



# IDL/SPEDAS講習会 (Araseデータを題材に)

名古屋大学宇宙地球環境研究所  
太陽圏サイエンスセンター



- IDL/SPEDASを使ってあらせ衛星データやその関連データをどうやって読み込み・プロット・操作するか、を学ぶ。



- 今は実践的なIDL/SPEDAS講習をやる時間であって、メールチェックやウェブサーフィンの時間ではない!
- 講師やtutor、隣に座っている(かもしれない)経験者となるべくコミュニケーションして、有用な知識・情報を得るように心がけること。
- 時間の制限上、今日の講習会で、用意してあるスライドのページ全てをカバーすることはできない。あとで各自で一通りスライドに目を通しておくことを推奨する。
- 講習は初～中級者に合わせて比較的ゆっくり進めていく予定。自分でできる人は自分のペースでどんどん練習していくってもらって構わない。



# 0. インストール



IDL/SPEDASを使うためには、最低限、以下の2つのソフトをPC or サーバーにインストールする必要がある：

- **Interactive Data Language (IDL)** 商用ソフト
  - <https://www.nv5geospatialsoftware.co.jp/Software-Technology/IDL>
- **最新バージョンのSPEDAS**
  - [http://spedas.org/wiki/index.php?title=Downloads\\_and\\_Installation](http://spedas.org/wiki/index.php?title=Downloads_and_Installation) (インストールの仕方が書かれている)



あらせデータの読み込み・解析のSPEDAS向け追加パッケージ:

- ERG plug-in tool for SPEDAS
  - [https://ergsc.isee.nagoya-u.ac.jp/data\\_info/howto.shtml.en](https://ergsc.isee.nagoya-u.ac.jp/data_info/howto.shtml.en)

その他の追加パッケージ:

Tsyganenko磁気圏磁場モデルの計算や磁力線トレーシングをやりたいなら、さらに下記のDLMが必要:

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

SPICE toolkitsを使いたいなら必要（惑星データ解析では必須、あらせでは不要）

- Icy DLM
  - [https://naif.jpl.nasa.gov/naif/toolkit\\_IDL.html](https://naif.jpl.nasa.gov/naif/toolkit_IDL.html)

# SPEDASのためのコード(プログラム)をどうやって書くか?



```
IDLWorkspace - /Users/horit/work/IDLWorkspace/horiidllib/ace_plot.pro - IDL Workbench
ace_plot.pro x
FUNCTION get_ace_timestamp, swe_Vp, tr=tr
get_data, swe_Vp, data=d
IF NOT KEYWORD_SET(tr) THEN avev=average(d.y,/nan)
tr=time_double(tr)
idx=WHERE( d.x GE tr[0] AND d.x LE tr[1], cnt)
IF cnt GT 2 THEN avev=average(d.y[idx],/nan)
ELSE avev=average(d.y,/nan)
ENDIF

dprint, 'Ave. Vsw for time shift: ', STRING(avev,FORMAT='(F7.3)
ts = ( 220.0 * 6371.2 ) / avev / 60. ;[min]
IF ts GT 15 AND ts LT 100 THEN RETURN, ts
dprint, 'unexpected ACE ts value:', ts, !CUse ts=60 min
RETURN, 60.
ENDIF

;RETURN
END

;PRO ace_plot, tr, force_ts=force_ts, noplots
; Load the ACE data from the CDAWeb site cda.cda
; ...
```

コンソール x コマンド履歴 Debug Problems カレントディレクトリ: /Users/horit

```
% Compiled module: TIME_DOUBLE.
% Compiled module: TIME_PARSE.
% Compiled module: PTRACE.
ERG_INIT(141):
ERG(Arase) in orbit for 2822 Days, 11 Hours, 26 Minutes, 44 Seconds since launch
% Compiled module: TIME_STAMP.
% Loading the Turbo color table
% Compiled module: CWD.
CWD(25): Directory changed to: /Users/horit
#####
you're all set.
ok. 3, 2, 1, let's jam!
Licensed for use by: Nagoya University
License: 766952
Expiration date: 03-Oct-2024
ERG>
```

Writable Smart Insert 1:1:0

文法チェック、色分け、括弧閉じチェック、字下げ・改行などの  
フォーマット一括適用、など

拡張機能 Marketplace で拡張機能を検索... 未設定 (ワークスペース)

インストール済み 39

- IDL for VSCode 125ms IDL syntax highlighting, code sni...
- NV5 Geospatial Software NV5 Geospatial Software
- indent-rainbow 7ms Makes indentation easier to read oderwat
- IntelliCode AI-assisted development Microsoft
- IntelliCode API Usage Examples See relevant code examples from... Microsoft

```
ace_plot.pro 4 x
horiidllib > ace_plot.pro > ...
Tomo Hori @ISEE, 5年前 | 2 authors (Tomo Hori and one other)
FUNCTION get_ace_timestamp, swe_Vp, tr=tr
tr=time_double(tr)
idx=WHERE( d.x GE tr[0] AND d.x LE tr[1], cnt)
IF cnt GT 2 THEN avev=average(d.y[idx],/nan) ELSE $
    avev=average(d.y,/nan)
ENDIF

dprint, 'Ave. Vsw for time shift: ', STRING(avev,FORMAT='(F7.3)
ts = ( 220.0 * 6371.2 ) / avev / 60. ;[min]
IF ts GT 15 AND ts LT 100 THEN RETURN, ts
dprint, 'unexpected ACE ts value:', ts, !CUse ts=60 min
RETURN, 60.
ENDIF
```

- IDLワークベンチ(IDLをインストールすると入っている)か、VS code + IDL for VSCode拡張を使うのがおすすめ



# 1. SPEDASとは?

# Space Physics Environment Data Analysis System (SPEDAS)



各種オンラインデータ  
リポジトリ



THEMIS



Van Allen  
Probes



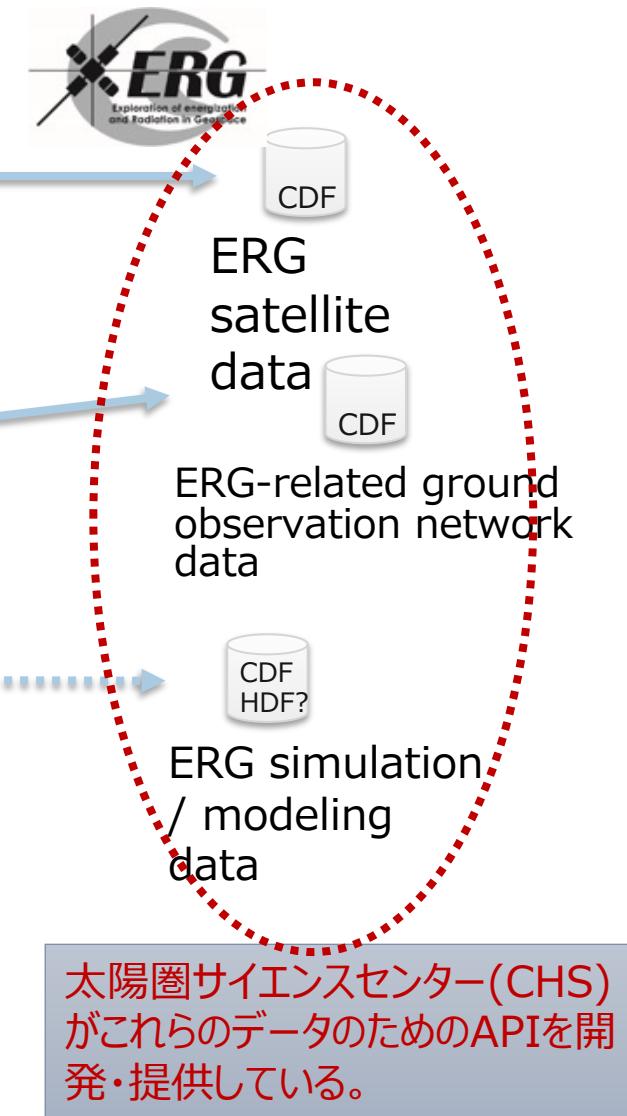
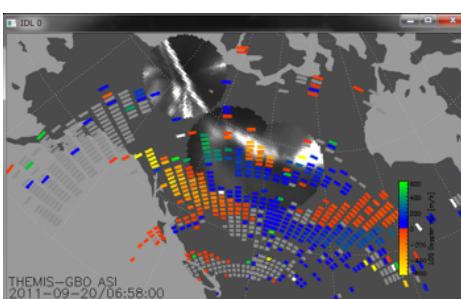
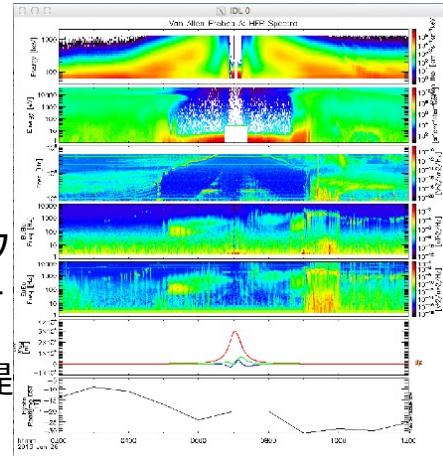
MMS



IUGONET

各衛星・観測プロジェクトが、データをダウンロード・読み込むためのAPIを開発してSPEDASに提供

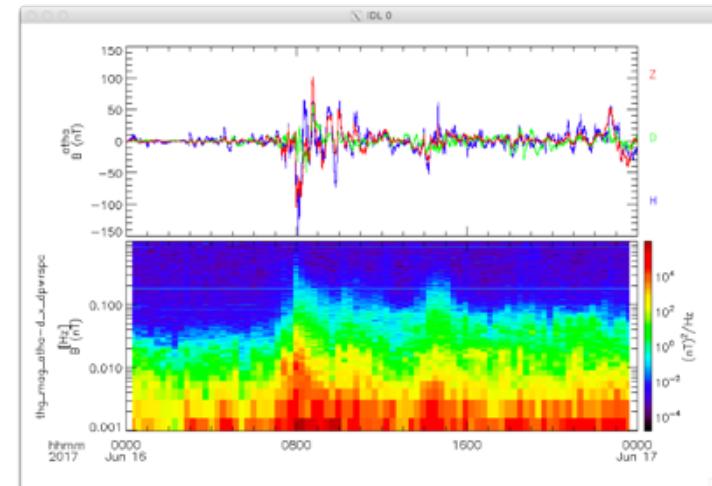
SPEDASツール



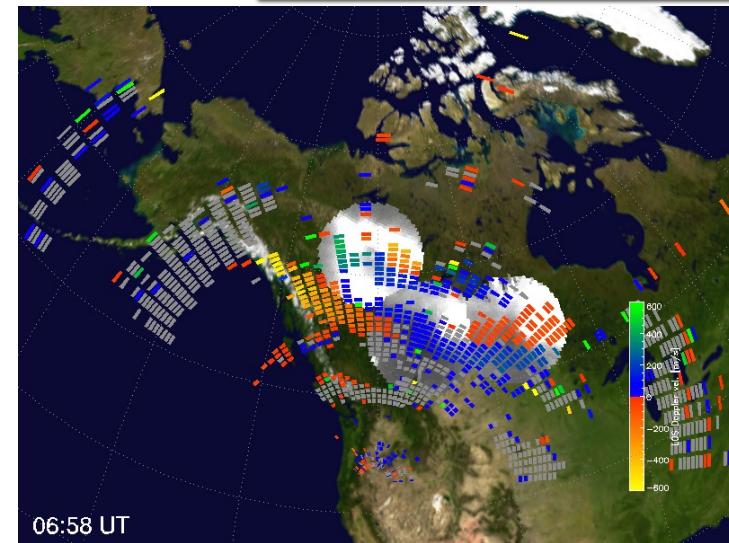
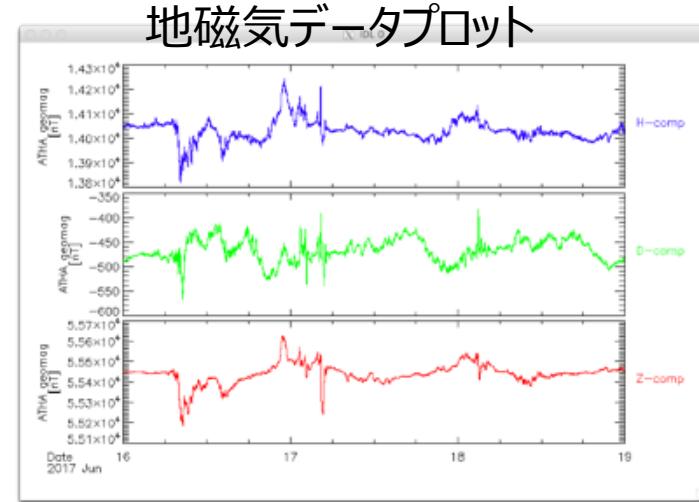
# SPEDASで何ができるか?



- ・あらせ衛星や他衛星・地上データの読込
- ・時系列データプロット
- ・データへの各種フィルターの適用
- ・周波数解析
- ・地上・電離圏へのマッピング
- ・...



地磁気データの周波数スペクトル



SuperDARN (大型電離圏短波レーダー)と全天カメラのデータを地図上に描画



# SPEDASの基礎： tplot と tplot変数

# SPEDASの使い方の前にIDLの基礎を少しだけ...



- コマンドと引数とキーワードの間にはカンマを入れる。

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

- 文字列は单・二重引用符で囲む ( ' ) or ( " )。

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

- 配列はカンマ区切りで角括弧で囲む。

```
IDL> arr1 = [ 2, 3, 4, 5 ]
```

```
IDL> string_arr1 = [ 'text1', 'text2', 'text3' ]
```

- 文字列と文字列配列を足すと、それぞれを接続した文字列配列になる：

```
IDL> strs = 'erg_mgf_' + [ 'bx', 'by', 'bz' ]
```

```
IDL> print, strs
```

```
erg_mgf_bx    erg_mgf_by    erg_mgf_bz
```

# SPEDASの使い方の前にIDLの基礎を少しだけ...(続き)



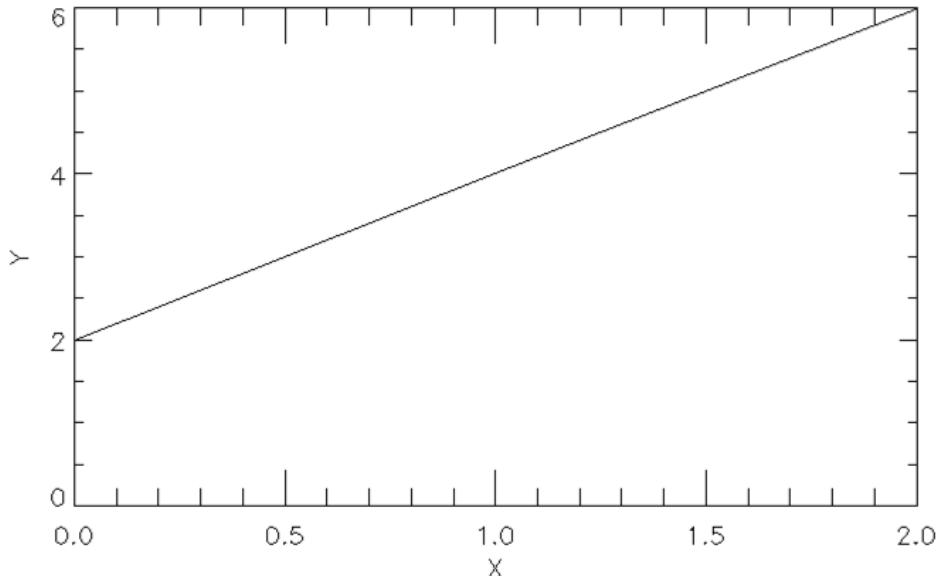
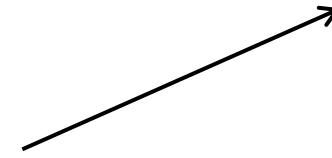
- 初心者がよく出会うIDL上のエラーメッセージ:

% Attempt to call undefined procedure: '????'.  
→多くの場合コマンドやルーチン名の綴りが間違っている。

% Syntax error.  
→ , ' ( ) [ ] の片方が脱落しているか、多すぎなことが多い。

- "plot" コマンド: 2-Dプロットの基本コマンド

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



- "append array" コマンド (SPEDASに収録):

```
IDL> arr = [1, 2, 3]
```

```
IDL> append_array, arr, [4, 5, 6] ;;; 既存の配列に、変数や配列を追加する便利コマンド
```

```
IDL> print, arr
```

```
1 2 3 4 5 6
```

- 矢印キーの↑を押すと直前のコマンドを再表示できる。

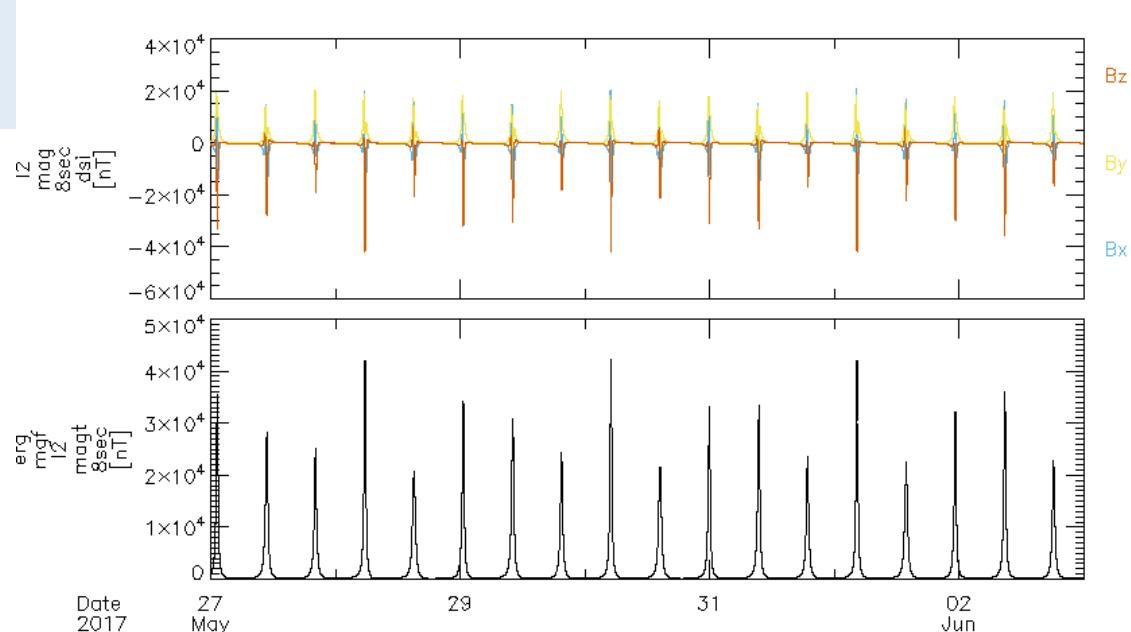
← →キー や バックスペースキーを使って編集して、異なるコマンドにして再実行可能

# SPEDASによるデータ処理の基本パターン



一番シンプルな点順は下記のような感じ

1. IDLを起動
2. SPEDAS環境の初期セットアップ
3. データの日付・時刻幅を設定
4. データを読み込む
5. (必要ならデータを加工)
6. データをプロット

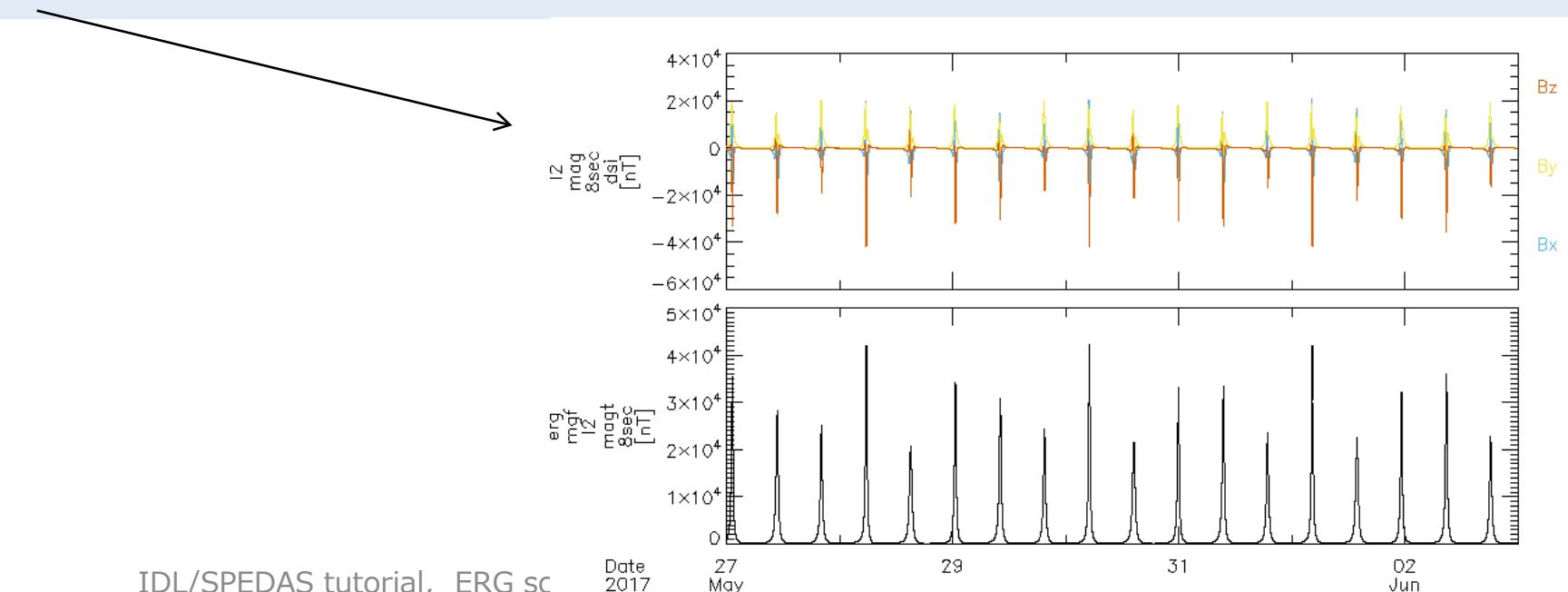


# SPEDASによるデータ処理の基本パターン (続き)



実際のコマンド例:

```
prompt> idl  
IDL> erg_init  
ERG> timespan, '2017-05-27', 7, /day  
ERG> erg_load_mgf  
(必要ならここでデータを加工)  
ERG> initct, 1080, line_clrs=7 ;; 色設定、必ずしもやらなくてもよい  
ERG> tplot, ['erg_mgf_l2_mag_8sec_dsi', 'erg_mgf_l2_magt_8sec']
```





ERG> **timespan**, timestr, N, option

timestr : 設定したい日時幅の開始時刻(UTC)を

'yyyy-mm-dd/hh:mm:ss'形式の文字列で指定

N : 長さを表す数字 (Default: 1)

option : ↑の長さの単位 (/day, /hour, /min, /sec, Default: /day)

例えば2017-05-27/00:00:00 UTCから1日分の範囲なら

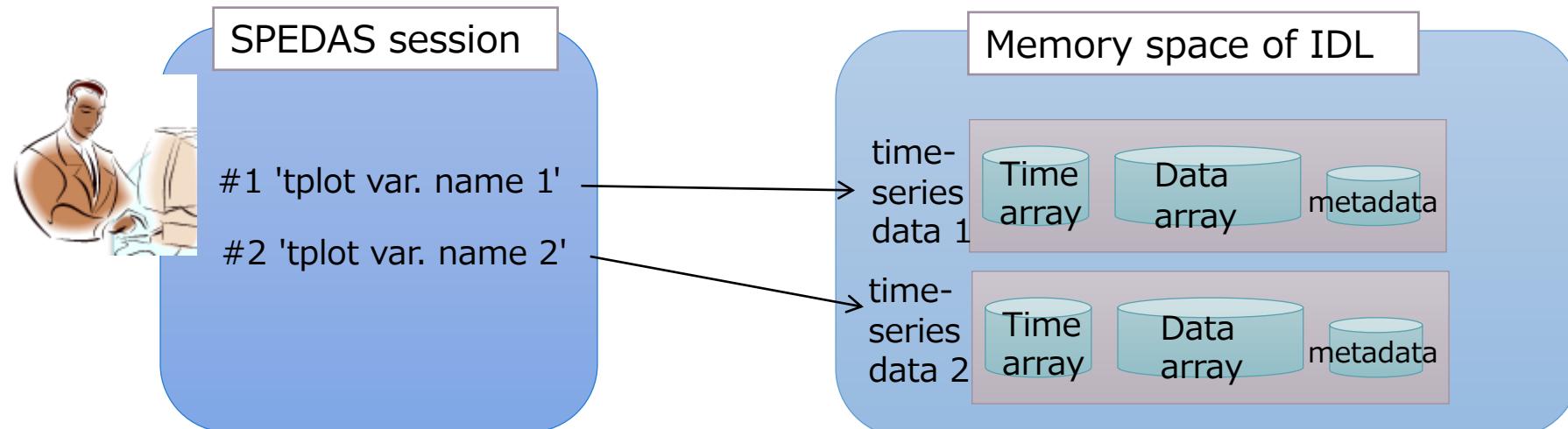
ERG> **timespan**, '2017-05-27'

2017-05-29/03:25:30 UTCから90分の範囲:

