



中級編-2:

2次元データ解析 1 (オーロラ画像、
SuperDARNレーダーデータ等)

堀 智昭 (名古屋大・STE研)



解析ツールを使った講習の前に...

- ▶ IDLが起動したままになっていたら、一旦終了→再起動させて下さい。
メモリーをクリア, カラーテーブル初期化
- ▶ IDLのコマンド検索パスの順番で、SDツールフォルダ、UDASフォルダがTDASフォルダより先、になっていることを確認して下さい。
- ▶ この講習で使用するSDツールはSPEDAS-j githubサイトにあります。
 - ▶ https://github.com/spedas-j/erg_sd_tool/tree/iugws_201408
 - ▶ またコマンドリストも以下にありますので、コピーして実行したりできるようにWebブラウザ等で開いておいて下さい。
 - ▶ https://github.com/spedas-j/erg_sd_tool/blob/iugws_201408/iugonet_ws_advanced_2_crib.pro
- ▶ 今回の講習後に自習したり辞書的に使ってもらえるように、この講習資料は網羅的に書いてあります。ただ今日の講習では時間が限られているので、RTIプロット、2D地図プロットのそれぞれ重要な箇所についてのみやります。



中級編-2 2次元データ解析1の内容

- ▶ 2次元データの可視化・解析をやってみる。
- ▶ 1では2次元データの例としてSuperDARNデータを用いる。(2では全天カメラデータ)
 - ▶ 典型的な可視化形態であるRTIプロット、2Dマッププロットを作成できるようにする。
 - ▶ 地図上にデータをプロットする方法を学ぶ。

- ▶ 使用するデータはERG Science Center (ERG-SC) (<http://ergsc.stelab.nagoya-u.ac.jp>)から公開されているCDF形式のCommon time fitacfデータ
 - ▶ 詳細はHori et al., Adv. Polar Sci., 2013 を参照のこと。
<http://journal.polar.gov.cn/EN/10.3724/SP.J.1085.2013.00069>

謝辞

- ▶ SuperDARNデータは各レーダーのPIから提供されている。
- ▶ この講習で使うIDLプログラムはERG-SCとIUGONETとの共同開発
- ▶ ERG-SC plug-inは、THEMISチームによって開発されたTHEMIS Data Analysis Software suite (TDAS, <http://themis.ssl.berkeley.edu/software.shtml>) をベースにしている



SuperDARNレーダーとは?

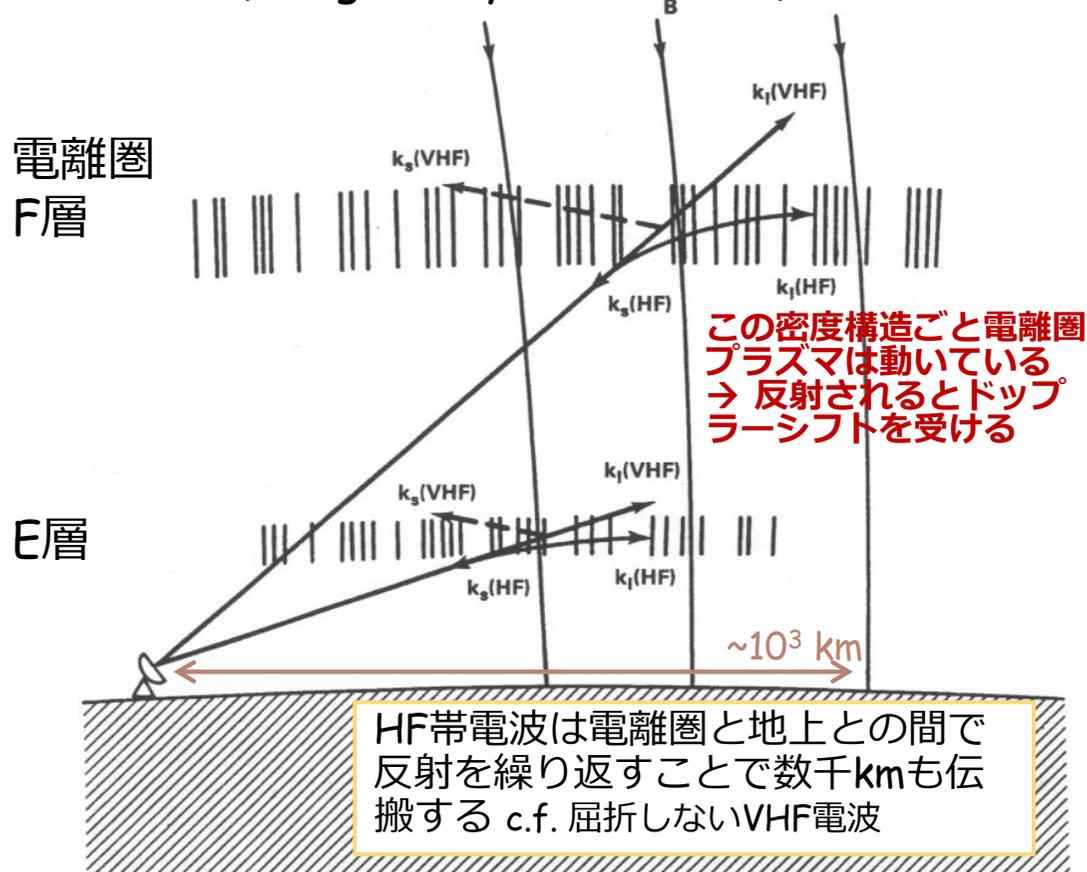


Hokkaido HF radar

- ▶ **Super Dual Auroral Radar Network (SuperDARN)**
 - ▶ HF帯電波を用いたcoherentレーダーの世界的ネットワーク
- ▶ **観測対象**
 - ▶ 電離圏F層、E層のプラズマの動き(電場ドリフト)
 - ▶ 中間圏エコー、熱圏風速(ある仮定の下で)、...

短波(HF)レーダーによる電離圏プラズマ速度計測

HF帯・VHF帯電波のray pathと電離圏電子密度揺らぎ(irregularity, 図中の縦縞)による反射

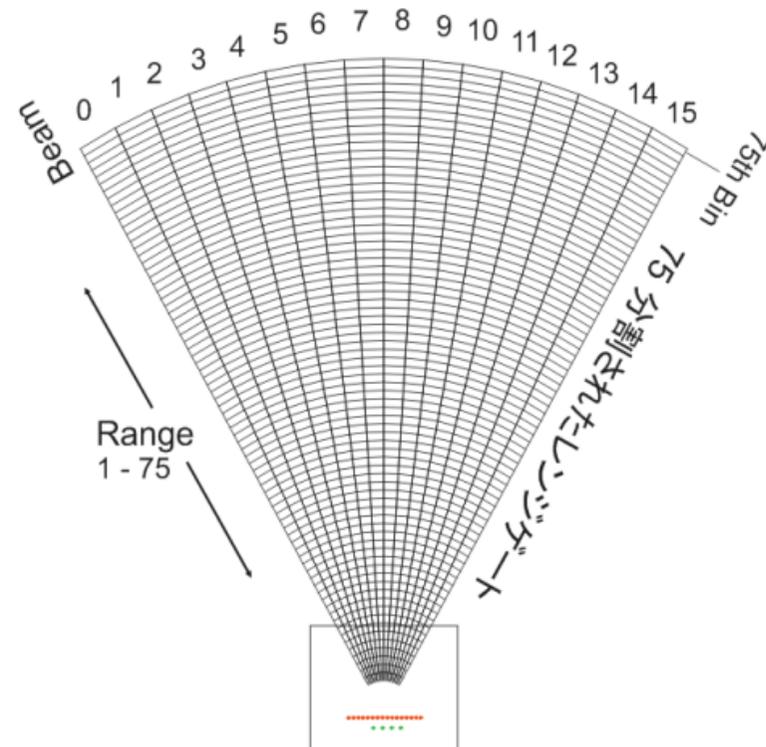
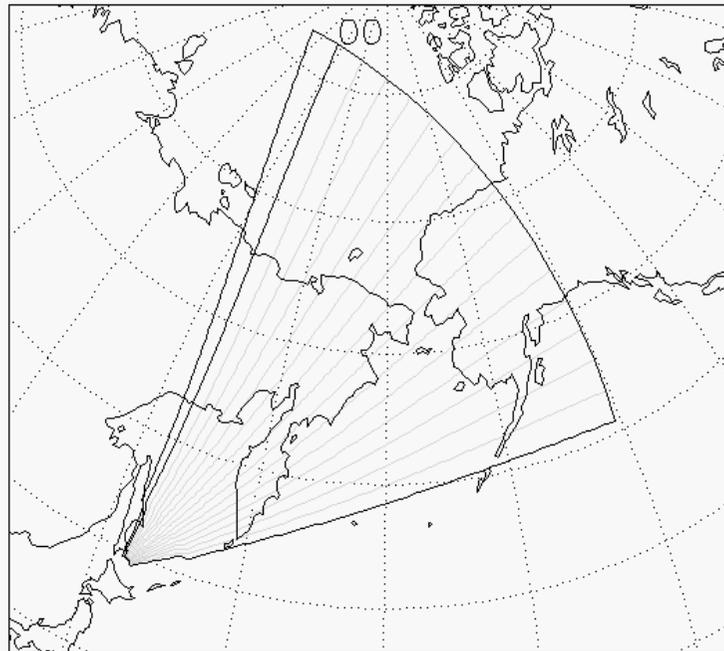


[Greenwald+1995]

- ▶ HF帯電波は電子密度の勾配で屈折する。
- ▶ 電離層中のdecameter ($\sim 10^1$ m) スケールのirregularityでBragg散乱を受け、極一部が後方散乱される。
- ▶ 散乱されて戻ってきた電波のドップラーシフトを計測することで電離圏プラズマの水平ドリフト速度を知ることができる。
- ▶ この密度縞が無いとエコーが返ってこないので計測できない。

