



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

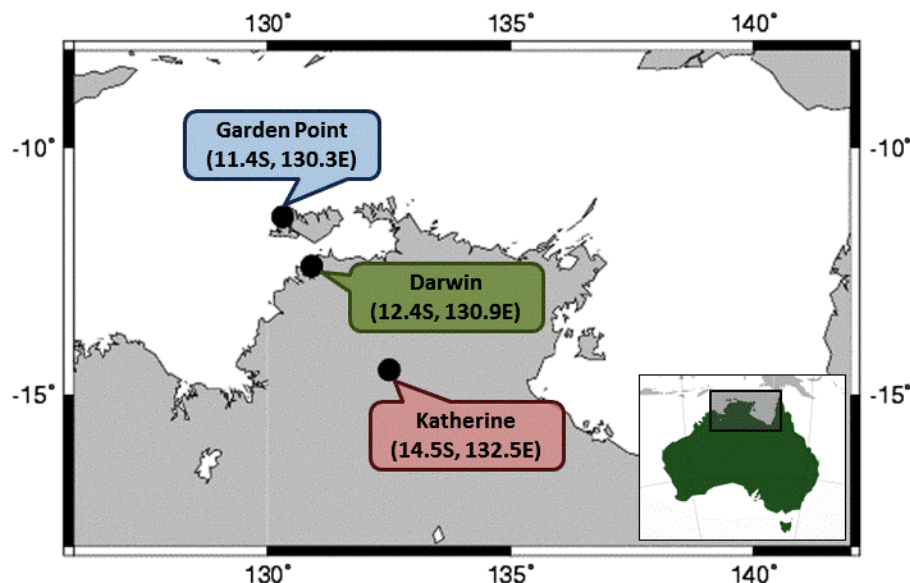
新堀 淳樹(京大生存研)

1. はじめに

- 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！

1. はじめに

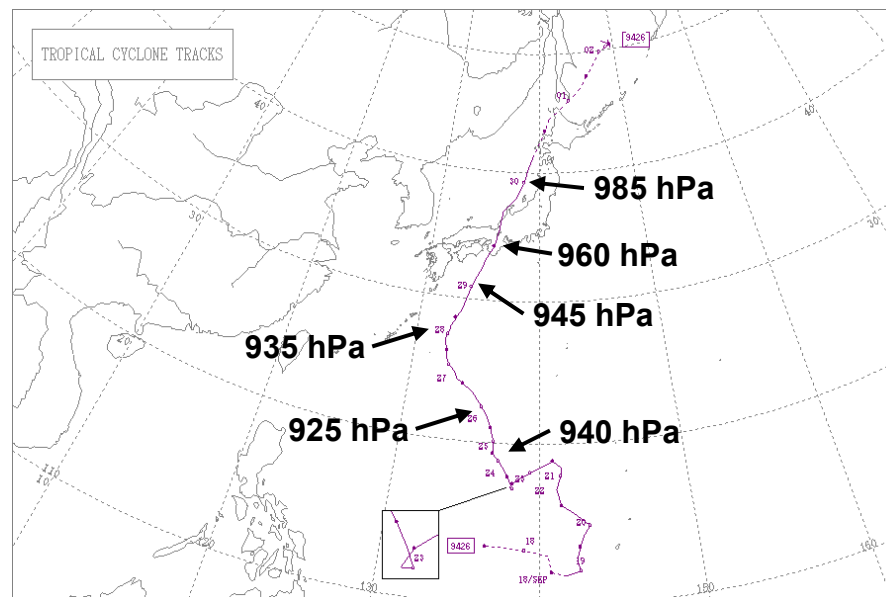
- 2001年10-12月の3期にわたって行われたダーウィンゾンデ観測キャンペーンデータ



<http://database.rish.kyoto-u.ac.jp/arch/iugonet/DAWEX/>

解析可能な観測パラメタ:
気圧、気温、相対湿度、露点温度
東西風速、南北風速

- 1994年9月29-30日に台風26号が信楽上空を通過した時に取得された自動気象観測データ



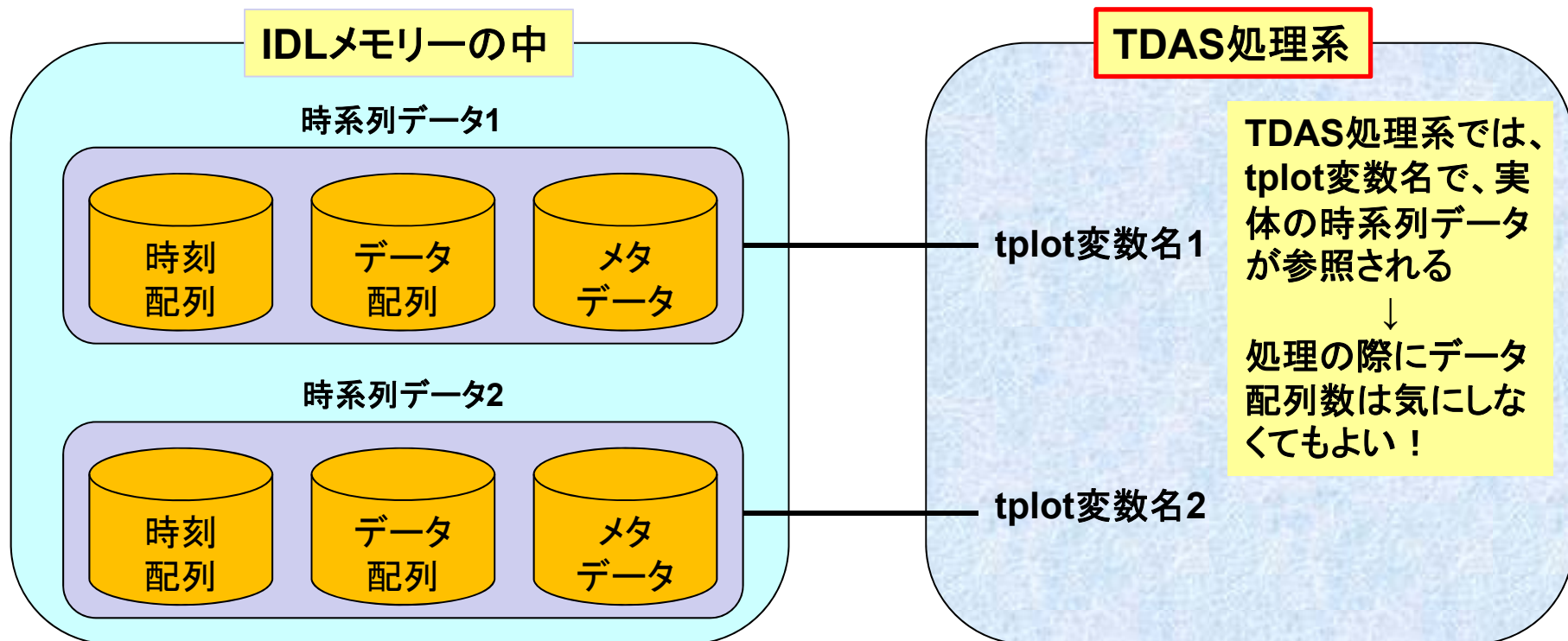
http://www.data.jma.go.jp/fcd/yoho/typhoon/route_map/bstv/1994.html

解析可能な観測パラメタ:
気圧、気温、相対湿度、東西風速、南北風速

2. tplot変数の取り扱いと演算

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`, '2001-12-11',5 時間幅として2001年12月11日から5日分を指定

THEMIS> `iug_load_radiosonde_rish`,site =['drw']