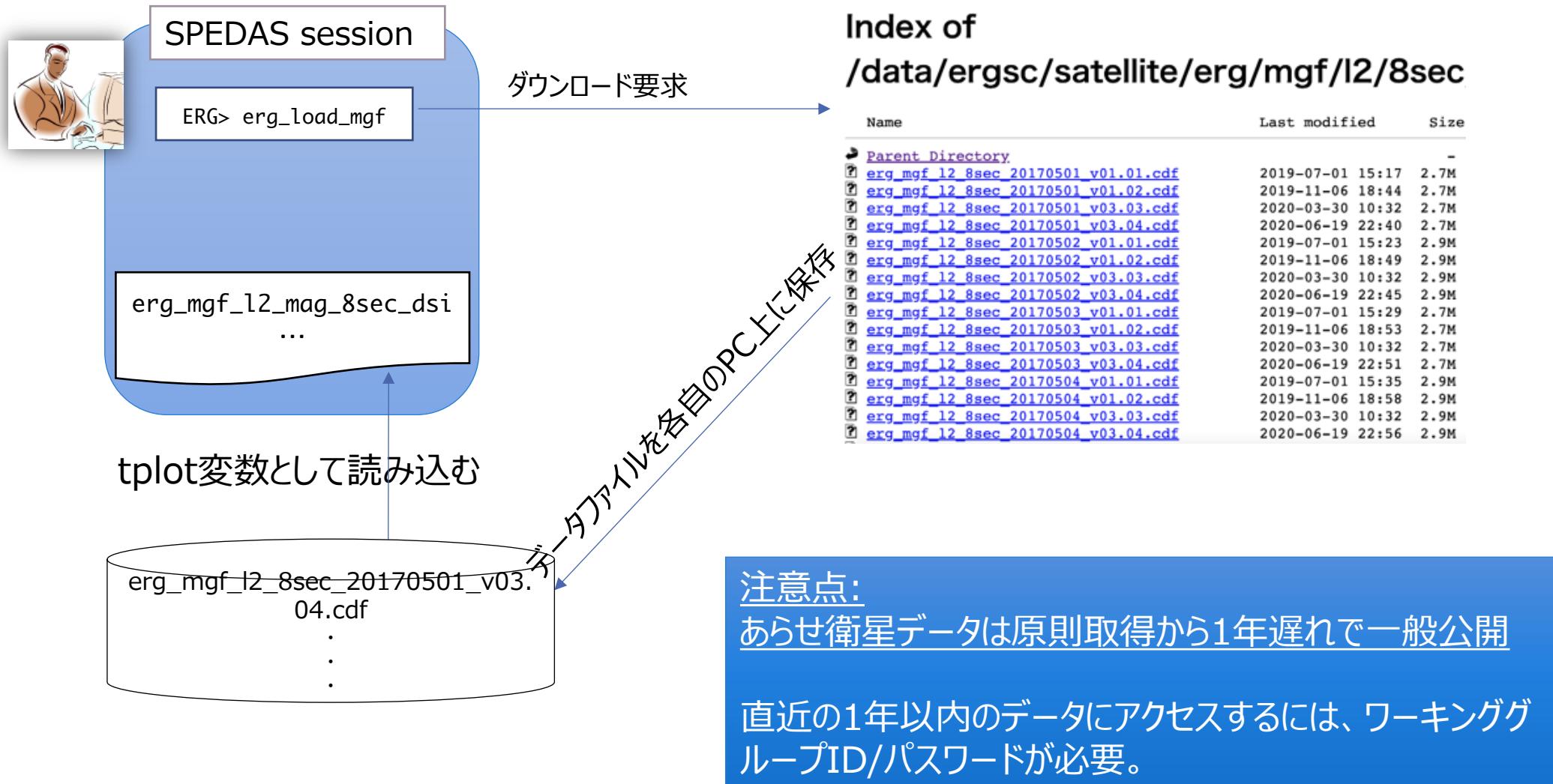
ERG> **timespan**, '2017-05-29/03:25:30', 90, /min



- 前ページの'erg\_mgf\_l2\_mag\_8sec\_dsi'のようなものは**tplot変数 (tplot variable)**と呼ばれる。
- "Tplot変数"は、**通し番号付きのtplot変数名(文字列)**と、実際の**メタデータ付き時系列データの構造体**を結びつけるものである。



# 例：MGFデータ読込ルーチンが実際には何をしているか



# tplot変数の中のデータ構造体にアクセスする

## - get\_data -



```
ERG> get_data, 'thg_mag_atha', data=data, dlimits=dlimits, lim=lim
```

data: これにtplot変数のデータ構造体が入る  
dlimits: ほとんどのメタデータが構造体の形で入る  
lim: プロット属性の情報が入る

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

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

"**get\_data**"によってtplot変数の中のデータを吸い出してIDL上の構造体(↑の例では"data")として保存できる。それにより、データの中身を普通の配列 "**data.x**" or "**data.y**" としてアクセスできるようになる。

```
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]
```

元データに付随していたメタデータやプロット属性の情報はキーワードで指定された "dlimits", "lim" という変数に構造体として格納される。

# 新しいtplot変数を作る – store\_data



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

varname: 新しく作るtplot変数の名前

timearr: 1-D array 時刻データの1次元配列 (SPEDAS時刻形式で与える)

datarr: 時系列データのデータ値。1次元配列でも多次元配列でもOK。ただし1つ目の次元の要素数は、時刻データであるtimearrの要素数と一致している必要がある。

- SPEDAS時刻形式は倍精度浮動小数点数で表したUNIX timeのこと。UNIX timeとは、1970年1月1日0時0分0秒 UTC からの経過時刻を秒で表したもの。
- 通常は `time_double()` という関数を使って'2017-06-16/12:30:00'のような文字列形式の時刻表記からSPEDAS時刻を計算する。
- SPEDAS時刻値は`time_string()`という関数を用いることで用意に文字列形式の日時に変換可能。

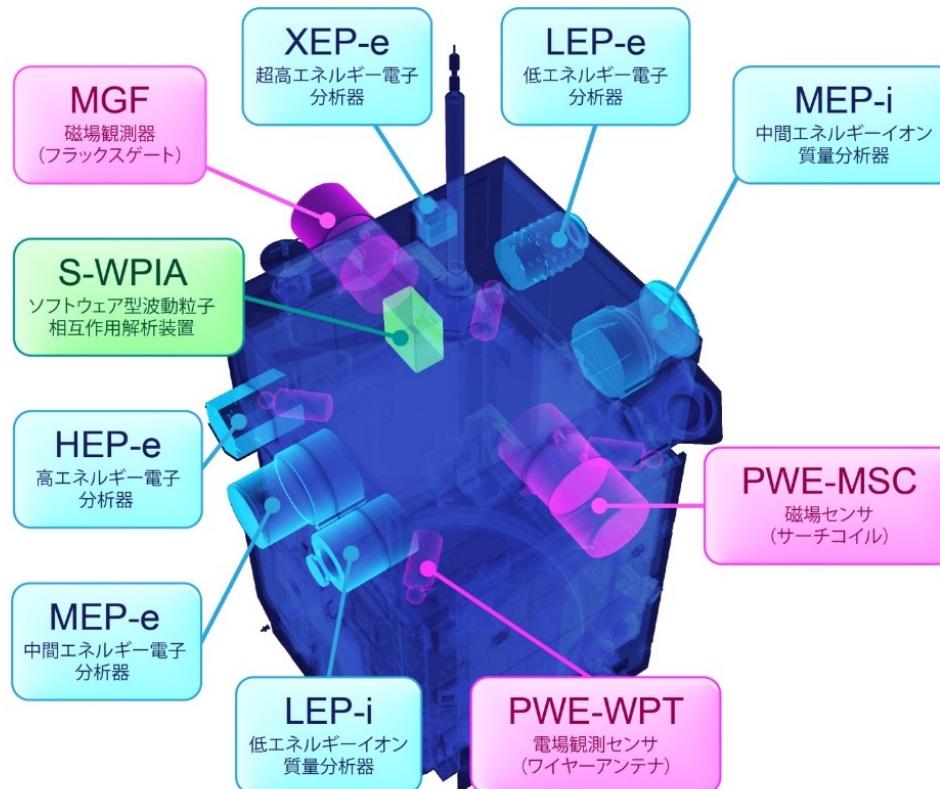
```
ERG> timestr='2017-06-16/12:30:00'
ERG> spedastime = time_double( timestr )
ERG> print, spedastime
1.4976162e+09
ERG> print, time_string( spedastime )
2017-06-16/12:30:00
```

SPEDASでの時刻値の取り扱いの詳細については  
[https://spedas.org/wiki/index.php?title=Time\\_handling](https://spedas.org/wiki/index.php?title=Time_handling)  
を参照のこと(ただし英語で記述されている)。

# Arase衛星の科学データプロダクト



- ERGプロジェクト全体についての論文
  - Miyoshi et al., EPS, 2018a
- ERGサイエンスセンターについての論文
  - 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

# tplot変数のリストを表示 & tplot変数の中身を見る



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>
```

通し番号付きでロード済みの  
tplot変数のリストが表示され  
る

```
[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.
```

tplot変数 'erg\_mgf\_l2\_mag\_8sec\_dsi' に紐づけられている実際のデータ構  
造体の情報が表示される。

X: 時刻配列 (SPEDAS時刻形式)、ここでは 75685要素の1次元配列

Y: 実際の時系列データの配列、ここでは 75685 × 3 要素の2次元配列

# tplot変数のリストを表示 & tplot変数の中身を見る

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: TPLLOT_NAMES.
1 erg_mgf_l2_mag_8sec_dsi
  DQ = STRUCT = TPLLOT_QUANT --(7 Tags/64 Bytes)-->
    NAME = STRING = 'erg_mgf_l2_mag_8sec_dsi'
    DH = POINTER = <PtrHeapVar65>
    *DHO = *PtrHeapVar65x = STRUCT = --(4 Tags/16 Bytes)-->
      X = POINTER = <PtrHeapVar344>
      *XO = *PtrHeapVar344x = DOUBLE[75685] = [1.4958432e+09, 1.4958432e+09, 1.4958432e+09, ...]
      X_IND = LONG = 75685
      Y = POINTER = <PtrHeapVar345>
      *YO = *PtrHeapVar345x = DOUBLE[75685,3] = [-495.70899, -498.25568, -502.07184, -505.67904, ...]
      Y_IND = LONG = 75685
    LH = POINTER = <PtrHeapVar66>
    *LHO = *PtrHeapVar66x = STRUCT = --(3 Tags/56 Bytes)-->
      LABELS = STRING[3] = ['Bx', 'By', 'Bz']
      COLORS = INT[3] = [2, 4, 6]
      LABFLAG = INT = 1
    DL = POINTER = <PtrHeapVar67>
    *(DL) = *PtrHeapVar67x = STRUCT = --(4 Tags/1528 Bytes)-->
      CDF = STRUCT = --(4 Tags/1504 Bytes)-->
        FILENAME = STRING = '/Volumes/HDD-LCU3/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 = 'ARASE(ERG)>Inner Magnetosphere'
          DATA_TYPE = STRING = '12_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 = '20180519'
          MDS = STRING = 'Created 05/2018'
          ADID_REF = STRING = ''
          LOGICAL_FILE_ID = STRING = 'erg_mgf_l2_8sec'
          LOGICAL_SOURCE = STRING = ''
          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] = [...] = [
            'For more information, see',
            'the ERG Science Center website',
            'https://ergsc.isee.nagoya-u.ac.jp',
            '8 s',
            '33P08573',
            '33F40050',
            '20170527 0000000000000000',
            '20170527 2359999999999999',
            '8 s average/start',
            'erg_mgf_l2_2017052700_v001.txt'
          ]
          STRING[16] = [...] = ...
          STRING = 'satellite/erg/mgf/makecdf_erg_mgf_l2_8sec.pro 1317'
          STRING[2] = [...] = ...
          STRING = ''
        ]
      ]
    ]
  ]

```

メタデータ (科学データに関する各種情報) が表示される

RULES\_OF\_USE という要素に "rules of the road" (データを使う・発表する際に守るべきルール) が入っている。

# Rules of the road (データポリシー)



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

- データを解析等に使う人は全員このルールを守らないといけない。
- 変なデータを見つけた場合は機器PIチームに報告することが推奨されている (erg\_xxx\_info @isee.nagoya-u.ac.jp)。



# カラーテーブルを使うまでのtips

カラーテーブルを定義するためのコマンドが主に2つある:

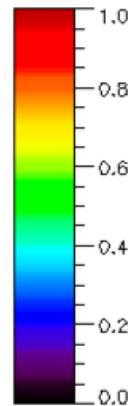
- `loadct_sd`
- `initct`

`loadct_sd, num (num: カラーーテーブルの番号)`

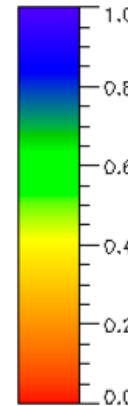
キーワード:

`center_hatched, hatched_width, hatched_color`

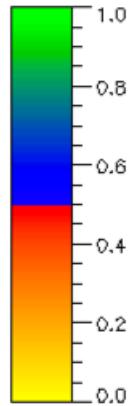
FAST-special  
(default)



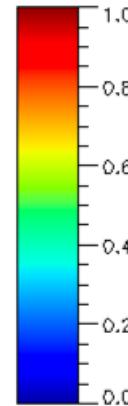
SD Cutlass



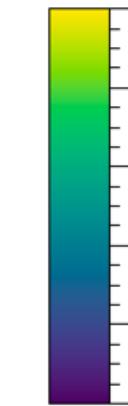
SD APL



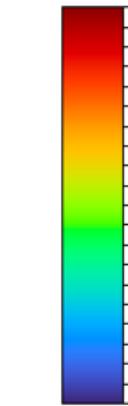
JET



Viridis



Turbo



`loadct_sd, 42 or 43`   `loadct_sd, 44`   `loadct_sd, 45`   `loadct_sd, 46`   `loadct_sd, 47`   `loadct_sd, 48`



色弱・盲の人にも見やすい

`loadct_sd` はテーブル  
No. 43–48 を先のテーブ  
ルに上書きする。

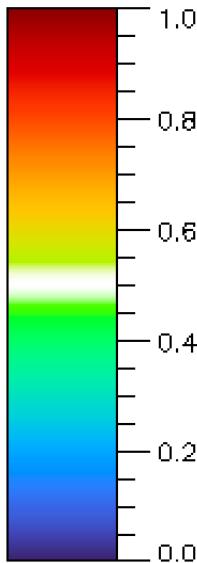
# カラーテーブル設定 by loadct\_sd



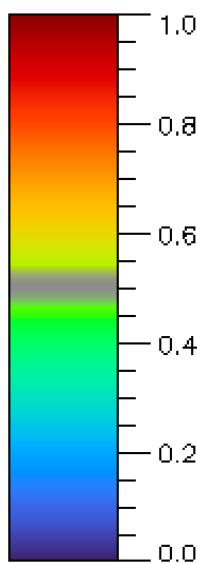
```
ERG> loadct_sd, 48, /center_hatched
```

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

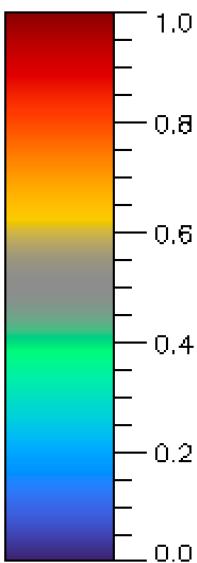
```
ERG> loadct_sd, 48, /center_hatched, hatched_color=5, hatched_width=60
```



/center\_hatched はテーブルの真ん中に「白色」を挿入する。



hatched\_color は挿入される白色を別の色に置き換える。

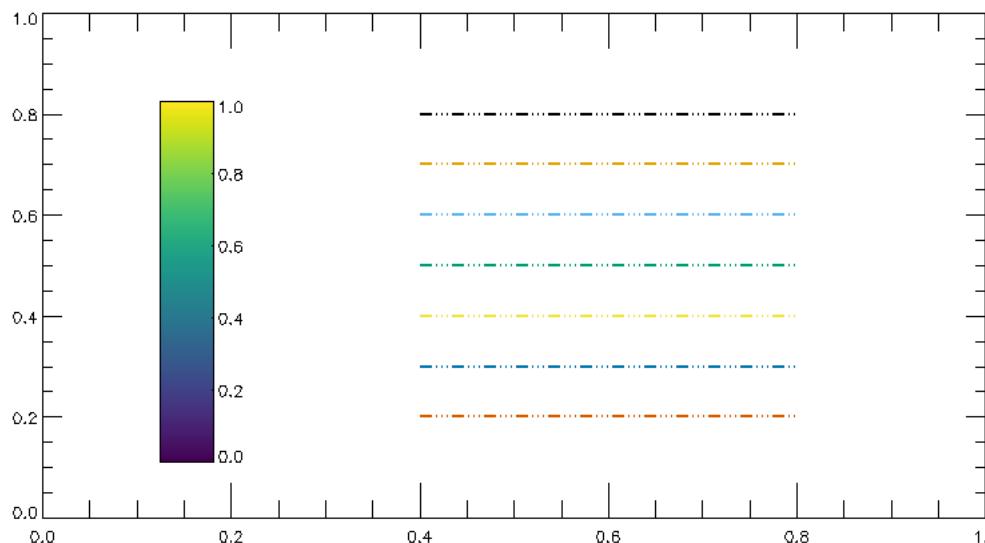


hatched\_width は真ん中に挿入される色の幅を指定する (default: hatched\_width = 20)。

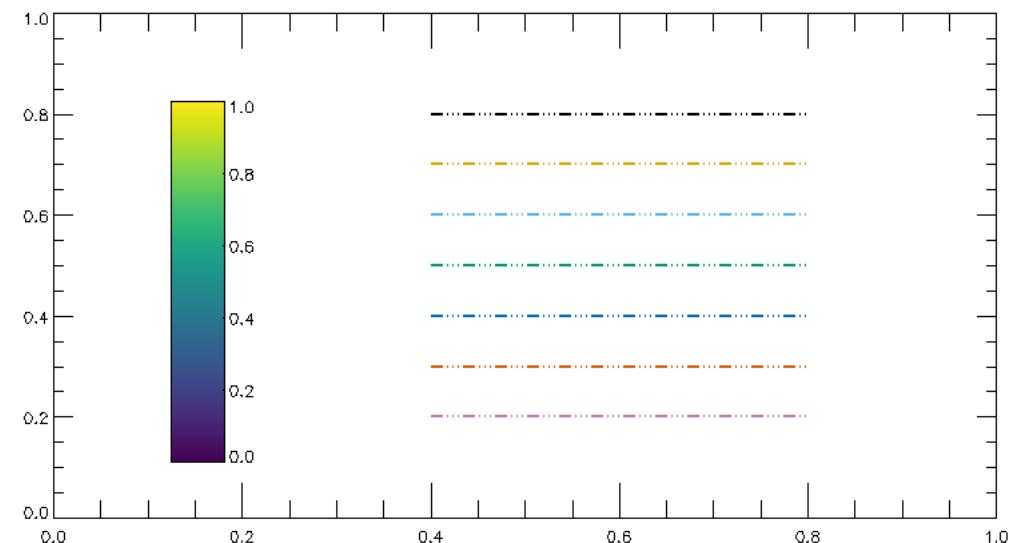
# より標準的なカラーテーブル設定方法 by initct



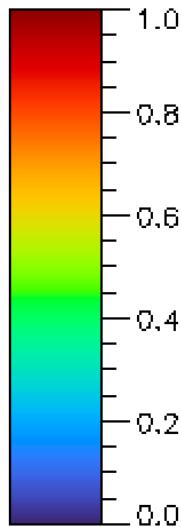
```
ERG> window  
ERG> initct, 1080 ;; 1000-1074はloadctの0-74と同じ(ただし41-43を除く)。1075-1118はinitctのオリジナルテーブル。  
  
ERG> initct, 1080, line_clrs=7  
ERG> plot, [0],[0] & draw_color_scale, rang=[0,1], pos=[0.2,0.2,0.25,0.8]  
ERG> for i=0, 6 do plots, [0.4,0.8],[0.8,0.8]-0.1*i, color=i, linestyle=4, thick=2  
  
ERG> initct, 1080, line_clrs=8  
ERG> plot, [0],[0] & draw_color_scale, rang=[0,1], pos=[0.2,0.2,0.25,0.8]  
ERG> for i=0, 6 do plots, [0.4,0.8],[0.8,0.8]-0.1*i, color=i, linestyle=4, thick=2
```



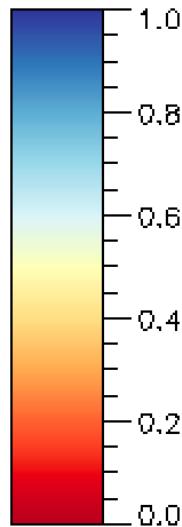
これらの7色は色弱・盲の人でもある程度区別できるように最適化されている [Wong, Nature Methods, 2011]。



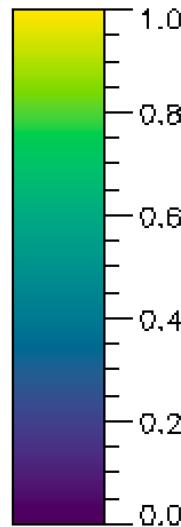
下記の4つのカラーテーブルを使うことにする。



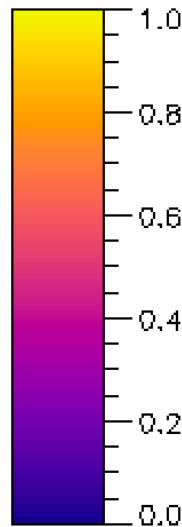
loadct\_sd, 48  
*Turbo*



initct, 1072  
*CB-RdYlBu*



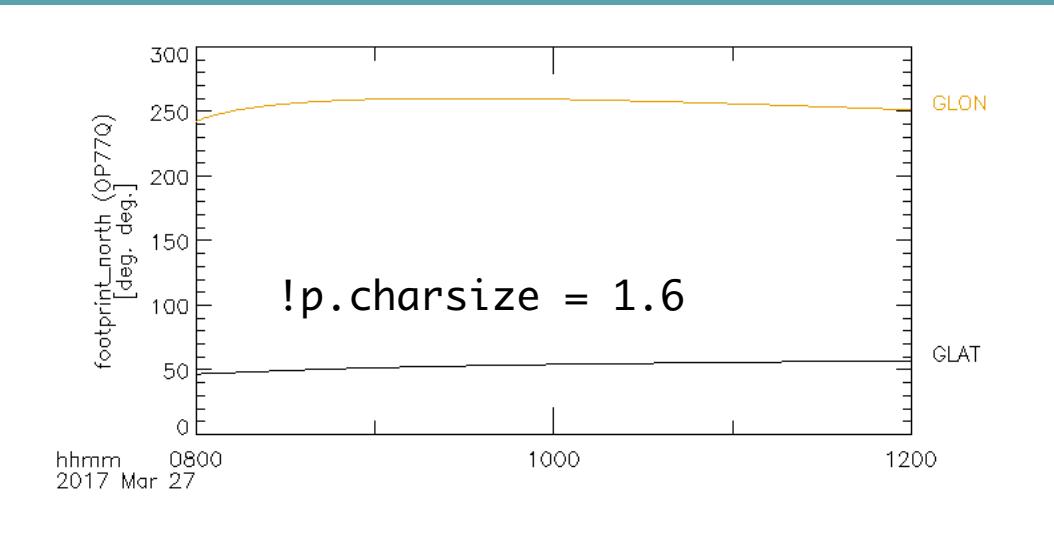
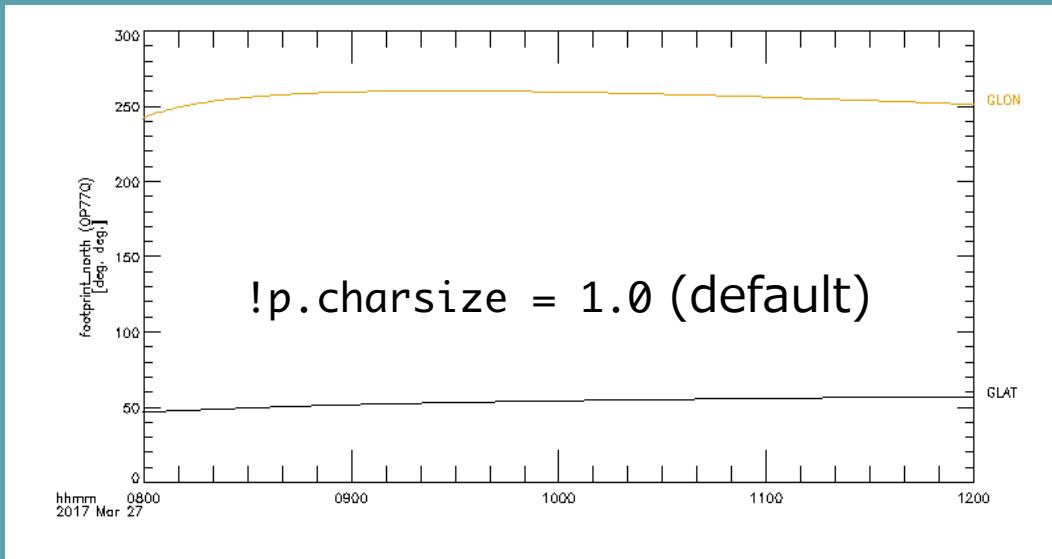
initct, 1080  
*Viridis*



initct, 1079  
*Plasma*

- カラーテーブルの詳細なリストと、あとPythonのmatplotlibのカラーテーブルを取り込む方法:
  - IDL memo by 京大花山天文台 西田さん  
<https://www.kwasan.kyoto-u.ac.jp/~nishida/idl/defaultcolortable.html>

しかしカラーテーブルだけでなく実際のスライドでは 文字の大きさ にも注意を！





# プロットのパネルを操作したり装飾したりする方法

# tplot変数のプロットを作る: *tplot*



tplot変数名を使ってプロット (文字列、文字列配列を与える)

```
ERG> tplot, 'erg_mgf_l2_mag_8sec_dsi'
```

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

tplot変数の通し番号を使ってプロット (整数、整数配列を与える)

```
ERG> tplot, 1
```

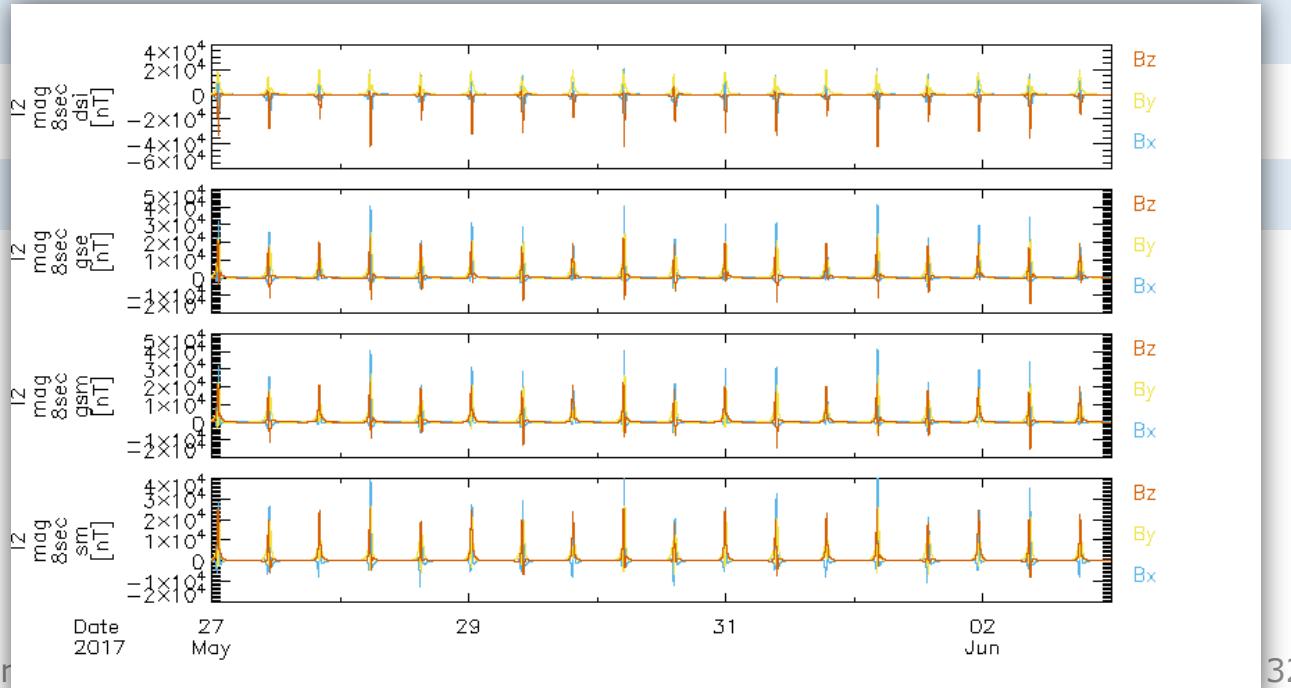
通り番号の配列を与えることで複数のtplot変数を1つのプロットウィンドウに並べてプロット