短波(HF)レーダーによる電離圏プラズマ速度計測

1 beamを3秒、扇型を一通りスキャンするのに1分かかる

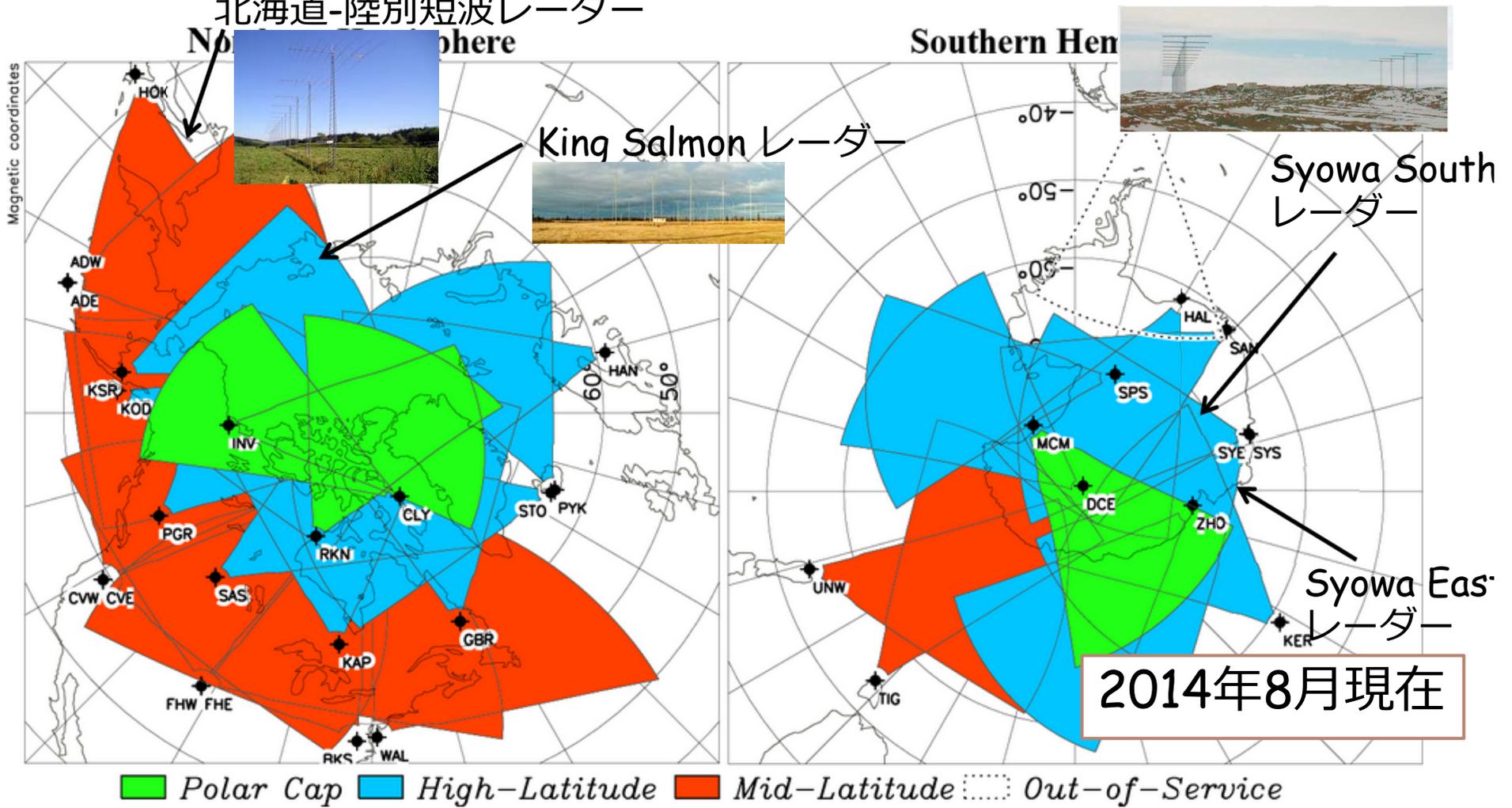


- ▶ ドップラー速度などの計測を、送信ビームに沿って75-110 range gateに分割した各ピクセル毎に行う(最近は24 Beamのレーダーも)。
- ▶ ビームを方位角方向に振ることで扇型の領域内をスキャンする。



全SuperDARNレーダーの視野

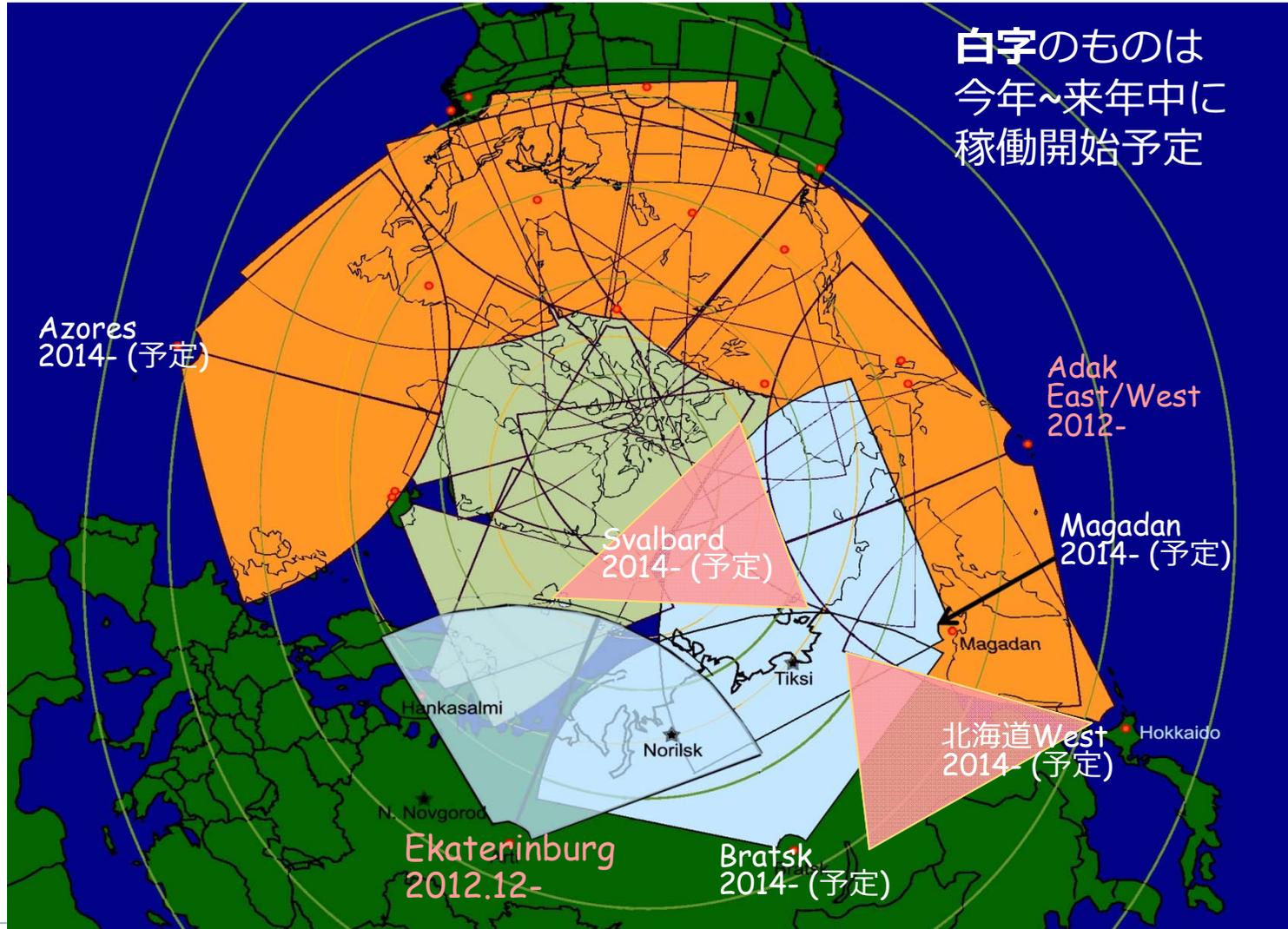
北海道-陸別短波レーダー



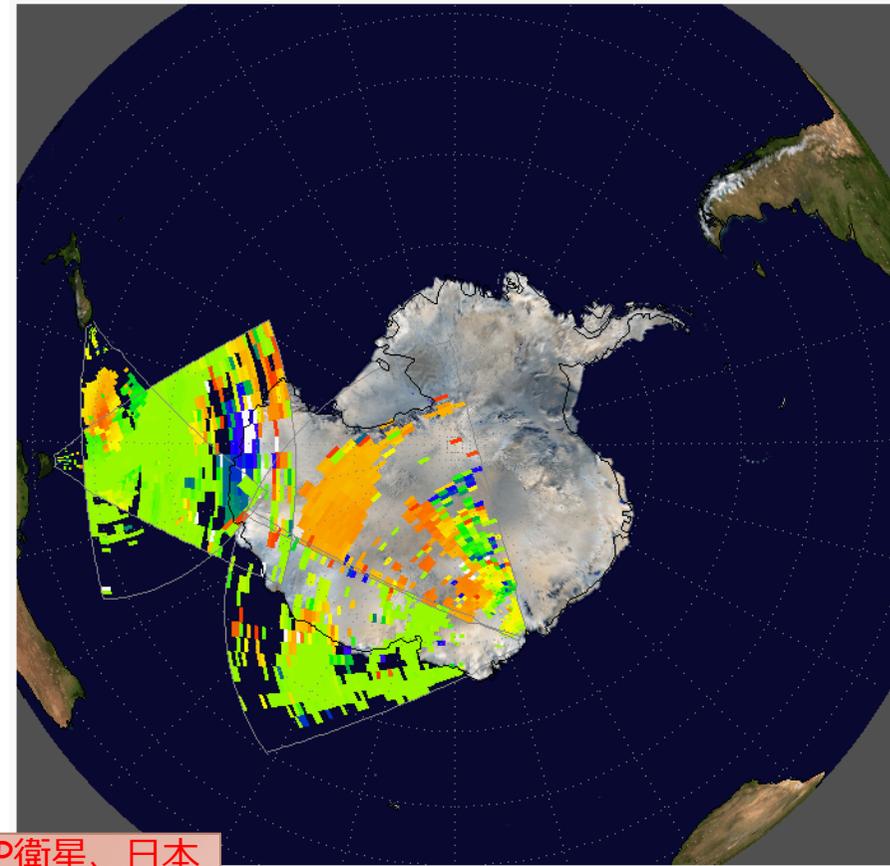
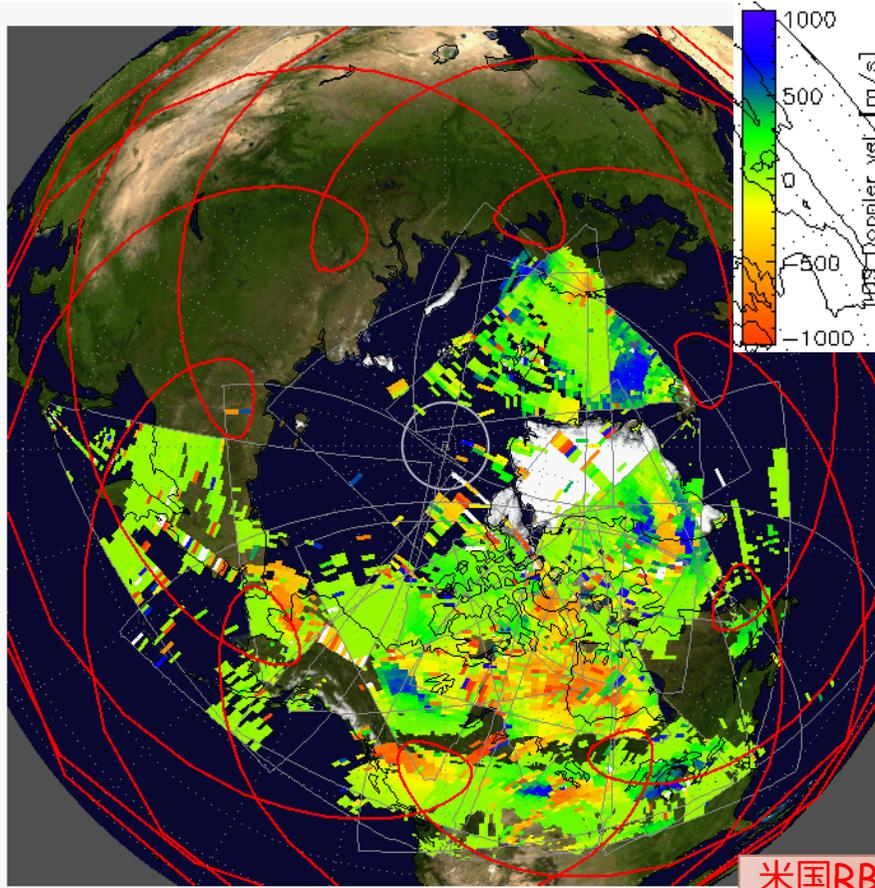
From <http://vt.superdarn.org/tiki-index.php?page=Radar+Overview>



