



CUIの使い方（後編）：calcコマンド、get\_dataや  
store\_dataの使い方、時系列データのフィルター  
処理、スペクトル解析方法

堀 智昭 (名古屋大STE研)

- UDAS環境をリセットしてください

```
THEMIS> .full_reset_session
```

IDL環境を初期化

先頭のピリオドを忘れずに！

```
IDL> thm_init
```

UDAS環境をセットアップ

```
THEMIS>
```

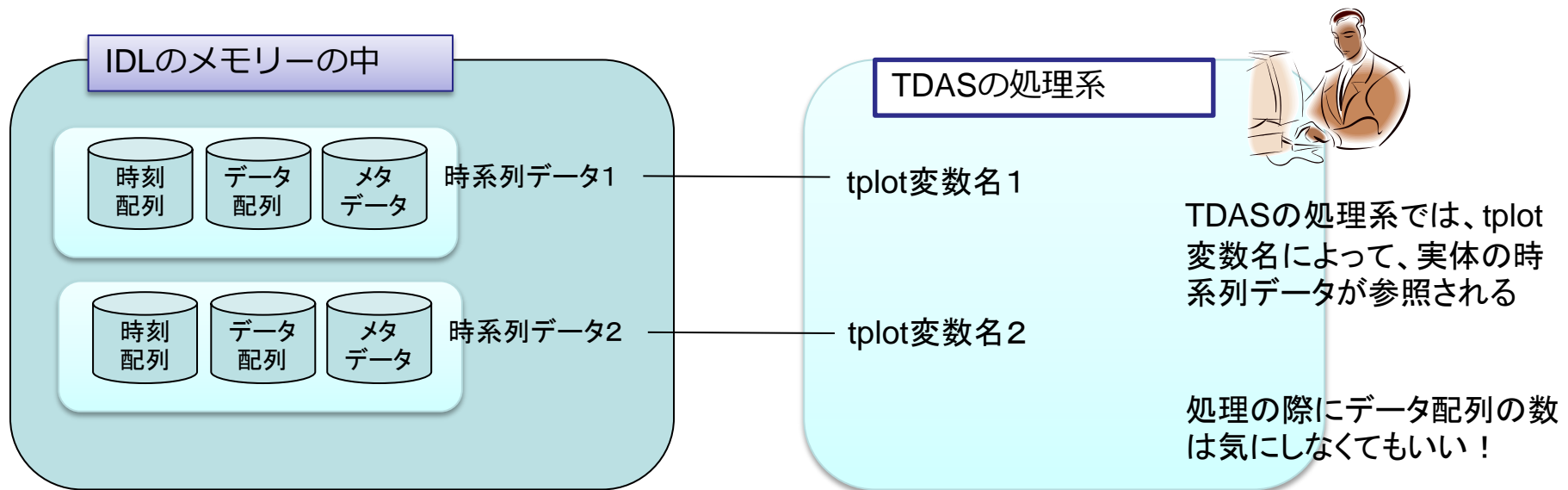
再びプロンプトがTHEMIS> になる

- 入門編・CUIの使い方（前編）では、データのロード、プロットの基礎を行った。
- CUIの使い方（後編）では...
  - UDAS上での汎用データ形式である "tplot変数" の中身について理解し、各自の手持ちのデータから独自の tplot変数 を生成する方法を学ぶ。
  - 非常に便利なtplot変数を使った演算について学ぶ。
  - スムージング、バンドパスフィルター、周波数スペクトル導出など、よく用いられる時系列解析のやり方を覚える。

GUIよりCUI(コマンドラインでの操作)の方が自由度が高く、UDASに慣れてくるとコマンドを使う方が断然便利！

# tplot変数の取り扱いと演算

- UDASのベースになっているTDAS (THEMIS Data Analysis Software)での、汎用時系列データ形式。
- IDL上では単なる文字列だが、tplot等のいわゆるtコマンドに与えると、tplot変数名に紐付けられた時系列データの実体に対して、コマンド処理が実行される。