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

ワイルドカード("\*", "?")も使える

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

**tplot**コマンドは様々な形式の引数をとってプロットできる。



# tplot用の便利コマンド: tplot\_remove\_panel, tnames\_cp()



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

```
ERG> tplot_names, /current
```

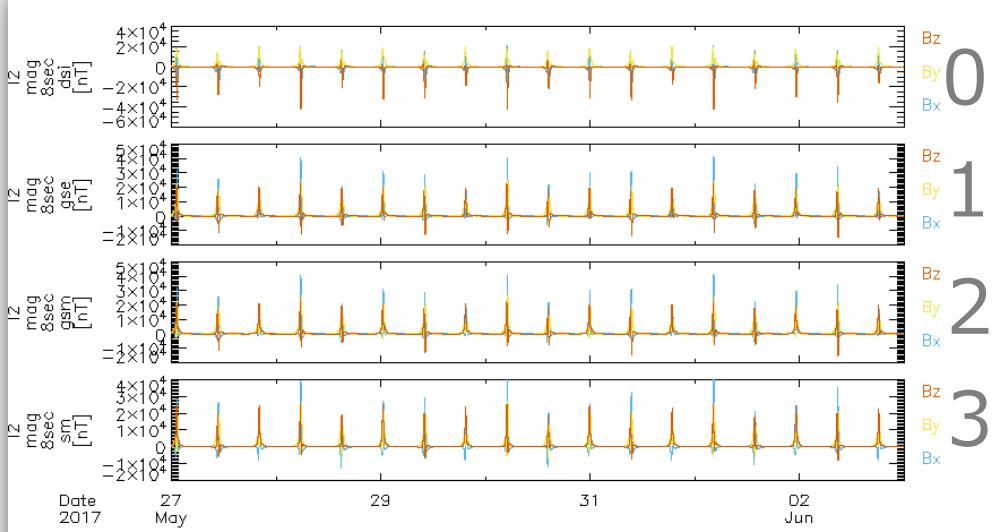
```
1 erg_mgf_l2_mag_8sec_dsi  
2 erg_mgf_l2_mag_8sec_gse  
3 erg_mgf_l2_mag_8sec_gsm  
4 erg_mgf_l2_mag_8sec_sm
```

```
ERG> tnames_cp( /tplot )
```

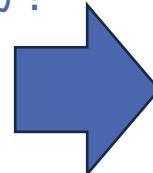
```
['erg_mgf_l2_mag_8sec_dsi', 'erg_mgf_l2_mag_8sec_gse', 'erg_mgf_l2_mag_8sec_gsm', 'erg_mgf_l2_mag_8sec_sm']
```

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

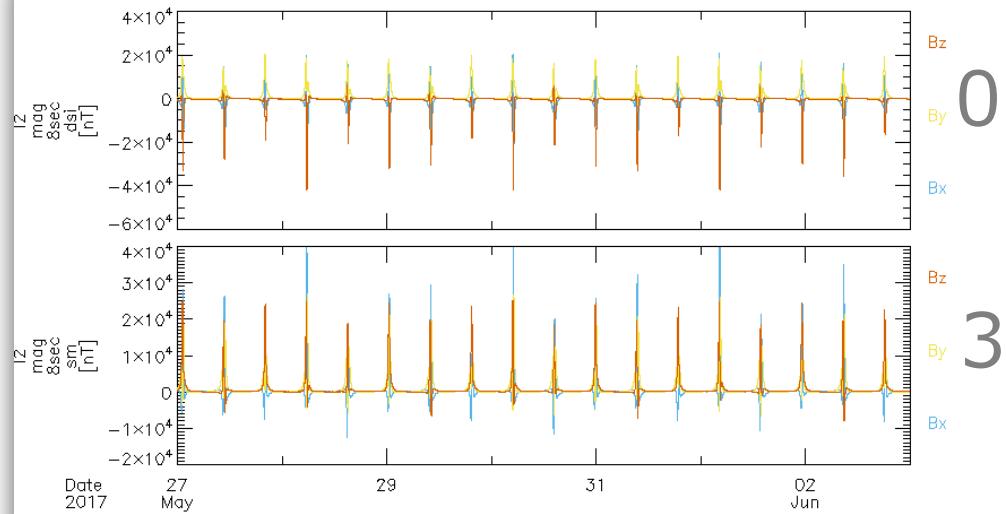
tplotにそのまま与える形で直前にプロットされた変数が表示されるので、コピー&貼付しやすい。



右のプロット  
をどう作る  
か?



または tplot\_remove\_panel も使える



# 各tplot変数のプロットパネルを装飾する

**options, varname, option1='...', option2='...', ...**

varname: tplot変数名 (配列でもワイルドカード入りでも)

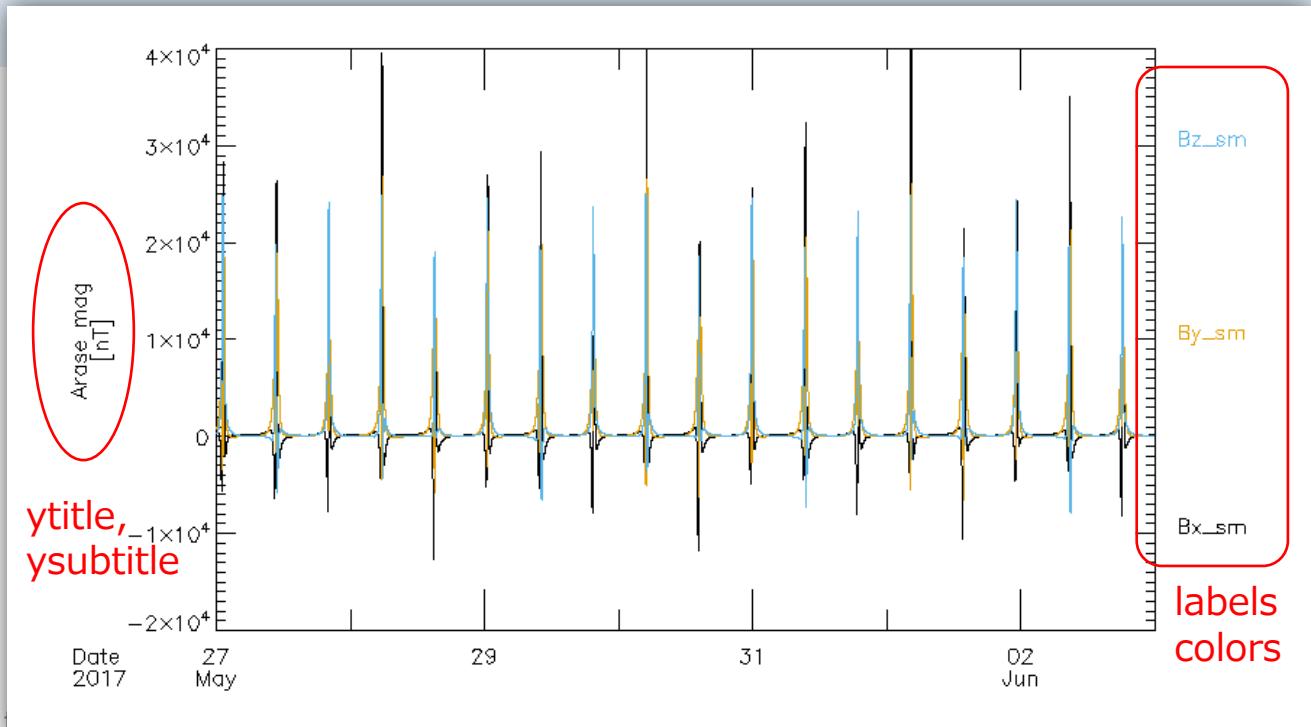
option?: プロット属性名

```
ERG> options, 'erg_mgf_l2_mag_8sec_*', ytitle= 'Arase mag' , ysubtitle='[nT]'
```

```
ERG> options, 'erg_mgf_l2_mag_8sec_sm', labels=['Bx_sm', 'By_sm', 'Bz_sm' ]
```

```
ERG> options, 'erg_mgf_l2_mag_8sec_sm', colors=[ 0, 1, 2 ]
```

```
ERG> tplot, 'erg_mgf_l2_mag_8sec_sm'
```



# プロットの時間幅を変更する



プロットウィンドウ上をマウスクリックすることで時間幅を指定する

```
ERG> tlimit
```

時間幅を明示的に与える

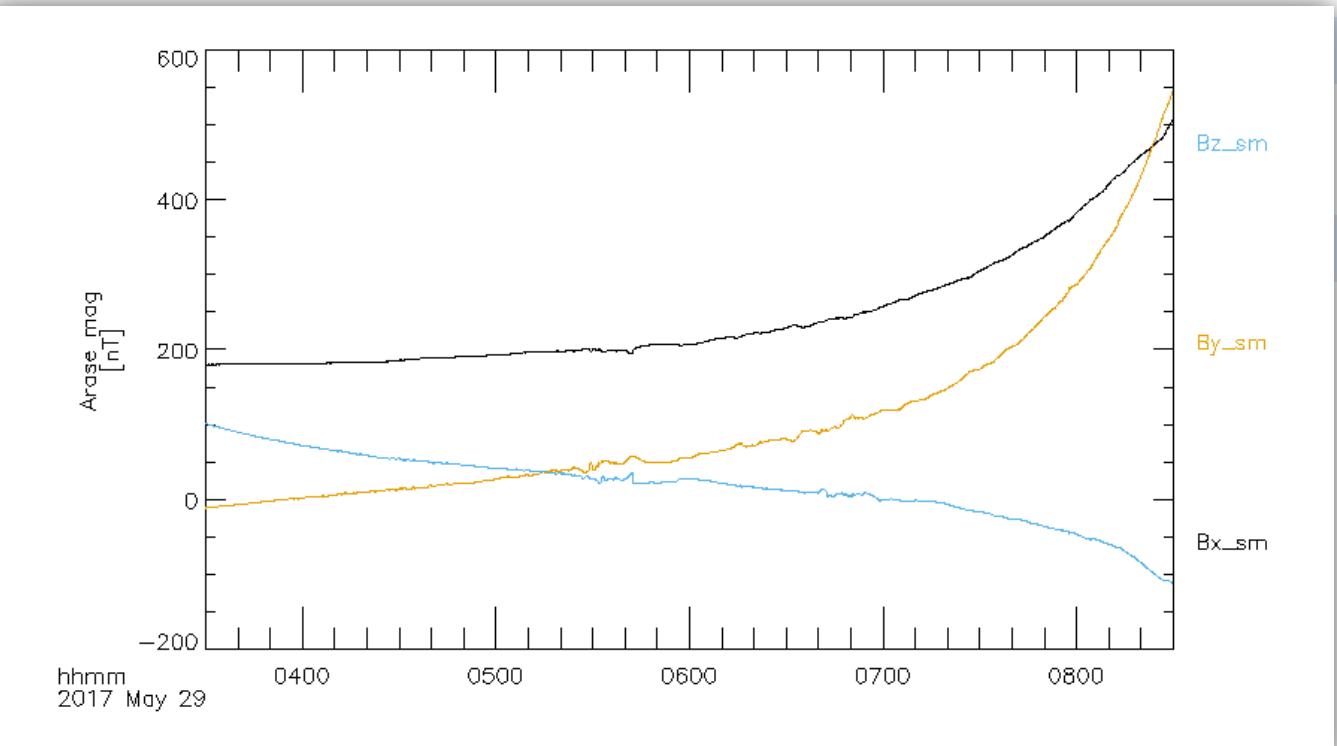
```
ERG> tlimit, '2017-05-29/03:30' , '2017-05-29/08:30'
```

直前にプロットした時間幅に戻して再プロット

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

timespan で最初に指定した  
オリジナルの時間幅に戻す

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



# ベクトル値入りのtplot変数を成分ごとに分離



```
ERG> split_vec, 'erg_mgf_12_mag_8sec_sm'
```

```
STORE_DATA(260): Creating tplot variable: 19 erg_mgf_12_mag_8sec_sm_x
```

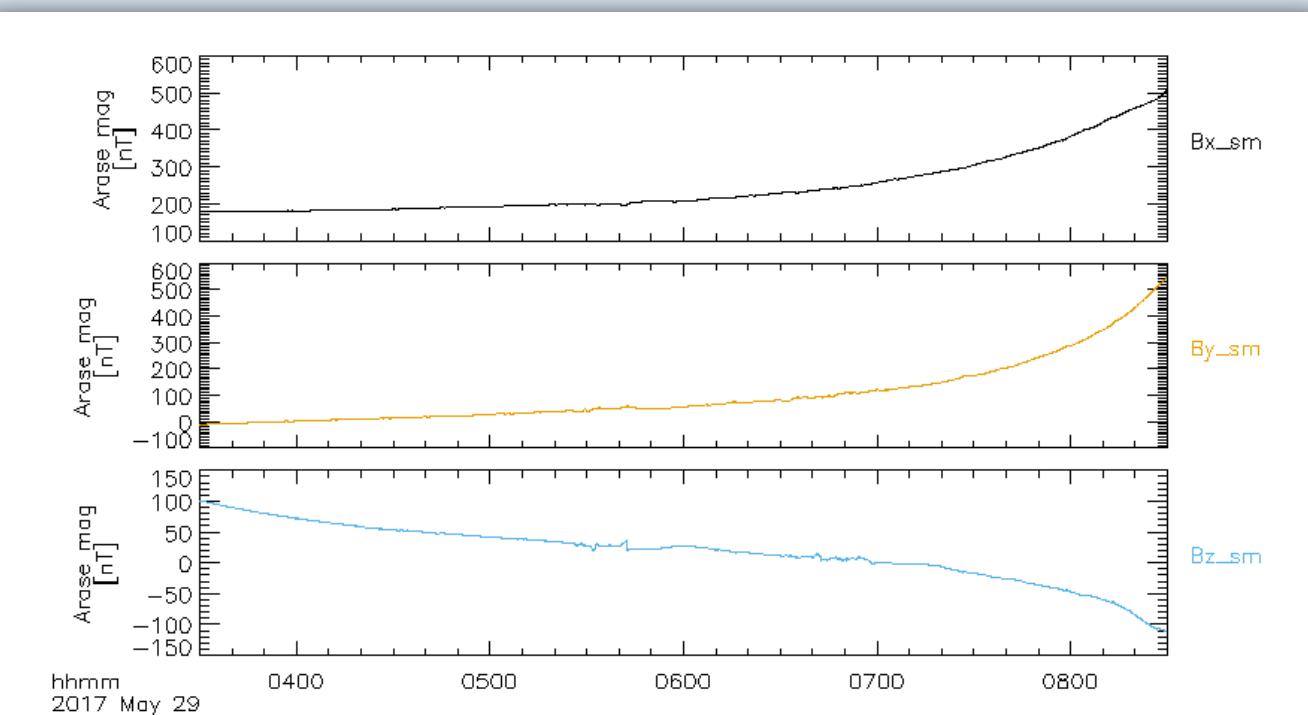
```
STORE_DATA(260): Creating tplot variable: 20 erg_mgf_12_mag_8sec_sm_y
```

```
STORE_DATA(260): Creating tplot variable: 21 erg_mgf_12_mag_8sec_sm_z
```

```
ERG> tplot, 'erg_mgf_12_mag_8sec_sm_?'
```

**split\_vec** はベクトル値や配列の時系列データを格納するtplot変数を、各成分ごとに別々のtplot変数にする。

- 別のパネルにして並べてプロットできる!
- 別々に装飾できる! (options)



# プロットの縦軸の範囲を変更する



`ylim, varname, ymin, ymax, logflag`

`varname` : tplot変数名、その配列

`ymin/ymax` : 縦軸の最小・大値

両方ゼロにすると自動でスケール調整

`logflag` : ゼロにすると線形スケール(default), 1 にするとログスケールで縦軸をプロット

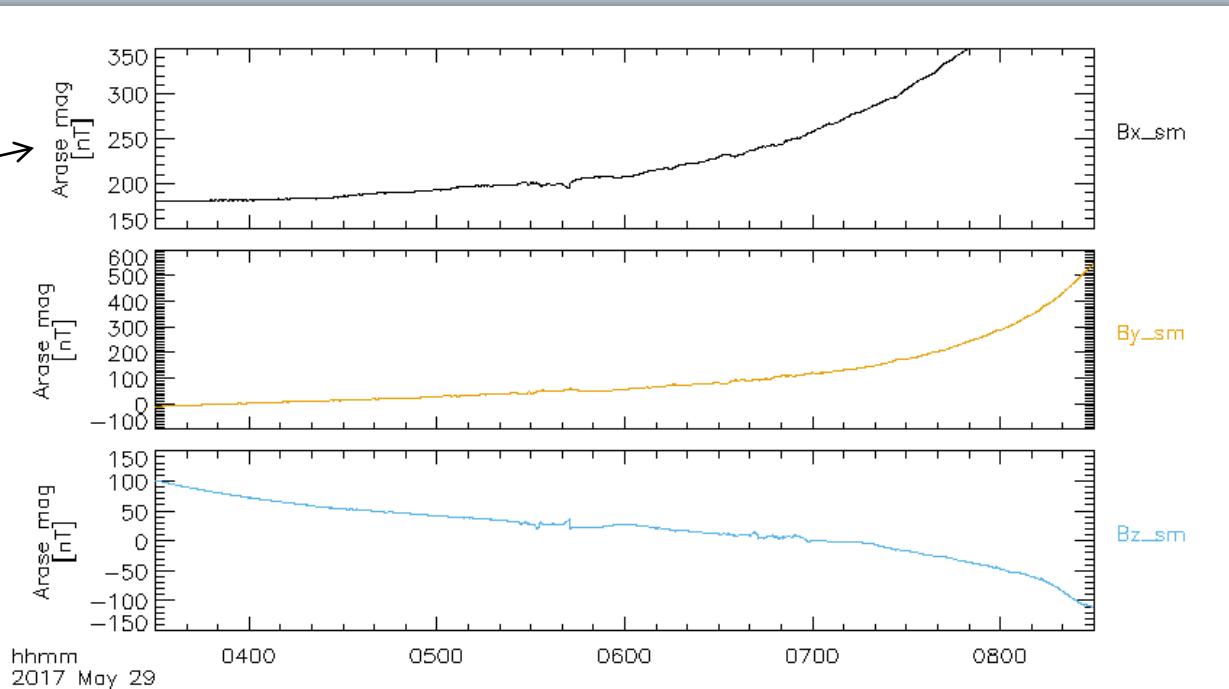
```
ERG> ylim, 'erg_mgf_l2_mag_8sec_sm_x', 150,350, 0
```

```
ERG> tplot
```

縦軸の範囲を変えることで、見たい範囲の変動をより詳細に調べることができる

自動スケール調整に戻る場合:

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



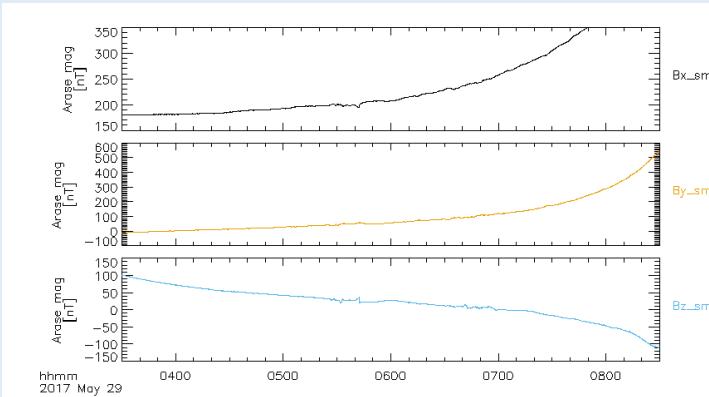
# pngファイル、postscriptファイル、テキストファイルに出力



pngファイル or postscriptファイル

```
ERG> cwd ;Display the current directory  
CWD(25): Directory changed to: /yyyy/yyyy  
ERG> tplot, 'erg_mgf_l2_mag_8sec_sm?'  
ERG> makepng, 'erg_mgf_plot' ;→ erg_mgf_plot.png  
  
ERG> popen, 'erg_mgf_plot'  
ERG> tplot ;Redo the last plot  
ERG> pclose ; → erg_mgf_plot.ps
```

/yyyy/yyyy/erg\_mgf\_plot.png



tplot変数の中身のデータをアスキーテキストとしてダンプ

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

```
kawa3:~ teramari$ cat erg_mgf_l2_mag_8sec_gse.txt  
2017-05-27/00:00:05.450 4.8450809e+02 7.9739140e+02 -6.6735940e+01  
2017-05-27/00:00:13.435 4.8518178e+02 8.0074490e+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 4.8998944e+02 8.2504219e+02 -6.8972403e+01  
2017-05-27/00:01:17.358 4.9012297e+02 8.2864665e+02 -7.1513794e+01  
2017-05-27/00:01:25.359 4.9170651e+02 8.3205503e+02 -6.7162655e+01  
2017-05-27/00:01:33.344 4.9212651e+02 8.3568878e+02 -6.8484317e+01  
2017-05-27/00:01:41.344 4.9235189e+02 8.3936319e+02 -7.1001025e+01  
2017-05-27/00:01:49.326 4.9343653e+02 8.4292241e+02 -6.9100879e+01  
2017-05-27/00:01:57.310 4.9437997e+02 8.4652938e+02 -6.8052480e+01  
2017-05-27/00:02:05.311 4.9460349e+02 8.5029266e+02 -7.0758300e+01  
2017-05-27/00:02:13.296 4.9510470e+02 8.5396844e+02 -7.1595251e+01  
2017-05-27/00:02:21.297 4.9673170e+02 8.5753736e+02 -6.7421116e+01  
2017-05-27/00:02:29.282 4.9693760e+02 8.6134959e+02 -7.0103764e+01  
2017-05-27/00:02:37.267 4.9711743e+02 8.6515922e+02 -7.2860997e+01  
2017-05-27/00:02:45.264 4.9883043e+02 8.6878014e+02 -6.8160833e+01  
2017-05-27/00:02:53.248 4.9923304e+02 8.7263199e+02 -6.9972202e+01  
2017-05-27/00:03:01.249 4.9946922e+02 8.7649747e+02 -7.2552333e+01
```



ブラウザで [/YOUR\\_SPEDAS\\_DIR/idl/\\_spd\\_doc.html](/YOUR_SPEDAS_DIR/idl/_spd_doc.html) with を開くと、SPEDASの各ルーチンのマニュアルを閲覧することができる。ただし各ルーチンのヘッダー部分を集めているだけなので、そのまま英語で記述されている。

The screenshot shows a Mac OS X-style web browser window with the title bar reading "YOUR\_SPEDAS\_DIR/idl/\_spd\_doc.html". The main content area displays the SPEDAS Software Help page for version r26714 from February 26, 2019. The page includes a header, a note about its creation by the IDL library routine `mk_html_help2`, and a timestamp for the last modification. Below these are links to various sections of the documentation, such as A through Z and specific modules like CDAWlib, GEOPACK, and CDF. A section titled "Directories Searched:" lists numerous external and general directory paths that were searched for help content.

SPEDAS SW Help for spdsrw26714\_2019-02-26

This page was created by the IDL library routine `mk_html_help2`.

Last modified: Tue Feb 26 18:16:03 2019.