拡大するSuperDARN (2012-2014)



SDERG-SC plug-inで使える SuperDARNレーダーデータの視野



米国RBSP衛星、日本
ERG衛星(2015年打
ち上げ予定)などの内
部磁気圏衛星の軌道

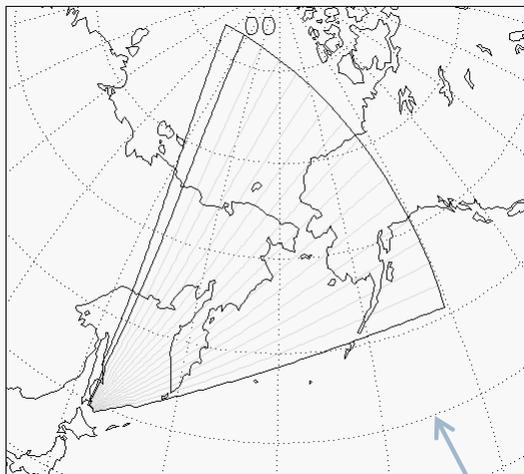
▶ これからも増える予定



SDによる観測



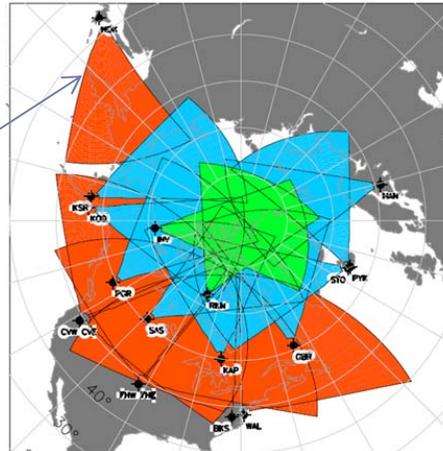
個々のレーダーによる視線方向Doppler速度、エコー強度、スペクトル幅の観測 (~10秒-1分)



1ビームに3秒→ 2次元スキャン(通常16ビーム)は1分で得られる

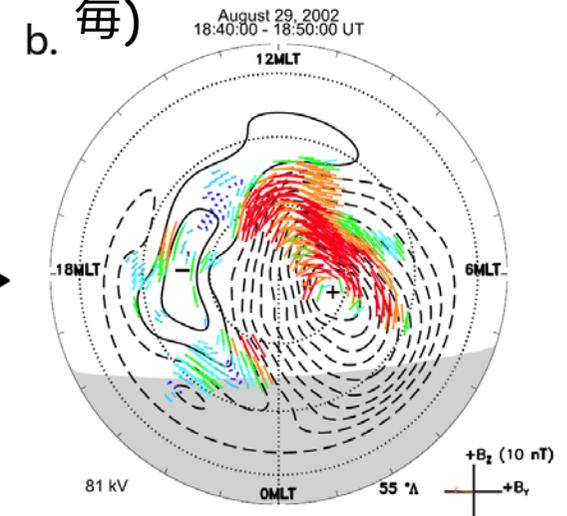
ERG-SC plug-in のSDライブラリ、及びこの講習資料はこっこのプロット形態用。

国際ネットワーク観測



SD観測から得られる対流パターンの経験モデル

データ同化で得られる対流パターン (2分毎)



[Cousins+2010 (the original technique developed by Ruohoniemi+1998)]

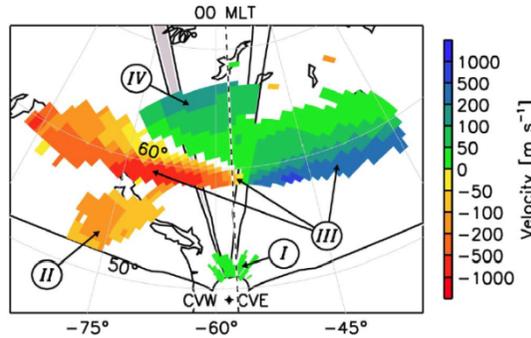
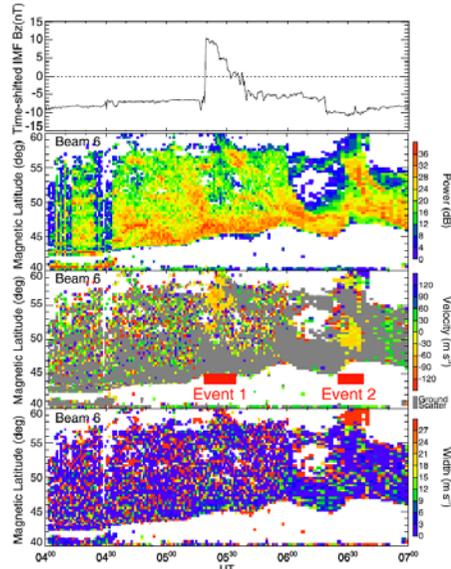
マップポテンシャル
と呼ばれる

マップポテンシャルプロットを作るには Virginia Tech のサイト(この資料の最後にある補足資料を参照)でやるのがよい。またSDレーダーPI に依頼してもよい。

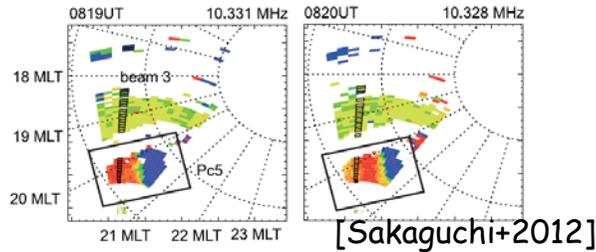
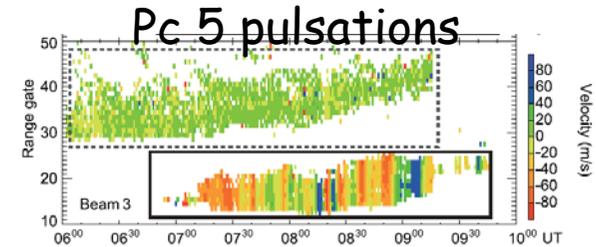


SDでどんな研究ができるか？

特に個々のレーダーのデータ

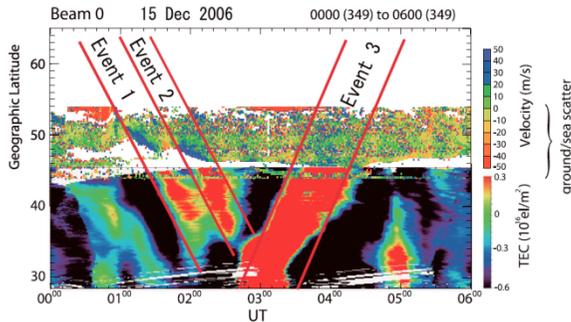


SAPS evolution [Clausen+2012]

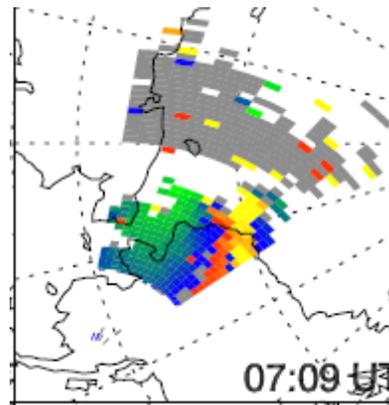


[Sakaguchi+2012]

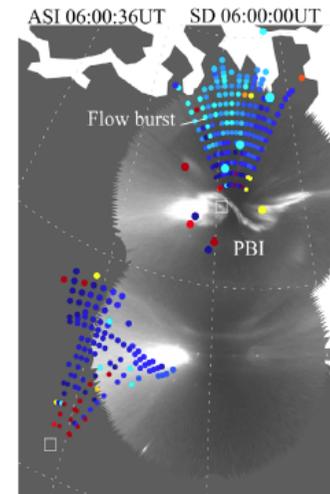
Storm-time convection [Ebihara+2008]



Large-scale traveling ionospheric disturbance (LSTID) [Hayashi+2010]



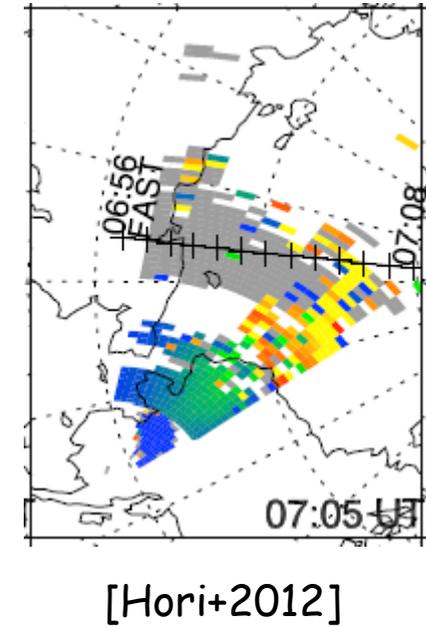
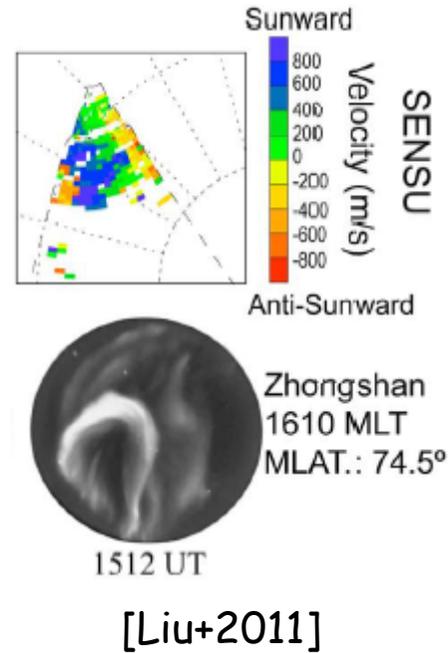
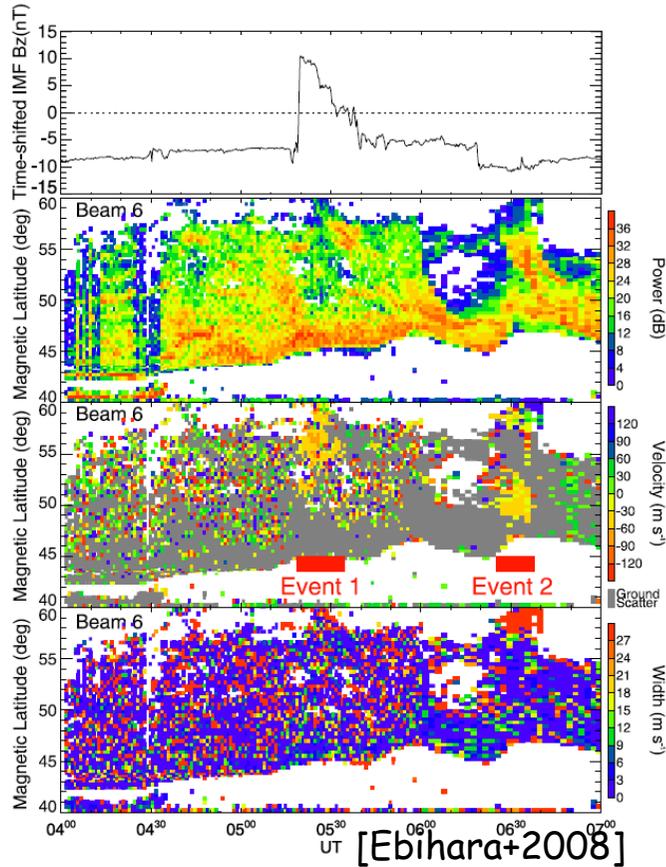
SC/SI-induced transient flows [Hori+2012]