気球観測点のDarwin (DRW) データをロード

THEMIS> `get_data`, 'iug_radiosonde_drw_temp', 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 <2488a5d0>, 3 tags, length=65920, data length=65920, refs=1:

X	DOUBLE	Array[40]	→	時刻データ(1次元)
Y	FLOAT	Array[40, 400]	→	温度データ(2次元)
V	FLOAT	Array[400]	→	高度データ(1次元)

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, d, /struct
```

```
** Structure <2488a5d0>, 3 tags, length=65920, data length=65920, refs=1:  
X          DOUBLE   Array[40]  
Y          FLOAT    Array[40, 400]  
V          FLOAT    Array[400]
```

tplot変数の実体のデータ構造体 (今の場合は d) は X, Y, V という3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 (1970-1-1 00:00:00 UTからの積算秒数)

この例では 40個の1次元配列。つまりデータのtime frame は40個ある。このデータは3時間値で7日分なので、1日 = 86400秒 / 10800秒 x 5 日分 で 40。

Y: 温度データが入っている配列

この場合、40*400の2次元配列。

V: 高度データが入っている1次元配列

この場合、400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 <24481a20>, 1 tags, length=32, data length=32, refs=7:  
DATA_ATT      STRUCT  -> <Anonymous> Array[1]
```

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
```

```
** Structure <13b5cf90>, 3 tags, length=40, data length=34, refs=2:  
YTITLE        STRING  'RSND-drw!CHeight!C[km]'  
ZTITLE        STRING  'Temp.!C[deg.]'  
SPEC          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:data1, v:data2},  
dlimits = dl, lim = lim
```

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> temp = d.y+273.15
```

```
THEMIS> height = d.v
```

```
THEMIS> store_data, 'iug_radiosonde_drw_temp_k', data = { x:time, y:  
temp, v: height}, dlimits = dl, lim = lim
```

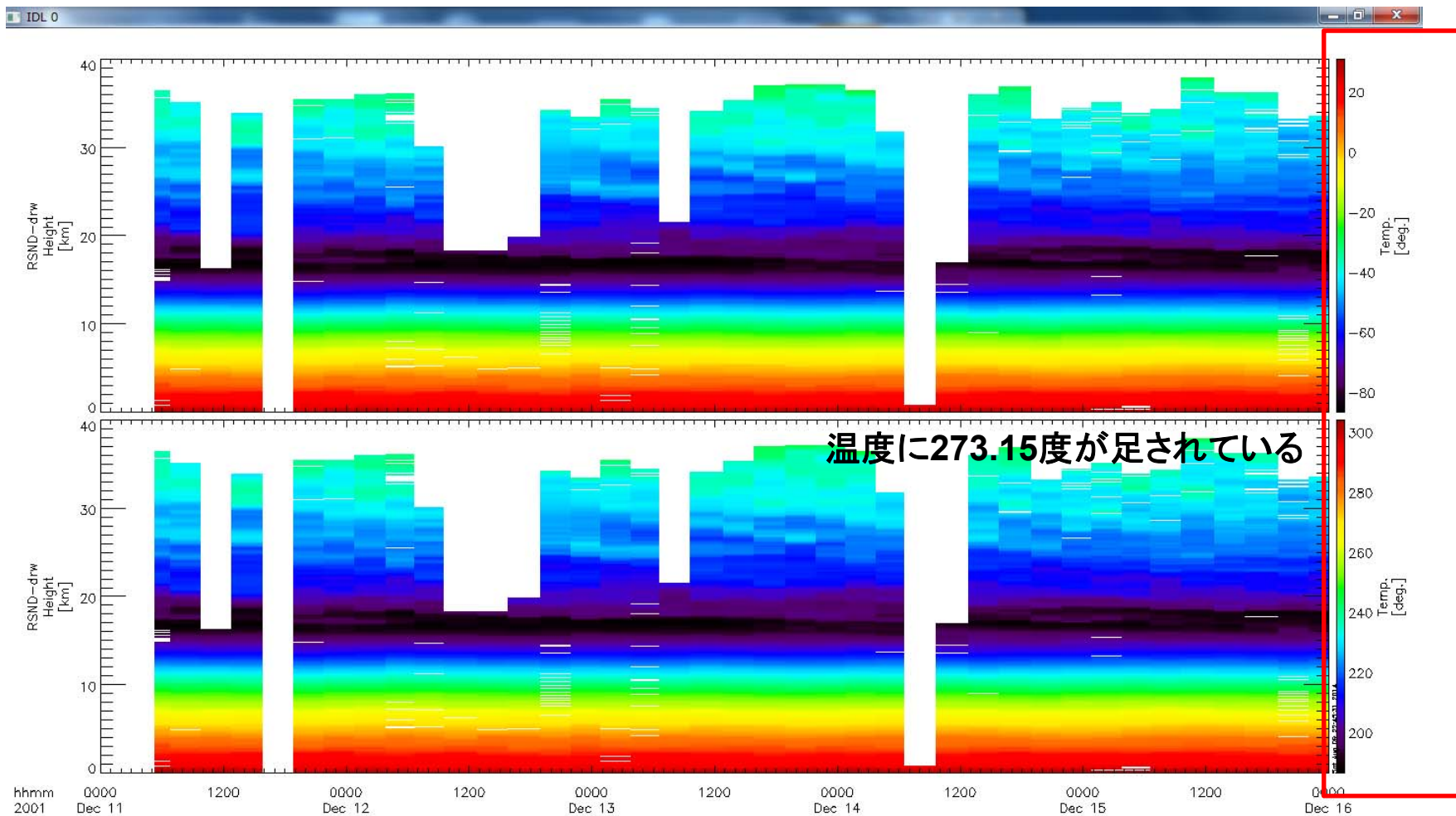
実際にtplotでプロットして確認してみる

```
THEMIS> tplot, ['iug_radiosonde_drw_temp',  
'iug_radiosonde_drw_temp_k']
```

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, ['iug_radiosonde_drw_temp', 'iug_radiosonde_drw_temp_k']



2. tplot変数の取り扱いと演算

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" = "iug_radiosonde_drw_temp" + 273.15 '