[A](#), [B](#), [C](#), [D](#), [E](#), [F](#), [G](#), [H](#), [I](#), [J](#), [K](#), [L](#), [M](#), [N](#), [O](#), [P](#), [Q](#), [R](#), [S](#), [T](#), [U](#), [V](#), [W](#), [X](#), [Y](#), [Z](#), [\\_](#), [◆](#)

**Directories Searched:**

- [external/CDAWlib](#)
- [external/CDAWlib/binning](#)
- [external/CDAWlib/jh](#)
- [external/CDAWlib/twins](#)
- [external/IDL\\_GEOPACK](#)
- [external/IDL\\_GEOPACK/examples](#)
- [external/IDL\\_GEOPACK/t01](#)
- [external/IDL\\_GEOPACK/t04s](#)
- [external/IDL\\_GEOPACK/t89](#)
- [external/IDL\\_GEOPACK/t96](#)
- [external/IDL\\_GEOPACK/trace](#)
- [external/IDL\\_GEOPACK/ts07](#)
- [external/IDL\\_ICY](#)
- [external/developers/outliers\\_and\\_convolution](#)
- [external/developers/solarwind](#)
- [external/misc](#)
- [external/spdfcdas](#)
- [external/spdfcdas/spd\\_cdawlib](#)
- [external/spdfssc](#)
- [general/CDF](#)
- [general/cotrans](#)
- [general/cotrans/aacgm](#)
- [general/cotrans/lmn\\_transform](#)
- [general/cotrans/special](#)
- [general/cotrans/special/enp](#)
- [general/cotrans/special/fac](#)
- [general/cotrans/special/minvar](#)
- [general/cotrans/special/rxy](#)
- [general/cotrans/special/sse](#)



# SPEDASの基礎: tplot変数の操作と各種時系列処 理フィルターの適用

# "calc" を使ってtplot変数を使った演算を行う



**calc, ' ... ... ... ... ... '**

'... ... ' はtplot変数を使った数式

```
ERG> timespan, '2017-05-29'
```

```
ERG> erg_load_mgf
```

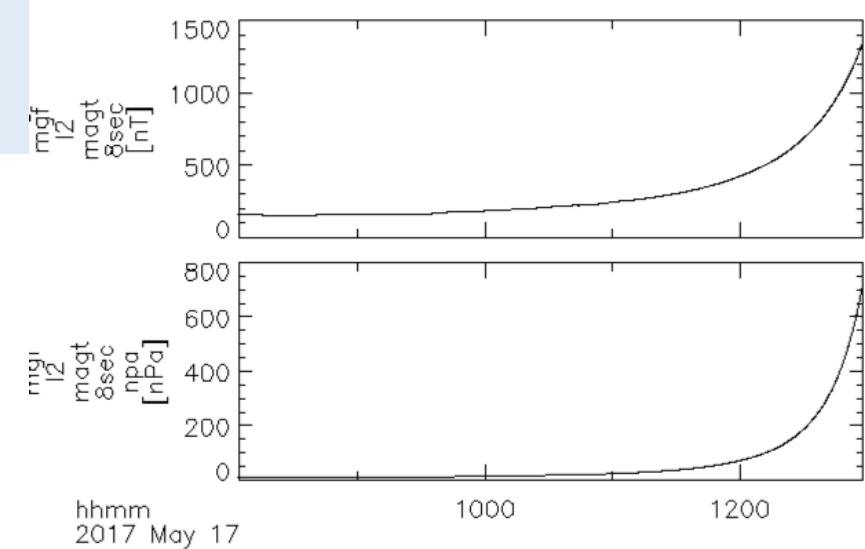
```
ERG> calc, ' "erg_mgf_l2_magt_8sec_npa" = "erg_mgf_l2_magt_8sec"^-2/(4*3.1416*1e-7) * 1e-9' ;; to in nPa
```

```
ERG> options, 'erg_mgf_l2_magt_8sec_npa', ysubtitle='[nPa]'
```

```
ERG> tplot, ['erg_mgf_l2_magt_8sec*']
```

上記の例ではMHDで言うところの磁気圧 ( $|B|^2/2\mu$ ) をcalcコマンドを使ってMGFデータから計算して、それは新たなtplot変数 (erg\_mgf\_l2\_magt\_8sec\_npa )に格納している。

tplot変数をそのまま数式の記述に使えるので、ある意味すっきり書けるし、プログラムの見通しも良くなる。



# "tinterpol\_mxN" を使って他のtplot変数の時刻ラベルに合わせて補間



```
IDL> erg_load_pwe_hfa, level='l3'      ;; UHR波動等から導出した電子密度データをロード  
IDL> tinterpol_mxN, 'erg_pwe_hfa_l3_1min_ne_mgf', 'erg_mgf_l2_magt_8sec_npa'  
IDL> get_data, 'erg_pwe_hfa_l3_1min_ne_mgf_interp', t_ne, dat_ne  
IDL> get_data, 'erg_mgf_l2_magt_8sec_npa', t_mag, dat_mag  
IDL> help, t_ne, t_mag, dat_ne, dat_mag  
  
T_NE          DOUBLE    = Array[10123]  
T_MAG         DOUBLE    = Array[10123]  
DAT_NE        DOUBLE    = Array[10123]  
DAT_MAG       DOUBLE    = Array[10123]
```

HFA Ne データは1分値、MGFデータは8秒値。同じ時刻範囲でも配列数がかなり異なる。

**tinterpol\_mxN**コマンドによって、MGFデータと同じ時刻ラベルを持つHFA Neデータを作成できる。1分値と1分値の間の値は補間計算して求められる。

# 等時間幅で平均する- avg\_data -



**avg\_data**, 'varname', timebin

varname: tplot変数名 (文字列、整数)

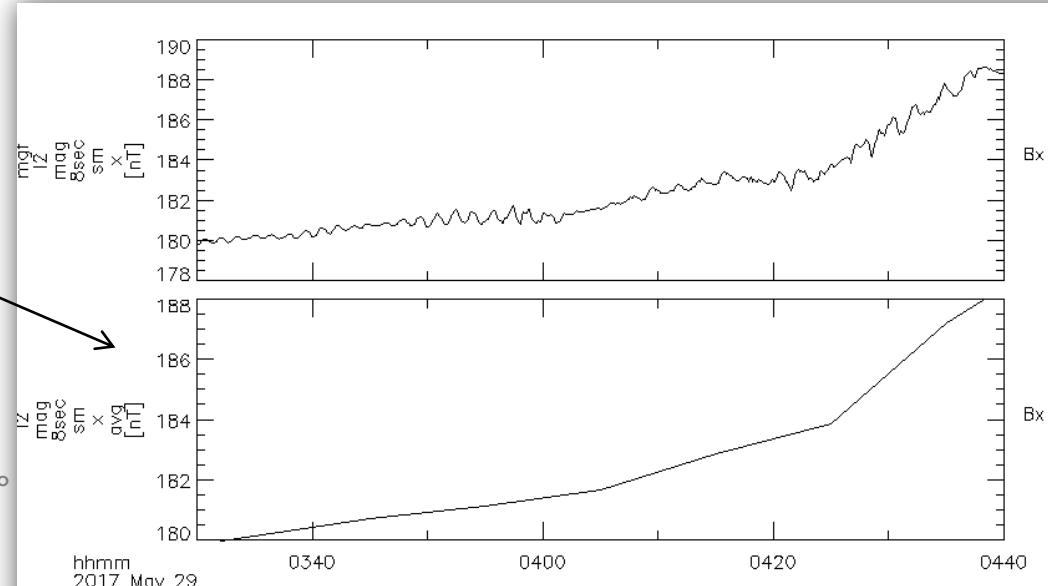
timebin : 平均する時間幅 (単位は秒)

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

tplot変数を一旦全削除して新たにロード

600秒ごとの平均値を求めている。  
こういう平均の仕方を英語で  
boxcar average と言う。

元は8秒ごとに値があつたものが、1分に1つ  
になって、結果としてデータ点数が減っている。



# データを時間方向にスムージング – tsmooth\_in\_time –



`tsmooth_in_time, 'varname', timebin`

`varname` : tplot変数名

`timebin` : 移動平均の時間幅 (単位は秒)

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

```
ERG> tplot_names
```

... ... ...  
19 erg\_mgf\_l2\_mag\_8sec\_sm\_x

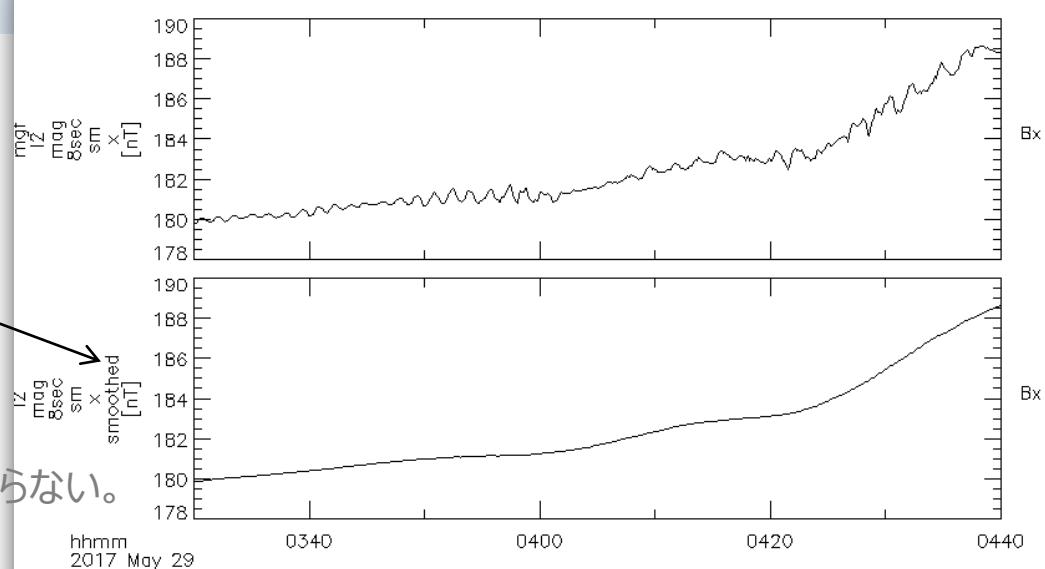
... ... ...  
23 erg\_mgf\_l2\_mag\_8sec\_sm\_x\_smoothed

```
ERG> tplot, [ 19 , 23 ]
```

600秒幅で移動平均を計算したもの。600秒より短い変動が平均化されて滑らかになるので、600秒でローパスフィルターをかけたような結果が得られる。

移動平均のことを英語ではrunning averageと言う。

データ点の数はフィルター前と変わらない。



# ハイパスフィルターをかける - `thigh_pass_filter` -



`thigh_pass_filter`, 'varname', timebin

varname : tplot変数名

timebin : 移動平均の時間幅 (単位は秒)

```
ERG> thigh_pass_filter, 'erg_mgf_l2_mag_8sec_sm', 600.
```

```
ERG> tplot_names
```

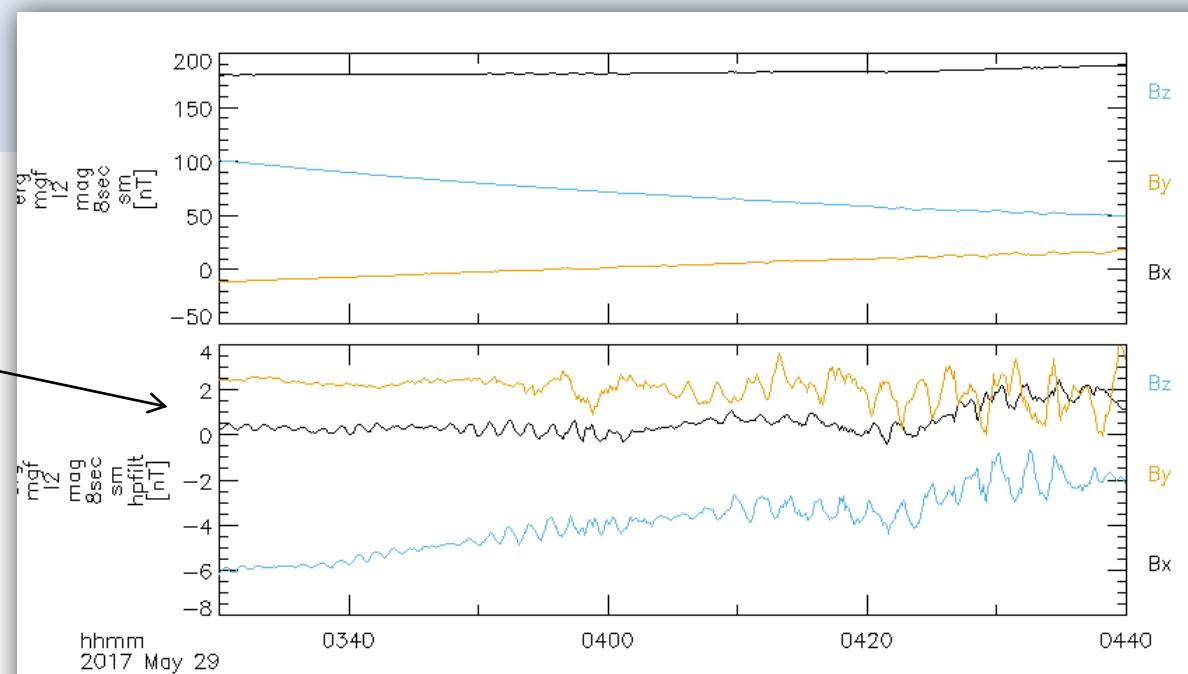
```
....  
4 erg_mgf_l2_mag_8sec_sm  
....
```

```
24 erg_mgf_l2_mag_8sec_sm_hpfilt
```

```
ERG> tplot, [ 4 , 24 ]
```

600秒より短いスケールの時間変動を抽出して表示している。

実はこのコマンドは`tsmooth_in_time`を使って作った移動平均を元のデータから差し引いているだけ。高速フーリエ変換(FFT)のような周波数フィルターは使っていない。





# SPEDASの基礎: tplot変数データの周波数解析

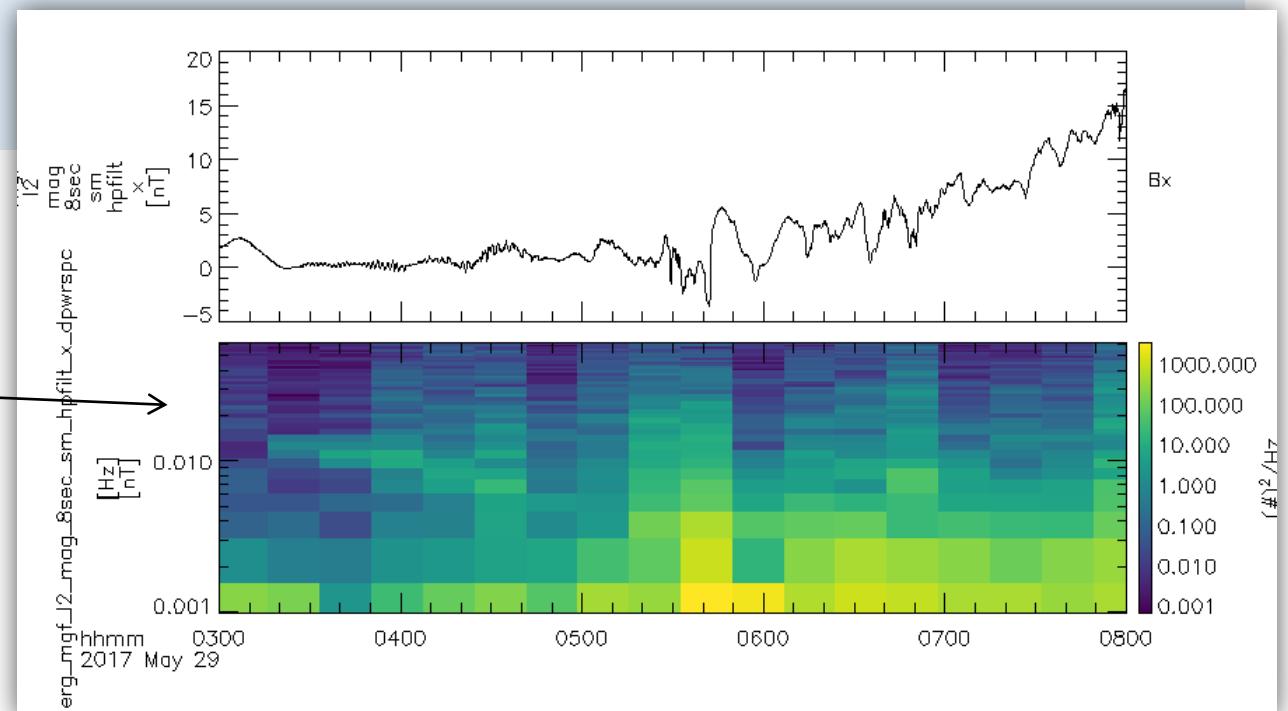
# ダイナミックスペクトル – tdpwrspc–



**tdpwrspc, 'varname'**  
varname : tplot変数名

```
ERG> tdpwrspc, 'erg_mgf_12_mag_8sec_sm_hpfilt'  
ERG> tplot_names  
... ... ...  
25 erg_mgf_12_mag_8sec_sm_hpfilt_x  
... ... ...  
28 erg_mgf_12_mag_8sec_sm_hpfilt_x_dpwrspe  
ERG> tplot, [ 25, 28 ]  
ERG> tlimit,'2017-05-29/03:00','2017-05-29/08:00'
```

高速フーリエ変換(FFT) + Hanning窓を使って、上の時系列データの周波数スペクトルを求めている。



# Waveletを用いた周波数スペクトル - wav\_data -



**wav\_data, 'varname'**

varname : tplot変数名

wav\_data は32768点以下のデータにのみ適用可。

```
ERG> wav_data, 'erg_mgf_l2_mag_8sec_sm_hpfilt_x'
```

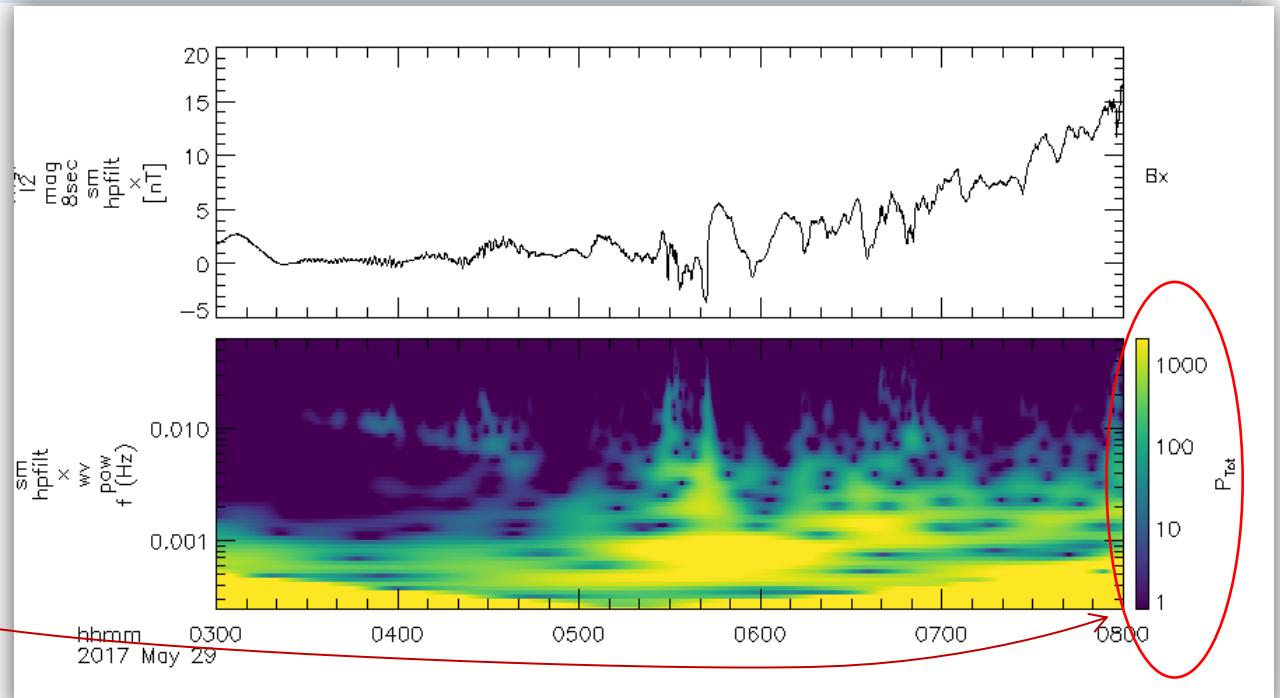
```
STORE_DATA(260): Creating tplot variable: 31 erg_mgf_l2_mag_8sec_sm_hpfilt_x_wv_pow
```

```
ERG> zlim, 31, 1, 2000, 1
```

```
ERG> tplot, ['erg_mgf_l2_mag_8sec_sm_hpfilt_x','erg_mgf_l2_mag_8sec_sm_hpfilt_x_wv_pow']
```

Wavelet解析を使って周波数スペクトルを求めている。

zlim は"ylim"と似た機能を持つ。ただし適用対象は縦軸ではなくカラースケールの上・下限値。





- **SPEDAS wiki**
  - [http://spedas.org/wiki/index.php?title=Main\\_Page](http://spedas.org/wiki/index.php?title=Main_Page)
    - ユーザーガイド、plug-in開発向けガイド、よく使うテクニック、crib sheet(チートシート)のリスト、…
- **SPEDAS開発版(bleeding edge版)の開発ログ** (残念ながら2023年8月で更新停止？)
  - <http://spedas.org/changelog/>
- **tplotに関するCrib sheetの例** (SPEDASディレクトリ/idl/general/examples/)
  - **crib\_tplot.pro** -- tplotの基本的な機能の説明
  - **crib\_tplot\_annotation.pro** -- プロット中のラベル、テキストなどの変更の仕方
  - **crib\_tplot\_export\_print.pro** -- tplotデータやtplotで作ったプロットを外部保存する方法
  - **crib\_tplot\_layout.pro** -- プロットのレイアウトの調整について
  - **crib\_tplot\_range.pro** -- プロットの範囲やスケールの考え方
  - **crib\_tplot\_ticks.pro** -- プロットの目盛り(ticks)の調整の仕方 (場所, 長さ, etc...)
- **SPEDAS-J wiki** 日本のSPEDASユーザーのための情報ページ
  - [https://github.com/spedas-j/member\\_contrib/wiki](https://github.com/spedas-j/member_contrib/wiki)



- ERG-SC ページ (plug-inのダウンロードページ、過去の講習会の資料など)
  - <https://ergsc.isee.nagoya-u.ac.jp>
- ERG-SC wiki
  - [https://ergsc.isee.nagoya-u.ac.jp/mw/index.php/Main\\_Page](https://ergsc.isee.nagoya-u.ac.jp/mw/index.php/Main_Page)

The screenshot shows two side-by-side web pages. On the left is the 'ERG Science Center' homepage, featuring a dark blue header with the ERG logo and 'ERG Science Center'. Below the header are navigation links for About ERG, Documents, Data, Software, ERG-SC Wiki, Bibliography, Press Release, About Us, and Site Map. A Japanese language link is also present. The main content area includes a 'PICK UP' section with news items about ion cyclotron waves and aurora time-scale beats, followed by a detailed description of the ERG project mission and its international partners (ISAS, JAXA, ISEE, Nagoya University). A 'News & Announcements' section is at the bottom. On the right is the 'ERG-SC wiki' main page, which has a similar header with the ERG logo and 'ERG-SC wiki'. It features a sidebar with links to Main page, Discussion, Read, View source, View history, and Search ERG-SC\_wiki. The main content area is titled 'Main Page' and contains a 'Contents' table of contents with sections like Overview, News, Data, and Related sites. Below the table of contents is a detailed description of the ERG project's scientific goals and observational capabilities.



# part\_productsライブラリの使い方 (ポイントのみ)

# "part\_products"とは何か?



- 粒子データの各種スペクトルプロットを作成するのに使用する、SPEDASに集録されているルーチン群のこと



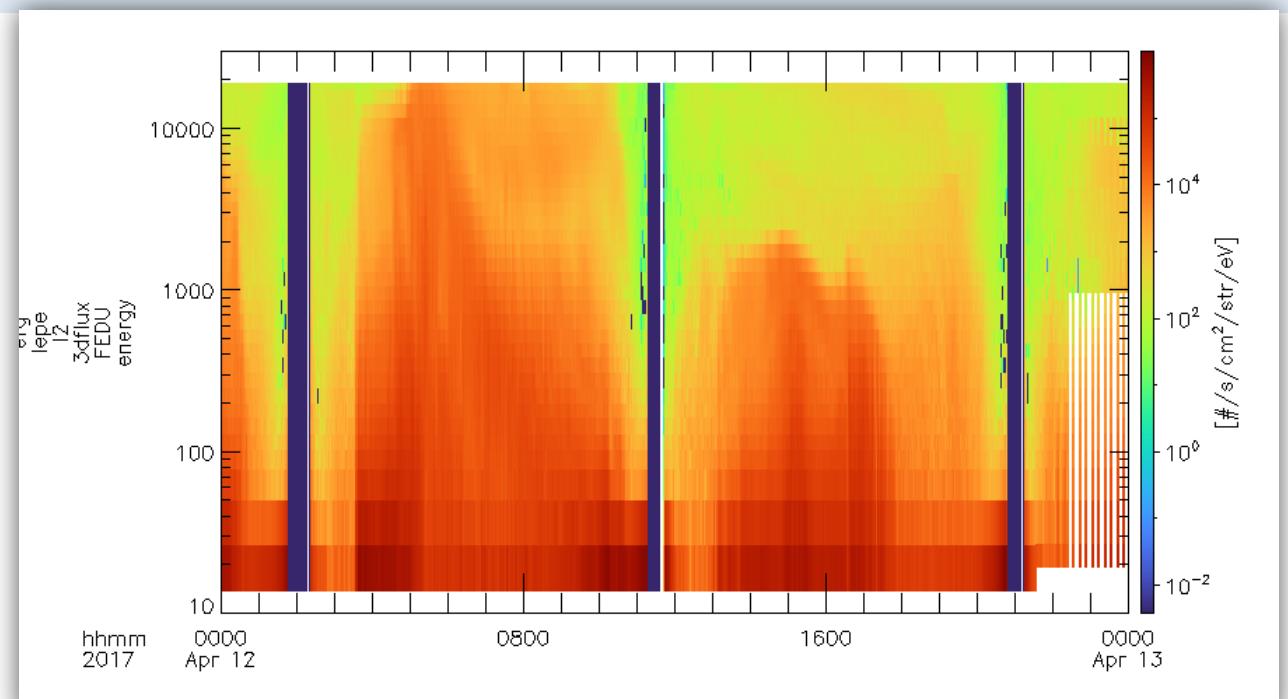
# part\_productsの基本: 平均フラックスのエネルギースペクトル



```
timespan, '2017-04-12'  
erg_load_lepe, datatype='3dflux', uname=username, pass=password, /no_sort_enebin  
  
erg_lep_part_products, 'erg_lepe_l2_3dflux_FEDU', output='energy'  
  
loadct_sd, 48 ;; using the color table Turbo  
tplot, 'erg_lepe_l2_3dflux_FEDU_energy'
```

時間 x エネルギー x 方位角 x 仰角  
↓  
時間 x エネルギー

(方位・仰角方向に粒子フラックスを全部平均)



