



IUGONET

Metadata DB for Upper Atmosphere

超高層大気長期変動の全地球上ネットワーク観測・研究
Inter-university Upper atmosphere Global Observation NETwork

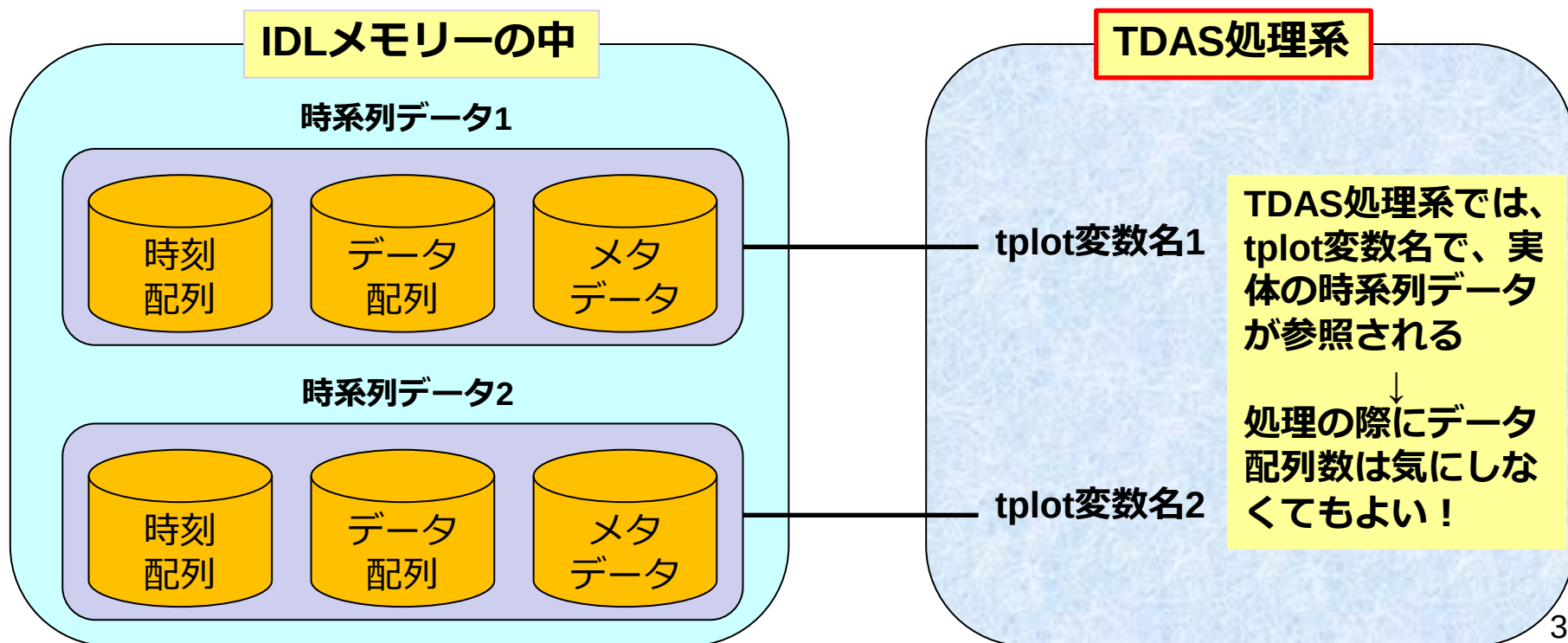
CUIの使い方（後編）：calcコマンド、get_data
やstore_dataの使い方、時系列データのフィル
ター処理、スペクトル/相関解析方法

新堀淳樹
(京大生存研)

- 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、およびプロットの画像出力方法などを行った。
- 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(足し算、引き算、掛け算、時間微分等)について学ぶ。
 - 移動平均、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)の方が自由度が高いことから、UDASに慣れてくるとコマンドを使う方が断然便利である！

2.1 tplot変数とは

- 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。



2. tplot変数の取り扱いと演算

2.2 get_data を用いてtplot変数の中身を見る

メタデータが入る

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

データ配列が入る

主に可視化情報が入る

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

```
THEMIS> timespan, '2012-11-11',7  時間幅として2012年11月7日から3日分を指定
THEMIS> iug_load_lfrto,site = 'ath'  LF電波観測点のAthabasca (ATH) データをロード
THEMIS> get_data, 'lfrto_ath_wwvb_pow30s', 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 <5841700>, 2 tags, length=239040, data length=239040, refs=1:
X      DOUBLE  Array[19920]
Y      FLOAT   Array[19920]
```

2.2 get_data を用いてtplot変数の中身を見る

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

```
** Structure <5841700>, 2 tags, length=239040, data length=239040, refs=1:  
X          DOUBLE   Array[19920]  
Y          FLOAT    Array[19920]
```

**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つのメンバーから構成されている。**

**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 19920個の1次元配列。**

**つまりデータのtime frame は19920個ある。 このデータは30秒値で7日分なので、
1日 = 86400秒 / 30秒 x 7 日分 で 19920。**

**Y: 実際にデータが入っている配列
この場合、19920の1次元配列。**

2. tplot変数の取り扱いと演算

2.2 get_data を用いてtplot変数の中身を見る

```
THEMIS> help, dl, /struct
```

```
** Structure <81c32f0>, 4 tags, length=1128, data length=1122, refs=3:
```

```
  CDF          STRUCT    -> <Anonymous> Array[1]
```

```
  SPEC         BYTE      0
```

```
  LOG          BYTE      0
```

```
  YSUBTITLE    STRING    '[dB]'
```