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, ' "iug_radiosonde_drw_temp_k" = "iug_radiosonde_drw_temp" +  
273.15 '
```

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" = "iug_radiosonde_drw_temp" + 273.15 '

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。

2.4 calcコマンドの練習

①飽和水蒸気圧 e_s を計算

$$e_s = 611 \times \exp \left[17.67 \times \frac{T-273.16}{T-29.66} \right] \text{ [hPa]}$$

②温位 θ を計算

$$\theta = T \left[\frac{1000}{P} \right]^{R/C_p} \text{ [K]}$$

③相当温位 θ_e を計算

$$\theta_e = \theta \cdot \exp \left(\frac{Lw_s}{C_p T_d} \right) \text{ [K]}, \quad w_s = 0.622 \times \frac{e_s}{P - e_s}$$

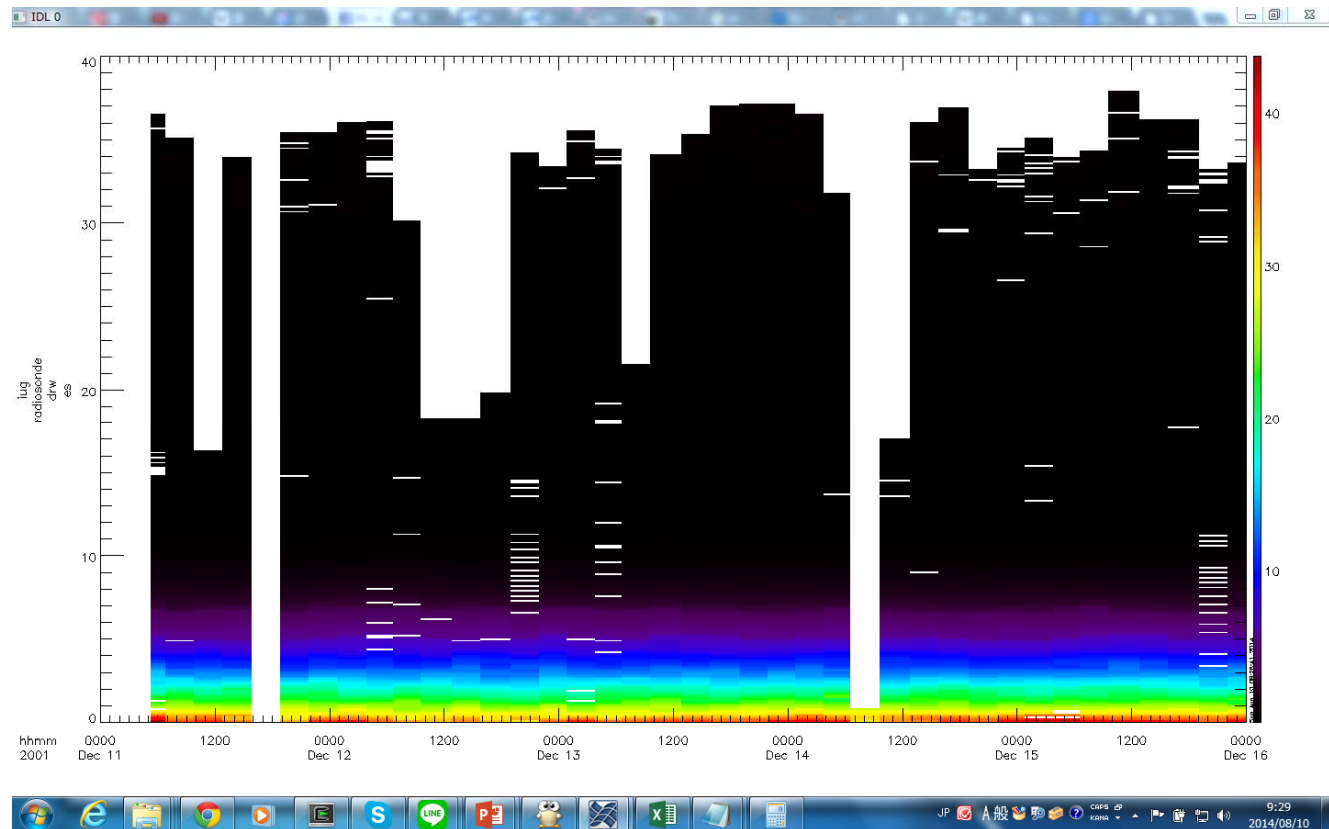
Lは凝結により放出される潜熱の定数値(約2500000)(J/kg)、 w_s は空気塊が持ち上げ凝結高度に達した時の飽和混合比、 T_d は空気塊の露点温度(K)、Rは気体定数(287.05)(J K⁻¹ kg⁻¹)、そして C_p は一定圧力での比熱容量(1004.675)(J K⁻¹ kg⁻¹)である

2. tplot変数の取り扱いと演算

2.4 calcコマンドの練習

```
THEMIS> calc,  
"'iug_radiosonde_drw_es'=6.11*exp(17.67*('iug_radiosonde_drw_temp_k'  
"-273.16)/('iug_radiosonde_drw_temp_k'-29.66))'
```

絶対温度に直した
温度データから飽
和水蒸気圧 e_s を
calc コマンドで求
めた。



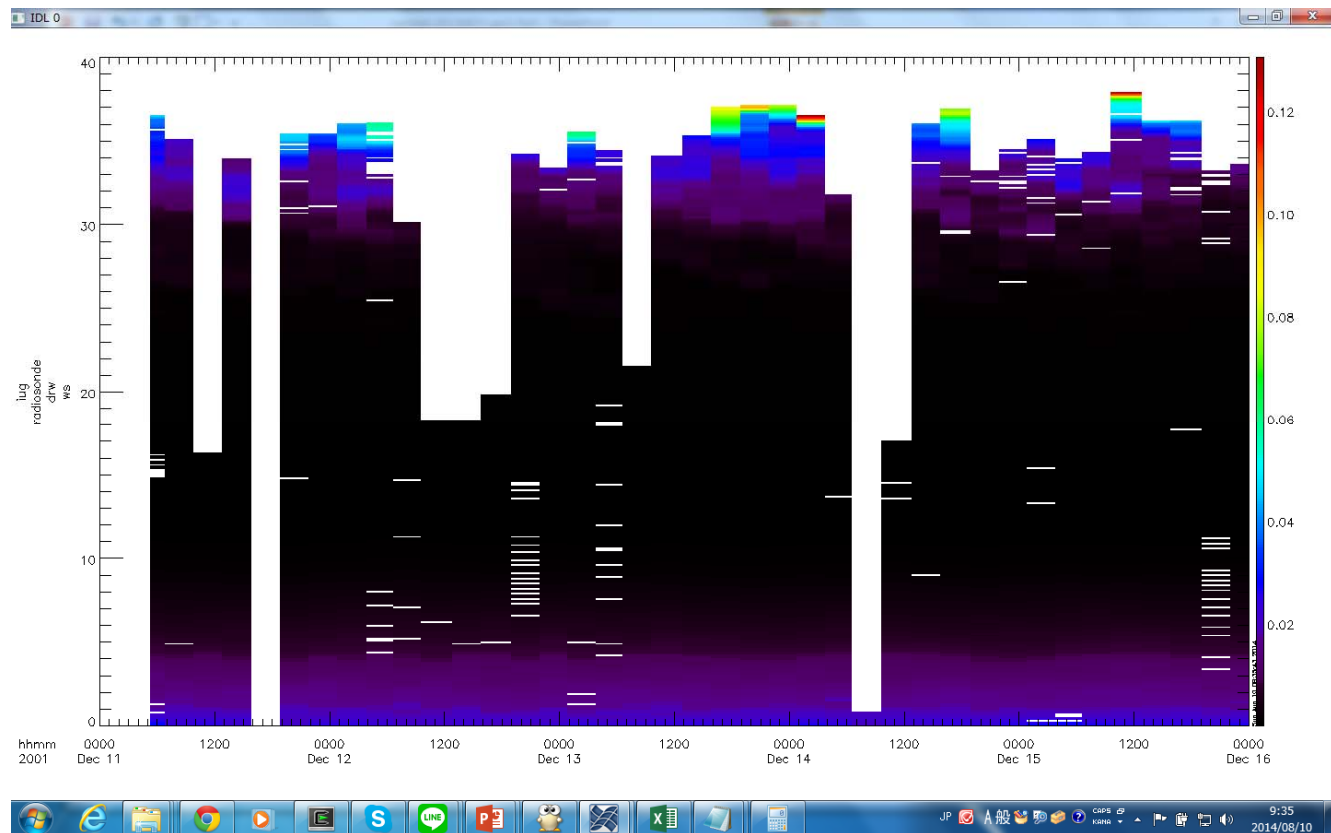
タイトルやラベルは後でoptions コマンドで適宜変更する。

2. tplot変数の取り扱いと演算

2.4 calcコマンドの練習