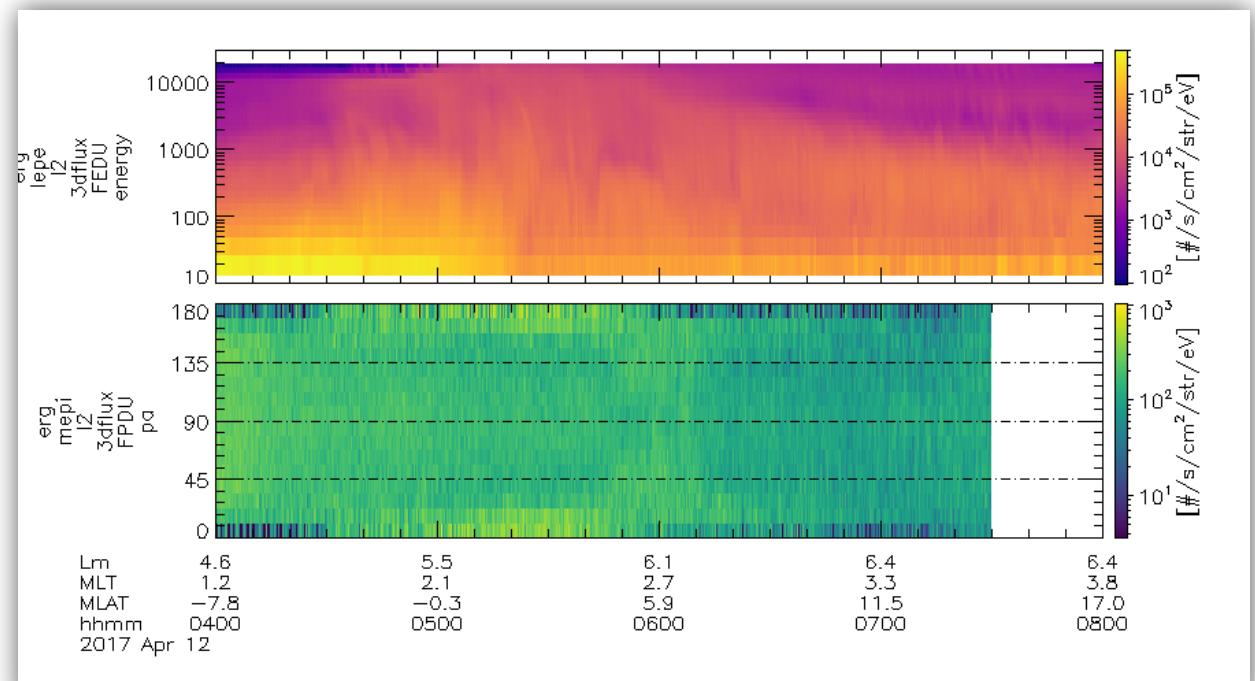
# part\_productsの基本: あるエネルギー範囲でのピッチ角スペクトル



```
timespan, '2017-04-12/04:00',4, /hour & get_timespan, tr  
erg_load_mgf & set_erg_var_label  
magvn = 'erg_mgf_l2_mag_8sec_dsi' & posvn = 'erg_orb_l2_pos_gse'  
erg_load_mepi_nml, datatype='3dflux'  
  
erg_mep_part_products, 'erg_mepi_l2_3dflux_FPDU', output='pa', $  
    energy=[8000., 13000.], trange=tr, mag=magvn, pos=posvn, /no_ang_weighting  
  
options, 'erg_lepe_l2_3dflux_FEDU_energy', 'color_table', 1079 ;; 2つのtplot変数を異なるカラースケールにする。  
options, 'erg_mepi_l2_3dflux_FPDU_pa', 'color_table', 1080 ;; 1079: Plasma, 1080: Viridis  
  
tplot, ['erg_lepe_l2_3dflux_FEDU_energy', 'erg_mepi_l2_3dflux_FPDU_pa']
```

時間 x エネルギー x 方位角 x 仰角  
↓  
時間 x ピッチ角

あるエネルギー範囲のフラックスだけを取り出す。  
磁場データを考慮して方位角x仰角を粒子のピッチ角に  
なおして、ピッチ角毎に平均フラックスを出す。



# part\_productsの基本: あるピッチ角範囲のフラックス値のエネルギースペクトル



```
erg_load_mepe, level='l2', datatype='3dflux'  
erg_mep_part_products, 'erg_mepe_l2_3dflux_FEDU', output='energy', $  
    trange=tr, mag=magvn, pos=posvn, pitch=[0, 7], /no_ang_weighting  
options, 'erg_mepe_l2_3dflux_FEDU_energy', color_table=1072, reverse_color_table=1  
tplot, ['erg_lepe_l2_3dflux_FEDU_energy', 'erg_mepi_l2_3dflux_FPDU_pa', $  
    'erg_mepe_l2_3dflux_FEDU_energy']
```

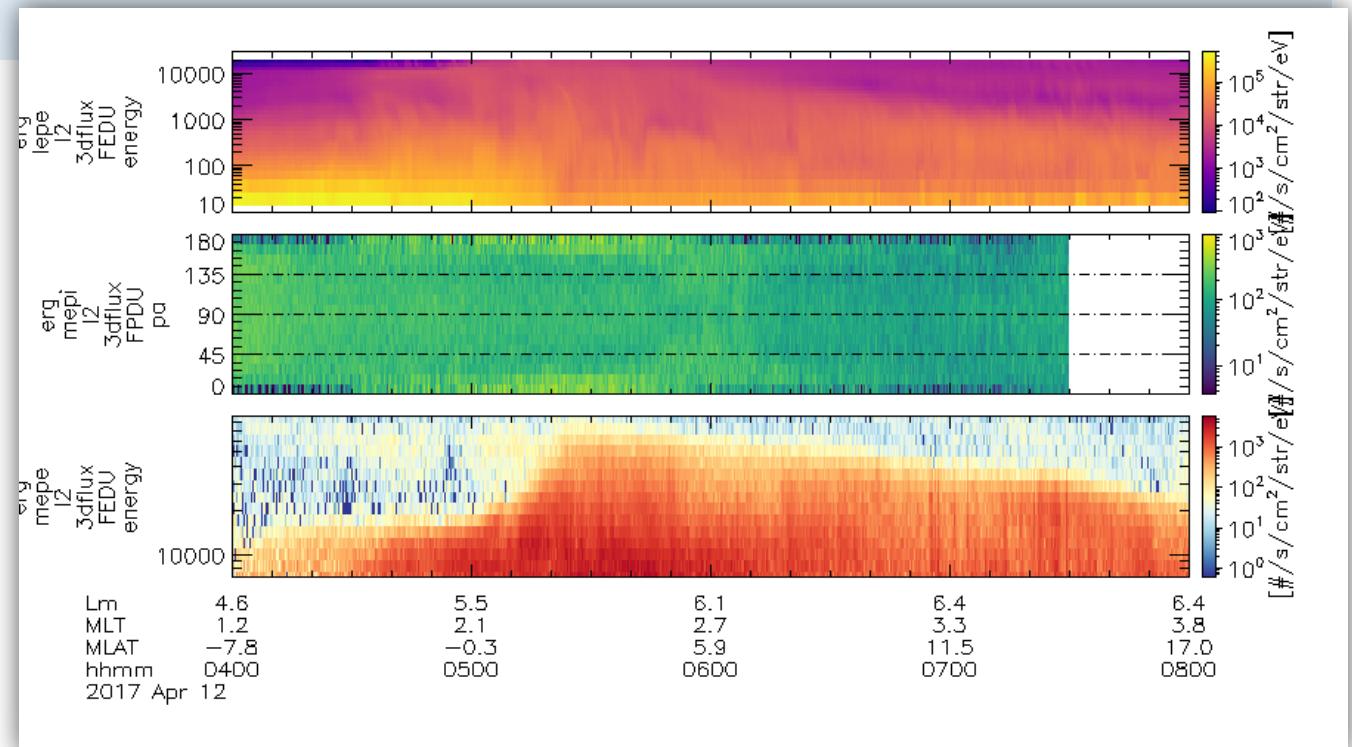
時間 × エネルギー × 方位角 × 仰角  
↓  
時間 × エネルギー

方位角と仰角からピッチ角を算出し、そのピッチ角が「ある範囲」に入っているフラックス値のみ抜き出して、そのエネルギースペクトルを求めている。

上記の例ではピッチ角が $0^\circ - 7^\circ$ の範囲を指定している  
(pitch=[0, 7])

Mar. 12, 2025

IDL/SPEDAS tutorial, ERG science meeting @ISEE, Nagoya U



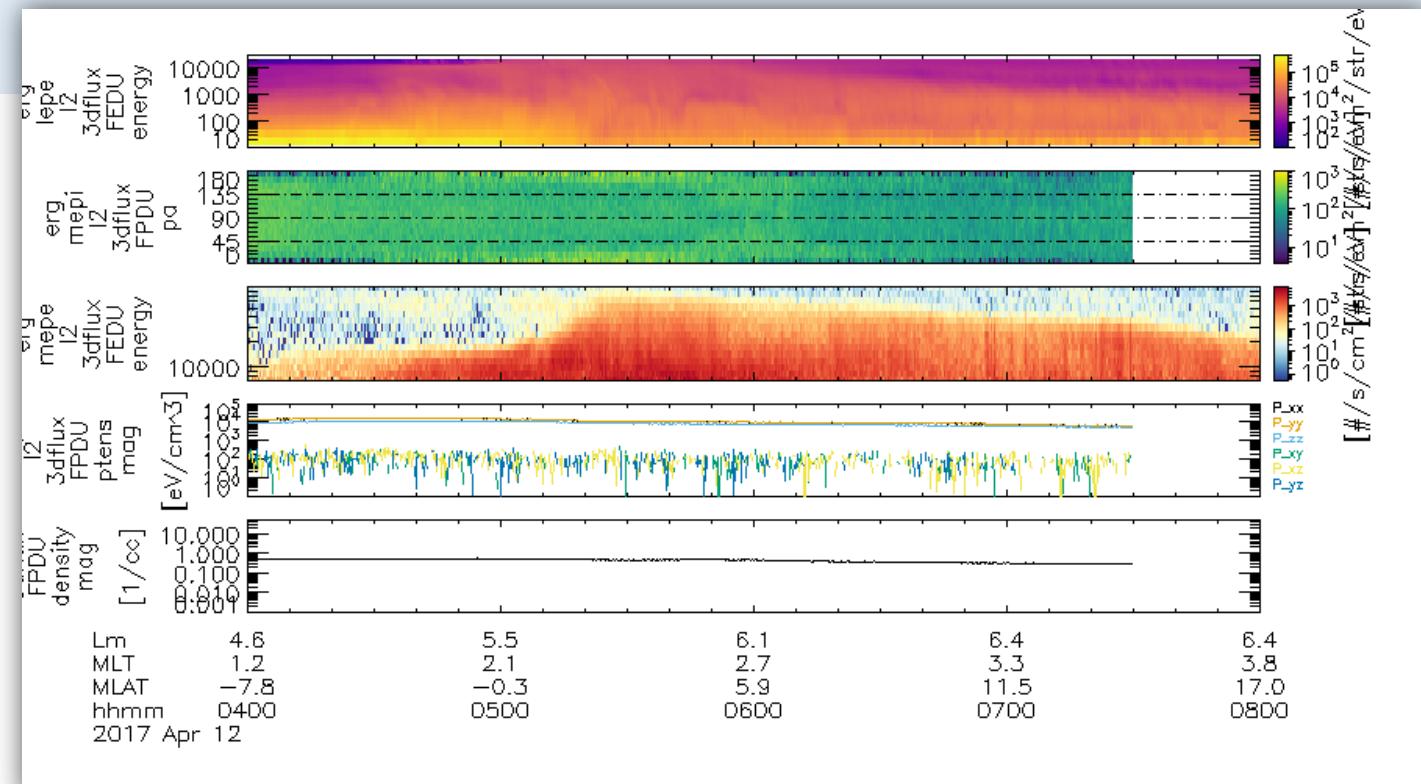
# part\_productsの基本: 速度モーメント値の計算



```
erg_mep_part_products, 'erg_mepi_l2_3dflux_FPDU', output='fac_moments', $  
    trange=tr, mag=magvn, pos=posvn, /no_ang_weighting  
  
options, 'erg_mepi_l2_3dflux_FPDU_ptens_mag', colors=[0,1,2,3,4,5] ;; プロット線の色を指定  
  
vns = tnames( /tplot ) ;; 直前にプロットした変数リストを得る  
  
tplot, [ vns, 'erg_mepi_l2_3dflux_FPDU_+'+['ptens_mag', 'density_mag'] ]
```

時間 × エネルギー × 方位角 × 仰角  
↓  
時間 × スカラー・ベクトル・テンソル量

エネルギー・方位角・仰角空間(つまり速度空間)で  
粒子速度分布関数を積分する。





- ERG-SCウェブページに、もう少し詳しいpart\_productsの使い方に関するスライドがある(ただし英語):

[https://ergsc.isee.nagoya-u.ac.jp/data\\_info/howto.shtml.en](https://ergsc.isee.nagoya-u.ac.jp/data_info/howto.shtml.en)

The screenshot shows the ERG Science Center website with a dark theme. At the top, there's a navigation bar with links for RESEARCH, About ERG-SC, Data, Meeting, Achievement, Contact, and Public Outreach. A yellow dashed oval highlights the "part\_products" section under the "Plots of ERG orbit and ionospheric footprint" heading.

- 4: Basics of SPEDAS: tplot and tplot variable #2 (テキスト p.16~20)
- 5: Basics of SPEDAS: tplot and tplot variable #3 (テキスト p.21~25)
- 6: Basics of SPEDAS: Various filtering routines for tplot data (テキスト p.26~29)
- 7: Basics of SPEDAS: Frequency analysis of tplot data (テキスト p.26~29)
- Plots of ERG orbit and ionospheric footprint
  - textbook  
SPEDAS training course: ERG orbit and footprint (Oct., 2021) (PDF)
  - Crib sheet  
SPEDAS training course: ERG orbit and footprint (Oct., 2021) (PDF)
  - movie  
Youtube Channel (link)
- ERG part\_products
  - textbook  
Brief tutorial for ERG part\_products (Oct., 2021) (PDF)
- ISEE\_3D
  - Visualization tool for three-dimensional plasma velocity distributions (ISEE\_3D) as a plug-in for SPEDAS
    - Reference: Keika et al., EPS (2017) <https://doi.org/10.11186/s40623-017-0761-9>

## Plug-In tools

- ERG Plug-In tool
  - Release



# あらせ衛星の軌道と電離圏フットプリントのプロット

# あらせ衛星の軌道データセット



@ ERG-SC website

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

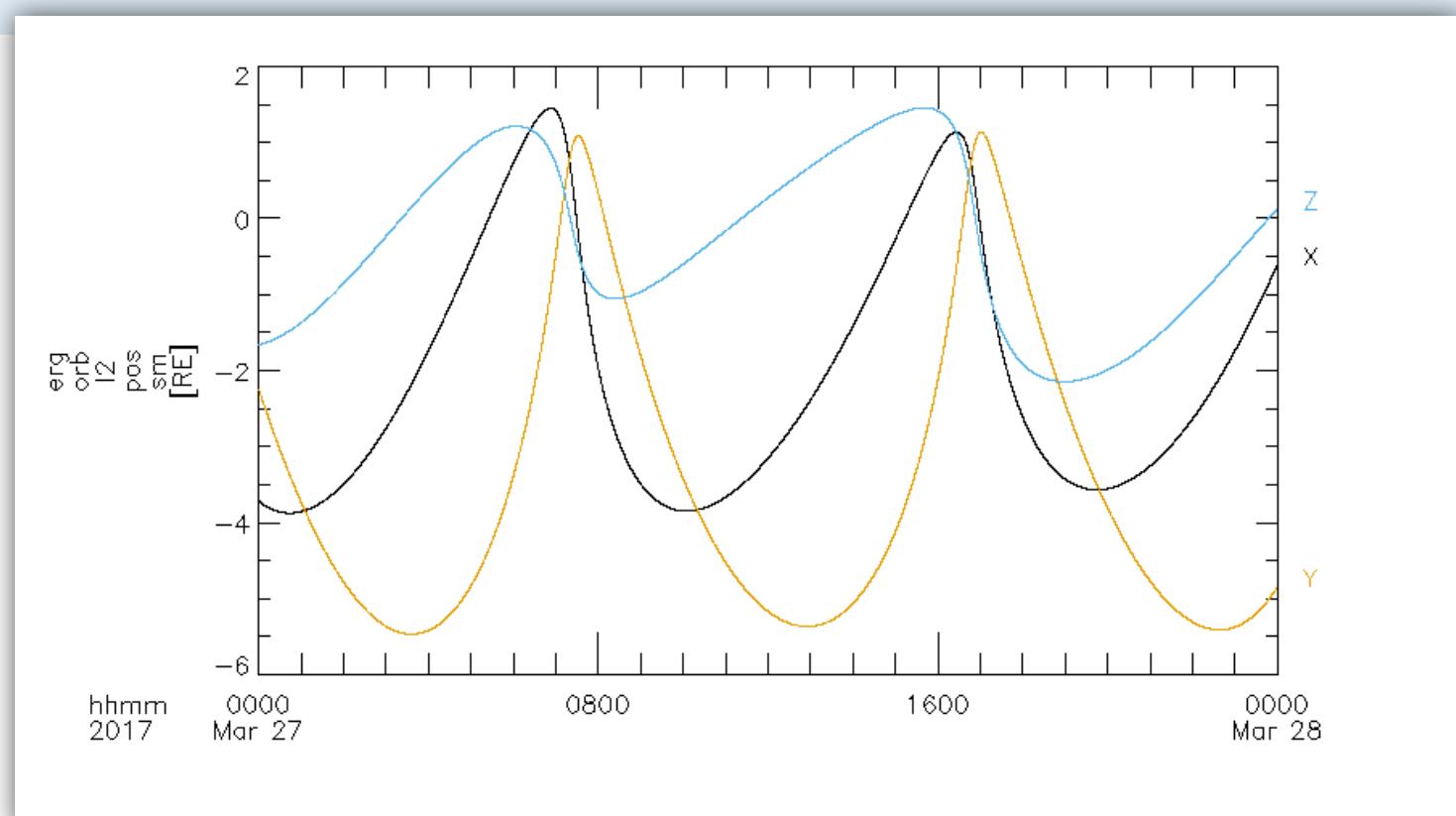
Level (model)	データタイプ	データ読み込みルーチン
Level 2 (IGRF)	確定軌道 Definitive	<code>erg_load_orb</code>
	予測軌道 Predicted (spre, mpre, lpre)	<code>erg_load_orb_predict, datatype='spre or mpre or lpre'</code>
Level 3 (IGRF + 外部磁 場モデル)	OP77Q 磁場モデル	<code>erg_load_orb_l3, model='op or t89 or ts04'</code>
	T89磁場モデル(Kpベース)	
	TS05 (TS04) 磁場モデル (IMF, Dst ベース)	

# あらせ軌道データを tplot でプロットする



```
timespan, '2017-03-27/00:00', 1, /day ;; 時刻幅を指定 2017-03-27 00:00-24:00 UT  
erg_load_orb      ;; データファイルをダウンロードしてIDL上にtplot変数としてロード  
options, 'erg_orb_12_pos_sm', colors=[0,1,2], labels=['X','Y','Z'] ;; 線の色と凡例  
tplot,'erg_orb_12_pos_sm'   ;; プロット!
```

右の例では、時々刻々と変わる  
Solar Magnetic (SM)座標系での  
あらせ衛星の位置 (X, Y, Z)を、線  
プロットで表示。



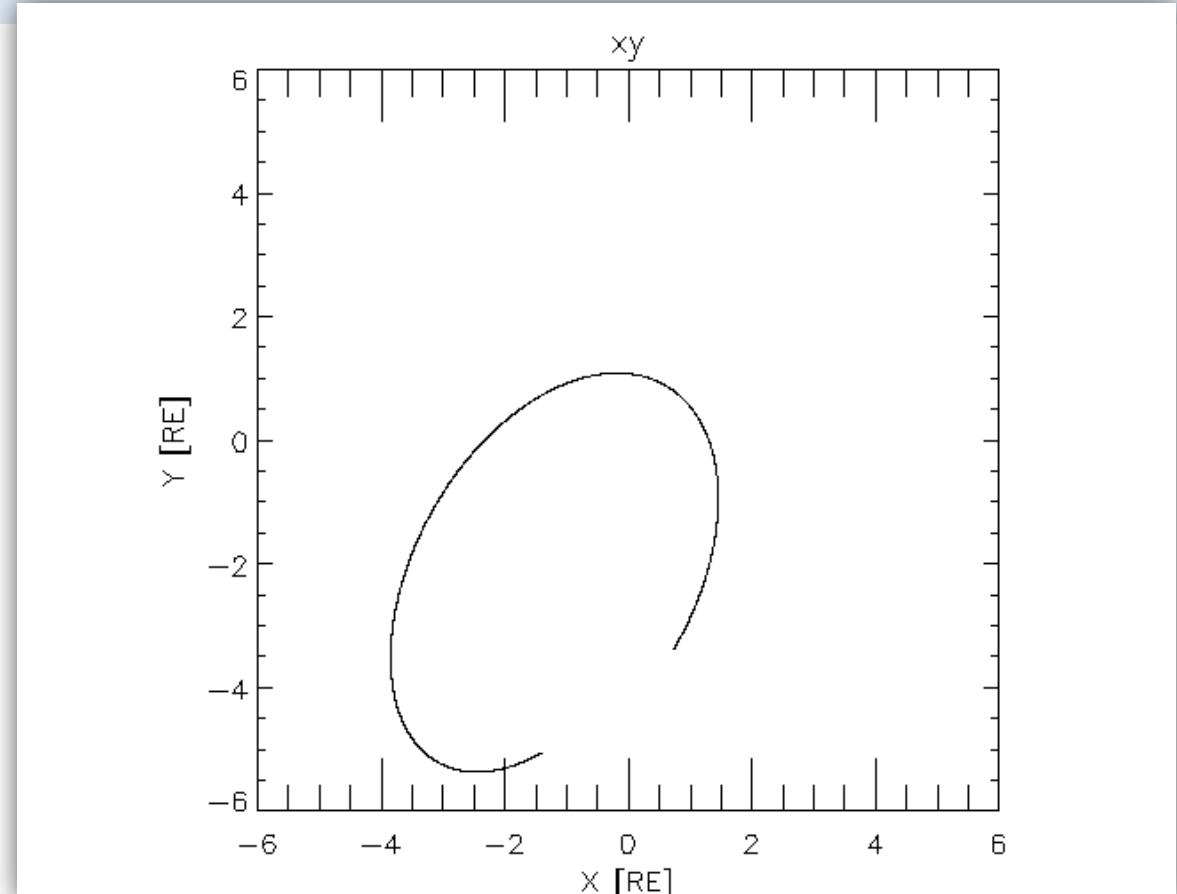
# あらせ軌道データを tplotxy でプロットする



```
timespan, '2017-03-27/06:00', 8, /hour ;; 時刻範囲 2017-03-27 06:00-14:00 UT  
tplotxy, 'erg_orb_l2_pos_sm', xrange=[-6, 6], yrange=[-6, 6] ;; プロット!
```

今度は全く同じデータを tplotxy コマンドでプロット。

defaultでは、tplotxy は 与えられたtplot変数のベクトル値のうち、最初と2つ目の次元の値を、2次元平面上でプロットする。ここではX座標とY座標になっている。



# あらせ軌道データを tplotxy でプロットする (続き)



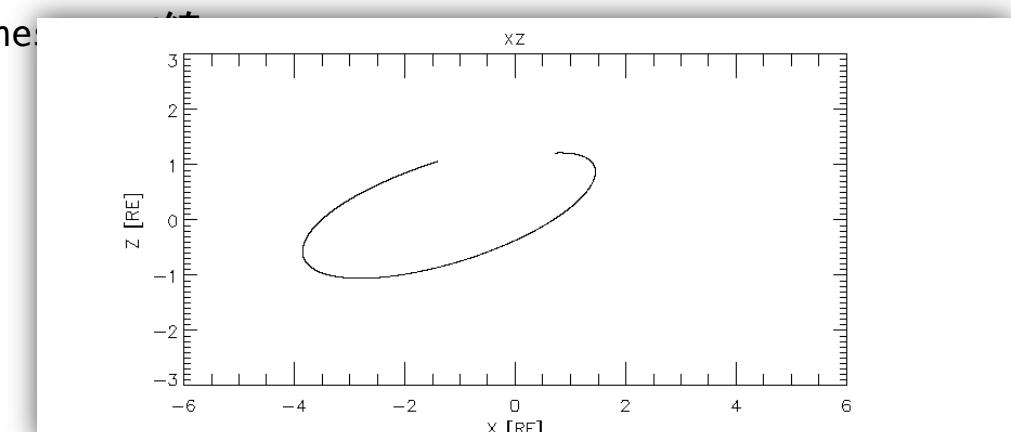
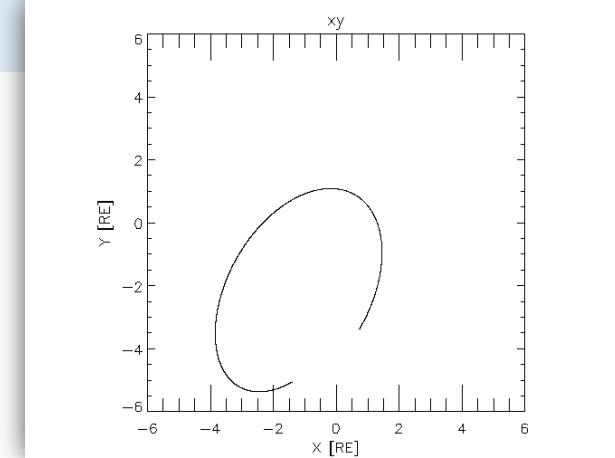
```
tplotxy, 'erg_orb_l2_pos_sm', xrange=[-6, 6], yrange=[-6, 6], versus='xy'    ;; SM X-Y 平面  
tplotxy, 'erg_orb_l2_pos_sm', xrange=[-6, 6], yrange=[-6, 6], versus='xz'    ;; SM X-Z 平面
```

tplotxy には多くのオプションを実装されている:

```
tplotxy, tplot_variable_name $  
, versus='xy' ; 2つのプロット軸 (e.g., xy or xz, ...)  
, multi= 'n_col n_raw' ; 複数プロット作成 ('列数 行数')  
, over = over ; プロットを削除せずに上書き  
, add = add ; 新しいプロットを追加
```