Flow evolution on substorm onset [Nishimura+2010]



よく使用するプロット形式は2つ



Range-Time-Intensity (RTI) プロット
 の例
 変動のタイミング・周期を詳細に見るために使われる

緯度-経度グリッド上での**2D地図プロット**の例

ある時刻の2D空間プロファイルを見たり、他の地上・衛星観測との位置関係を見るために使われる

SDERG-SC plug-in のSDライブラリに集録されているSD関係のツール



データ読込

- ▶ `erg_load_sdfit`
SuperDARNデータをダウンロード・読込

データ加工・RTIプロット

- ▶ `splitbeam`
tplot変数をビーム別に分割
- ▶ `set_coords`
縦軸の座標系を変更
- ▶ `loadct_sd`
カラーテーブルをセット
- ▶ `get_fixed_pixel_graph`
1つのピクセルの値を独立したtplot変数にする
- ▶ `get_sd_ave (*)`
ある緯度、経度範囲の平均値を計算
- ▶ `get_sd_lat_profile (*)`
緯度プロファイルのテオグラムを作成
- ▶ `asciidump_scan`
データをアスキーダンプ

2D地図プロット環境設定

- ▶ `map2d_init (旧sd_init)`
2D地図描画関係の変数初期化
- ▶ `map2d_time (旧sd_time)`
2D地図プロット用の時刻をセット

- ▶ `map2d_set (旧sd_map_set)`
地理座標及びAACGM座標緯度-経度グリッドを定義
- ▶ `map2d_coord`
2D地図プロット用の座標系をセット
- ▶ `map2d_grid`
緯度-経度グリッドを点線で描画

SuperDARNデータの2D地図プロット

- ▶ `plot_map_sdfit`
SDデータを2D地図上にプロットするall in one コマンド
- ▶ `overlay_map_sdfit`
データを緯度-経度グリッド上にプロット
- ▶ `overlay_map_sdfov`
レーダーの視野をプロット
- ▶ `make_fanplot_pictures`
2Dプロットをまとめて作成

2D地図プロット関連

- ▶ `overlay_map_coast`
地図(海岸線)をプロット
- ▶ `overlay_map_vec (*)`
2D地図上に線を描画

(*) 今回は資料に入れられなかったもので、興味のある方はソースコード先頭のドキュメントをご覧ください。

データの読み込み



レーダーのデータを読み込む

- ▶ **erg_load_sdfit** を使う
 - ▶ `get_support_data` キーワードをセットすることで、座標変換に必要なパラメータなども一緒に読み込む
- `site` キーワードでレーダー名を指定する (以下以外もある)
- `hok`: 北海道-陸別短波レーダー
 - `ksr`: King Salmon レーダー (アラスカ)
 - `sye`: Syowa East レーダー (南極昭和基地)
 - `sys`: Syowa South レーダー (")
 - `pyk`: Pykkvibaer (Iceland East) レーダー (アイスランド)
 - `han`: Hankasalmi レーダー (フィンランド)

`erg_load_sdfit` コマンド 1 つで

`fitacf` データを CDF ファイルにしたものを自動ダウンロード (ローカル PC 上に保存される)

↓
各種パラメータを `tplot` 変数として IDL 上に読み込む

をやってくれる

データの保存先は
Windows: `c:\¥data`
Unix系: `~/data` の下

```
THEMIS> timespan, '2013-09-23'
THEMIS> erg_load_sdfit, site='pyk',
/get_support_data
...
(ロードの過程がいっぱい表示される)
...
##### RULES OF THE ROAD
#####
Data distributed with this CDF file can be used
##### RULES OF THE ROAD
#####
THEMIS>
THEMIS> tplot_names
1 sd_pyk_cpid_2
2 sd_pyk_channel_2
3 sd_pyk_int_time_2
4 sd_pyk_azim_no_2
5 sd_pyk_pwr_2
6 sd_pyk_pwr_err_2
7 sd_pyk_spec_width_2
8 sd_pyk_spec_width_err_2
9 sd_pyk_vlos_2
...
..
```

Rules of the roadはデータ使用上の注意ですので、必ず内容を熟読してからデータをお使い下さい

`tplot_names` で現在メモリ上に読み込まれているデータ変数(`tplot`変数)をリスト表示

Range-Time-Intensity (RTI) プ ロットの作成



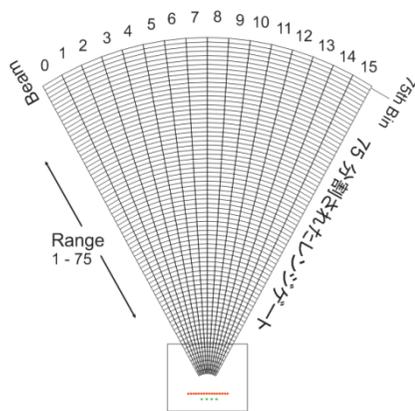
Range-Time-Intensity (RTI)プロット

- ▶ Backscatter power, Line-of-sight Doppler velocity (LOS V), Spectral width の時間変動をプロット

```
THEMIS> tplot, 'sd_pyk_pwr_2'
...
TPLOT(398): 5 sd_pyk_pwr_2
...

THEMIS>

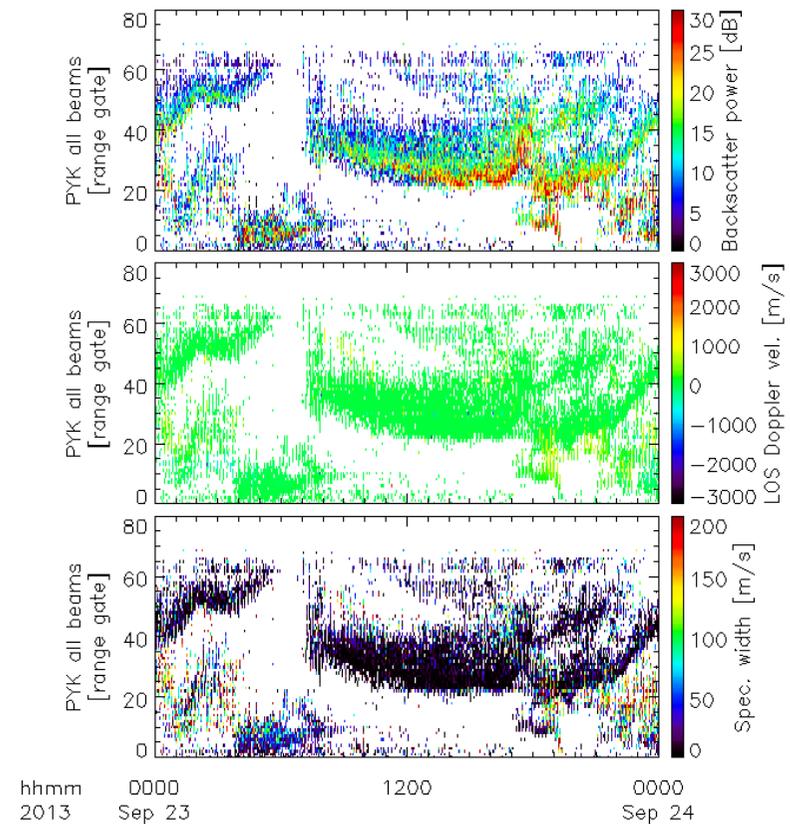
THEMIS> tplot, ['sd_pyk_pwr_2',
'sd_pyk_vlos_2', 'sd_pyk_spec_width_2']
TPLOT(410): 5 sd_pyk_pwr_2
TPLOT(410): 9 sd_pyk_vlos_2
TPLOT(410): 7 sd_pyk_spec_width_2
THEMIS>
```



Beam0, Beam1, Beam2, ..., Beam15 の順に1ビーム約3秒ずつ観測していく。

右図ではビーム順に時間方向に並べてプロットしている。

この時Pykkvibaerレーダーは70レンジゲートモードだった。





Range-Time-Intensity (RTI)プロット

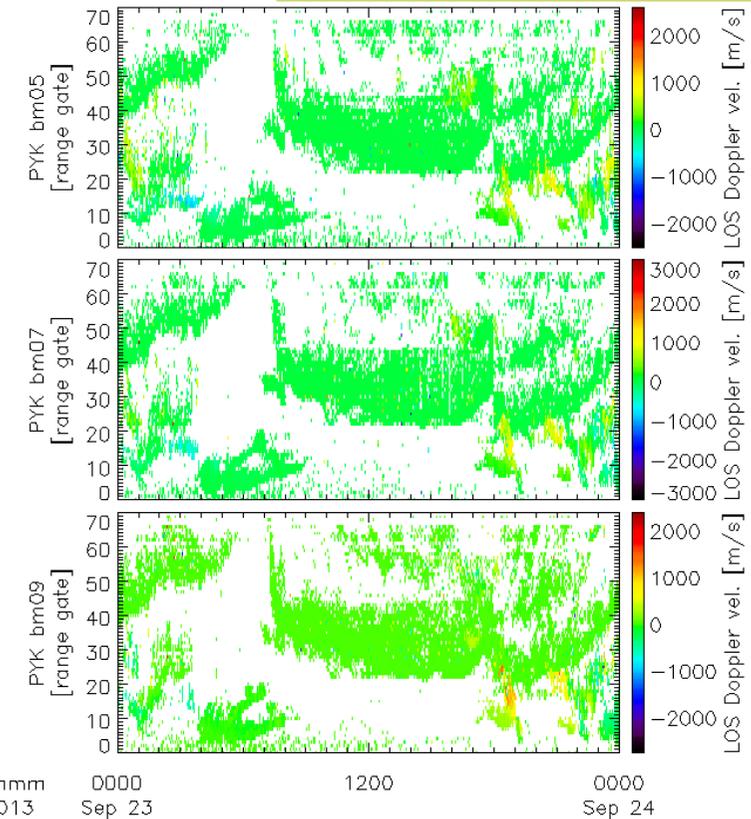
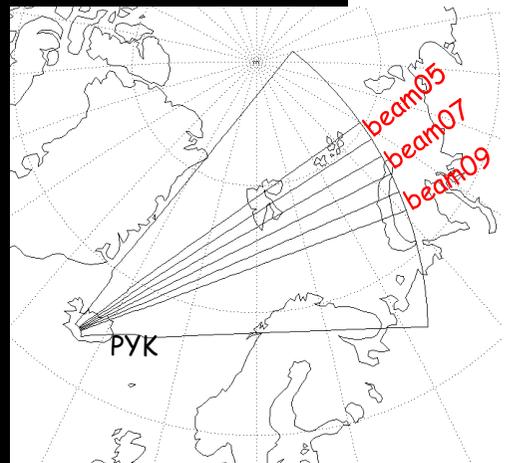
▶ Beamに分割して複数beamを並べる

RTIプロットでは通常各beamごとに時間変化を見ることが多い。

```

THEMIS> splitbeam, 'sd_pyk_vlos_2'
THEMIS> tplot_names
40 sd_pyk_vlos_2_azim00
41 sd_pyk_vlos_2_azim01
42 sd_pyk_vlos_2_azim02
43 sd_pyk_vlos_2_azim03
44 sd_pyk_vlos_2_azim04
45 sd_pyk_vlos_2_azim05
...
...
49 sd_pyk_vlos_2_azim09
50 sd_pyk_vlos_2_azim10
51 sd_pyk_vlos_2_azim11
52 sd_pyk_vlos_2_azim12
53 sd_pyk_vlos_2_azim13
54 sd_pyk_vlos_2_azim14
55 sd_pyk_vlos_2_azim15
THEMIS>
THEMIS> tplot,
['sd_pyk_vlos_2_azim05', 'sd_pyk_vlos_2_azim07', 'sd_pyk_vlos_2_azim09']
;長いけど tplot から ..azim20*]
まで1行で書く
TAPLOT(410): 56 sd_pyk_vlos_2_azim05
TAPLOT(410): 58 sd_pyk_vlos_2_azim07
TAPLOT(410): 60 sd_pyk_vlos_2_azim09
THEMIS>

```



Beam05, 07, 09 を並べてみた。しかしカラーバーのスケールが合っていない...