`get_data, 'tplot変数名', data = d, dlimits = dl, lim = lim`

'tplot変数名' のところはインデックス番号でもよい。その場合はシングルクォーテーションは不要

メタデータが入る

データ配列が入る

主に可視化情報が入る

```
THEMIS> timespan, '2007-06-01', 3          時間幅として2007年6月1日から3日分を指定
THEMIS> iug_load_gmag_mm210, site = 'rik'  210MM地磁気チェーンのRikubetsu (RIK) データをロード
THEMIS> get_data, 'mm210_mag_rik_1min_hdz', data = d,
        dlimits = dl, lim = lim
THEMIS> help, d, /struct  help コマンドは変数・構造体の情報を表示する。/struct キーワードを
                             付けると、構造体内の配列情報を表示する。

** Structure <5bd10a0>, 2 tags, length=138240, data length=138240, refs=1:
X      DOUBLE   Array[4320]
Y      DOUBLE   Array[4320, 3]
```

```
THEMIS> help, d, /struct
```

```
** Structure <5bd10a0>, 2 tags, length=138240, data length=138240, refs=1:  
X          DOUBLE  Array[4320]  
Y          DOUBLE  Array[4320, 3]
```

tplot変数の実体のデータ構造体 (今の場合は d ) は  
X, Y という2つのメンバーを持っている。

X: 倍精度浮動小数点で表したUnix time (1970年1月1日0時0分0秒UTからの積算秒数)  
この例では 4320個の1次元配列。つまりデータのtime frame は4320個ある。  
このデータは1分値で3日分なので、1日=1440分 x 3 日分 で 4320。

Y: 実際にデータが入っている配列  
この場合、4320 x 3 の2次元配列。ちなみに第2次元の3 は地磁気H, D, Z の3成分。

THEMIS> **help**, dl, /struct

\*\* Structure <5b83c18>, 4 tags, length=952, data length=950, refs=2:

```
CDF          STRUCT  -> <Anonymous> Array[1]
SPEC         BYTE    0
LOG          BYTE    0
YSUBTITLE   STRING  '[nT]'
```

Dlimits構造体にはメタデータ(データに関する各種情報)が格納される。

例えば CDF はこれ自体も構造体であり、元データファイルであるCDFファイルの情報(ファイルのセーブ場所など)が格納されている。

もっと他にも入ることがある

THEMIS> **help**, lim, /struct

\*\* Structure <5b6b178>, 3 tags, length=48, data length=44, refs=2:

```
COLORS      INT      Array[3]
LABELS      STRING   Array[3]
LABFLAG     INT      1
```

lim 構造体の方には主にプロット等に可視化する際に必要な情報が入っている。

例えば tplot コマンドがtplot変数をプロットする場合、この情報を参照して、線の色や縦軸のラベル、凡例 等を描画する。



```
store_data, 'tplot変数名', data = {x:time, y:data }
```

time: データの時刻ラベルを倍精度浮動小数点のUnix time の配列にしたもの  
1次元配列 [N] N: 時刻ラベル数

val: データの配列。スカラーデータの場合は [N] (timeと同じサイズ)、1次元ベクトル  
データの場合は [N][ J ] (J がベクトルの成分数) という配列。

というような time, val を用意すればtplot変数を作成できる。

(前ページの続きで)

```
THEMIS> time = d.x
```

```
THEMIS> val = d.y / 2.0
```

```
THEMIS> store_data, 'mm210_mag_rik_1min_hdz_half',  
data = { x:time, y:val }
```