```
THEMIS> calc,  
"iug_radiosonde_drw_ws"=0.622*("iug_radiosonde_drw_es")/("iug_radiosonde_drw_press"-  
"iug_radiosonde_drw_es")'
```

飽和水蒸気圧と
気圧データから飽
和混合比 w_s をcalc
コマンドで求めた。



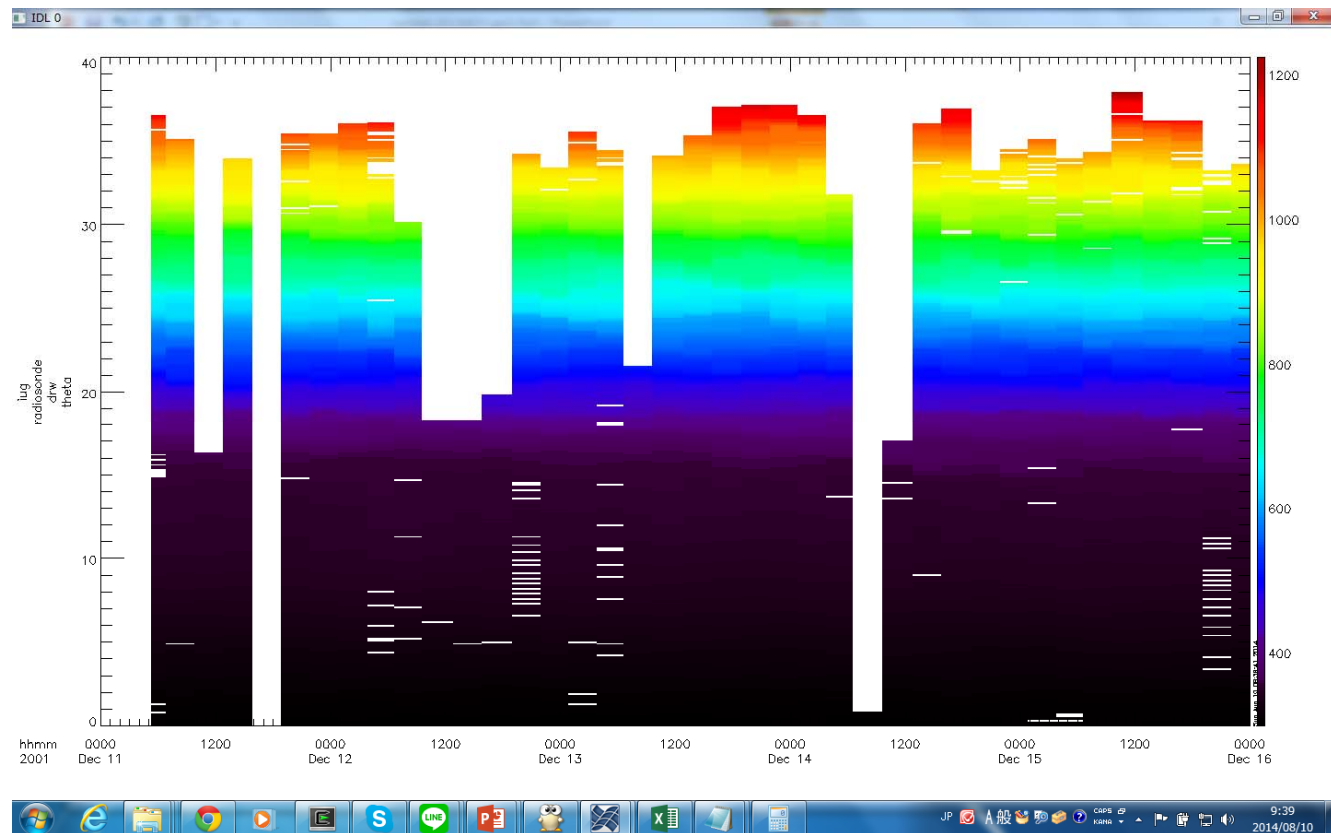
タイトルやラベルは後でoptions コマンドで適宜変更する。

2. tplot変数の取り扱いと演算

2.4 calcコマンドの練習

```
THEMIS> calc,  
"iug_radiosonde_drw_theta"=("iug_radiosonde_drw_temp_k")*(1000/"iug_  
radiosonde_drw_press")^0.2857'
```

絶対温度に直した
温度と気圧データ
から温位 θ をcalc
コマンドで求めた。



タイトルやラベルは後でoptions コマンドで適宜変更する。

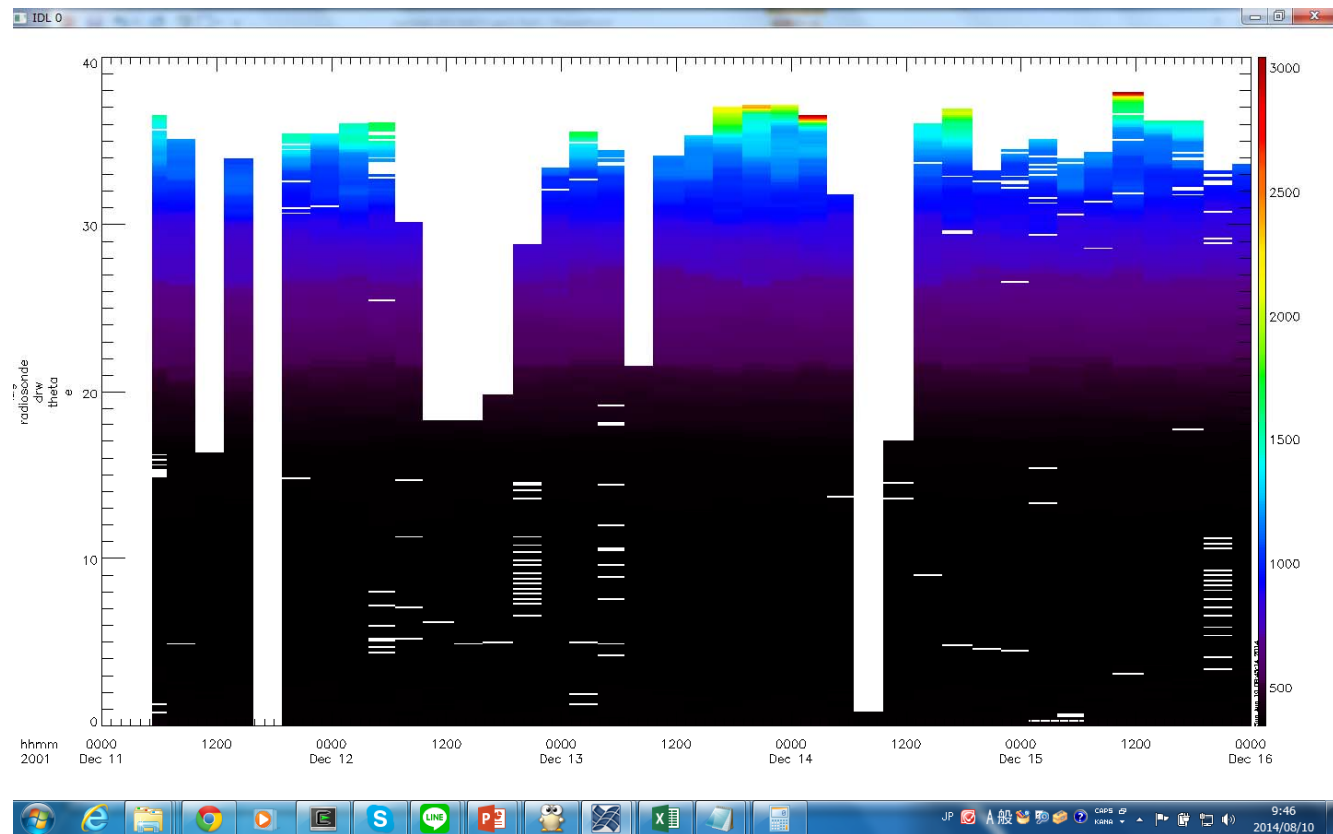
2. tplot変数の取り扱いと演算

2.4 calcコマンドの練習

THEMIS> `calc,`