Range-Time-Intensity (RTI)プロット

▶ カラーバーのスケールを変える

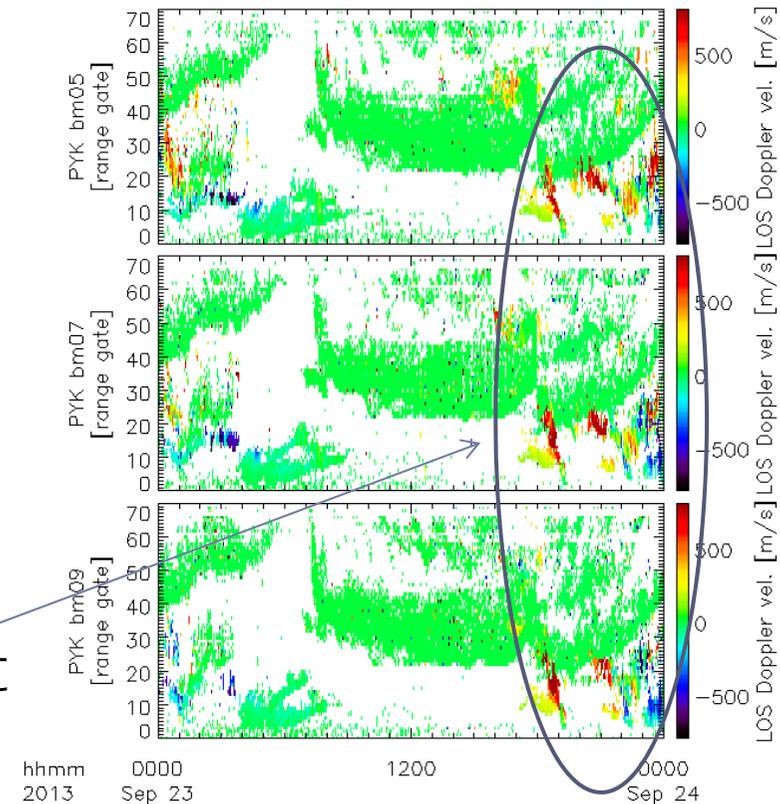
```

THEMIS> zlim, 'sd_pyk_vlos_2_azim??', -800, 800
THEMIS>
THEMIS> tplot :tplotのみだと前回のプロットしたtplot変数を再プロットする
TPLOT(486): 45 sd_pyk_vlos_2_azim05
TPLOT(486): 47 sd_pyk_vlos_2_azim07
TPLOT(486): 49 sd_pyk_vlos_2_azim09
THEMIS>

```

zlim, 'sd_pyk_vlos_2_azim??' のようにワイルドカードを使うことで、...azim00 ~ ...azim15 全てについて、カラーバーの上下限を設定できる。
 * (アスタリスク)もワイルドカードとして使える。

カラーバーの上限下限を-800, +800 [m/s] にしたらDoppler速度の変動が浮き出てきた！





Range-Time-Intensity (RTI)プロット

▶ プロットの時間幅を変える

```

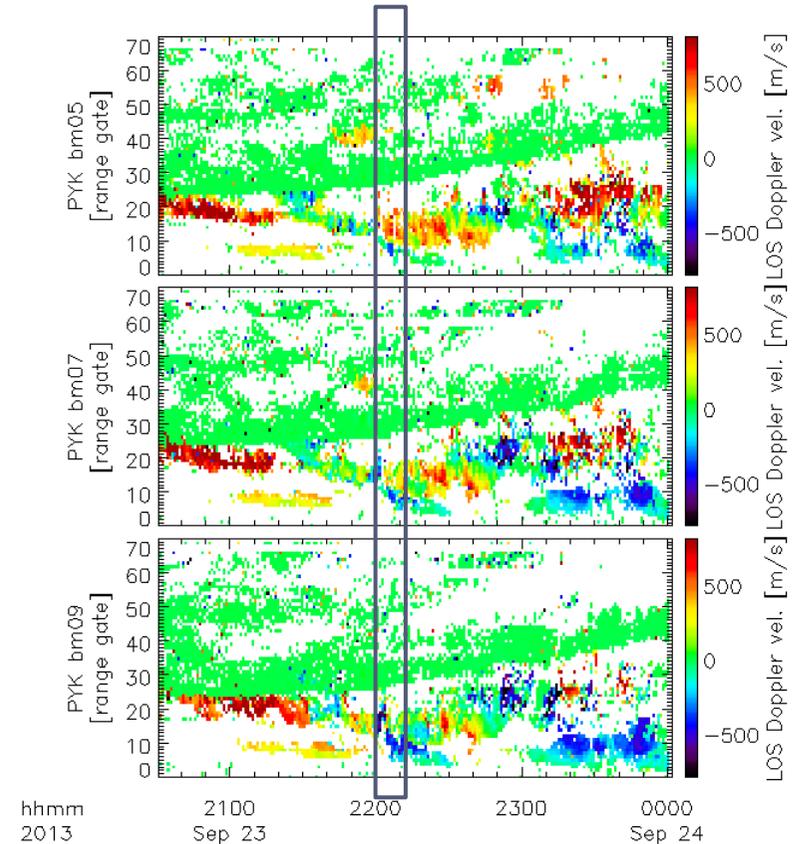
THEMIS> tlimit
;ウィンドウに十字スケールが出てくるので、プロットしたい時間帯の
;最初の辺に合わせて左クリック1回
;最後の辺りに合わせて左クリック1回、 で時間幅を選択できる
;右の例では 20:30, 00:00(右端) 辺りをクリックして得られたプロット

THEMIS> tlimit, /last ;時間幅を前回プロットした時の時間幅に戻す
THEMIS> tlimit, /full ;時間幅をtimespan で指定してある幅に戻す
THEMIS> tlimit, '2013-09-23/20:30', '2013-09-24/00:00'
;時間幅を直書きして指定することもできる

```

tlimit + マウスクリック
tlimit, /last
tlimit, /full
tlimit, [開始時刻, 終了時刻]
を駆使することで、プロットの時間幅を自由に設定できる

よくやるのは、
tlimit+マウスクリック → 狙った時刻幅を少し外
したので tlimit, /last で元に戻す → 再度 tlimit,
とか、自由自在にプロット時刻幅を変更できる。



このように拡大すると、22 UT直後のドップラー速度が
beam05→07→09と赤色主体から青色になっていることがわかる。
→ 異なるbeam(つまり異なる方向)で速度が反転している！



Range-Time-Intensity (RTI)プロット

▶ 電離圏エコー、地上エコーを区別してプロット

sd_pyk_vlos_bothscat_2 は sd_pyk_vlos_iscat_2 と sd_pyk_vlos_gscat_2 の 2 つを指すマルチtplot変数なので、以下のコマンドで iscat, gscat 両方をビーム毎に分割することができる

```
THEMIS> splitbeam, 'sd_pyk_vlos_bothscat_2'
```

; 電離圏エコーのみをプロット

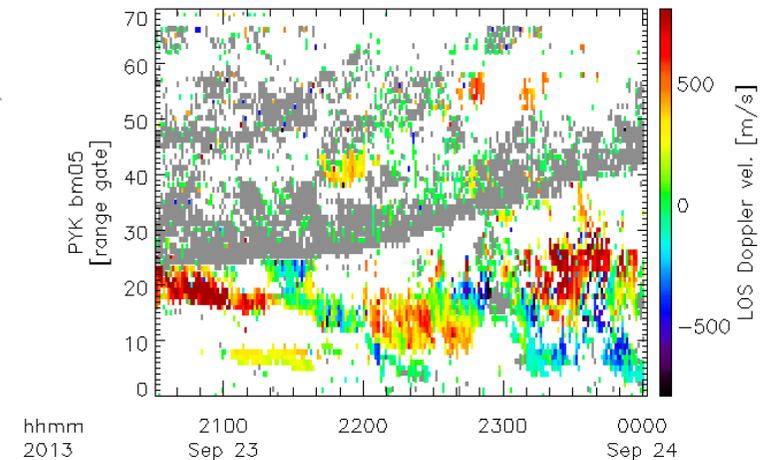
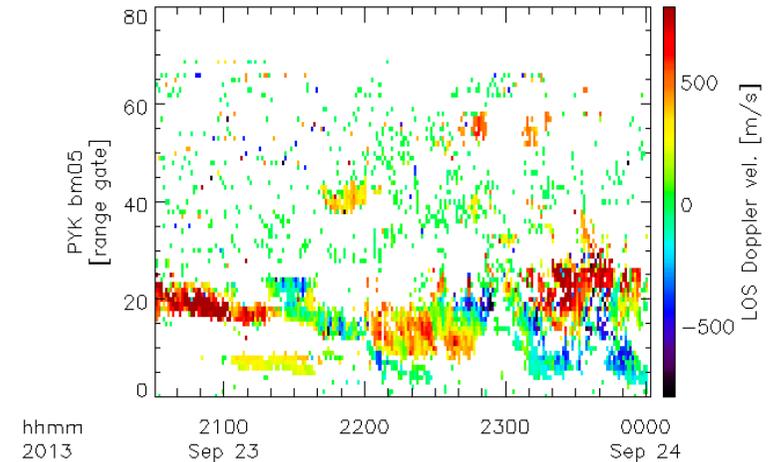
```
THEMIS> zlim, 'sd_pyk_vlos_*scat*', -800,800
```

```
THEMIS> tplot, 'sd_pyk_vlos_iscat_2_azim05'
```

; 電離圏エコーと地上エコー(灰色でマスク)をプロット

```
THEMIS> loadct_sd, 43 ; 灰色入りカラーテーブルをセット
```

```
THEMIS> tplot, 'sd_pyk_vlos_bothscat_2_azim05'
```



ただしここでの地上エコーの判定は経験式に基づいている. 実際にはエコー強度, Doppler速度, スペクトル幅などから総合的に判断する必要がある.