他にも色々: title(タイトル), xrange(X方向の範囲), color(色), line(線種), などなど

今度は全く同じデータを tplotxy コマンドでプロット。



# あらせ軌道データを tplotxy でプロットする（続き）



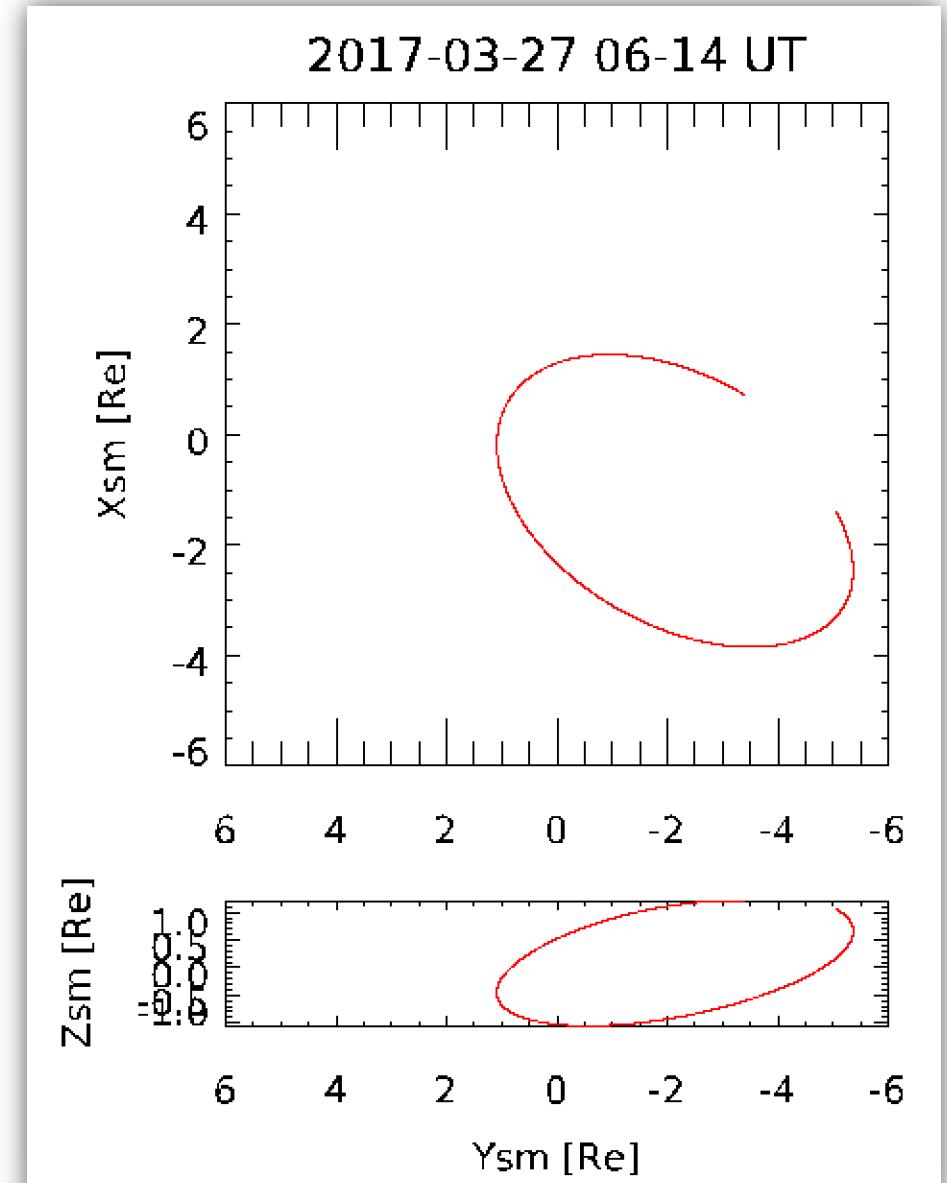
```
;; 時刻範囲 2017-03-27 06:00-14:00 UT
timespan, '2017-03-27/06:00', 8, /hour
erg_load_orb

;; Y-X plot and Y-Z plot of the orbit

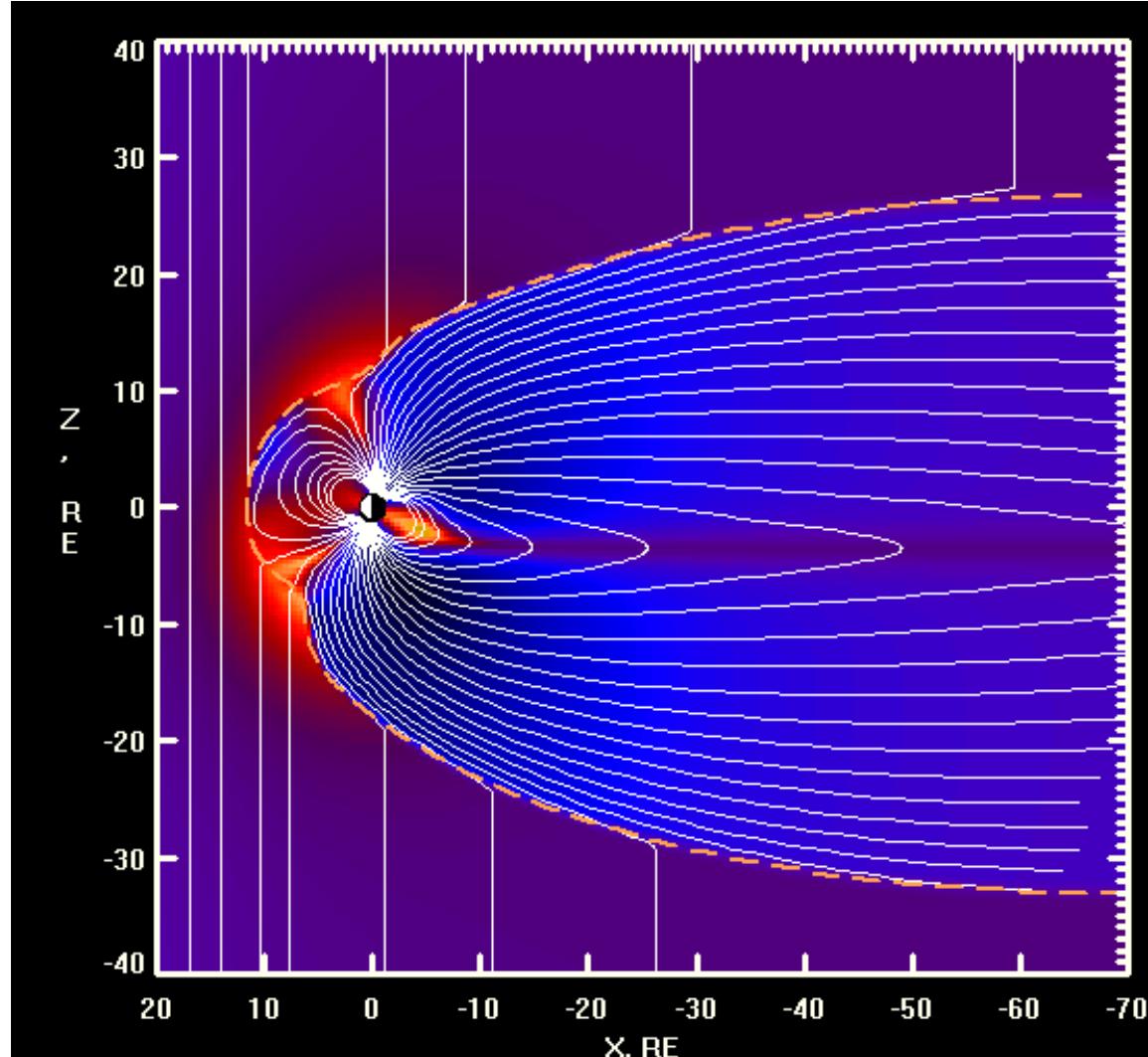
tplotxy, 'erg_orb_l2_pos_sm', $
  xrange=[-6, 6], yrangle=[-6, 6], versus='yrx', $
  xtitle=' ', ytitle='Xsm [Re]', multi='1 2', $
  color='r', xmargin=[0.2, 0.3], $
  title='2017-03-27 06-14 UT'

tplotxy, 'erg_orb_l2_pos_sm', $
  xrange=[-6, 6], versus='yrz', $
  xtitle='Ysm [Re]', ytitle='Zsm [Re]', $
  /add, color='r', ymargin=[0.8, 0.0], $
  xmargin=[0.2,0.3], title=' '
```

オプションを駆使すると様々な軌道プロットを作成できる。オプションの詳細についてはtplotxy.pro のソースの最初についているマニュアルを参照のこと。



# あらせ衛星の電離圏フットプリント



adapted from:

Modeling the Earth's Magnetosphere Using Spacecraft Magnetometer Data

<https://geo.phys.spbu.ru/~tsyganenko/modeling.html>

Mar. 12, 2025

衛星位置と、そこから磁力線をトレースしてたどり着く電離圏上の場所(電離圏フットプリント、フットポイントなどと呼ばれる)との対応を調べることは重要  
(オーロラ粒子降下、Whistler・EMIC波、沿磁力線電流、...)

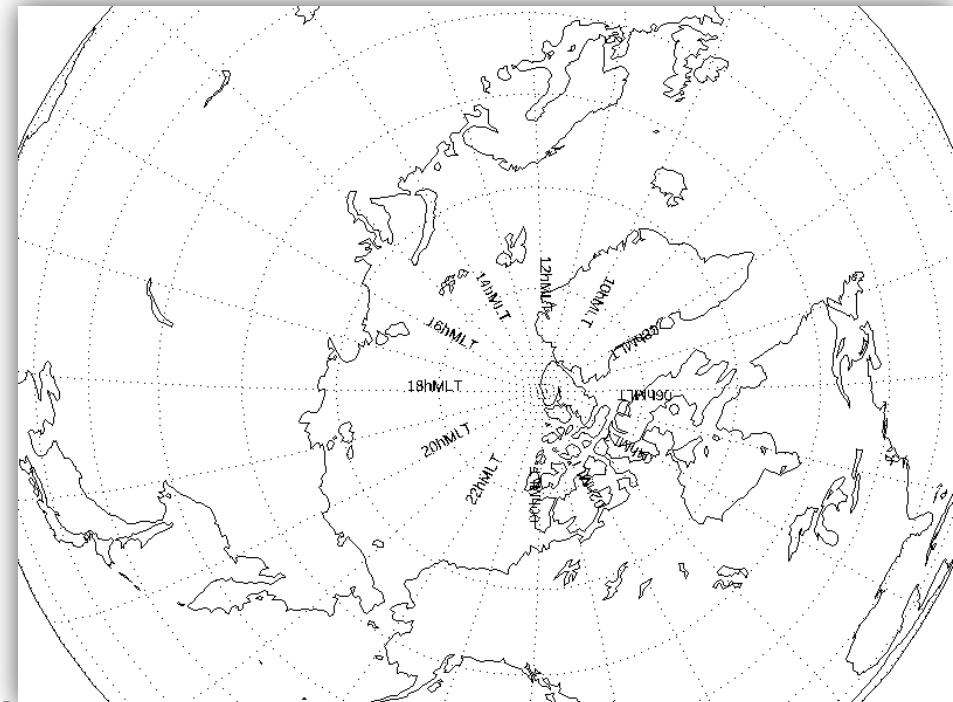
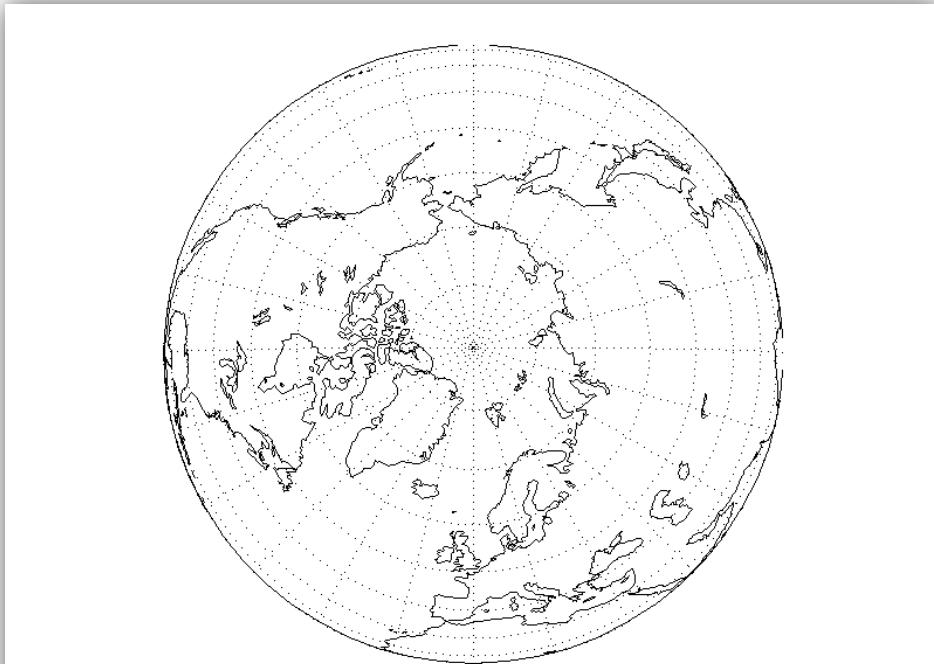
磁気圏の磁力線の構造は時々刻々と変わる。  
衛星も動いていく。

→ 電離圏フットプリントも動いていく。(さらに地球も回る)

# あらせ衛星の電離圏フットプリントの描画 (map2dライブラリを使って)



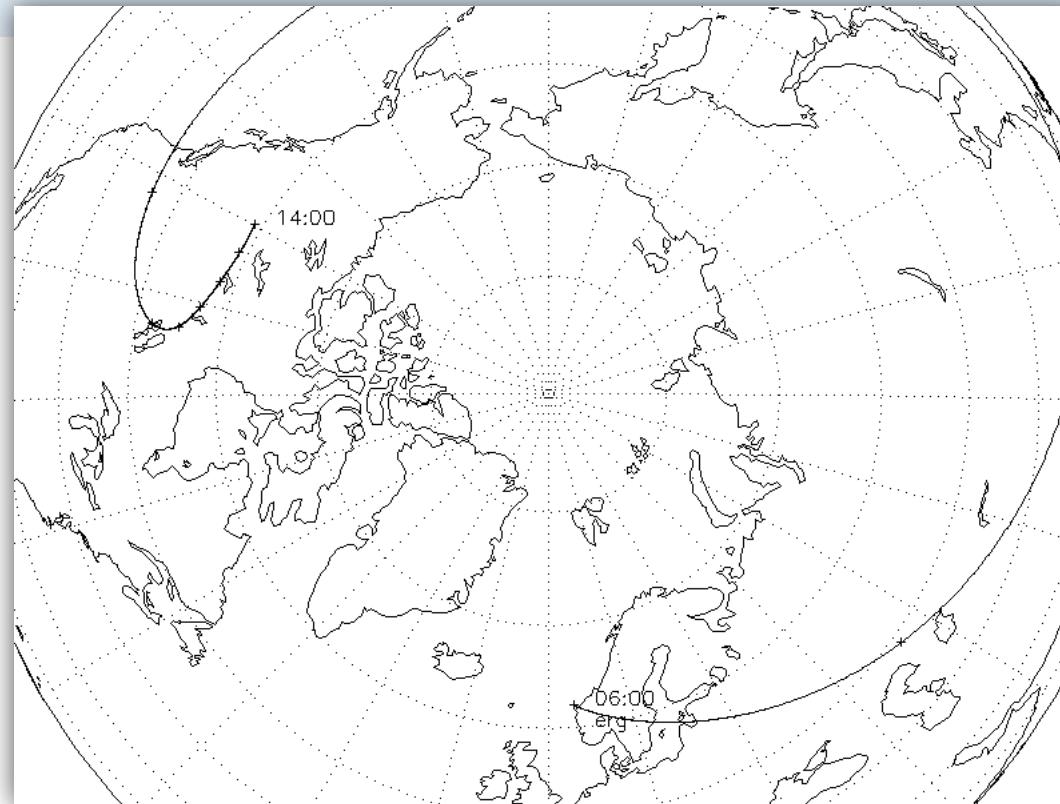
```
map2d_init & map2d_coord, 'geo'  
map2d_set, /erase & overlay_map_coast  
  
map2d_coord, 'aacgm'  
map2d_time, 1000 ;; どの日時の方向で地図を描くか指定する必要 (10:00 UT on 2017-03-27)  
map2d_set, /erase & overlay_map_coast
```



# あらせ衛星の電離圏フットプリントの描画 (map2dライブラリを使って)



```
split_vec, 'erg_orb_l2_pos_iono_north' ;; 地理緯度と経度を別々にtplot変数に  
map2d_coord, 'geo'  
map2d_set, /erase & overlay_map_coast  
overlay_map_sc_ifoot, 'erg_orb_l2_pos_iono_north_0', 'erg_orb_l2_pos_iono_north_1'
```



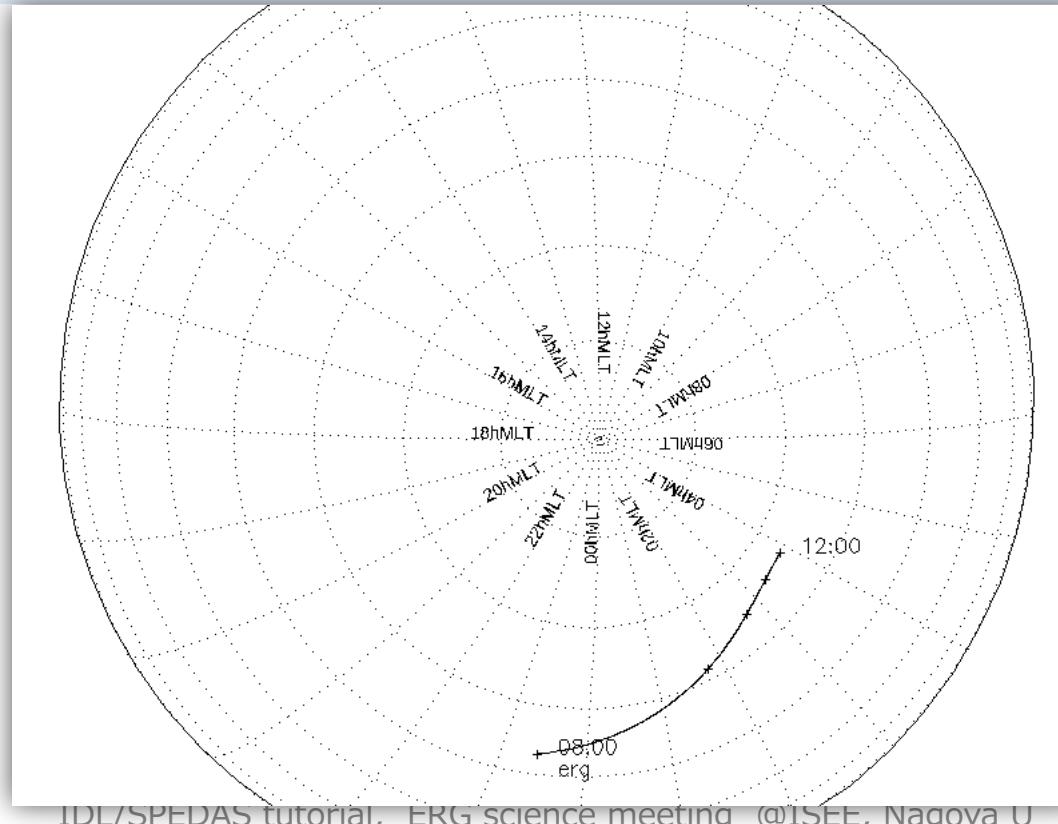
地理座標であらせ衛星のフットプリントを世界地図上に描画

~6 UTには北欧辺りに位置していたフットプリントが~10 UT以降は北米に移動している。

# あらせ衛星の電離圏フットプリントの描画 (map2dライブラリを使って)



```
map2d_coord, 'aacgm' ;; AACGMに描画座標系を切り替え  
timespan, '2017-03-27/08:00', 4, /hour ;; 時刻範囲指定: 08:00-12:00 UT on 2017-03-27  
map2d_time, 1000 ;; 10:00 UTの時点での地図を描く、という指定  
map2d_set, /erase, /mlt, scale=60e+6  
overlay_map_sc_ifoot, 'erg_orb_12_pos_iono_north_0', 'erg_orb_12_pos_iono_north_1'
```



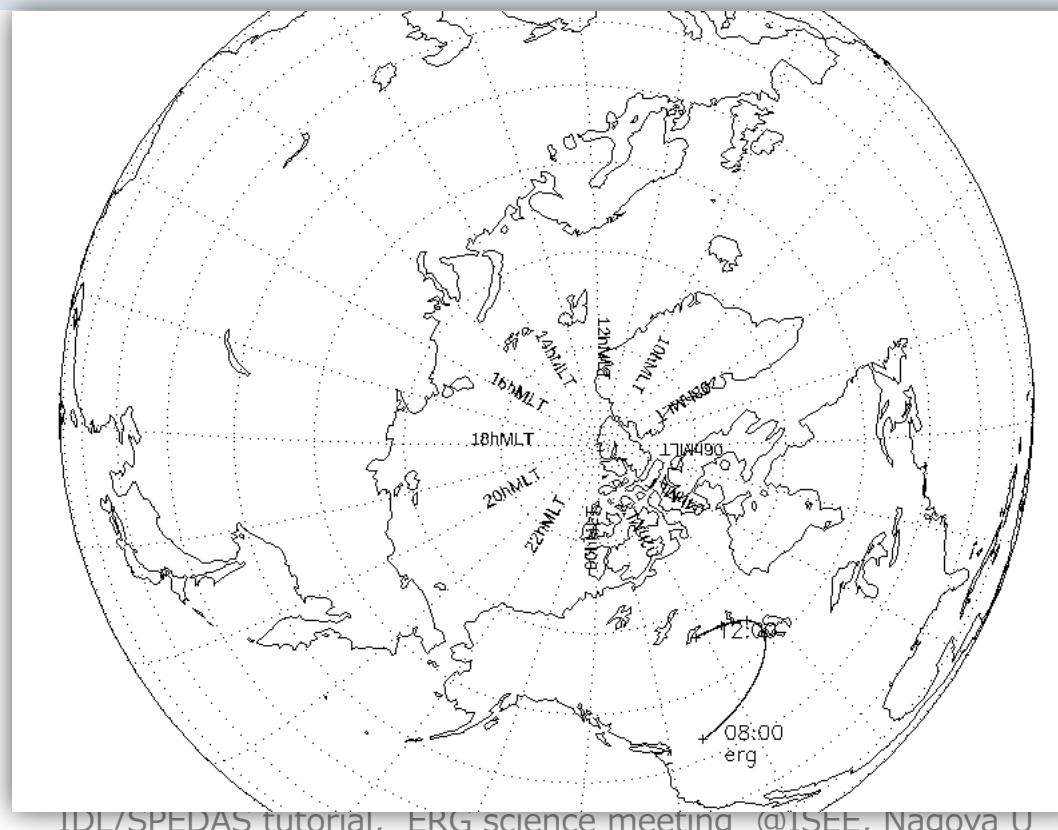
あらせ衛星の電離圏フットプリントをAltitude-adjusted corrected geomagnetic (AACGM)座標系(地磁気座標系の1つ)のMlat-MLT グリッド 上にに描画。

# あらせ衛星の電離圏フットプリントの描画 (map2dライブラリを使って)



```
map2d_coord, 'aacgm' & map2d_time, 1000 ;; 10:00 UT時点の地図に描画、座標はAACGM  
map2d_set, /erase, /mlt, scale=60e+6  
overlay_map_coast  
overlay_map_sc_ifoot, 'erg_orb_12_pos_iono_north_0', 'erg_orb_12_pos_iono_north_1', plottime=!map2d.time
```

どうしてフットプリントの軌道の形が前ページのものと異なるのか？

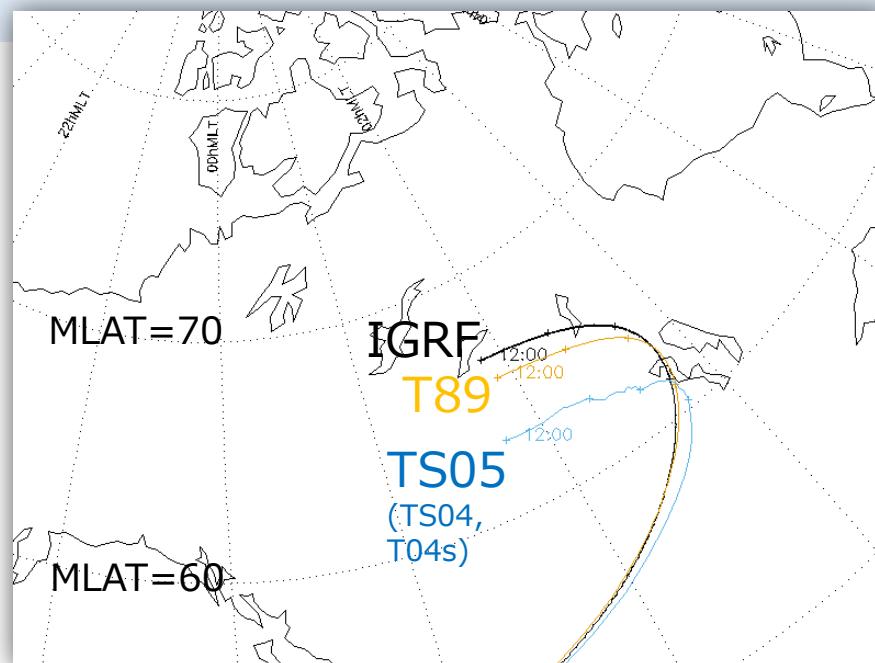


あらせ衛星の電離圏フットプリントを、Altitude-adjusted corrected geomagnetic (AACGM)座標系(地磁気座標系の1つ)で10:00 UT時点での世界地図上に描画。

# あらせ衛星の電離圏フットプリントの描画 (map2dライブラリを使って)



```
erg_load_orb_l3, model='t89' & erg_load_orb_l3, model='ts04'  
split_vec, 'erg_orb_l3_pos_iono_north_*'  
map2d_time, 1000 ;; 描画の日時を10:00 UTに  
map2d_set, /erase, /mlt, scale=20e+6, glatc=60., glonc=250.  
overlay_map_coast  
overlay_map_sc_ifoot, 'erg_orb_l2_pos_iono_north_0', 'erg_orb_l2_pos_iono_north_1', $  
    plottime!=map2d.time, trace_color=0  
overlay_map_sc_ifoot, 'erg_orb_l2_pos_iono_north_t89_0', 'erg_orb_l2_pos_iono_north_t89_1', $  
    plottime!=map2d.time, trace_color=1  
overlay_map_sc_ifoot, 'erg_orb_l2_pos_iono_north_TS04_0', 'erg_orb_l2_pos_iono_north_TS04_1', $  
    plottime!=map2d.time, trace_color=2
```



電離圏フットプリントの位置は、あらせ衛星の位置から電離圏への磁力線トレーシングに用いる磁場モデルに依存する。

異なる磁場モデルを使えば、フットプリントの位置も異なる。



# Appendix

# Appendix A-1: part\_productsで計算される速度モーメント値のリスト



```
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_mtens  
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>
```

part\_productsで計算されるモーメント値のうち主なもの:

- density: 数密度 (/cc)
- avgtemp: 温度 (eV) (!)
- velocity: バルク速度 (km/s)
- vthermal: 热速度 (km/s)
- mtens: 運動量輸送テンソル
- ptens: 圧力テンソル (eV/cm<sup>3</sup>)
- t3: 温度テンソル (eV) (!)
- magt3: 磁場垂直・平行方向の温度 (!)
- flux: 粒子フラックス
- eflux: 粒子エネルギー・フラックス

すべてのベクトル・テンソル量はDSI座標系で与えられる。

(!) 通常の温度定義(Maxwell分布の幅)とは異なる定義で計算された温度であることに注意。単にエネルギー密度を数密度で割り算したもの。

## Appendix A-2: SPEDASで使われる粒子3次元分布関数の共通データ形式



erg\_lepe\_get\_dist() のような関数に  
よって作ることができる

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

LEP-e 3-D flux データの例:

**dists** is 構造体の配列になっていて、配列の要素は、各スピニンごとの分布関数データが格納されている構造体になっている。

"**DATA**" は32 エネルギーチャンネル × 16 スピンセクター × 12 センサーの電子フラックス値配列

**ENERGY** と **DENERGY** は各エネルギーチャンネルの中央値とエネルギー幅の配列

**PHI**, **DPHI**, **THETA**, and **DTHETA** は粒子観測器によって計測された、電子フラックスが流れている方向の中心方位角、方位角幅、中心仰角、仰角幅をそれぞれ格納している配列

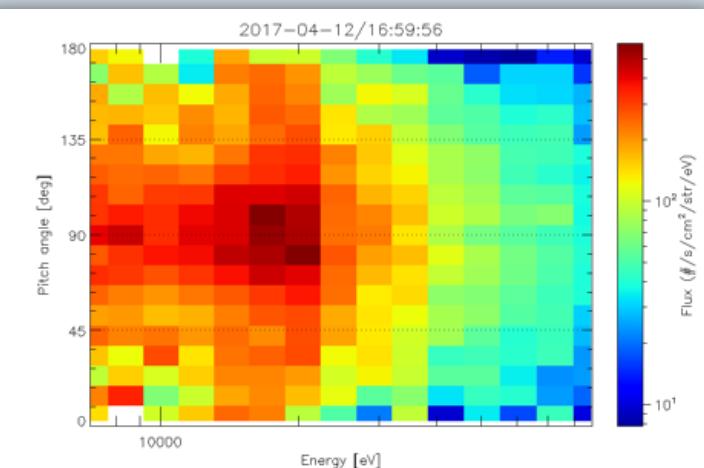


`erg_part_en_pa_spec_plot, dist $`

- , `time=time $` ; プロットを描画する時刻または時刻幅
- , `units=units $` ; 描画する物理量を指定 'flux', 'eflux', 'df\_km', or 'df\_cm'
- , `with_contour=with_contour $` ; 等高線を上書きする
- , `zrange=zrange $` ; カラースケールの上・下限値を指定
- , `npabin=npabin $` ; ピッチ角を何個のbinに分割するか (default: 19)
- , `rslt=rslt $` ; 描画したデータを配列として得たい場合に指定
- , `noplot=noplot` ; プロットを抑制 (↑のrsltだけを得たい場合などに使用)

他の粒子データにも使用できる:

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



# Appendix A-4: LEP-eデータの分布関数の解析



```

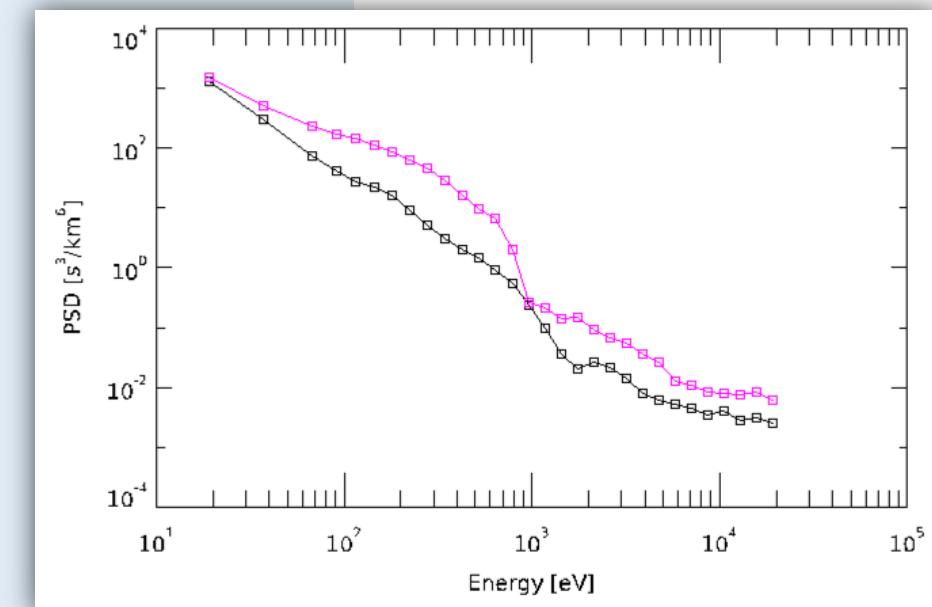
ERG> timespan, '2017-04-12/16:00', 2, /hour
ERG> erg_load_lepe, datatype='3dflux', /no_sort_enebin
ERG> get_timespan, tr
ERG> dists = erg_lepe_get_dist( 'erg_lepe_l2_3dflux_FEDU', trange=tr )
;; ここまでLEP-eデータ読み込み、3D分布データ形式へ変換
;; ピッチ角-エネルギープロットのデータを作成、rsltに保存
ERG> erg_part_en_pa_spec_plot, dists, $
    time=['2017-04-12/17:05:35', '2017-04-12/17:06:08'], $
    /noplott, rsdt=rsdt, units='df_km'
ERG> help, rsdt ;; rsdtのデータ構造を確認
;; rsdtの中の配列からピッチ角30°, 90°のものを選択して線プロット
yid = nn( rsdt.y_pitchangle, 30. ) ;; for PA=30 deg
plot, rsdt.x_energy, rsdt.z_hist[*, yid], /xlog, /ylog, psym=-6, $
    xtitle='Energy [eV]', ytitle='PSD [s!U3!N/km!U6!N]'
yid2 = nn( rsdt.y_pitchangle, 90 ) ;; for PA=90 deg
oplot, rsdt.x_energy, rsdt.z_hist[*, yid2], psym=-6, color=spd_get_color('purple')

```

```

ERG> help, rsdt
** Structure <2191318>, 3 tags, length=2476,
   X_ENERGY      FLOAT      Array[30]
   Y_PITCHANGLE  FLOAT      Array[19]
   Z_HIST        FLOAT      Array[30, 19]
ERG>

```



Kazama+2018の解析を再現

`erg_part_en_pa_spec_plot` で位相空間密度(PSD)のピッチ角-エネルギープロットのデータを作成して、その中からピッチ角30°と90°のエネルギースペクトルをプロット。

# Appendix A-4:

## LEP-eデータの分布関数の解析



```
ERG> erg_part_maxwellian_fit, rslt.x_energy[2:14], rslt.z_hist[2:14, yid2], n90, kbt90  
ERG> df_km = erg_part_get_maxwellian( rslt.x_energy, n90, kbt90 )  
ERG> oplot, rslt.x_energy, df_km, color=spd_get_color('purple'), $  
    linestyle=2, thick=2.5  
ERG> print, 'n = ', n90, ' [/cc]', ' kBT = ', kbt90, ' [eV] '
```

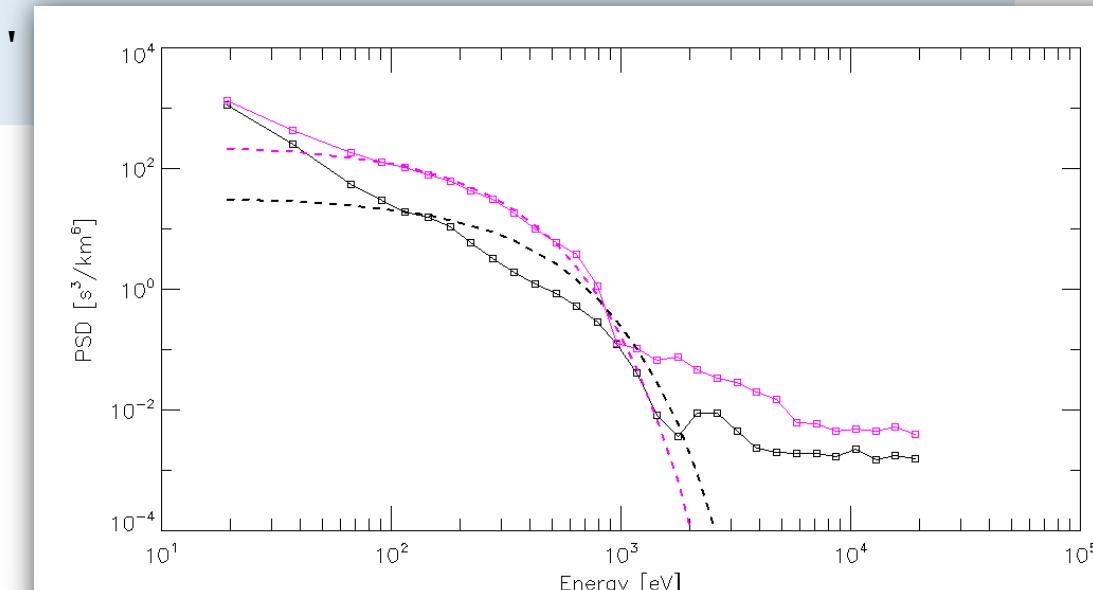
さらに67–963 eVの範囲の  
PSDにMaxwell分布をfitting  
して、その結果を重ねプロット

```
ERG> erg_part_maxwellian_fit, rslt.x_energy[2:14], rslt.z_hist[2:14, yid], n30, kbt30  
ERG> df_km = erg_part_get_maxwellian( rslt.x_energy, n30, kbt30 )  
ERG> oplot, rslt.x_energy, df_km, color=spd_get_color('black'), $  
    linestyle=2, thick=2.5  
ERG> print, 'n = ', n30, ' [/cc]', ' kBT = ', kbt30, ' [eV] '
```

```
ERG> print, 'n = ', n90, ' [/cc]', ' kBT = ', kbt90, ' [eV] '  
n = 0.46398556 [/cc] kBT = 137.79072 [eV]
```

```
ERG>  
ERG> print, 'n = ', n30, ' [/cc]', ' kBT = ', kbt30, ' [eV] '  
n = 0.11406248 [/cc] kBT = 203.69873 [eV]
```

```
ERG> print, rslt.x_energy[ [2,14] ]  
67.0372 963.422  
ERG>
```

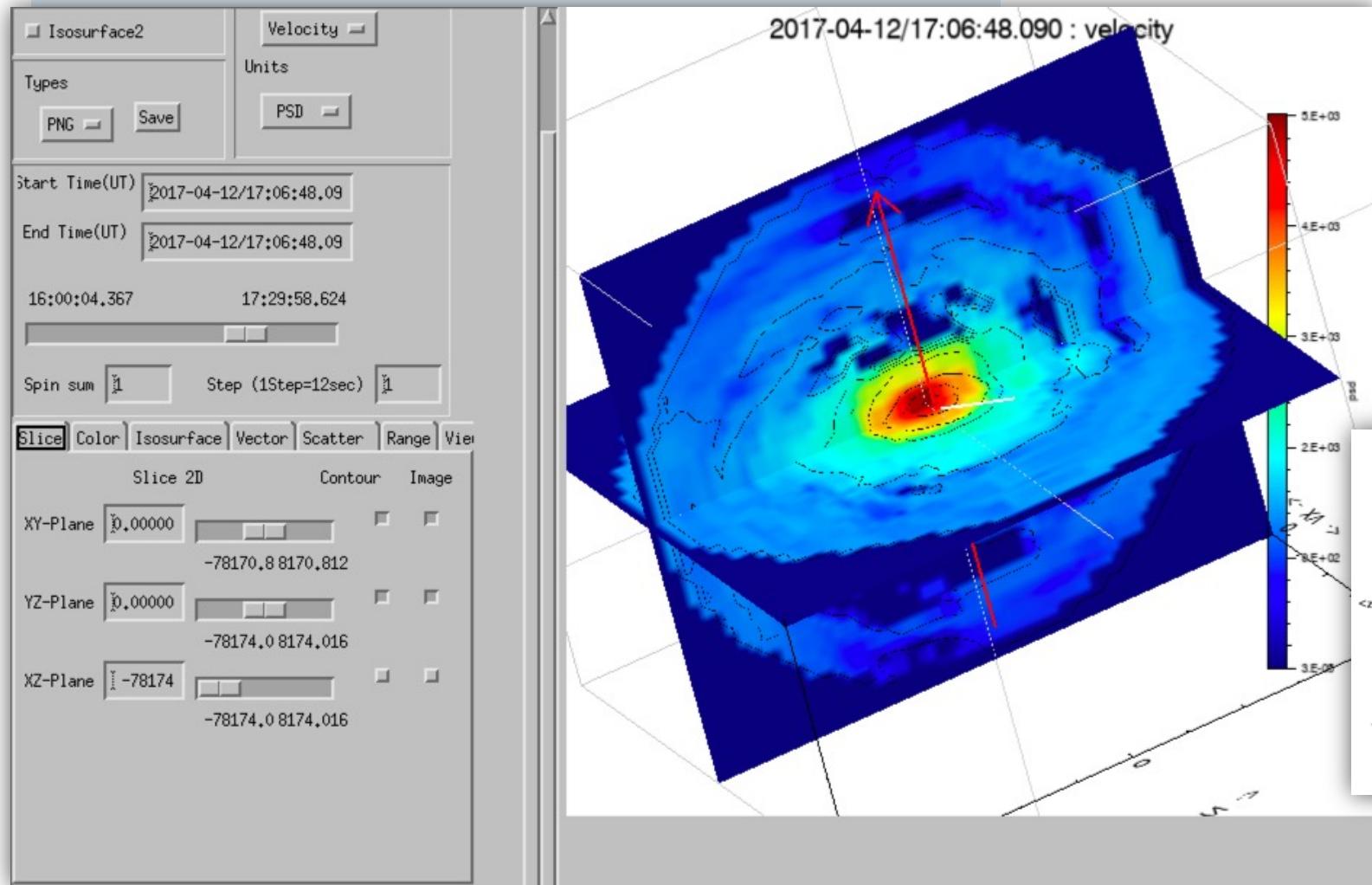


Kazama+2018の解析を再現

# Appendix A-5: ISEE3D 3次元速度分布関数可視化ツール



```
ERG> erg_crib_lepe_isee3d, $  
      trange='2017-04-12/'+'[16:00','17:30']
```



このツールの詳細は Keika et al. [EPS, 2018]に記述されている。

左上のスクリプトはLEP-eデータ用だが、簡単に LEP-i用に書き換えることができる。

# Appendix A-5: ISEE3D 起動画面

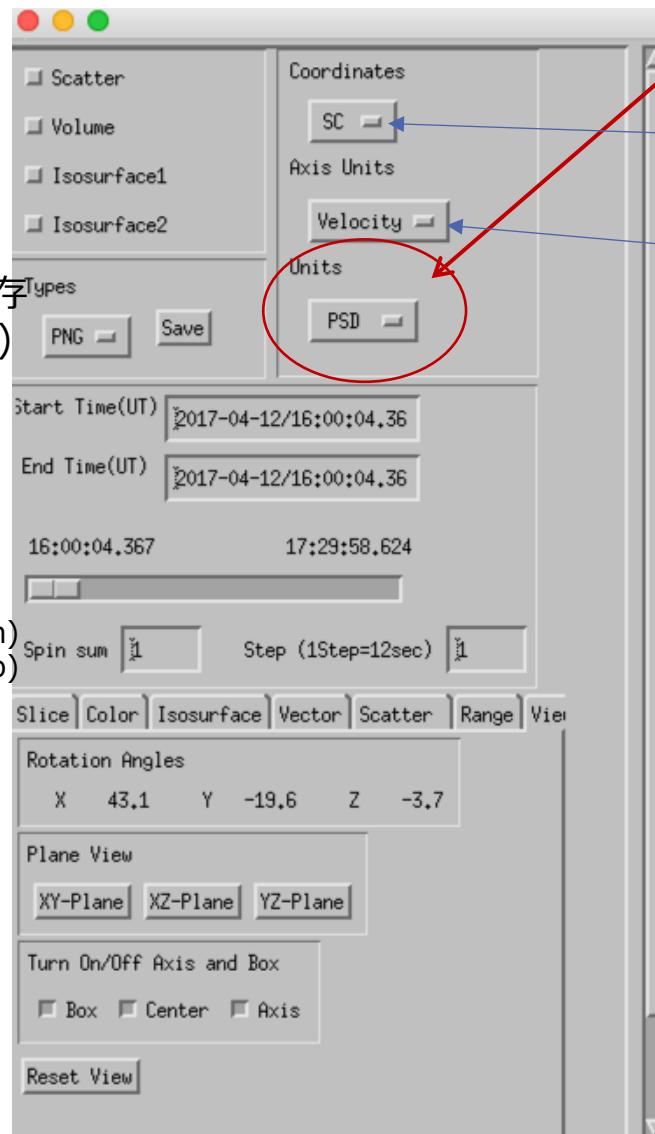
散布図、ボリュームレンダリング、等値面の表示  
On/Off

図のファイル保存  
(png, ps, ...)

プロット日時範囲

時刻選択バー

何スピン分平均するか (Spin sum)  
何スピン分ずつ時刻を進めるか (Step)



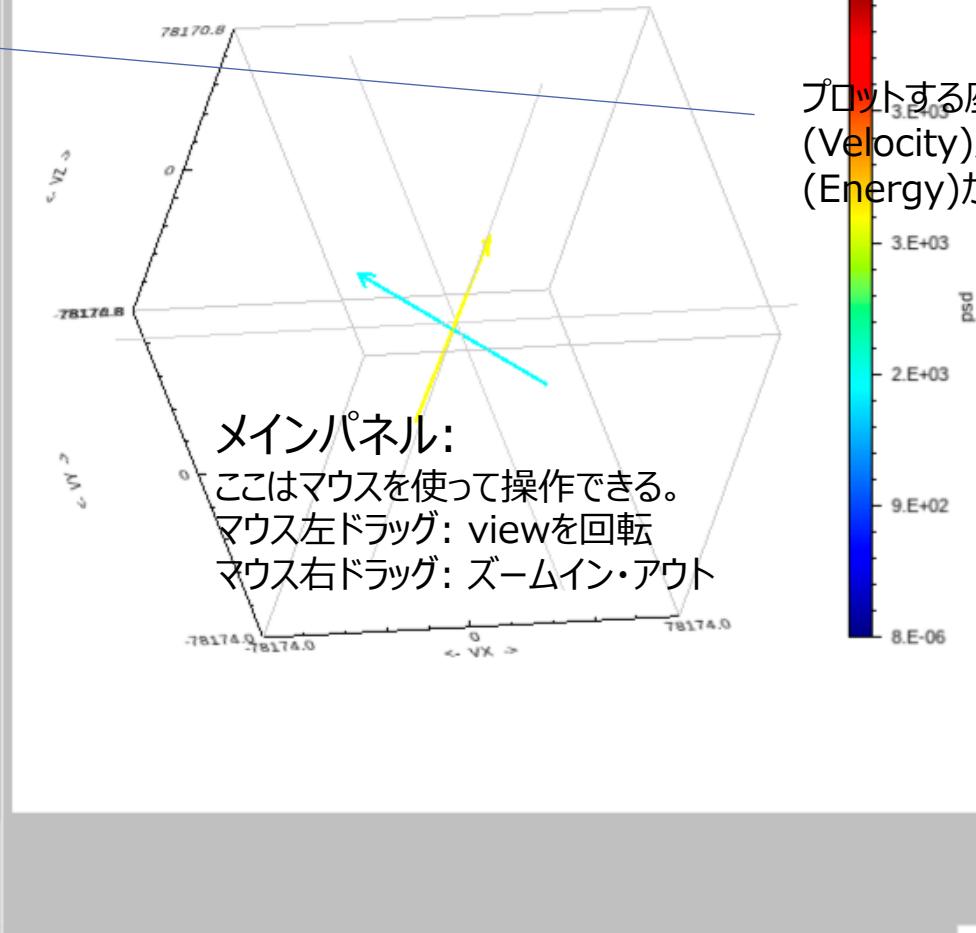
注意！

ここを必ず"PSD"にしておくこと。countだとエラーで動かない！

2017-04-12/16:00:04.367 - 17:29:58.624 (velocity)

座標系をDSI座標(SC)か磁場  
座標(Mag)から選択

プロットする座標の単位を速度  
(Velocity)かエネルギー  
(Energy)から選択



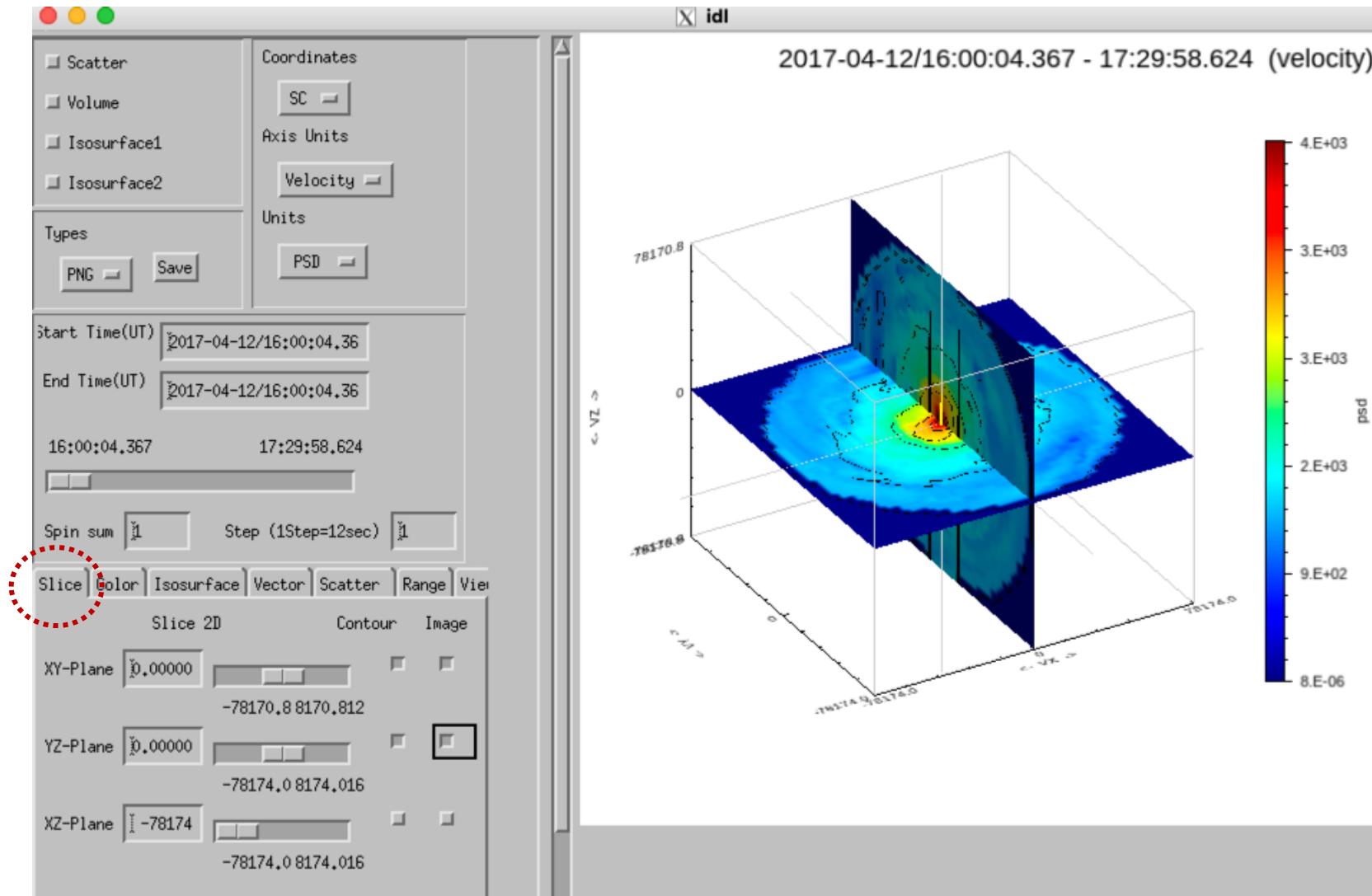
# Appendix A-5: ISEE3D Sliceペイン

恐らく多くのユーザーが最初に分布関数を見る時に使う機能。

XY面、YZ面、XZ面の分布関数の断面を表示できる。

スライドバーで断面の場所を変更可能。

Imageをチェック  
→ 断面をピクセル描画  
Contourをチェック  
→ 等高線を重ねて描画

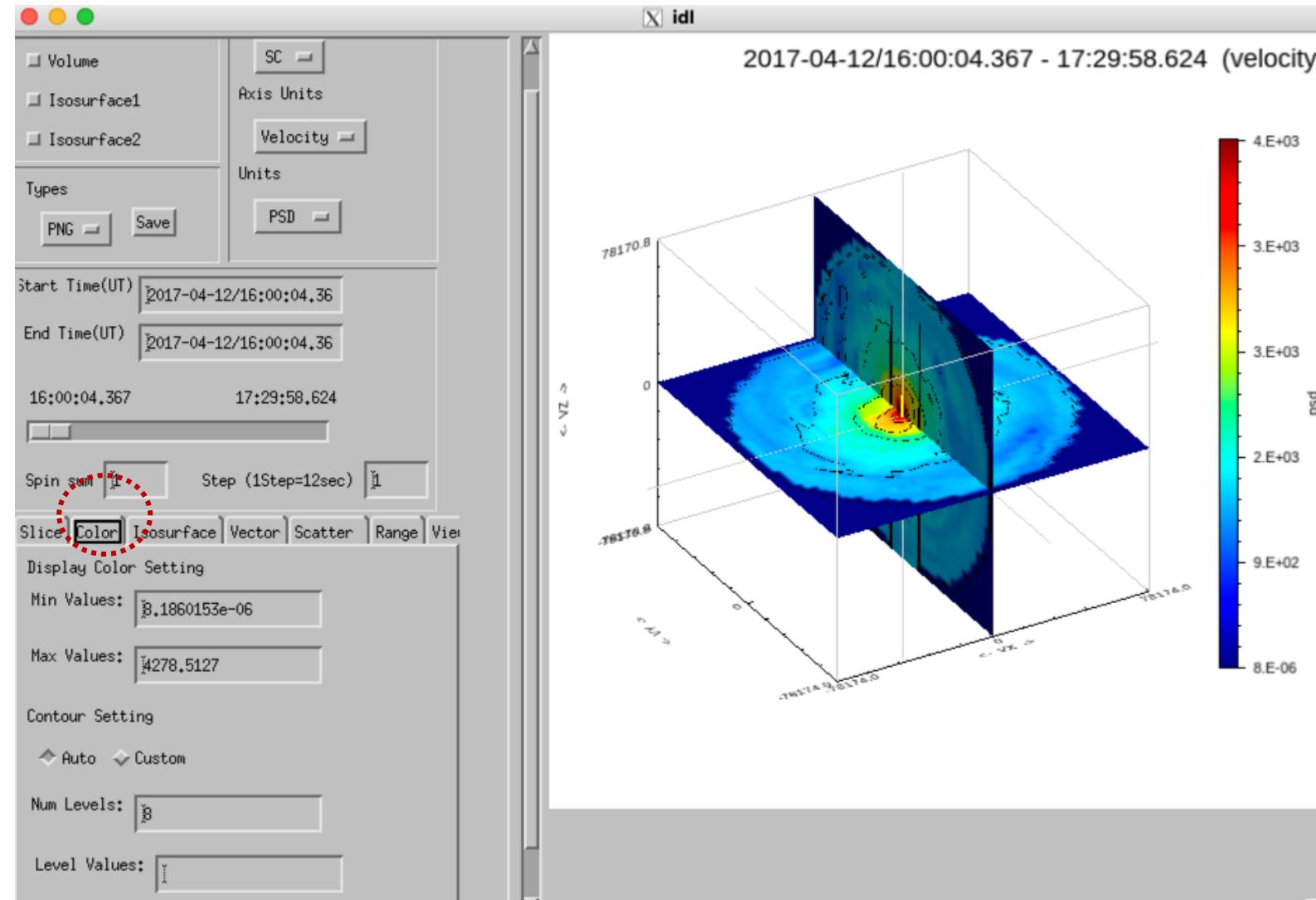


# Appendix A-5: ISEE3D Colorペイン

Min Values, Max Valuesで、メインパネルのカラースケールの上限値と下限値を手動で変更できる。

Contour Settingのところは、Auto をチェックすると自動で適当な間隔の等高線を引いてくれる。Num Levelsの値を変えると等高線の数を増減できる。