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。

```
THEMIS> help, lim, /struct
```

```
LIM          LONG      =      0
```

lim 構造体の方には主にプロット等に可視化する場合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。

2.3 store_dataで新規tplot変数を作成

```
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。

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

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

```
THEMIS> store_data, 'lfrto_ath_wwvb_pow30s_half', data = { x:time, y:val }
```

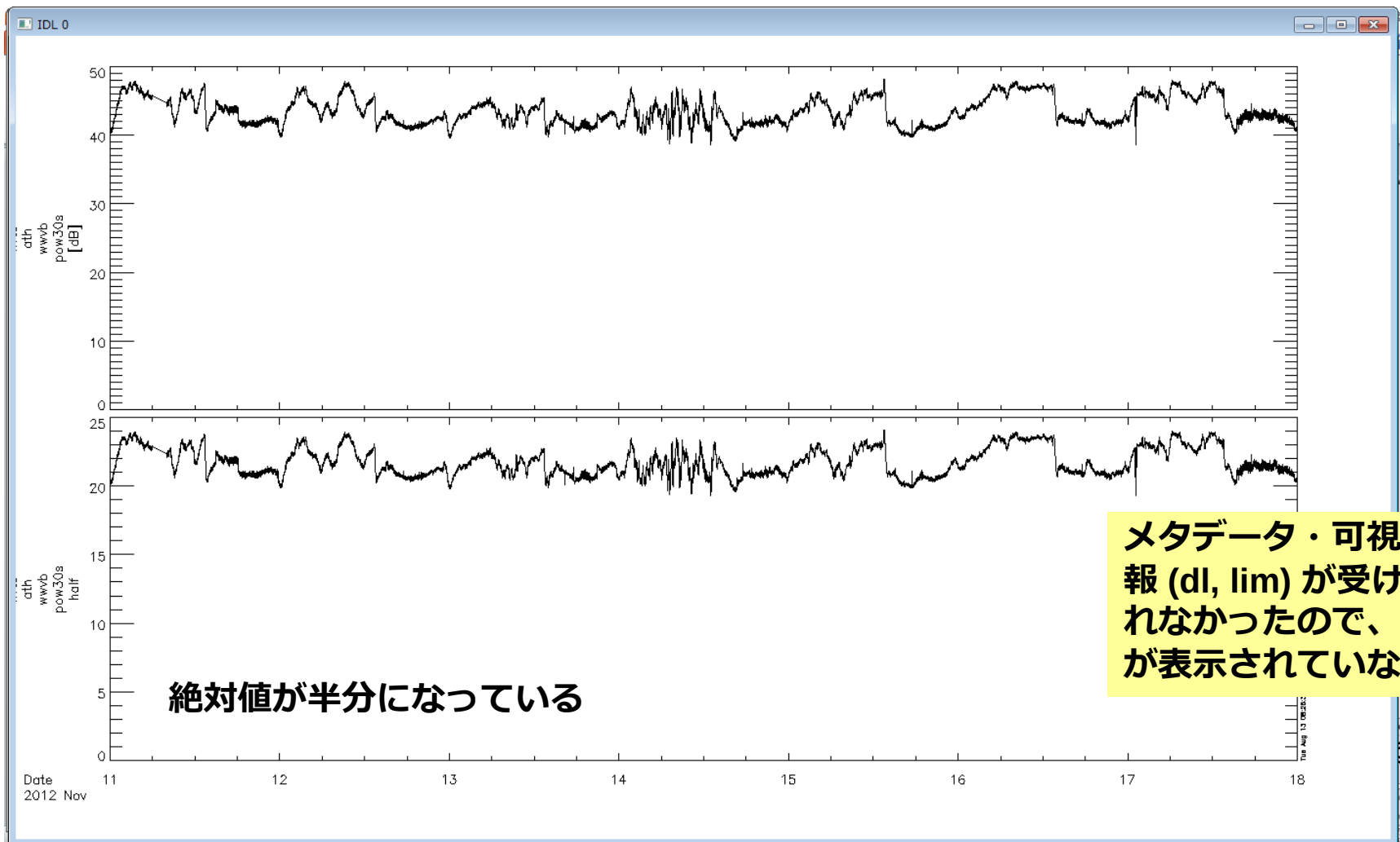
```
THEMIS> tplot, ['lfrto_ath_wwvb_pow30s', 'lfrto_ath_wwvb_pow30s_half' ]
```

実際にtplotでプロットして確認してみる

2. tplot変数の取り扱いと演算

2.3 store_dataで新規tplot変数を作成