Range-Time-Intensity (RTI)プロット

▶ 縦軸をRange gateから地理緯度、地磁気緯度にする

```

; 縦軸を地理緯度に変換
THEMIS> set_coords, ['sd_pyk_vlos_2_azim??'], 'glat'
...
(縦軸をGeographical Latに変更, という表示)

; 再プロット (縦軸がrange gateから地理緯度になる)
THEMIS> tplot, ['sd_pyk_vlos_2_azim05', 'sd_pyk_vlos_2_azim09']

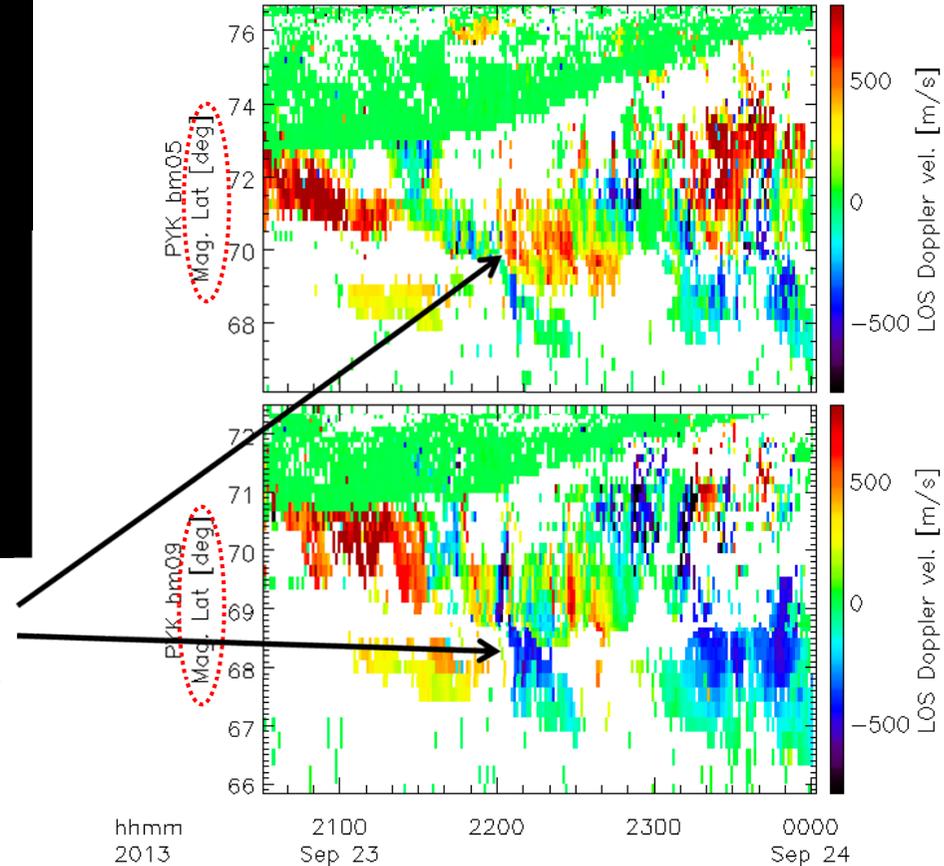
; 縦軸をAACGM緯度に変換
THEMIS> set_coords, ['sd_pyk_vlos_2_azim??'], 'mlat'
...
(縦軸をAACGM Latに変更, という表示)

THEMIS> tplot, ['sd_pyk_vlos_2_azim05', 'sd_pyk_vlos_2_azim09']

```

AACGM緯度~69°を境に、それより高緯度側(beam05)ではレーダーに向かう方向(つまり正の)のDoppler速度、低緯度側(beam09)ではレーダーから離れる向きの(負の)Doppler速度となるようなflow shear が観測されている!

AACGM: Altitude-Adjusted Corrected GeoMagnetic coordinates
 磁力線マッピングを考慮した地磁気座標の1つ
 [Baker+, JGR, 1989]
 (<http://www.aer.com/science-research/space/space-weather/altitude-adjusted-corrected-geomagnetic-coordinates>)



ylim コマンドを使えば縦軸の範囲も自由に変更可能!



Range-Time-Intensity (RTI)プロット

- ▶ 1つのpixelの時間変化を線プロットで描画する

```

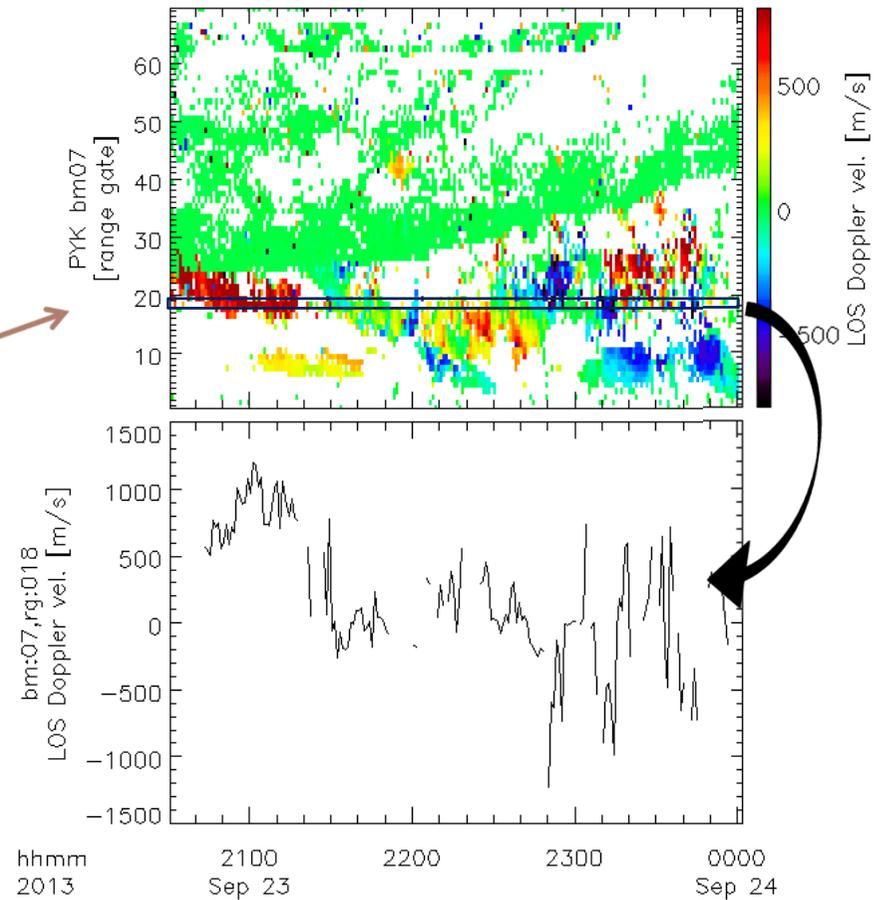
THEMIS> set_coords, 'sd_pyk_vlos_2_azim??', 'gate'
;再び縦軸をrange gateに変更

THEMIS> get_fixed_pixel_graph, 'sd_pyk_vlos_2', $
beam=7, range_gate=18
; beam07, range gate 18のpixelを選択

THEMIS> tplot, ['sd_pyk_vlos_2_azim07', $
'sd_pyk_vlos_2_bm07rg018 ' ]

; get_fixed_pixel_graph
; beam07, range gate 18 のpixel の値を取り出して、新しい
; tplot変数に格納する

```



速度の値をグラフで確認することができる

```
THEMIS> get_data, 'sd_cvw_vlos_8_bm17rg032', data=data
```

で、値を通常のIDL変数(実際には構造体)に代入することも可能



Range-Time-Intensity (RTI)プロット

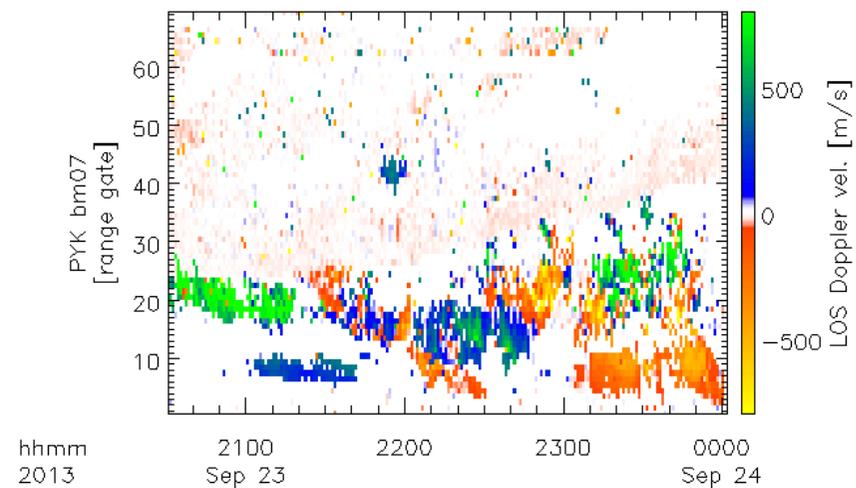
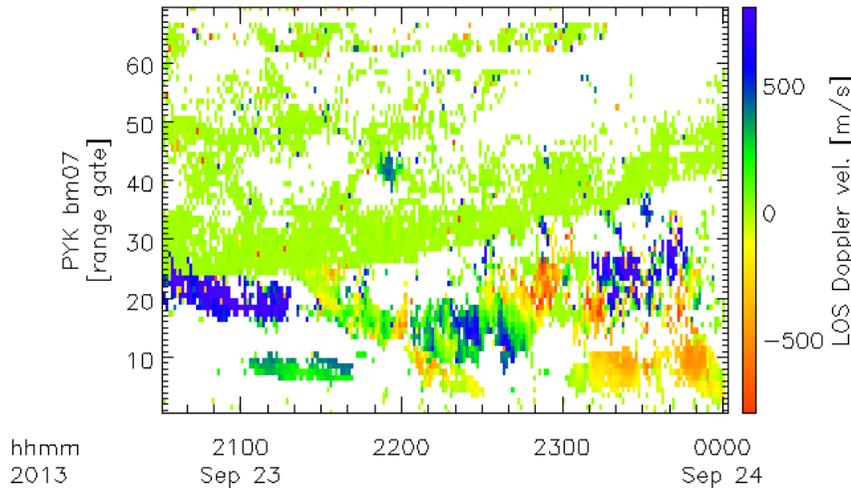
▶ カラーテーブルを変えてプロット

`/center` を付けるとカラーバーの真ん中辺りを透明にする。→ ゼロ付近を目立たないようにできる。他のカラーバーにも適用できる。

```
THEMIS> loadct_sd, 44
THEMIS> tplot, 'sd_pyk_vlos_2_azim07'
```

```
THEMIS> loadct_sd, 45, /center
THEMIS> tplot, 'sd_pyk_vlos_2_azim07'
```

次項の講習のために `loadct_sd, 43` で元に戻しておいて下さい



正の値を寒色系、負の値を暖色系で描画。`SD`プロットでは慣習的にこっちを使うことが多い。

ゼロのところを寒色系-暖色系を切り替える → 値の正負をわかりやすくする



LOSVDデータをアスキーダンプする

```
THEMIS>
asciidump_scan, 'sd_pyk_vlos_2'

The ascii files have been generated in
/home/idl/work/IDL8_project/tmp/asci
idump
Program finished.
```