Customを選択してLevel Valuesを指定すると任意のレベルの等高線を描けるはずだが、うまく機能しない？

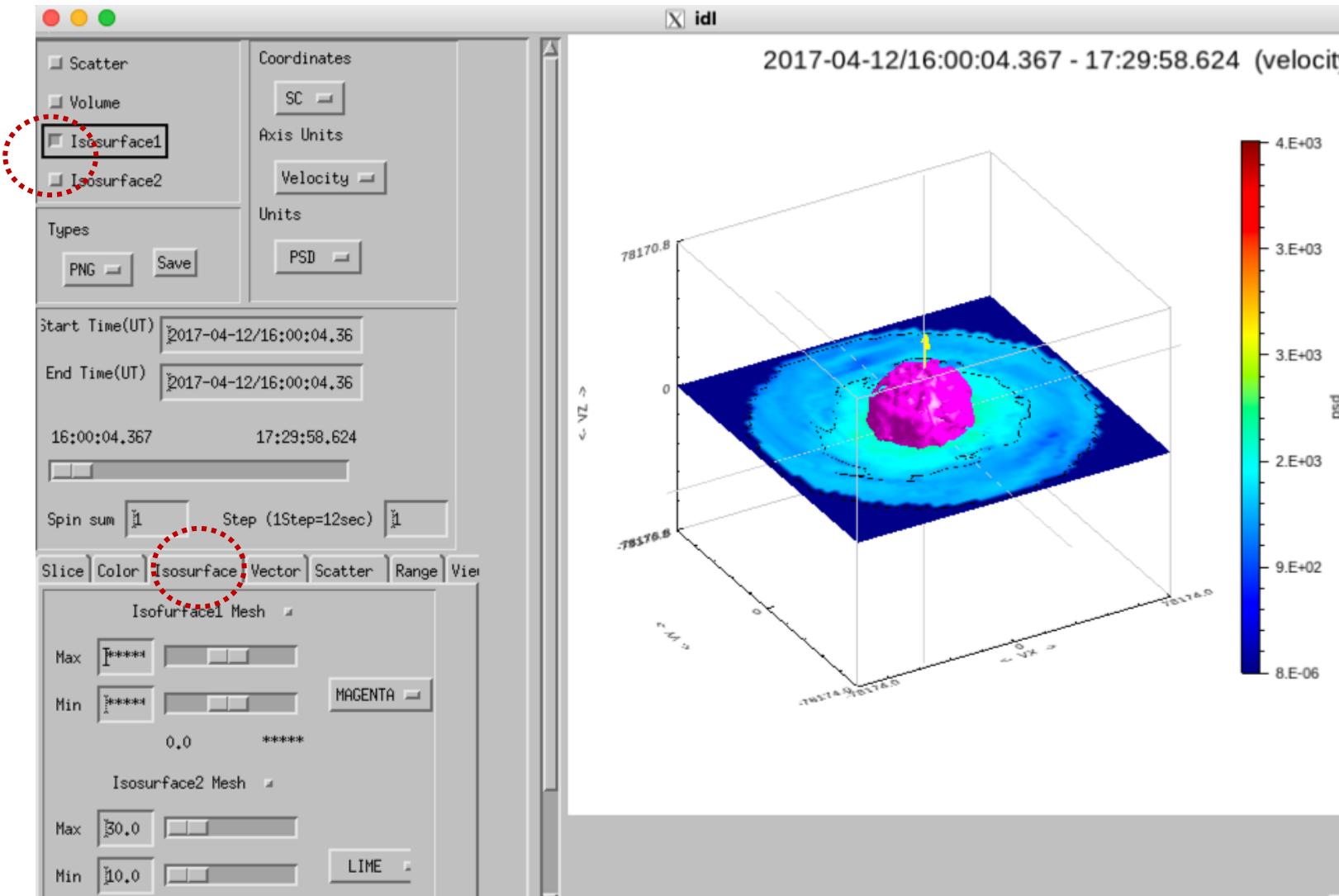


# Appendix A-5: ISEE3D Isosurfaceペイン

等値面(isosurface)を最大2つ(Isosurface1, Isosurface2)描画できる。実際に表示させるには、左上のチェックボタンを押す必要がある。

それぞれ、Max, Minの値を左の枠にタイプして入力する。表示行数が不足するバグで\*\*\*\*\*と表示されてしまうが、値はちゃんと入力されている。

右側のMAGENTA, LIMEは色を選択できるプルダウンメニューになっている。

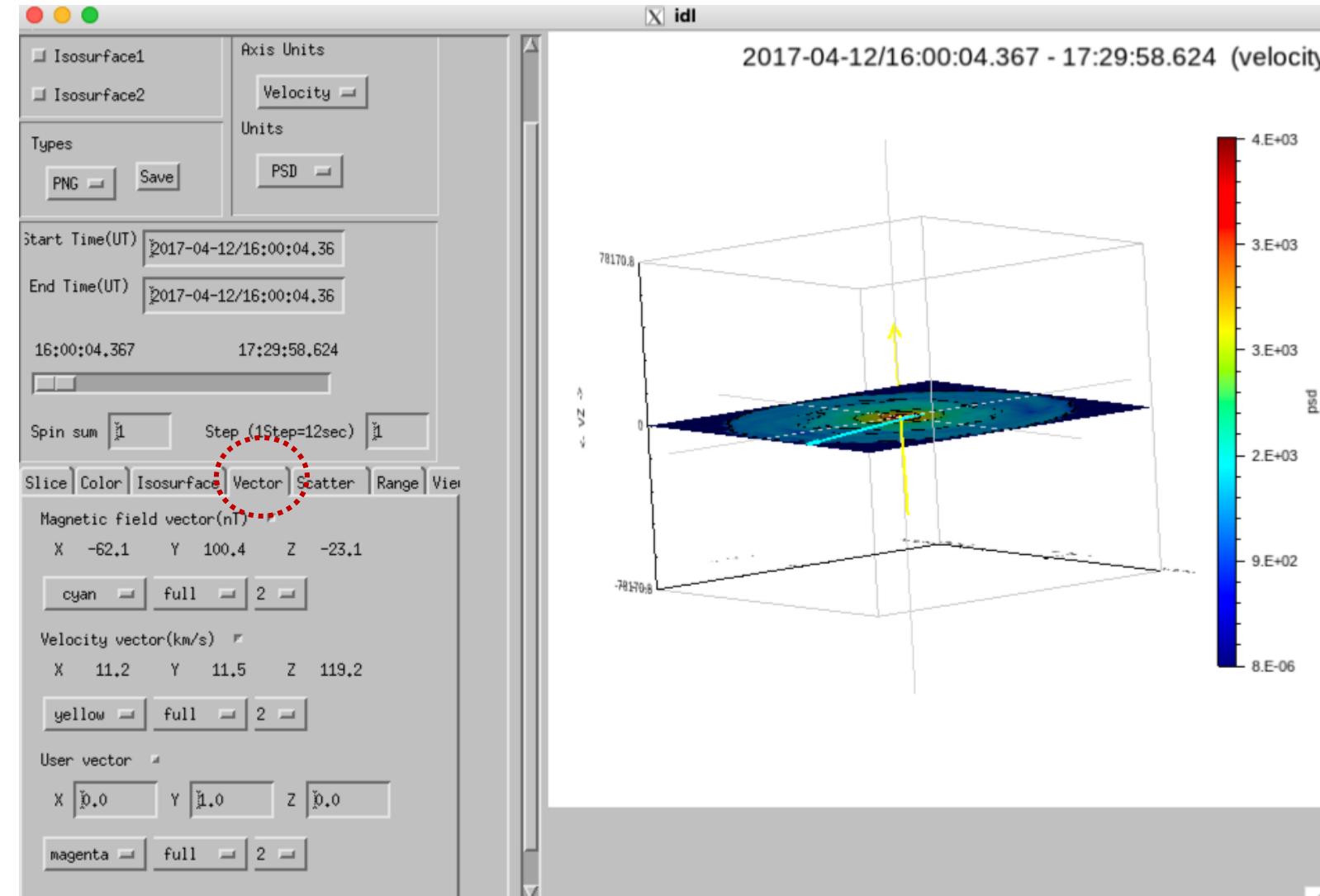


# Appendix A-5: ISEE3D Vectorペイン

メインパネルの中に、磁場ベクトル、バルク速度ベクトル、及び任意のベクトルを矢印として表示できる。

Magnetic field vector(nT) のすぐ右にある小さいボタンを押すと表示、戻すと非表示にできる。また、色、長さ(full, half, quarterから選択)、ベクトルの太さを数字で指定できる。

任意のベクトルは一番したのUser vectorとして指定する。

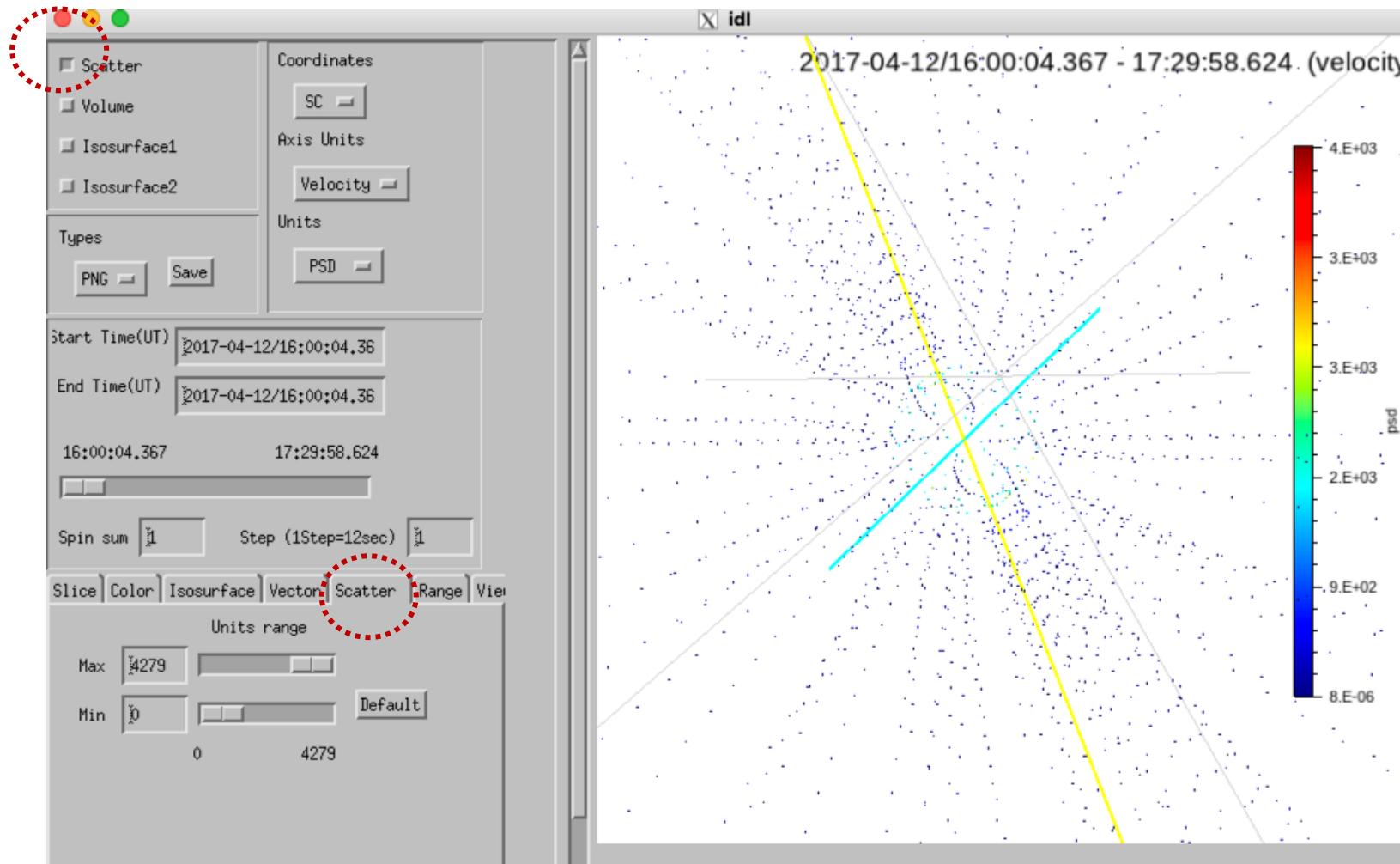


# Appendix A-5: ISEE3D Scatterペイン

左上のScatterボタンを押すと、全データ点を点として表示できる。

Max, Minを指定することで、位相空間密度がその値の範囲に入った点だけを選択表示できる。

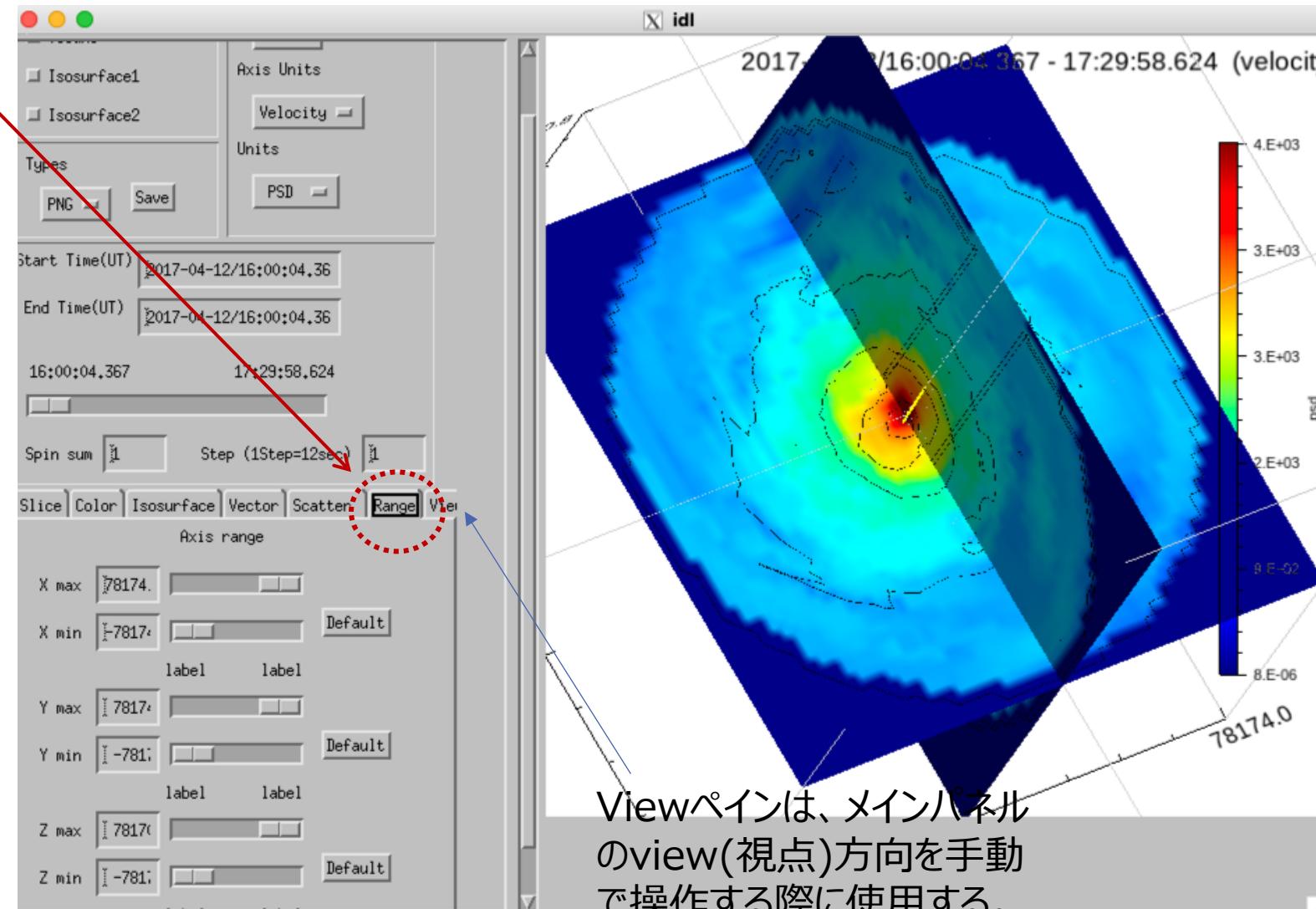
ユースケースは少ないが、速度空間の中のある領域に本当に実測値があるかどうかを確認する際に使うことがある。



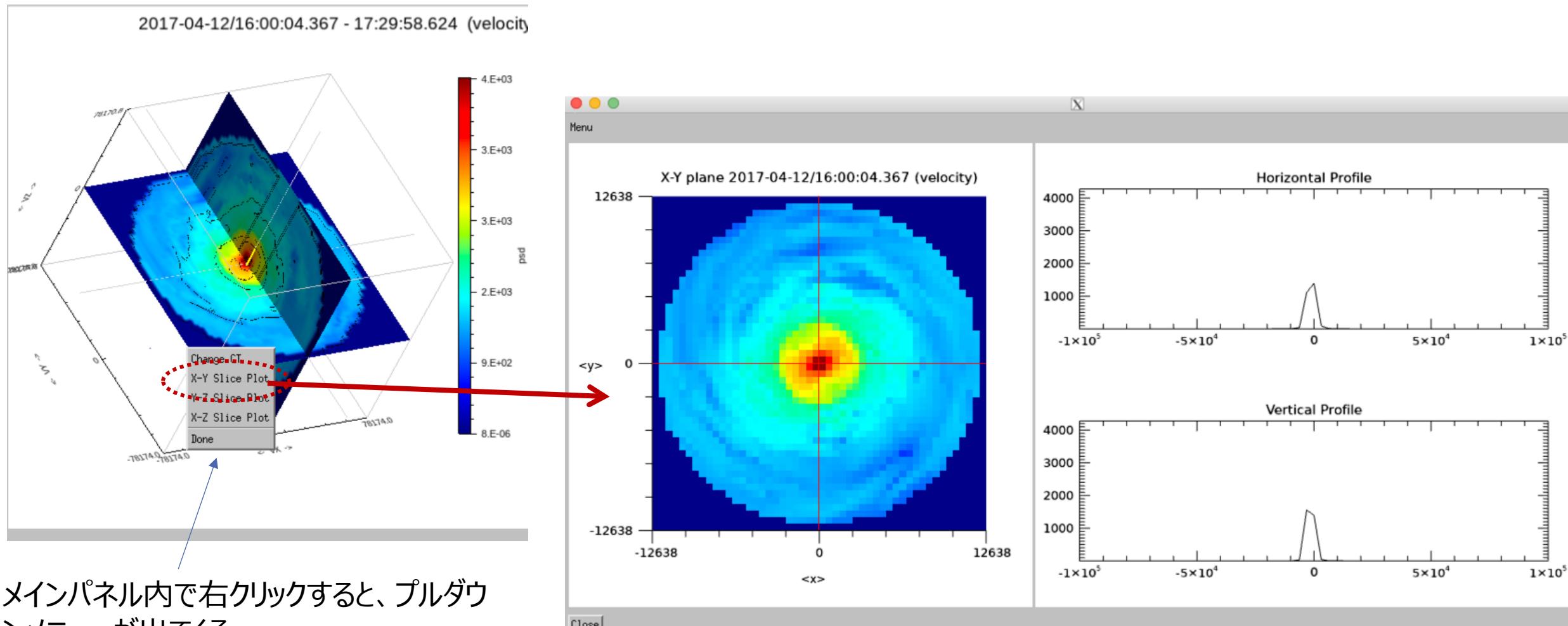
# Appendix A-5: ISEE3D Rangeペイン

X軸、Y軸、Z軸のプロット範囲を明示的に指定できる。  
それぞれの軸について、  
max, min値を指定する。

Defaultボタンを押すと、最初の設定に戻る。



# Appendix A-5: ISEE3D XY, YZ, XZ平面内の1次元カット



左のカラーの図の中にマウスポインタを入れると、位相空間密度のX軸カット  
およびY軸カットを右側の折れ線グラフに表示する。



# 演習問題

## 問題 1:

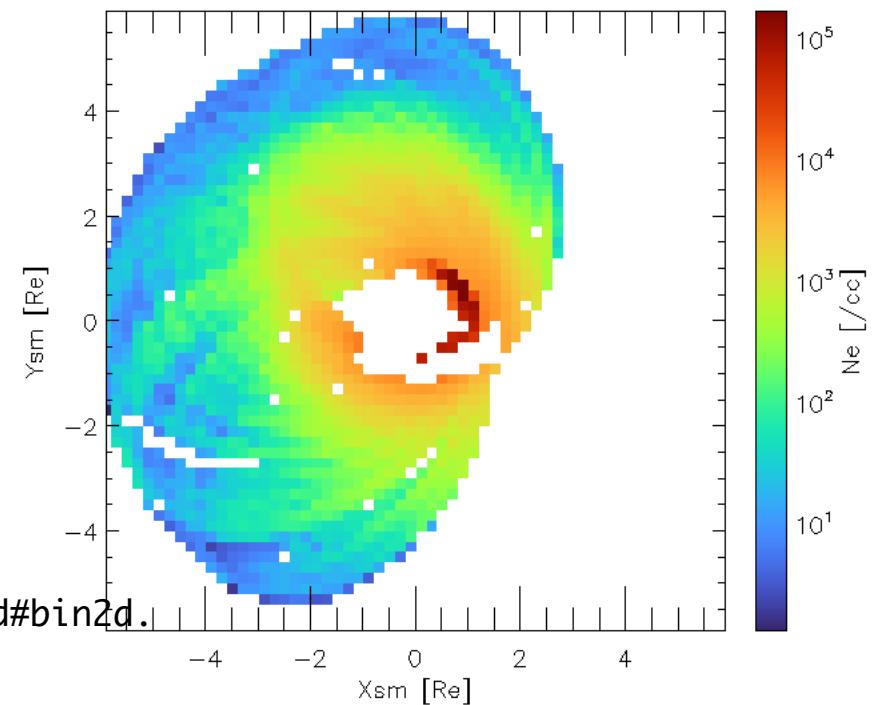
2017-04-01 ~ 2017-10-08 (200日分)のPWE/HFAのLv.3  
電子密度データを解析して、平均電子密度をSM X-Y面上に描画せ  
よ。その際に1つのピクセルは $0.5 \text{ Re} \times 0.5 \text{ Re}$ とし、各ピクセル内の  
データを平均せよ。

使用するルーチン:

- erg\_load\_pwe\_hfa, level='l3'
- erg\_load\_orb
- bin2d
- plotxyz

bin2d, plotxyz コマンドの情報はここにある

[https://github.com/spedas-j/member\\_contrib/wiki/spedas\\_useful\\_command#bin2d](https://github.com/spedas-j/member_contrib/wiki/spedas_useful_command#bin2d).



# 問題1の解答例



```
dens = ''
xsm = '' & ysm = ''

;; Loop for reading data for each day
for i=0, 200 do begin    ;; 200 days
  timespan, time_double('2017-04-01') + i *
86400L
  del_data, 'erg_*
  erg_load_pwe_hfa, level='l3'
  erg_load_orb
  if tnames('erg_pwe_hfa_l3_1min_ne_mgf') eq
'' or tnames('erg_orb_l2_pos_sm') eq '' then
continue

  get_data, 'erg_pwe_hfa_l3_1min_ne_mgf',
data=d_ne
  tinterpol_mx, 'erg_orb_l2_pos_sm',
'erg_pwe_hfa_l3_1min_ne_mgf'

  get_data, 'erg_orb_l2_pos_sm_interp',
data=d_pos

  append_array, dens, d_ne.y
  append_array, xsm, d_pos.y[* ,0]
  append_array, ysm, d_pos.y[* ,1]
endfor

bin2d, xsm, ysm, dens, xrange=[-6,6],
yrange=[-6,6], binsize=[0.2, 0.2], xc=xc,
yc=yc, ave=ave, binhist=num

plotxyz, xc, yc, ave, /zlog, xtitle='Xsm [Re]',
ytitle='Ysm [Re]', ztitle='Ne [/cc]'

end
```