```
"iug_radiosonde_drw_theta_e"="iug_radiosonde_drw_theta"*exp(2500000*"iug_radiosonde_drw_ws"/(1004.675*("iug_radiosonde_drw_dewp"+273.15)))'
```

温位、飽和混合比と露点温度データから相当温位 θ_e をcalcコマンドで求めた。



タイトルやラベルは後でoptions コマンドで適宜変更する。

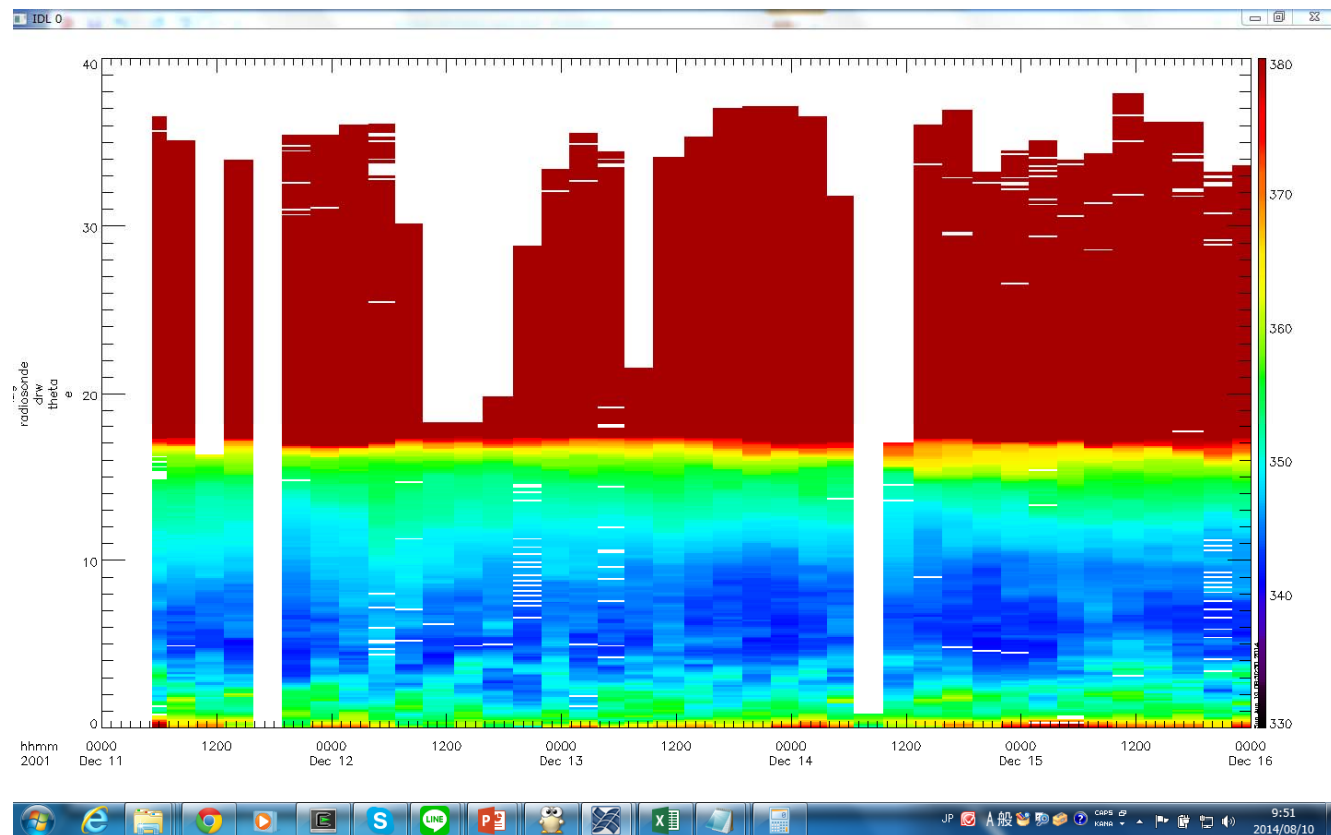
2. tplot変数の取り扱いと演算

2.4 calcコマンドの練習

THEMIS> **zlim**, ["iug_radiosonde_drw_theta_e"], 330, 380

zlimコマンドでカ
ラスケールを変
更した。

相当温位の時間-高
度分布図を用いて**集
中豪雨**を解析するこ
とも可能。



相当温位は、気温が高いほど、また湿度が高いほど、大きくなる。気温・湿度ともに高度が高くなるほど低下するため、相当温位は高度とともに減少する。しかし、実際の大気では、対流圏中層への暖湿流の流入や、下層への乾燥大気の流入などの移流によって、不均一な状態になることが多く、時に逆転する。

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変数を用いた各種データ解析

3.1 tsub_average で平均値を差し引く

自動気象観測データのロード

```
THEMIS> timespan, '1994-9-28', 3
```

```
THEMIS> iug_load_aws_rish,site=['sgk']
```

```
THEMIS> tplot_names
```

22 iug_aws_sgk_press	→	気圧データ
23 iug_aws_sgk_temp	→	温度データ
24 iug_aws_sgk_rh	→	相対湿度データ
25 iug_aws_sgk_uwnd	→	東西風データ
26 iug_aws_sgk_vwnd	→	南北風データ

```
THEMIS> tplot, ['iug_aws_sgk_press', 'iug_aws_sgk_temp', 'iug_aws_sgk_rh',  
'iug_aws_sgk_uwnd', 'iug_aws_sgk_vwnd']
```

3. tplot変数を用いた各種データ解析

3.1 tsub_average で平均値を差し引く

tsub_average, 'tplot変数名'

(例) tsub_average, 'iug_aws_sgk_press'

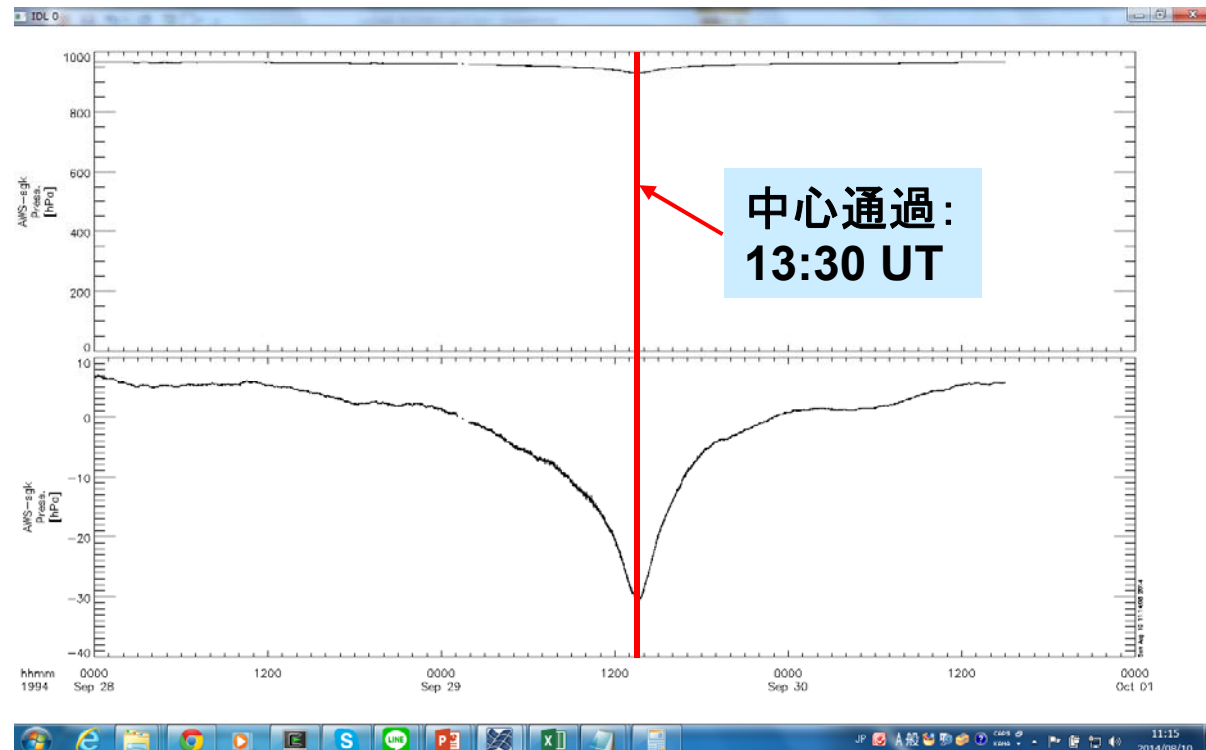
```
THEMIS> tsub_average, 'iug_aws_sgk_press'
```