- ▶ `asciidump_scan`コマンドを使うと、カレントディレクトリに`asciidump`というディレクトリが作成され、その下に各スキャン毎にアスキーファイルが作成される。
- ▶ 多数のファイルができるので注意(1スキャン1分 → 1日で1440個!)
- ▶ 今のところLOSVDデータ (`sd_STN_vlos_?` STNはレーダー名) のみに対応。

```
IDL - IDL 127.0.0.1:20000 - Tera Term VT
ファイル(F) 編集(E) 設定(S) コントロール(O) ウィンドウ(W) ヘルプ(H)
sd_pyk_vlos_2_201309231110.dat* sd_pyk_vlos_2_201309232237.dat*
sd_pyk_vlos_2_201309231111.dat* sd_pyk_vlos_2_201309232238.dat*
sd_pyk_vlos_2_201309231112.dat* sd_pyk_vlos_2_201309232239.dat*
sd_pyk_vlos_2_201309231113.dat* sd_pyk_vlos_2_201309232240.dat*
sd_pyk_vlos_2_201309231114.dat* sd_pyk_vlos_2_201309232241.dat*
sd_pyk_vlos_2_201309231115.dat* sd_pyk_vlos_2_201309232242.dat*
sd_pyk_vlos_2_201309231116.dat* sd_pyk_vlos_2_201309232243.dat*
sd_pyk_vlos_2_201309231117.dat* sd_pyk_vlos_2_201309232244.dat*
sd_pyk_vlos_2_201309231118.dat* sd_pyk_vlos_2_201309232245.dat*
sd_pyk_vlos_2_201309231119.dat* sd_pyk_vlos_2_201309232246.dat*
sd_pyk_vlos_2_201309231120.dat* sd_pyk_vlos_2_201309232247.dat*
sd_pyk_vlos_2_201309231121.dat* sd_pyk_vlos_2_201309232248.dat*
sd_pyk_vlos_2_201309231122.dat* sd_pyk_vlos_2_201309232249.dat*
sd_pyk_vlos_2_201309231123.dat* sd_pyk_vlos_2_201309232250.dat*
sd_pyk_vlos_2_201309231124.dat* sd_pyk_vlos_2_201309232251.dat*
sd_pyk_vlos_2_201309231125.dat* sd_pyk_vlos_2_201309232252.dat*
sd_pyk_vlos_2_201309231126.dat* sd_pyk_vlos_2_201309232253.dat*
sd_pyk_vlos_2_201309231127.dat* sd_pyk_vlos_2_201309232254.dat*
[Horit@surge ...tmp/asciidump]$ cat sd_pyk_vlos_2_201309232212.dat
# Radar code: pyk
# Scan time (ave. time of beams for a scan): 20130923 221222
# iscat_flag 1:ionospheric echo, 0:ground scatter, -1:no data
### pyk_vishell_2
# beam_range_gate LOSV[m/s] iscat_flag MLTdir-bmdir_angle[deg] Glat[deg] Glon[deg] ML
at[deg] MLT[chr]
0 0 NaN -1 61.40 64.7 339.3 66.3 22.04
0 1 NaN -1 62.36 65.2 339.5 66.8 22.07
0 2 NaN -1 63.70 65.6 339.7 67.2 22.11
0 3 NaN -1 64.18 66.1 339.9 67.6 22.14
0 4 NaN -1 64.33 66.5 340.0 68.1 22.17
0 5 NaN -1 64.30 66.9 340.2 68.5 22.20
0 6 2010 NaN -1 64.16 67.3 340.3 68.9 22.23
0 7 2010 NaN -1 63.95 67.8 340.4 69.3 22.26
0 8 306.9 1 63.68 68.2 340.6 69.7 22.29
0 9 306.8 1 63.37 68.6 340.7 70.0 22.32
0 10 334.2 1 70.64 68.9 340.8 70.4 22.35
0 11 341.8 1 78.72 69.2 340.8 70.6 22.36
0 12 444.1 1 86.37 69.4 340.7 70.8 22.37
0 13 452.6 1 95.88 69.5 340.5 71.0 22.37
0 14 385.4 1 74.06 69.7 340.4 71.2 22.38
0 15 292.8 1 59.22 70.1 340.4 71.6 22.41
0 16 NaN -1 59.02 70.5 340.7 72.0 22.45
0 17 NaN -1 58.74 70.9 340.9 72.4 22.50
0 18 NaN -1 58.41 71.4 341.2 72.7 22.54
0 19 NaN -1 58.03 71.8 341.4 73.1 22.59
```

2次元地図プロットの実成

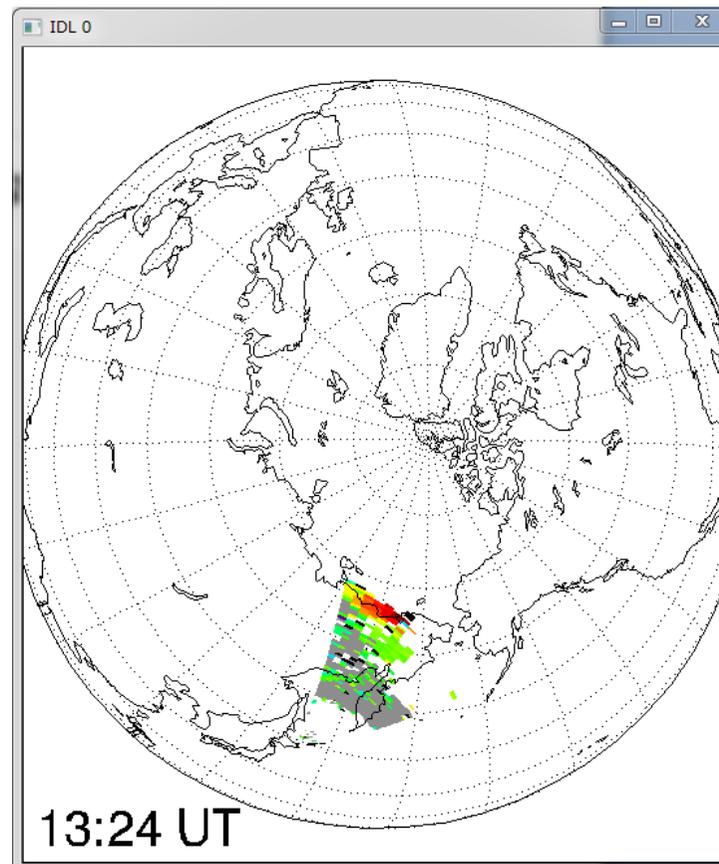


2次元地図プロット

- ▶ ある時刻の2次元スキャンのデータを、緯度・経度グリッド(+世界地図)上に描画する

観測値の2次元空間分布がわかる

他の観測データを重ね描きすれば位置関係を調べることができる



ある1つの時刻のデータしかプロットすることができない
(異なる時刻の複数のプロットを作る必要)

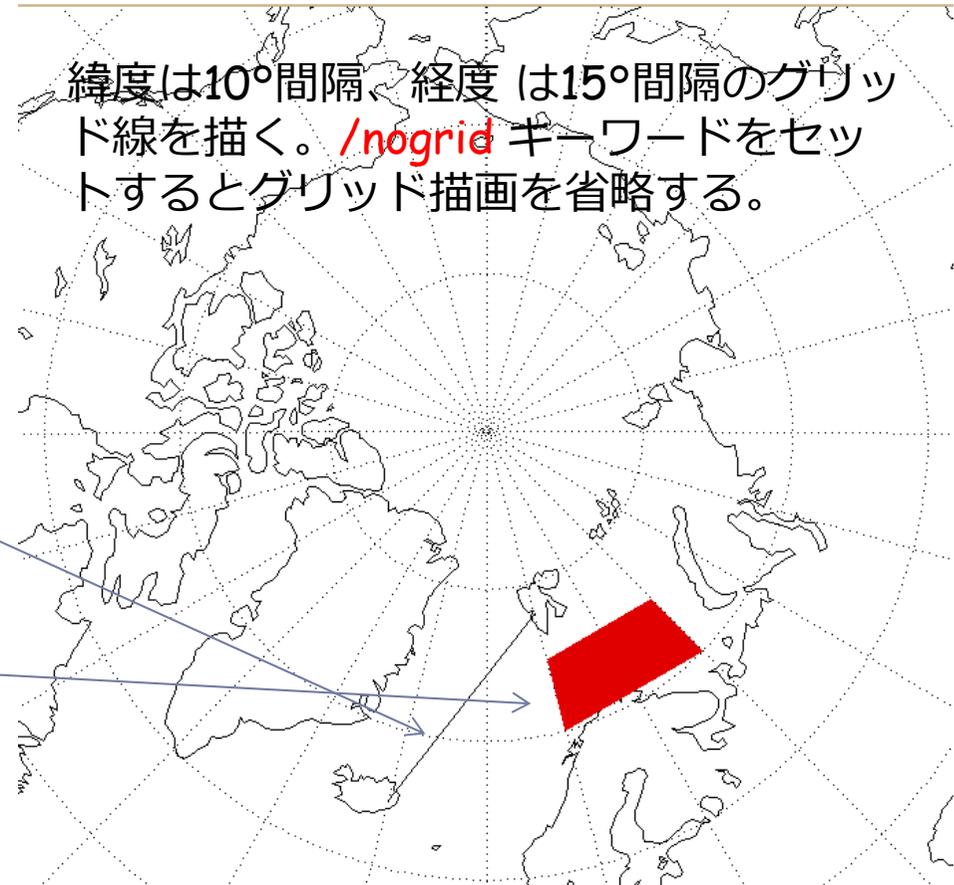


地理緯度-経度グリッド上での描画例

```
THEMIS> map2d_init
THEMIS> timespan, '2013-09-23'
THEMIS> map2d_time, 2200 ;描画時刻をセット
plot_time: 2013-09-23/22:00:00
THEMIS> map2d_set, /erase
;デフォルトで地理緯度経度グリッドを定義
THEMIS> overlay_map_coast ;地図を描画

THEMIS> plots, [345., 15], [65,78]
; plotsコマンドにより (地理緯度, 経度)で
; (75°, 345°) → (78°, 15°) に直線を描く

THEMIS> polyfill, $
[15., 45., 45., 15.], $
[70., 70., 75., 75.], color = 240
; (60°, 17h), (60°, 19h), (70°, 19h), (70°, 17h) の4点で
; 囲まれる台形を 150番の色 (この場合は黄緑) で塗りつぶす
```