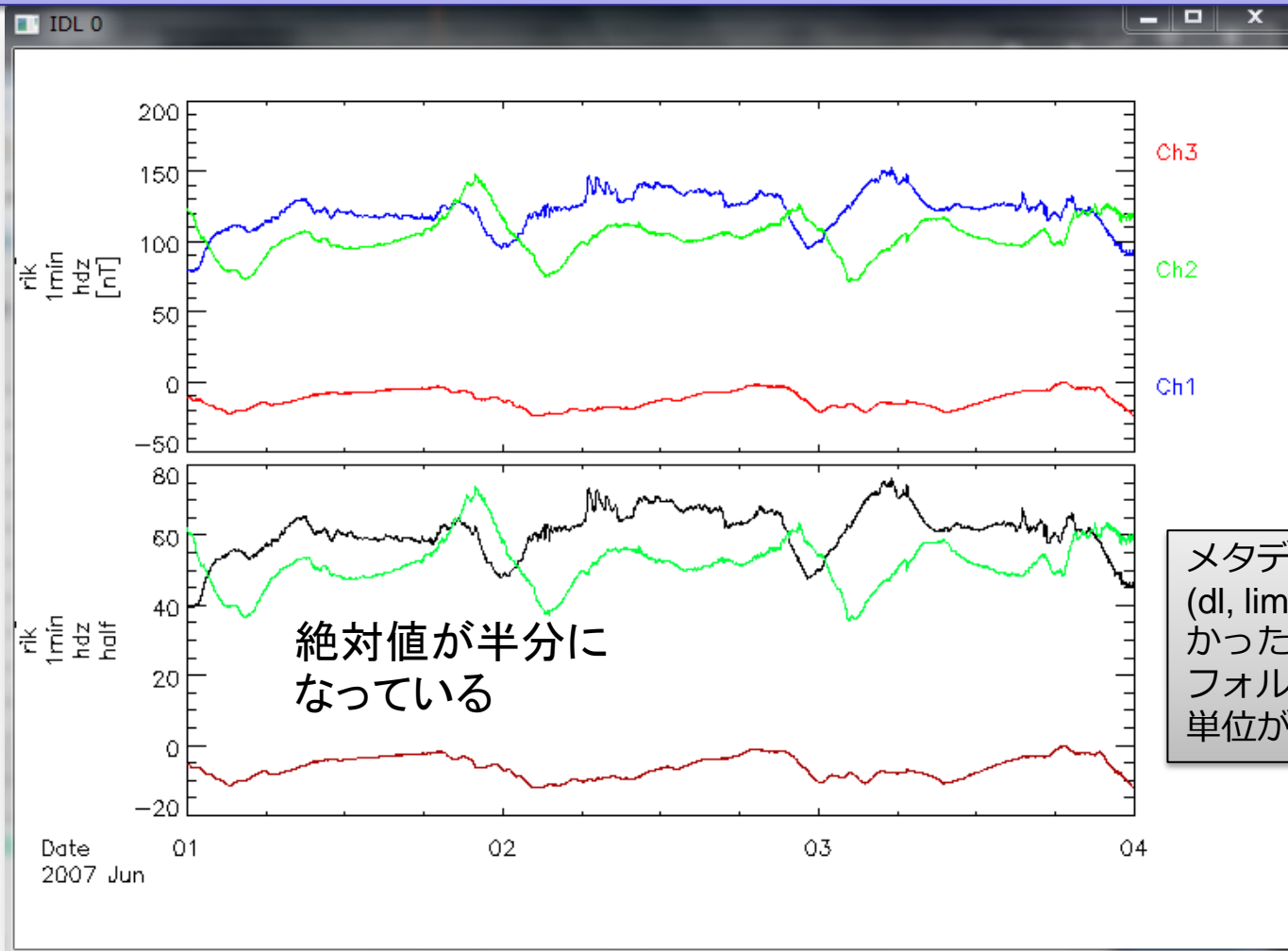
```
Creating tplot variable: 3 mm210_mag_rik_1min_hdz_half
```

```
THEMIS> tplot, ['mm210_mag_rik_1min_hdz',  
                'mm210_mag_rik_1min_hdz_half' ]
```

実際にtplotでプロットして確認してみる

# store\_dataで新規tplot変数を作成

```
THEMIS> tplot, ['mm210_mag_rik_1min_hdz',
                'mm210_mag_rik_1min_hdz_half']
```



メタデータ・可視化情報 (dl, lim) が受け継がれなかったため、線の色がデフォルトになり、凡例や単位が表示されていない

```
calc, ' "新tplot変数名" = ... 計算式 ... '
```

```
(例) calc, ' "newvar" = "mm210_mag_rik_1min_hdz" + 40. '
```

時系列データであるtplot変数全体を使った演算を、直感的にわかり易い形で書いて実行することができる！

実は、前頁のstore\_data を使ってやったことは、

```
calc, ' "mm210_mag_rik_1min_hdz_half" =  
      "mm210_mag_rik_1min_hdz" / 2.0 '
```

と、わずか1行で実行できる！

```
calc, ' "新tplot変数名" = ... 計算式 ... '
```

```
(例) calc, ' "newvar" = "mm210_mag_rik_1min_hdz" + 40. '
```

## 計算式のルール

- フォーマットは普通の計算式と同じ。全体を単引用符(')で囲む。tplot変数は二重引用符(")で囲む。
- 使用可能な演算: 四則(+-\* /), べき乗, sin/cos/tan(), exp(), log(), abs(), min(), max(), total(), mean(), median(), ...