```
THEMIS> tplot, ['lfrto_ath_wwvb_pow30s', 'lfrto_ath_wwvb_pow30s_half']
```



2.4 calcコマンドによるtplot変数の演算

calc, ' "新tplot変数名" = ... 計算式 ... '

(例) calc, ' "newvar" = "lfrto_ath_wwvb_pow30s" + 20. '

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, ' "lfrto_ath_wwvb_pow30s_half" = "lfrto_ath_wwvb_pow30s" / 2.0 '

と、わずか1行で実行できる！

2.4 calcコマンドによるtplot変数の演算

calc, ' "新tplot変数名" = ... 計算式 ... '

(例) calc, ' "newvar" = "lfrto_ath_wwvb_pow30s" + 20. '

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

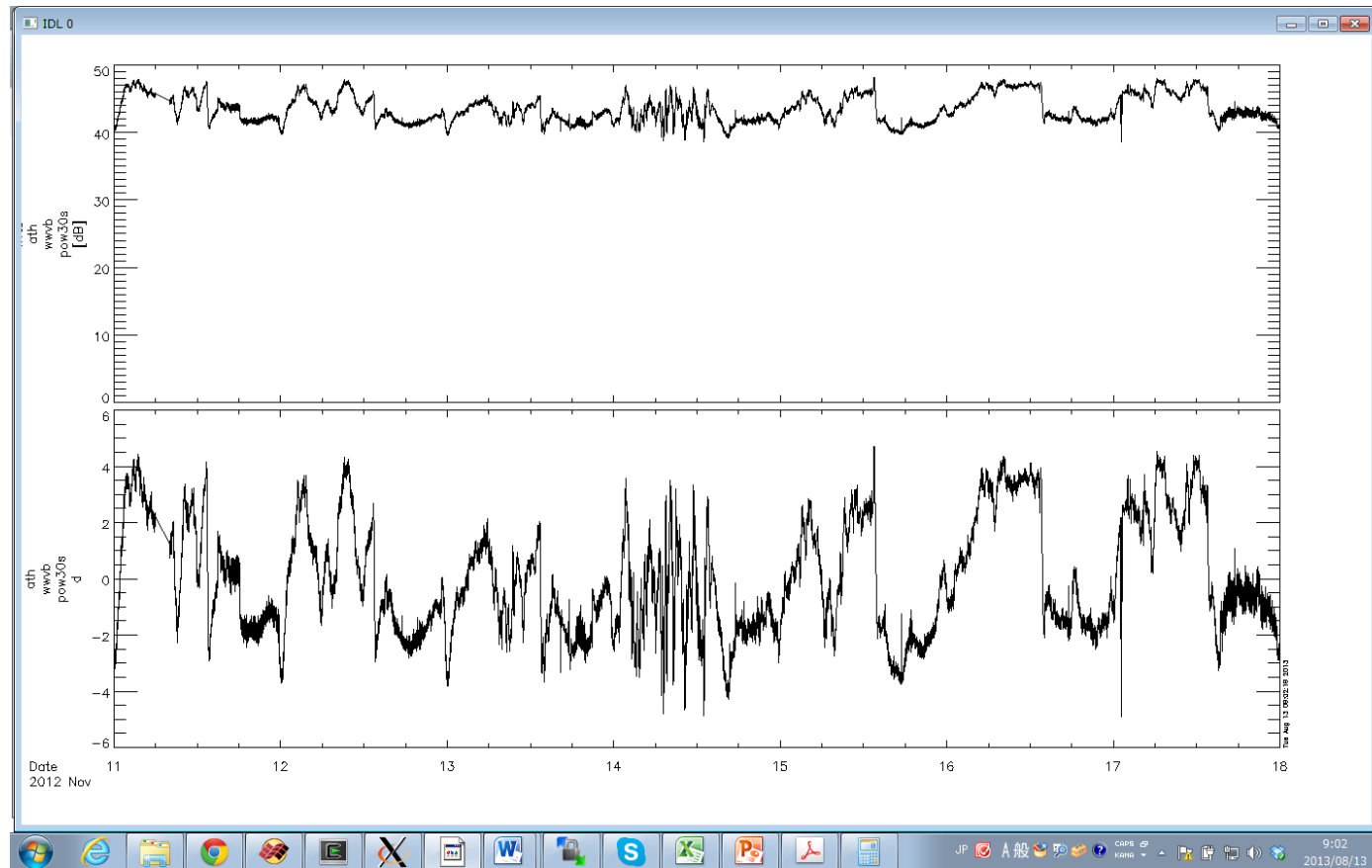
- 複数のtplot変数を演算に使う場合、実体の配列のサイズ・次元が同一でないといけない。データの時刻数が異なる、データの次元が異なる(スカラーデータとベクトルデータの混在など)とエラーになる。

2. tplot変数の取り扱いと演算

2.4 calcコマンドの練習

```
THEMIS> calc, "'lfrto_ath_wwvb_pow30s_d' = 'lfrto_ath_wwvb_pow30s'-  
mean('lfrto_ath_wwvb_pow30s')"
```

LF電波強度の平均
値を計算し、それ
を元データから差
し引く演算をcalc
で求めた。



タイトルやラベルは後でoptions コマンドで適宜変更する。

2.5 calcコマンドの応用

電離圏Pedersen, Hall伝導度からCowling電気伝導度を導出

```
calc, ' "sigmaC" = "sigmaP" + ("sigmaH" ^2 / "sigmaP")'
```

注) sigmaP: Pedersen伝導度、 sigmaH: Hall伝導度

$$\Sigma_C = \Sigma_P + \frac{\Sigma_H^2}{\Sigma_P}$$

太陽風観測から太陽風動圧を導出