```
THEMIS> tplot, ['iug_aws_sgk_press', 'iug_aws_sgk_press-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。

9月29日13時過ぎに気圧が最も低くなっている

→**台風26号の中心通過**



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

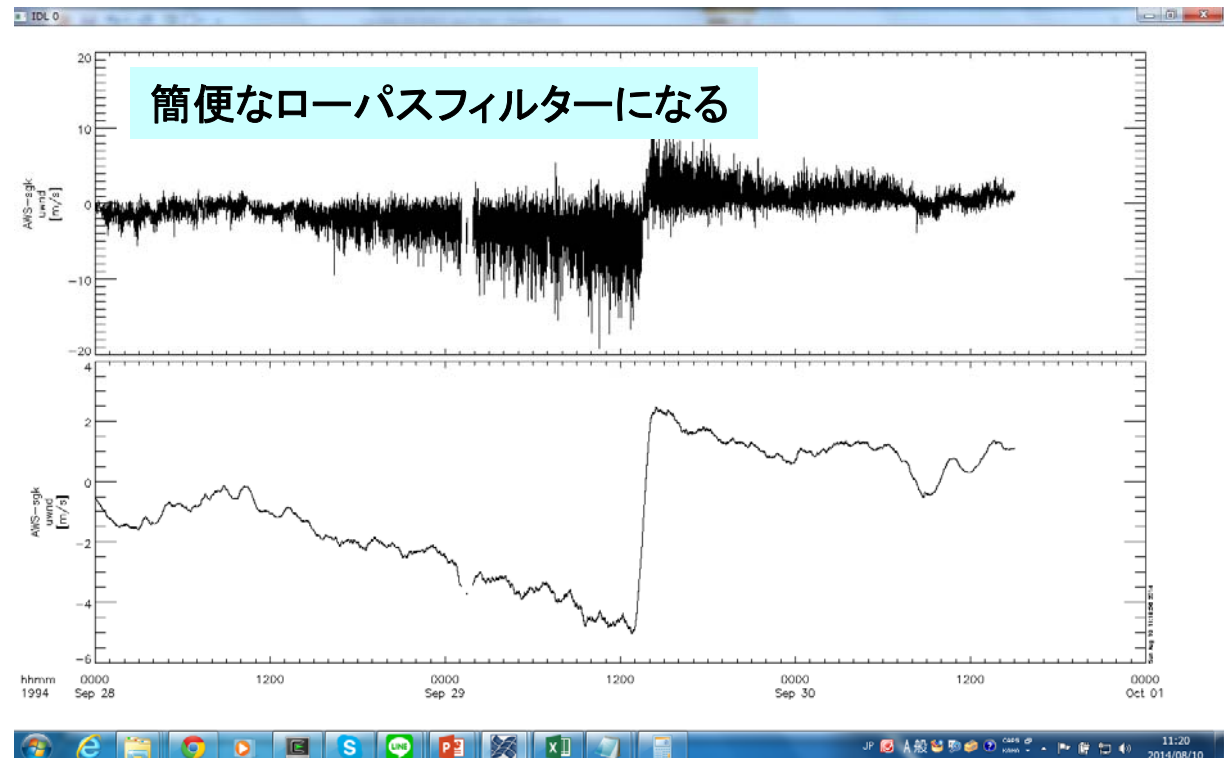
tsmooth_in_time, 'tplot変数名', 平均幅[秒]

(例) tsmooth_in_time, 'iug_aws_sgk_uwnd', 3600

```
THEMIS> tsmooth_in_time, 'iug_aws_sgk_uwnd', 3600
```

```
THEMIS> tplot, ['iug_aws_sgk_uwnd', 'iug_aws_sgk_uwnd_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

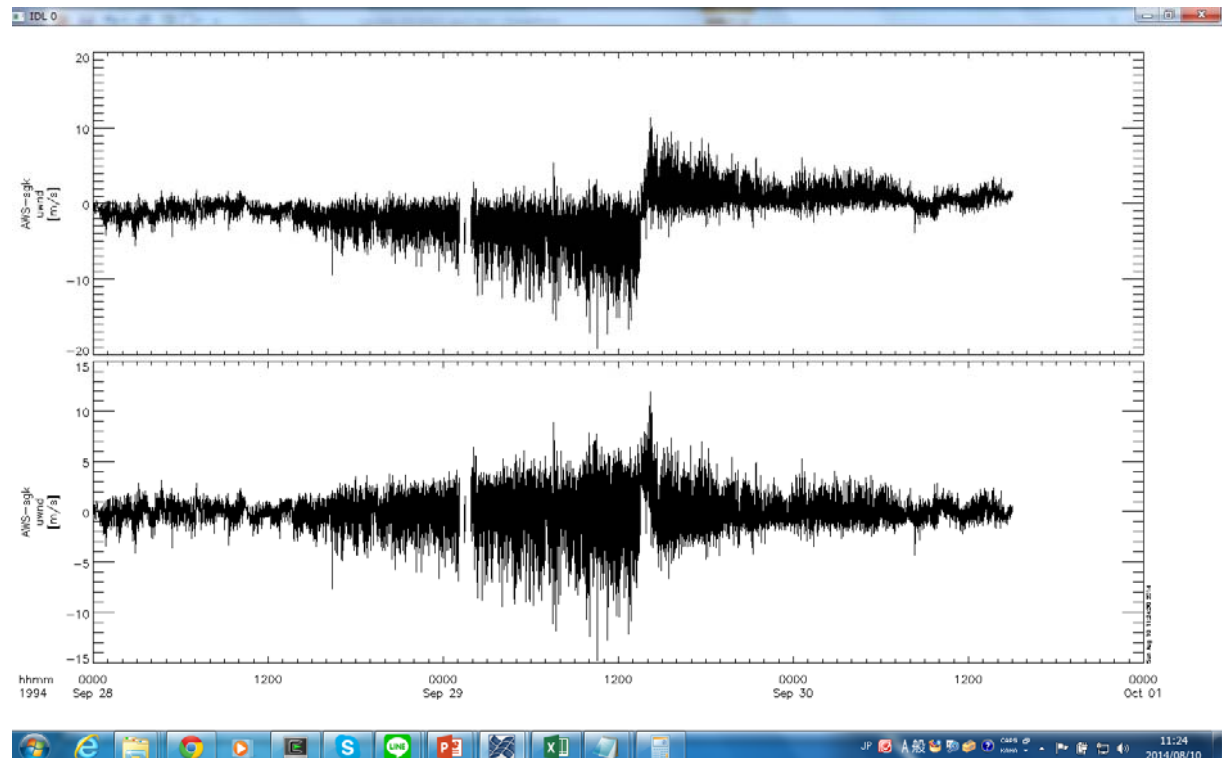
thigh_pass_filter, 'tplot変数名', 下限周期[秒]

(例) thigh_pass_filter, 'iug_aws_sgk_uwnd', 3600