## 注意点

- 複数のtplot変数を演算に使う場合、実体の配列のサイズ・次元が同一でないといけない。データの時刻数が異なる、データの次元が異なる(スカラーデータとベクトルデータの混在など)とエラーになる。

```
THEMIS> split_vec, 'mm210_mag_rik_1min_hdz' HDZ3成分を別々のtplot変数に分解する
```

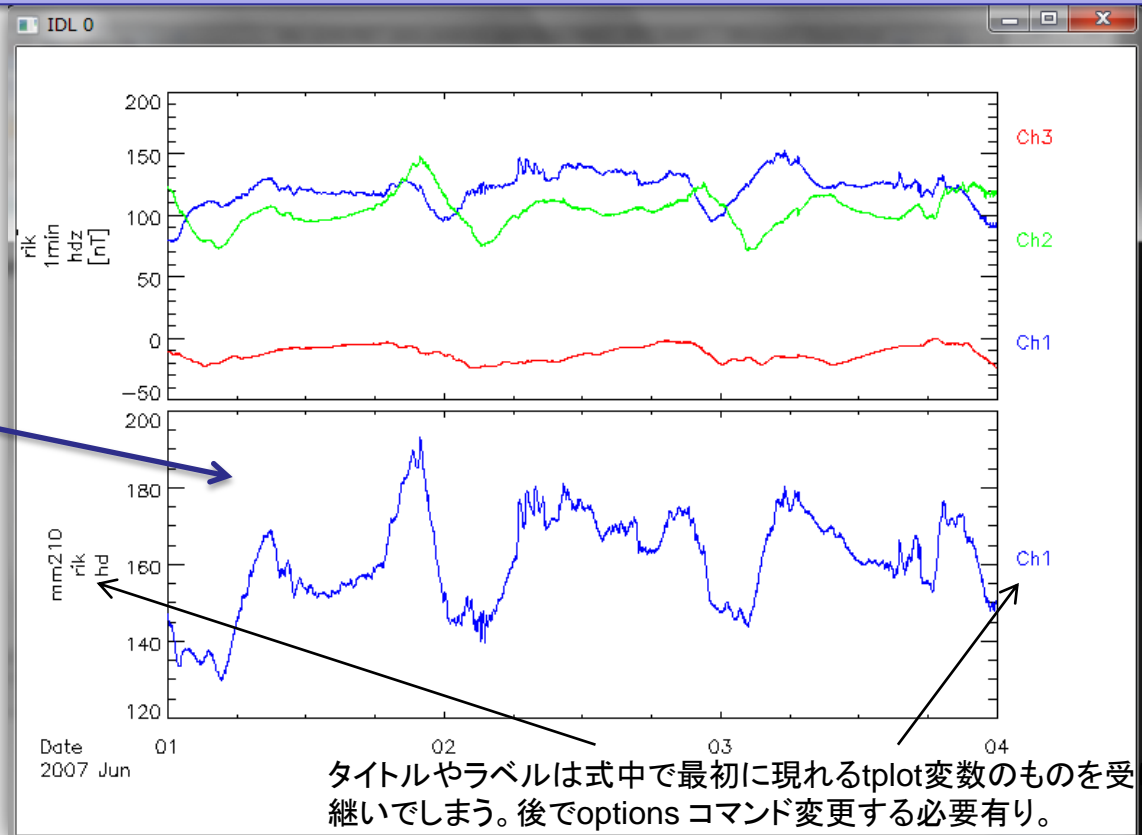
```
THEMIS> calc, ' "mm210_rik_hd" = sqrt( "mm210_mag_rik_1min_hdz_x"^2 +  
"mm210_mag_rik_1min_hdz_y"^2 ) '
```

```
THEMIS> tplot, [ 'mm210_mag_rik_1min_hdz', 'mm210_rik_hd' ]
```

- IDLでは^はべき乗演算の意味。

$$\sqrt{x^2 + y^2}$$

- 地磁気3成分のうち  
の水平2成分(H成分, D成分)の2乗和平方根をcalcで求めた。



[Robinson+, JGR, 1987]

オーロラ降下粒子観測から電離圏電気伝導度を導出

$$\text{calc, ' "sigmaP" = 40.* "prep_avgE" * \sqrt{"prep_itgEf"} / (16. + "prep_avgE"^2)'}$$

$$\Sigma_P = \frac{40E}{16 + E^2} \Phi_E^{1/2}$$

prep\_avgE: 降下粒子の平均エネルギー (E) [keV]、 prep\_itgEf: 積分エネルギーフラックス ( $\Phi_E$ ) [mW/m<sup>2</sup>]

太陽風観測から太陽風動圧を導出

$$\text{calc, ' "Pdyn" = "ace_Np" * "ace_Vp"^2 * 1.6726 * 1e-6 '$$

ace\_Np: 太陽風密度 [/cc]、 ace\_Vp: 太陽風速度 [km/s]