```
calc, ' "Pdyn" = "ace_Np" * "ace_Vp"^2 * 1.6726 * 1e-6 '
```

注) ace_Np: 太陽風密度 [/cc]、 ace_Vp: 太陽風速度 [km/s] ← プロトンの質量

$$P_{dyn} = N_p * M * V_p^2$$

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。

3. tplot変数への各種フィルター処理

3.1 tsub_average で平均値を差し引く

tsub_average, 'tplot変数名'

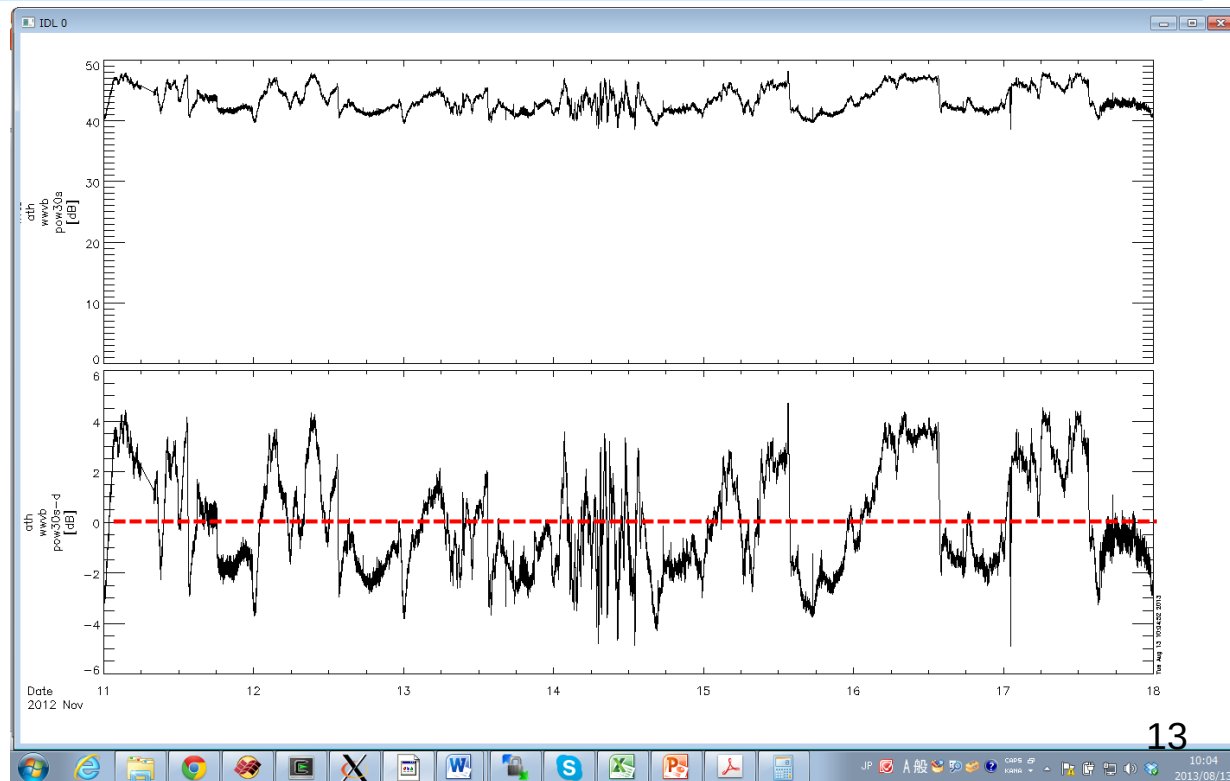
(例) tsub_average, 'lfrto_ath_wwvb_pow30s'

THEMIS> **tsub_average**, 'lfrto_ath_wwvb_pow30s'

THEMIS> **tplot**, ['lfrto_ath_wwvb_pow30s', 'lfrto_ath_wwvb_pow30s-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。



3. tplot変数への各種フィルター処理

3.2 tsmooth_in_time でスムージング

tsmooth_in_time, 'tplot変数名', 平均幅[秒]

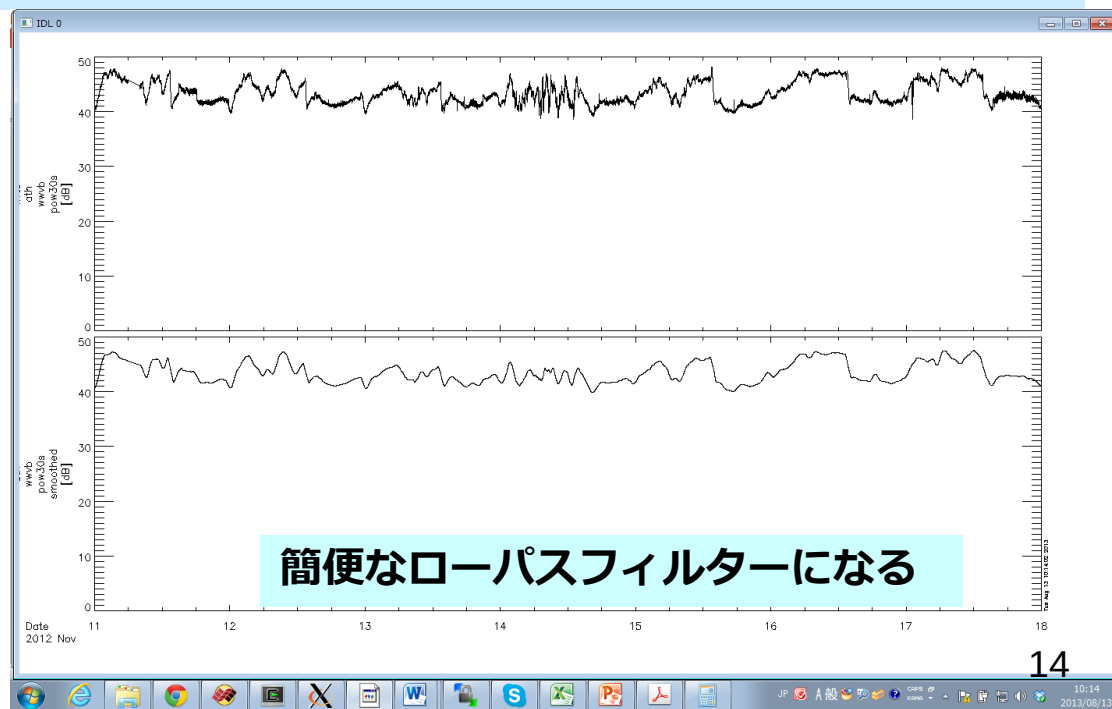
(例) tsmooth_in_time, 'lfrto_ath_wwvb_pow30s', 3600

