (roughly westward) instead.
  - X-axis: $e_y \times e_z$ (roughly radially outward for `phigeo` and radially inward for `mphigeo`)
- `'(m)phism'`
  - Y-axis: the vector product of Z-axis and the phi direction (roughly eastward) in the solar-magnetic (SM) coordinate system at a satellite location. `mphism` uses the negative phi direction (roughly westward) instead.
  - X-axis: $e_y \times e_z$ (roughly radially outward for `phism` and radially inward for `mphism`)

Other sources about part_products

- (SPEDAS top)/idl/general/science/spd_part_products/
  - The directory in which common routines of the part_products library are placed.
- Brief description about the 3-D data structure format on SPEDAS wiki
  - http://spedas.org/wiki/index.php?title=3D_data_structures
- SPEDAS-j wiki?
  - https://github.com/spedas-j/member_contrib/wiki/top

    Nothing now, but some may appear in future…
- What else?