陽子質量の小数部分

$$P_{dynamic} = N_p M V_p^2$$

2つ目の例のace\_Np, ace\_Vp というデータは、TDASに収録されている  
ace\_swe\_load, datatype='h0' というコマンドでロードできる。

# tplot変数への各種フィルター処理

## tsub\_average, 'tplot変数名'

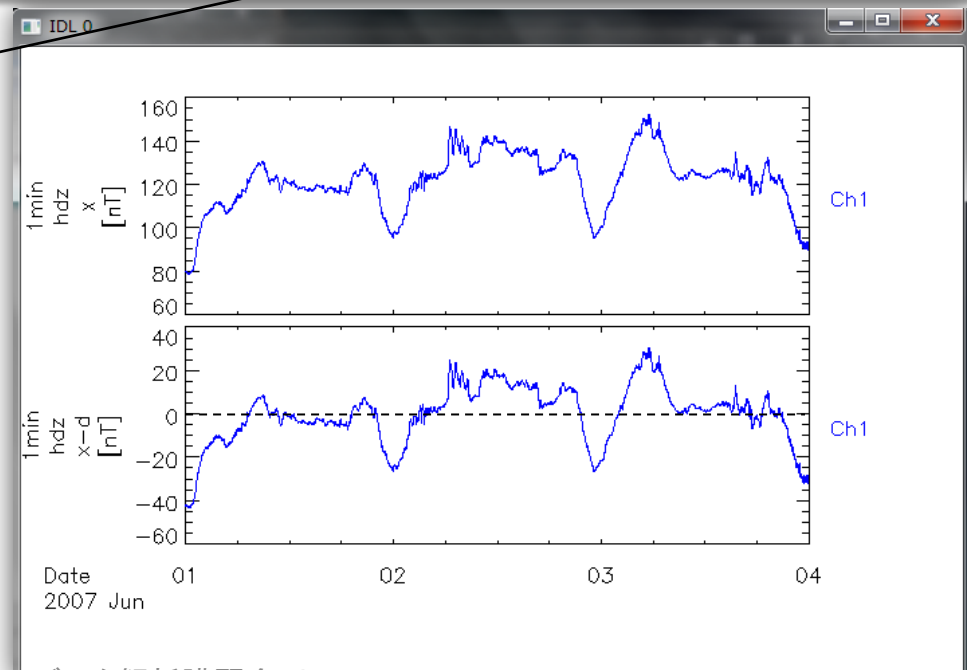
(例) tsub\_average, 'mm210\_mag\_rik\_1min\_hdz'

```
THEMIS> tsub_average, 'mm210_mag_rik_1min_hdz_x'
```

```
STORE_DATA(221): Creating tplot variable: 8 mm210_mag_rik_1min_hdz_x-d
```

```
THEMIS> tplot, [ 'mm210_mag_rik_1min_hdz_x', 'mm210_mag_rik_1min_hdz_x-d' ]
```