```
THEMIS> tsmooth_in_time, 'lfrto_ath_wwvb_pow30s', 3600
```

```
THEMIS> tplot, ['lfrto_ath_wwvb_pow30s',  
                'lfrto_ath_wwvb_pow30s_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。



3. tplot変数への各種フィルター処理

3.3 thigh_pass_filter でハイパス・フィルター

thigh_pass_filter, 'tplot変数名', 下限周期[秒]

(例) thigh_pass_filter, 'lfrto_ath_wwvb_pow30s', 3600

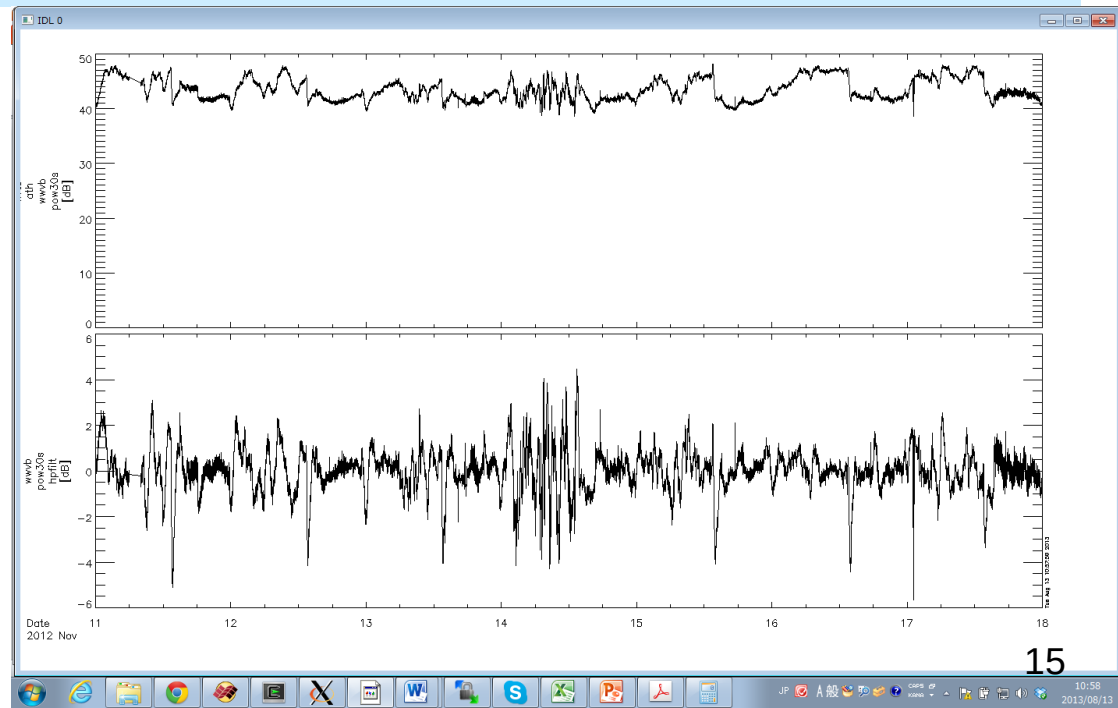
```
THEMIS> thigh_pass_filter, 'lfrto_ath_wwvb_pow30s', 3600
```

```
THEMIS> tplot, ['lfrto_ath_wwvb_pow30s',  
                'lfrto_ath_wwvb_pow30s_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。



3.4 avg_dataで~分値、~時間値に平均

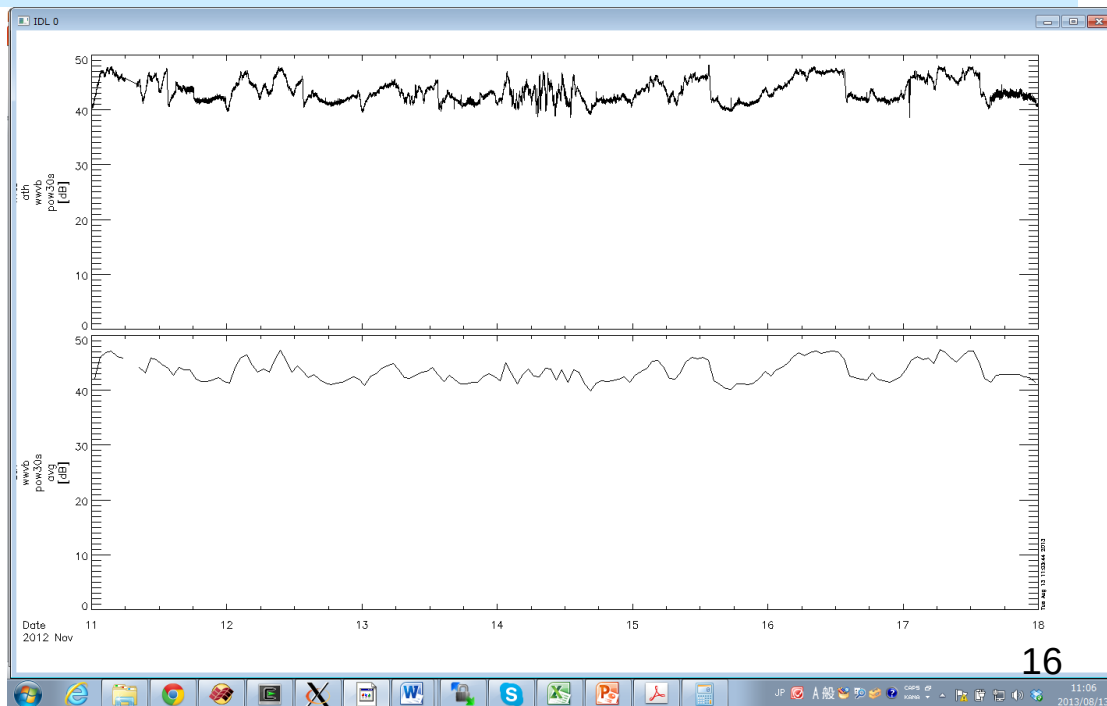
avg_data, 'tplot変数名', 平均時間幅[秒]

(例) avg_data, 'lfrto_ath_wwvb_pow30s', 3600