```
THEMIS> thigh_pass_filter, 'iug_aws_sgk_uwnd', 3600
```

```
THEMIS> tplot, ['iug_aws_sgk_uwnd', 'iug_aws_sgk_uwnd_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. tplot変数への各種フィルター処理

3.4 avg_dataで~分値、~時間値に平均

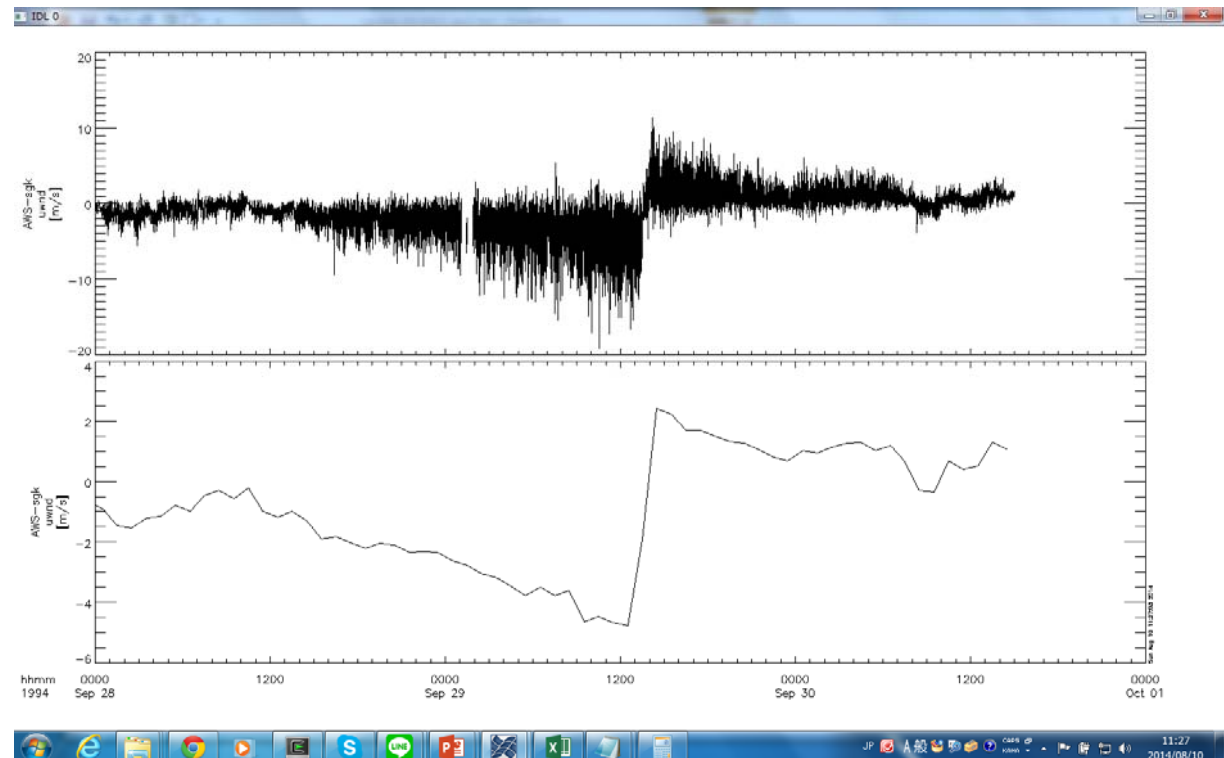
avg_data, 'tplot変数名', 平均時間幅[秒]

(例) avg_data, 'iug_aws_sgk_uwnd', 3600

```
THEMIS> avg_data, 'iug_aws_sgk_uwnd', 3600
```

```
THEMIS> tplot, ['iug_aws_sgk_uwnd', 'iug_aws_sgk_uwnd_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. 周波数スペクトル解析

4.2 フーリエスペクトル解析 tdpwrspec

tdpwrspec, 'tplot変数名'

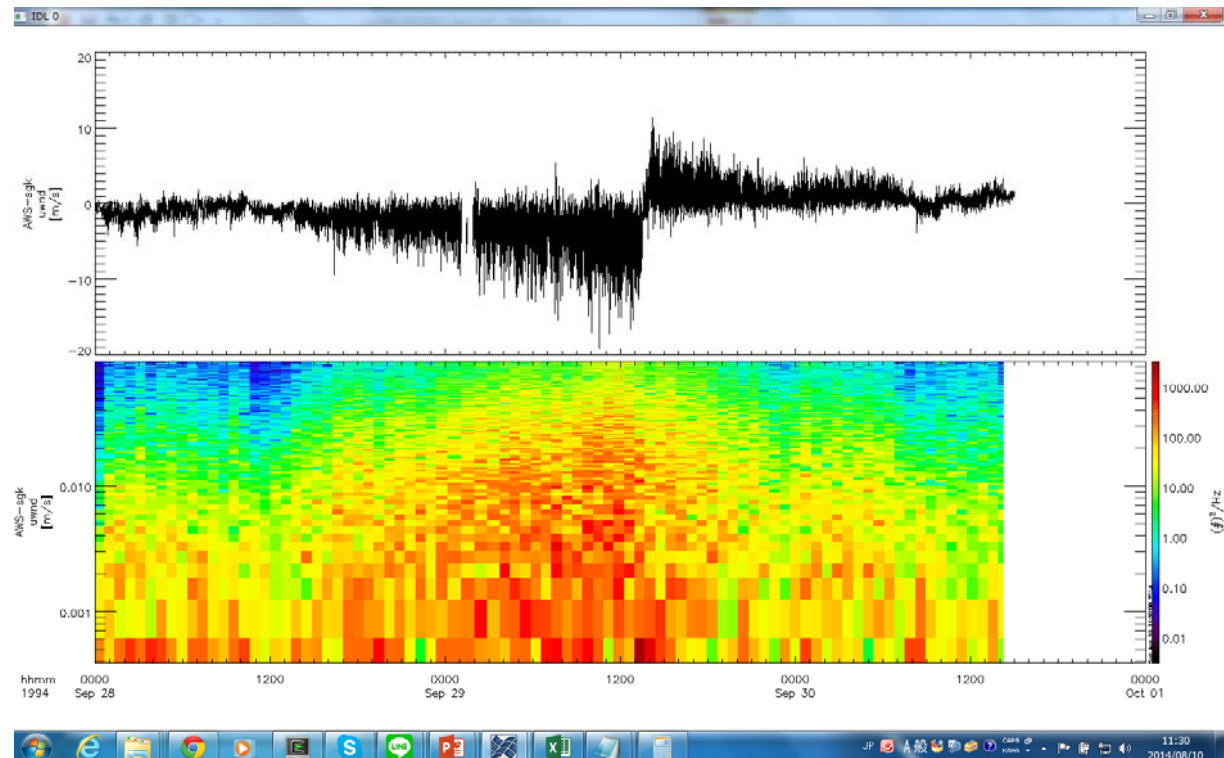
(例) tdpwrspec, 'iug_aws_sgk_uwnd'

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> tdpwrspec, 'iug_aws_sgk_uwnd'
```

```
THEMIS> tplot, ['iug_aws_sgk_uwnd', 'iug_aws_sgk_uwnd_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 dB²/Hz)
- 縦軸のキャプションは、optionsコマンドで適宜修正する。



4. 周波数スペクトル解析

4.2 ウェーブレット変換 wav_data

wav_data, 'tplot変数名'

(例) wav_data, 'iug_aws_sgk_uwnd'

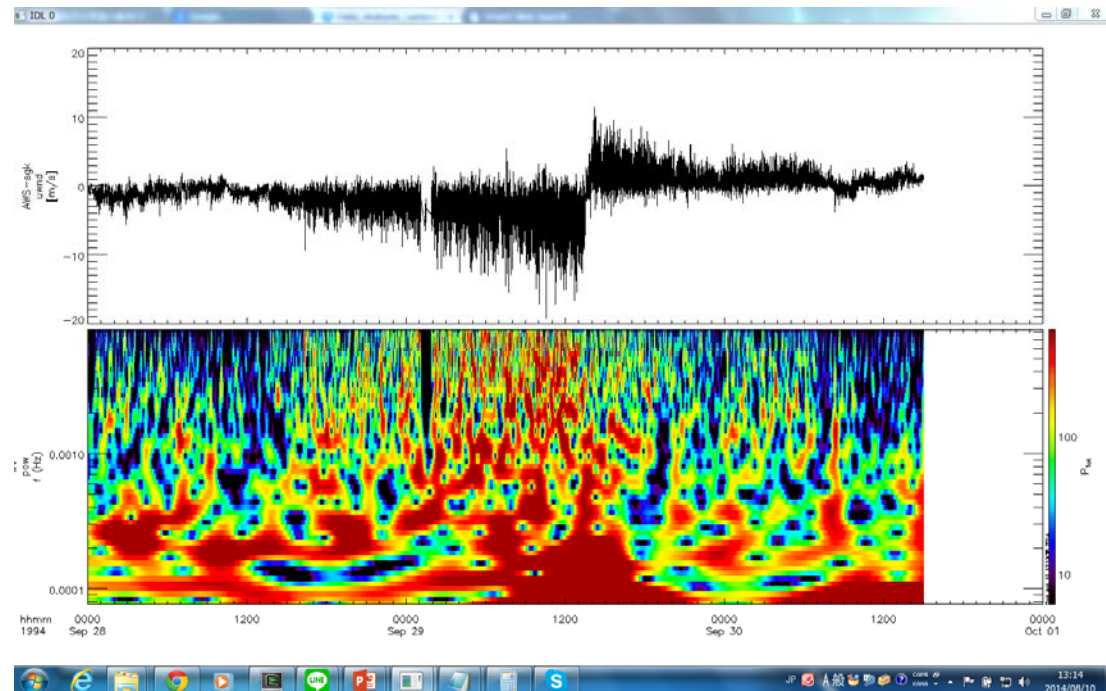
Wavelet変換で周波数
スペクトルを求める

THEMIS> avg_data, 'iug_aws_sgk_uwnd', 60 1分平均値の計算

THEMIS> wav_data, 'iug_aws_sgk_uwnd_avg'

THEMIS> tplot, ['iug_aws_sgk_uwnd', 'iug_aws_sgk_uwnd_avg_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1万点程度にした方がよい。**



4. 周波数スペクトル解析

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`, 'iug_aws_sgk_uwnd', /sampling, /abs