- 元の変数名に **-d** を付けた新しいtplot変数に結果が格納される。
- プロットする際にゼロ線を揃えたり周波数解析の前処理などで多用される。





## tsmooth\_in\_time, 'tplot変数名', 平均幅[秒]

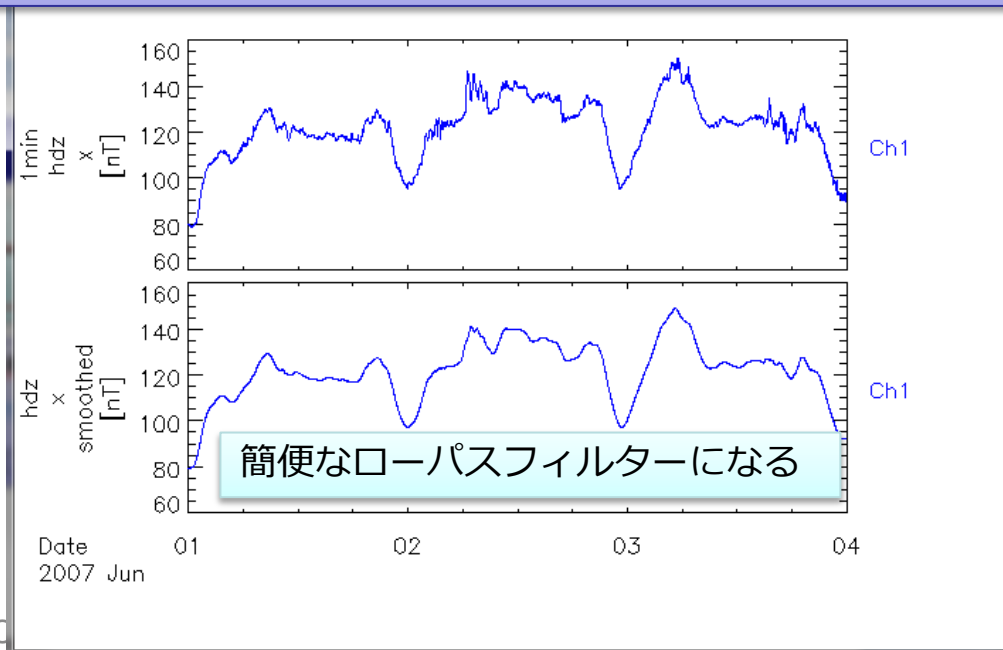
(例) tsmooth\_in\_time, 'mm210\_mag\_rik\_1min\_hdz', 3600.

```
THEMIS> tsmooth_in_time, 'mm210_mag_rik_1min_hdz_x', 3600.
```

```
STORE_DATA(...): Creating tplot variable: 11 mm210_mag_rik_1min_hdz_x_smoothed
```

```
THEMIS> tplot, [ 'mm210_mag_rik_1min_hdz_x',  
                 'mm210_mag_rik_1min_hdz_x_smoothed' ]
```

- 指定された**時間幅で移動平均**することでスムージングされた結果が...**\_smoothed** という名前の新しいtplot変数に格納される。
- **平均幅を秒数で与える**点に注意。  
上の例は3600秒=1時間幅で移動平均している。



## thigh\_pass\_filter, 'tplot変数名', 下限周期[秒]

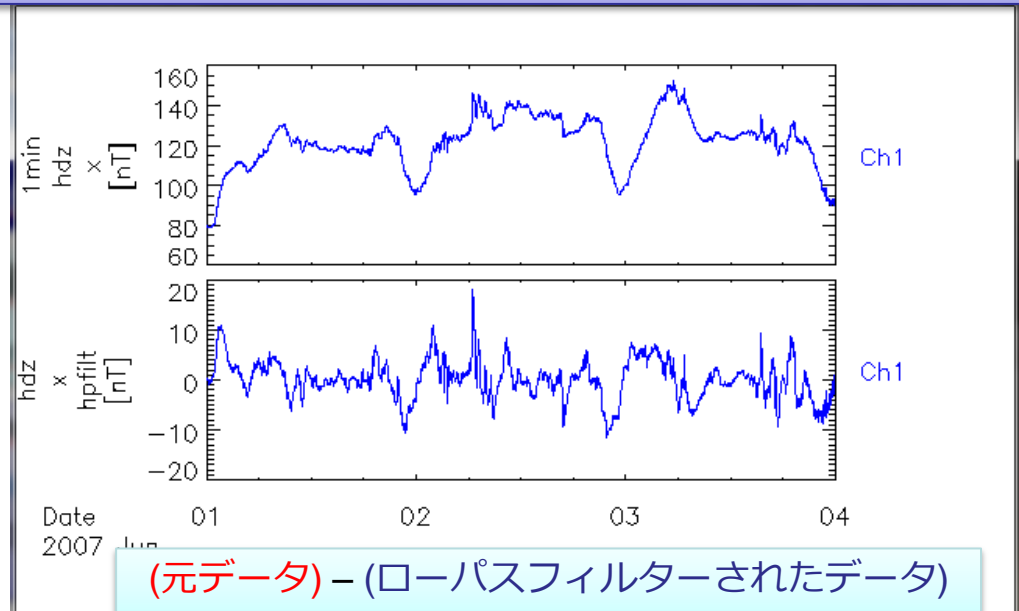
(例) thigh\_pass\_filter, 'mm210\_mag\_rik\_1min\_hdz', 3600.

```
THEMIS> thigh_pass_filter, 'mm210_mag_rik_1min_hdz_x', 3600.
```

```
STORE_DATA(...): Creating tplot variable: 12 mm210_mag_rik_1min_hdz_x_hpfilt
```

```
THEMIS> tplot, [ 'mm210_mag_rik_1min_hdz_x',  
                 'mm210_mag_rik_1min_hdz_x_hpfilt' ]
```