```
THEMIS> avg_data, 'lfrto_ath_wwvb_pow30s', 3600
```

```
THEMIS> tplot, ['lfrto_ath_wwvb_pow30s',  
                'lfrto_ath_wwvb_pow30s_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。



4.2 フーリエスペクトル解析 tdpwrspc

tdpwrspc, 'tplot変数名'

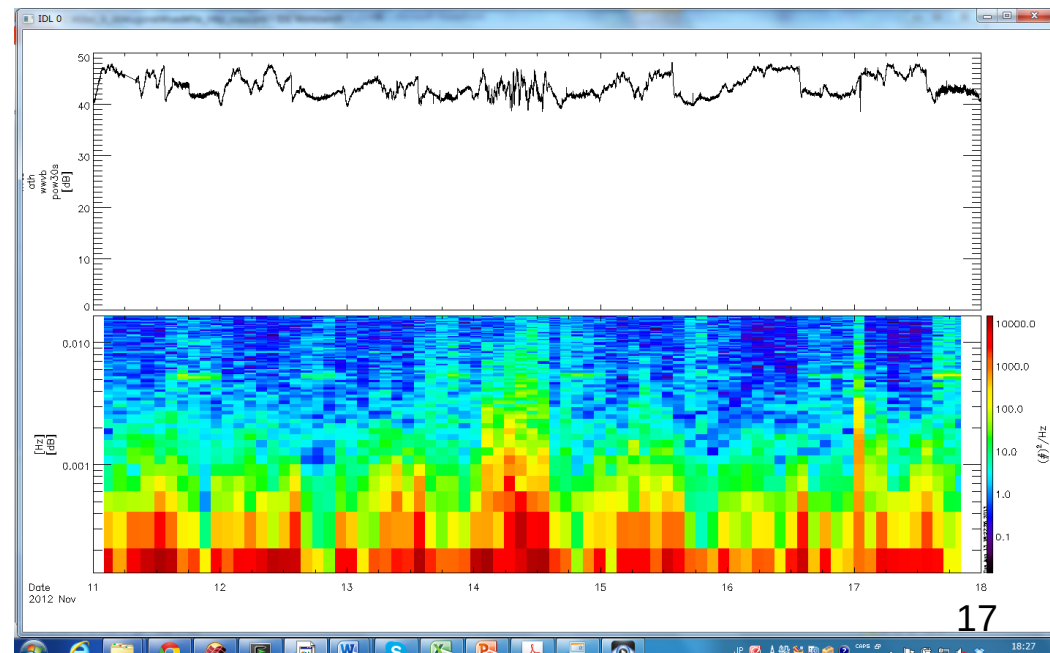
(例) tdpwrspc, 'lfrto_ath_wwvb_pow30s'

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

THEMIS> tdpwrspc, 'lfrto_ath_wwvb_pow30s'

THEMIS> tplot, ['lfrto_ath_wwvb_pow30s',
'lfrto_ath_wwvb_pow30s_dpwrspec']

- **ハニング窓+FFTでダイナミックスペクトル求め, ..._dpwrspec** 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: dB $\Rightarrow \text{dB}^2/\text{Hz}$)
- 縦軸のキャプションは、optionsコマンドで適宜修正する。



4.2 ウェーブレット変換 wav_data

wav_data, 'tplot変数名'

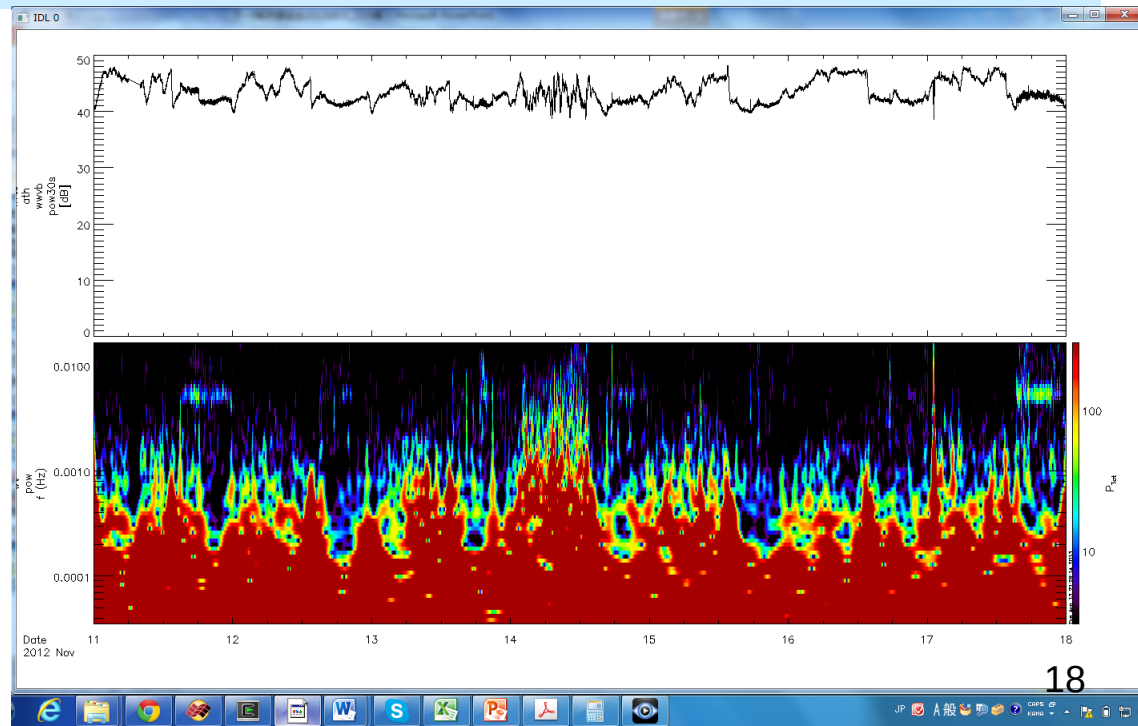
(例) wav_data, 'lfrto_ath_wwvb_pow30s'