```
THEMIS> avg_data, 'iug_aws_sgk_uwnd', 60
```

1分平均値の計算

```
THEMIS> ustrans_pwrspc, 'iug_aws_sgk_uwnd_avg'
```

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

単位の変更

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

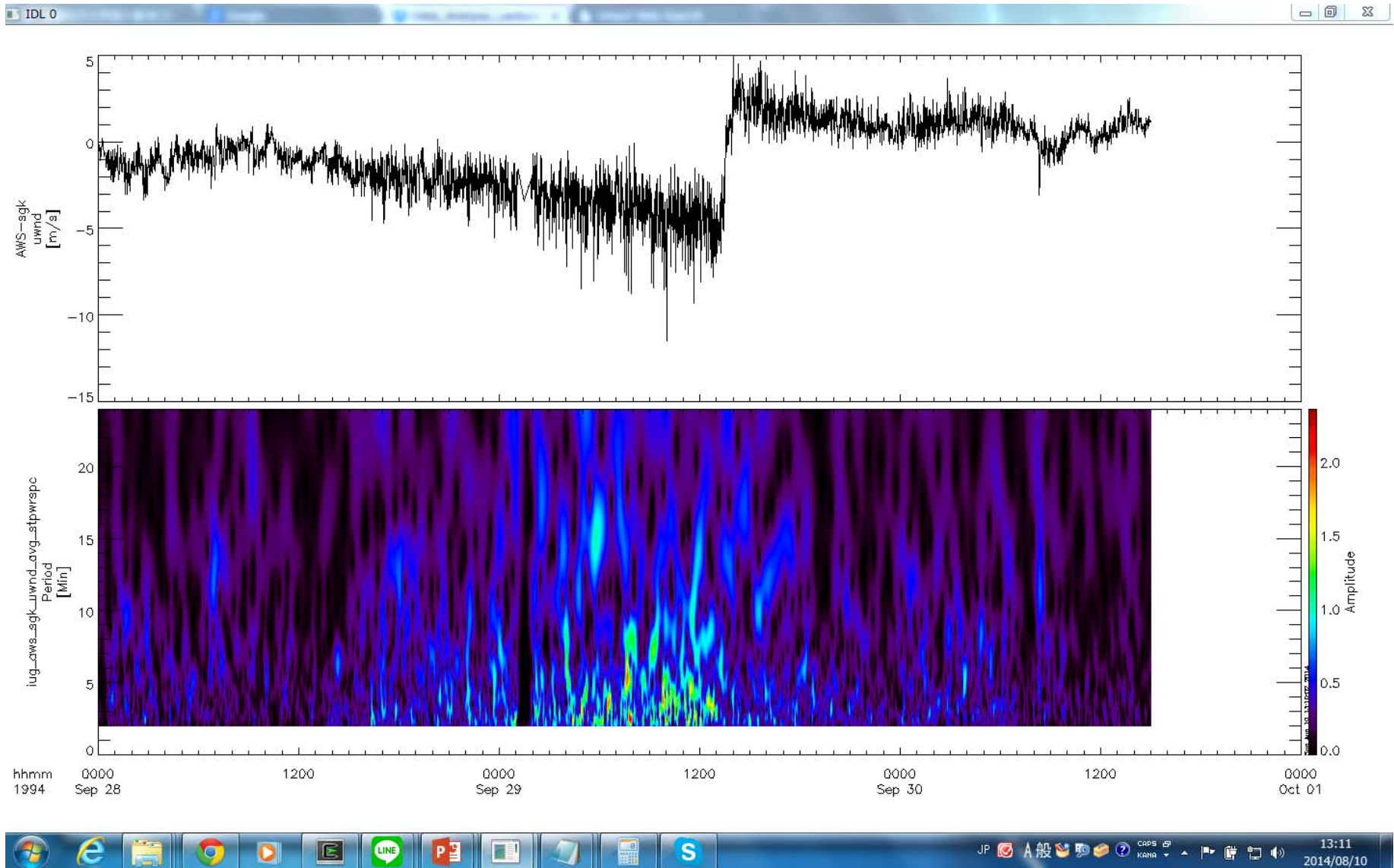
Y軸の範囲変更

```
THEMIS> tplot, ['iug_aws_sgk_uwnd_avg',  
                'iug_aws_sgk_uwnd_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1万点程度にした方がよい。**

4. 周波数スペクトル解析

4.3 S(Stockwell)変換 ustrans_pwrspc



- 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。