- 結果が ...**hpfilt** という名前の新しいtplot変数に格納される。
- ただしデジタルフィルターではなく、簡易的なもの。
- 実際は前頁のtsmooth\_in\_time でローパスフィルターされたデータを元データから差し引いている。



## avg\_data, 'tplot変数名', 平均時間幅[秒]

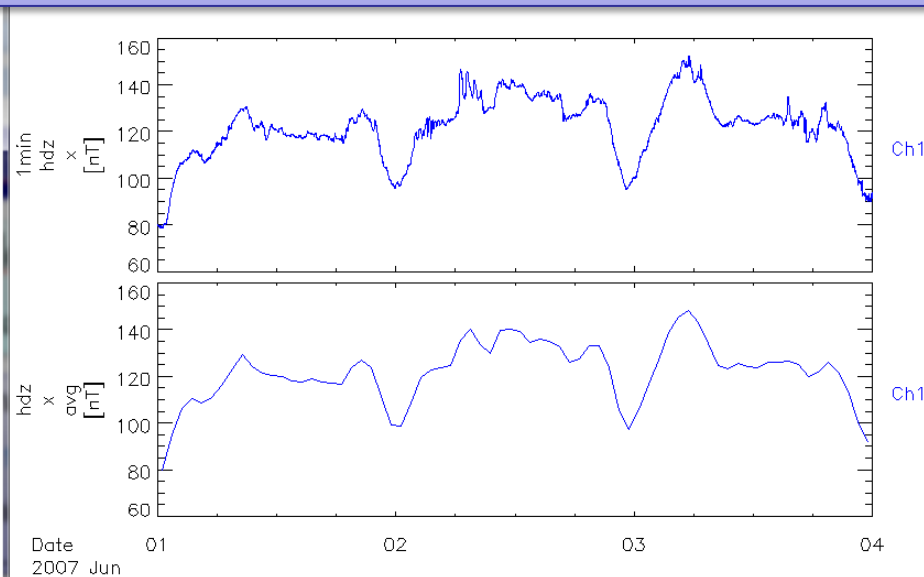
(例) avg\_data, 'mm210\_mag\_rik\_1min\_hdz', 3600.

```
THEMIS> avg_data, 'mm210_mag_rik_1min_hdz_x', 3600.
```

```
STORE_DATA(...): Creating tplot variable: 12 mm210_mag_rik_1min_hdz_x_avg
```

```
THEMIS> tplot, [ 'mm210_mag_rik_1min_hdz_x',  
                 'mm210_mag_rik_1min_hdz_x_avg' ]
```

- 結果が **...\_avg** という名前の新しいtplot変数に格納される。
- 第2引数に平均の時間幅を与える。3600[秒]にすれば1時間平均、60にすれば1分平均。
- 元データの時間分解能より小さい時間幅を与えると、結果が歯抜けデータになってしまうので注意。



# 周波数スペクトル解析

## tdpwrspc, 'tplot変数名'

窓幅のデータ点数、ハニング窓を使う/使わない、など色々オプションがある

(例) tdpwrspc, 'mm210\_mag\_rik\_1min\_hdz\_x'

```
THEMIS> tdpwrspc, 'mm210_mag_rik_1min_hdz_x'
```

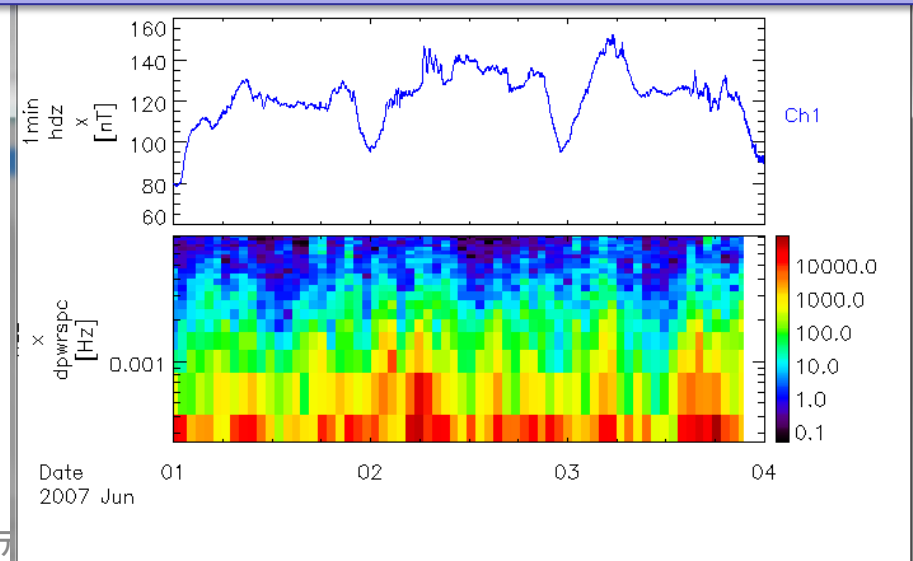
```
STORE_DATA(...): Creating tplot variable: 12 mm210_mag_rik_1min_hdz_x_dpwrspc
```

```
THEMIS> options, 'mm210_mag_rik_1min_hdz_x_dpwrspc', 'ysubtitle', '[Hz]'
```