Wavelet変換で周波数
スペクトルを求める

THEMIS> wav_data, 'lfrto_ath_wwvb_pow30s'

THEMIS> tplot, ['lfrto_ath_wwvb_pow30s',
 'lfrto_ath_wwvb_pow30s_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。
- その代わり処理に時間がかか
るので、**1度に変換するのは1
万点くらいにしておいた方が
よい。**



4.3 S(Stockwell)変換 `ustrans_pwrspc`

`ustrans_pwrspc`, 'tplot変数名', /sampling, /abs

S変換で周波数スペクトルを求める

(例) `ustrans_pwrspc`, 'lfrto_ath_wwvb_pow30s',
/sampling, /abs

THEMIS> `avg_data`, 'lfrto_ath_wwvb_pow30s', 60

1分平均値の計算

THEMIS> `ustrans_pwrspc`, 'lfrto_ath_wwvb_pow30s_avg'

THEMIS> `options`, 'lfrto_ath_wwvb_pow30s_avg_stpwrspc',
'ysubtitle', '[Min]'

単位の変更

THEMIS> `ylim`, 'lfrto_ath_wwvb_pow30s_avg_stpwrspc', 0, 24

Y軸の範囲変更

THEMIS> `zlim`, 'lfrto_ath_wwvb_pow30s_avg_stpwrspc', 0, 1