縦軸の単位を[Hz]に変更する

```
THEMIS> tplot, [ 'mm210_mag_rik_1min_hdz_x',  
                 'mm210_mag_rik_1min_hdz_x_dpwrspc' ]
```

- ハニング窓+FFTでダイナミックスペクトル求め, ... **dpwrspc** という名前のtplot変数に結果を格納する。
- tplotによりカラーコンターでプロットされる。コンターの単位は元の値の単位の2乗/Hz (元: nT → nT<sup>2</sup>/Hz)



## wav\_data, 'tplot変数名'

(例) wav\_data, 'mm210\_mag\_rik\_1min\_hdz\_x'

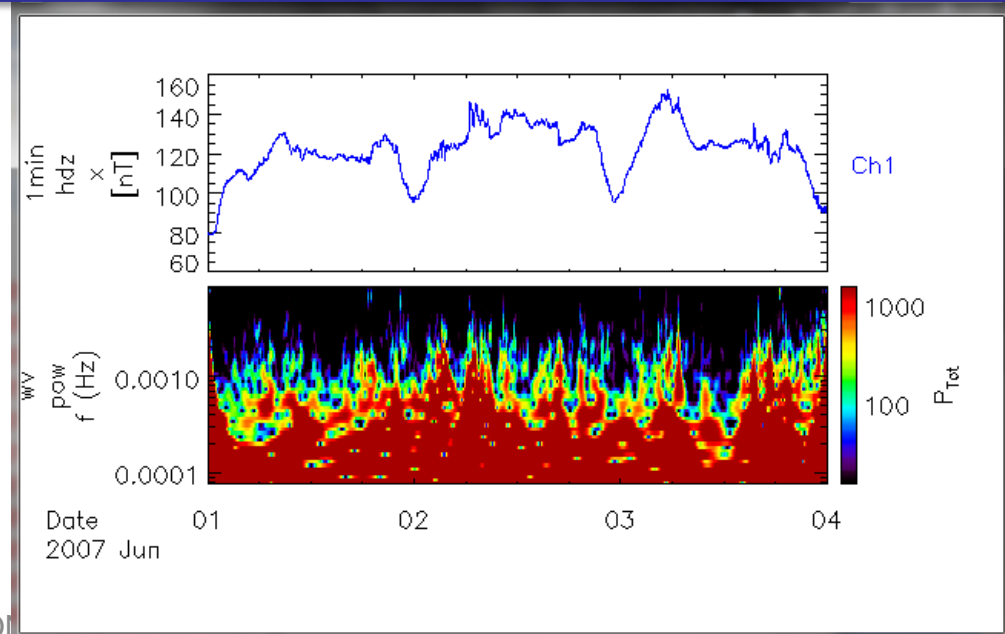
Wavelet変換で周波数  
スペクトルを求める

```
THEMIS> wav_data, 'mm210_mag_rik_1min_hdz_x'
```

```
STORE_DATA(...): Creating tplot variable: 12 mm210_mag_rik_1min_hdz_x_wv_pow
```

```
THEMIS> tplot, [ 'mm210_mag_rik_1min_hdz_x',  
                 'mm210_mag_rik_1min_hdz_x_wv_pow' ]
```

- **ウェーブレット変換**を用いるので、tdpwrspcよりは速い時間変動にも追従できる。
- その代わりに処理に時間がかかる。一度に変換するのは1万点くらいにしておいた方がよい。



- **tplot変数**とはTDAS上の時系列データ参照の概念であり、IDLのメモリー上にその実体となる**メタデータ付きデータ構造体**がある。
- `get_data`および`store_data`によりIDLの通常の配列とのやり取りが可能。
- `Calc` コマンドによりtplot変数の演算ができる。
- 各種フィルター処理やスペクトル解析を行うことができる。