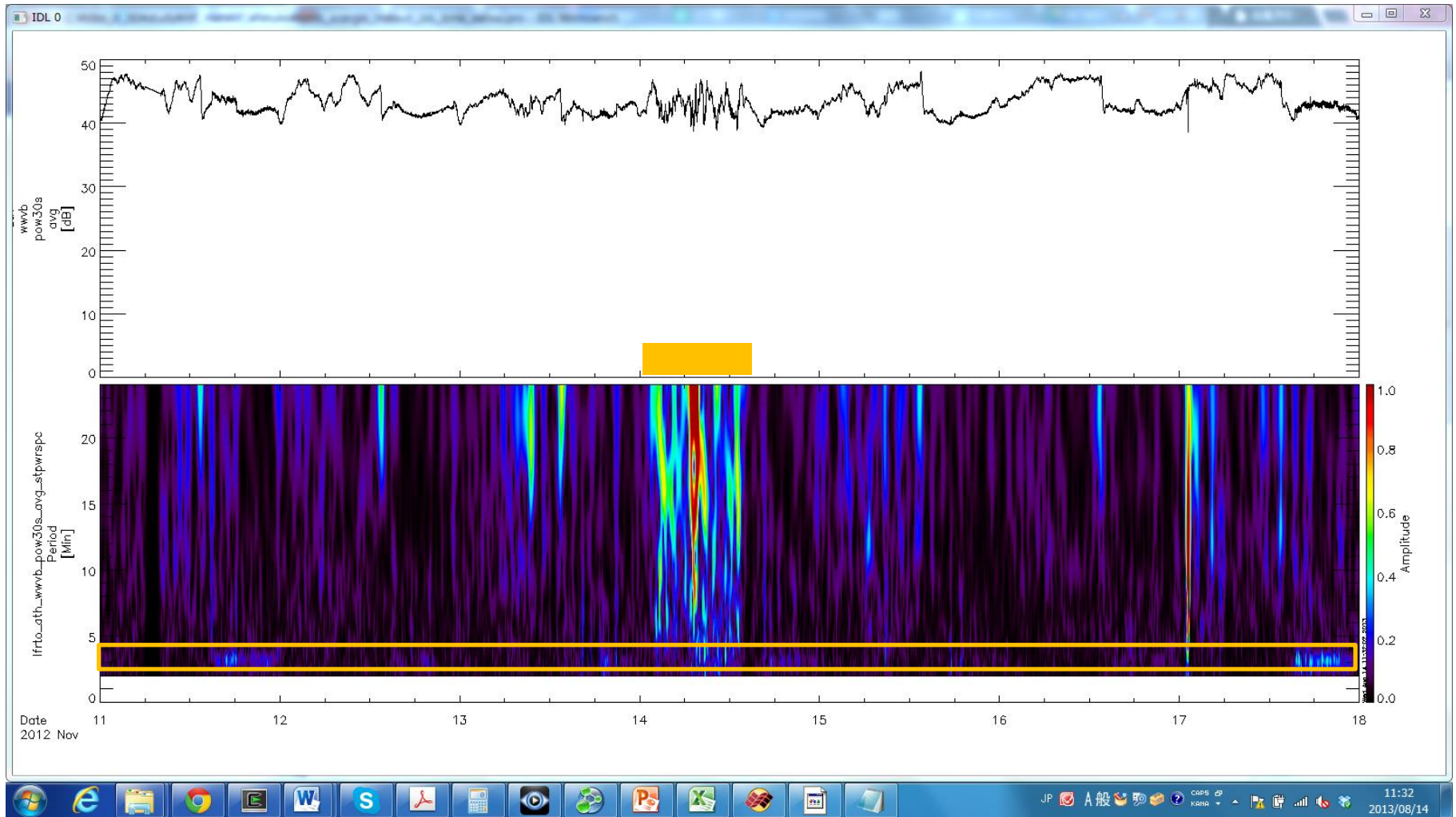
Z軸の範囲変更

THEMIS> `tplot`, ['lfrto_ath_wwvb_pow30s_avg',
'lfrto_ath_wwvb_pow30s_avg_stpwrspc']

•引数/absの代わりに/powerとすると、振幅ではなくパワー値を算出する。

•処理に時間がかかるので、1度に変換するのは1万点くらいにしておいた方がよい。

4.3 S(Stockwell)変換 ustrans_pwrspc



5. 他のデータとの比較

5.1 地磁気指数との比較解析

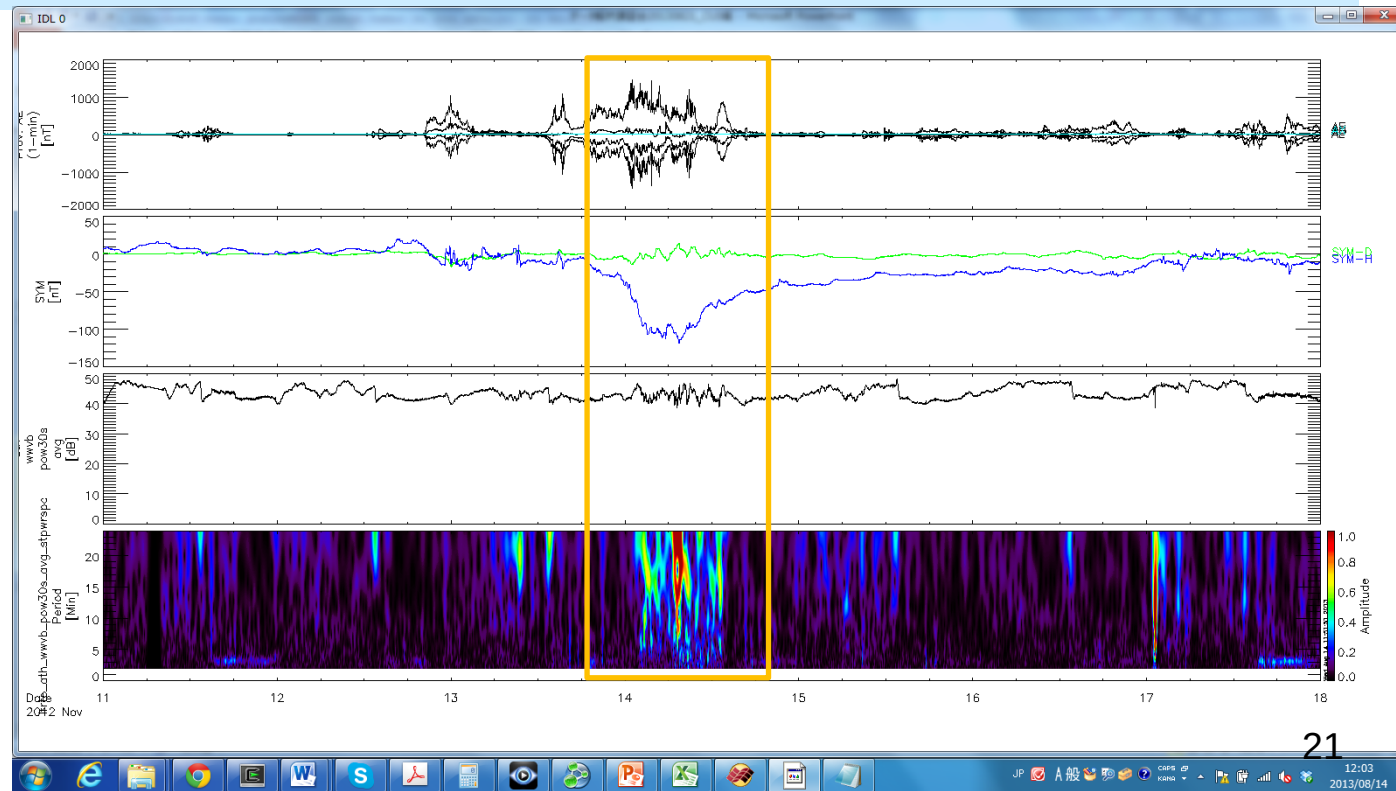
THEMIS> iug_load_gmag_wdc, site= ['ae', 'sym']

地磁気指数(AE、SYM)のロード

THEMIS> tplot, ['wdc_mag_ae_prov_1min', 'wdc_mag_sym', 'lfrto_ath_wwvb_pow30s_avg', 'lfrto_ath_wwvb_pow30s_avg_stpwrspec']

上記でロードしたデータと先ほどのS変換解析結果との並列プロット

•11月13-14日に発生した磁気嵐に対応してLF電波強度の変調が多様な周波数域で発生していることが分かる



- 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。
- UDAS3.00.1以降のバージョンでは、IUGONETで独自に開発した描画や解析ツール(相互相関・無相関検定、コヒーレンス解析、トレンド検定)などが付け加わっている。