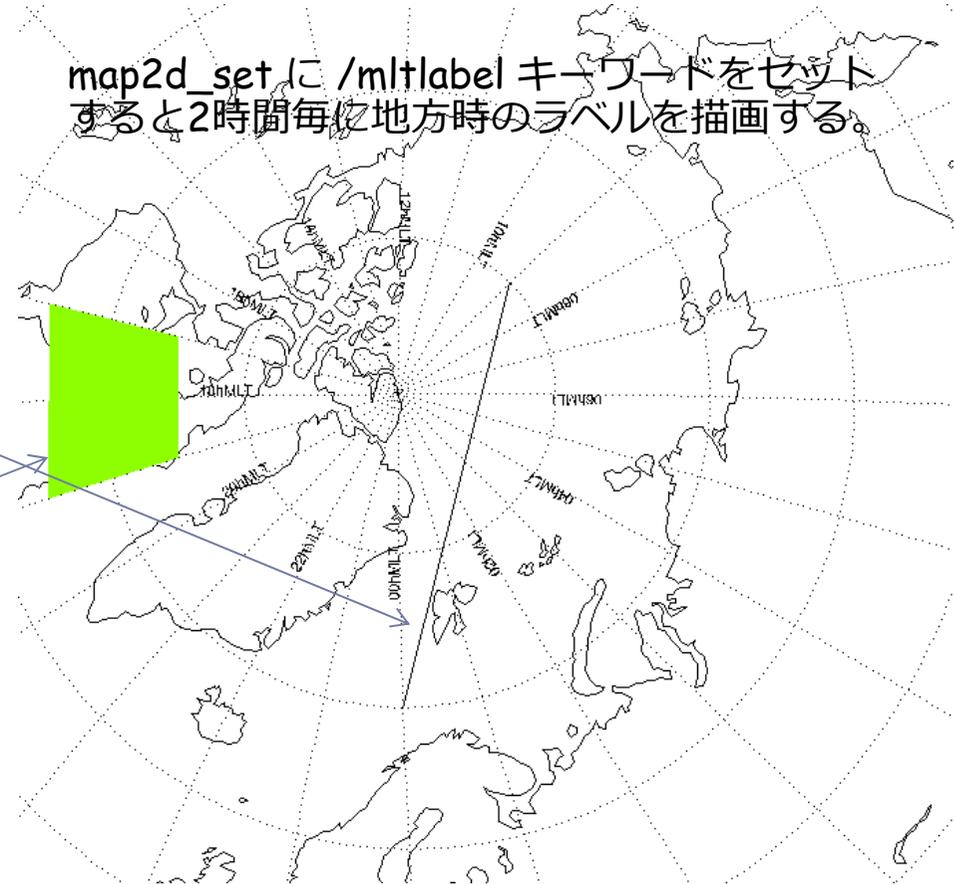
AACGM緯度-LTグリッド上での描画例

```
map2d_init, map2d_time は既に設定済みなので省略
THEMIS> map2d_coord, 'aacgm' ;AACGM座標を指定
THEMIS> map2d_set, /erase, /mltlabel
                ;AACGM緯度-LTグリッドを定義
THEMIS> overlay_map_coast ;地図を描画

THEMIS> plots, [0., 9] /24.*360., [70., 80.]
                ; plotsコマンドで(AACGM緯度,LT)で
                ; (70°, 0.0h) → (80°, 9.0h) に直線を描く

THEMIS> polyfill, $
                [17., 19., 19., 17.] /24.*360., $
                [60., 60., 70., 70.], color = 150
                ; (60°, 17h), (60°, 19h), (70°, 19h), (70°, 17h) の4点で
                ; 囲まれる台形を 150番の色 (この場合は黄緑) で塗りつぶす
```

map2d_set に /mltlabel キーワードをセットすると2時間毎に地方時のラベルを描画する。



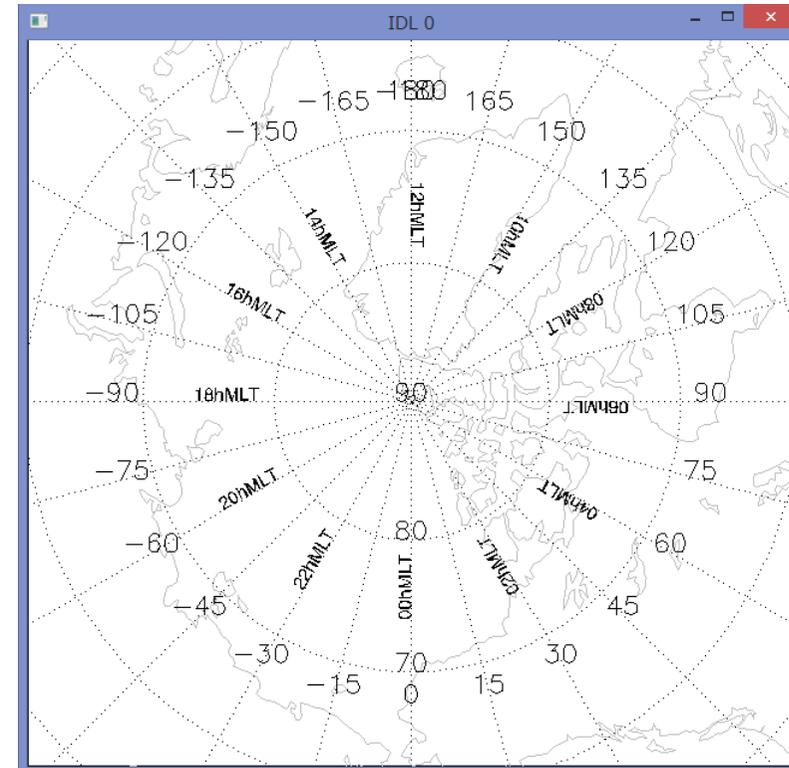
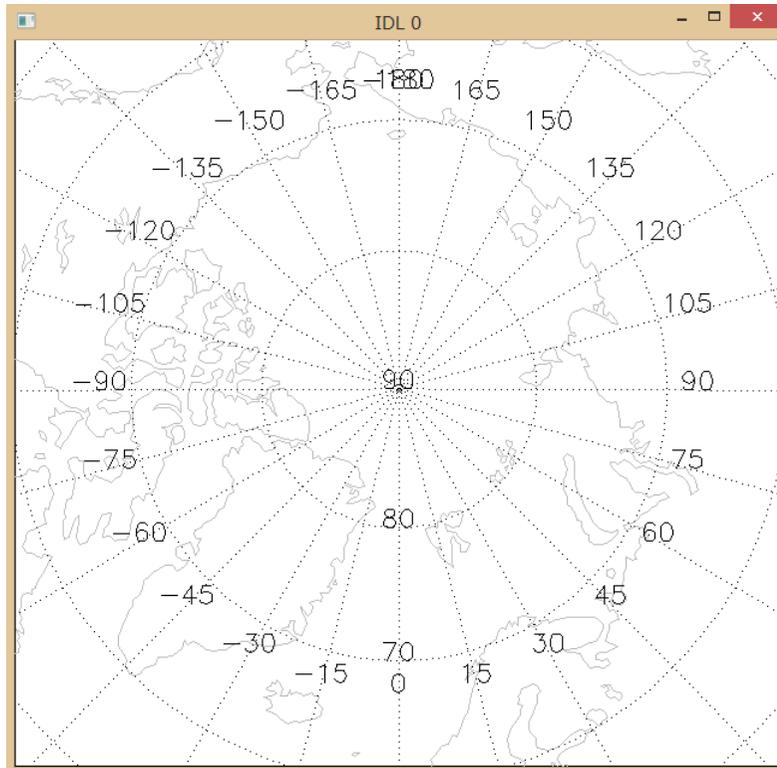
- ▶ map2d_set の他のオプションはヘッダードキュメント等を参照



map2dによる2次元地図プロット

map2d_coord,'geo' & map2d_set (地理座標用)

map2d_coord,'aacgm' & map2d_set (AACGM座標用)



- ▶ map2d_setでは↑のような緯度-経度グリッドが定義される。地理座標では、plot, (経度), (緯度) のようにすればそのままプロットできる。座標系はmap2d_coord コマンドで指定する。
- ▶ AACGM座標の場合、緯度はAACGM緯度、AACGM地方時を0h→0°, 9h→90°, 12h→180°, 18h→270°のように変換した仮想的な経度にして、plot, (AACGM緯度), (仮想的な経度) とすれば対応する場所に作図できる。
- ▶ overlay_map_???? 系のコマンドは内部で緯度,経度を上記のように計算して、点・線などを描画している。
- ▶ つまり同じように緯度,経度を与えるようにすれば、自作のプログラムでも同様なプロットをすることができる。



2次元地図プロット 1発コマンド, plot_map_sdfitを使う

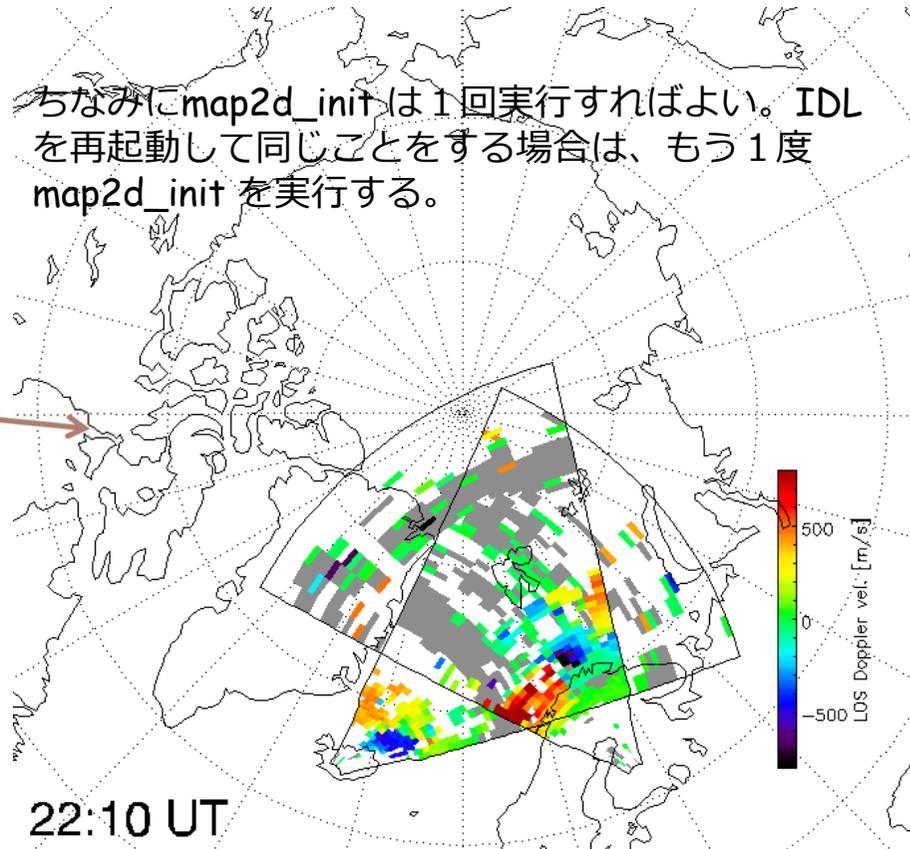
```
;環境をセットアップする
THEMIS> map2d_init

;プロットする時刻を22:10 UTに指定する
THEMIS> map2d_time, 2210

;指定時刻のLOS velocityデータを描画する
THEMIS> plot_map_sdfit, 'sd_pyk_vlos_2'

; coast キーワードをセットすると世界地図を重ねて描く
THEMIS> plot_map_sdfit, 'sd_pyk_vlos_2', /coast

;HANデータをロードして一緒に描画する
THEMIS> eng_load_sdfit, site='han', /get_support
THEMIS> zlim, 'sd_han_vlos_2', -800,800
;スケールを揃える
THEMIS> plot_map_sdfit, ['sd_pyk_vlos_2', $
' sd_han_vlos_2'], /coast ; pykとhan両方をプロット
; 座標を地理座標に変えてプロット
THEMIS> map2d_coord, 'geo' ;座標系切替コマンド
THEMIS> plot_map_sdfit, ['sd_pyk_vlos_2', $
' sd_han_vlos_2'], /coast
THEMIS> overlay_map_sdfov, site='pyk han'
;レーダーの視野を重ね描き
```





2次元地図プロット 1発コマンド, plot_map_sdfitを使う

```

:clip キーワードをセットするとズームイン。
:レーダー視野を外れることがあるので center_glat,
:center_glon キーワードで描画中心の地理緯度経度を指定
:する。

```

```

THEMIS> map2d_coord, 'aacgm' ;AACGMに戻す
THEMIS> plot_map_sdfit, ['sd_pyk_vlos_2', $
'sd_han_vlos_2'], /coast, /clip, $
center_glat=75, center_glon=0
;$(ダラー)を付けると1行を分割できる

```

```

:拡大するとカラーバーがはみ出ることが多いので、手動で位置
を調整するとよい。 colorscalepos キーワードで指定する。

```

```

THEMIS> plot_map_sdfit, ['sd_pyk_vlos_2', $
'sd_han_vlos_2'], /coast, /clip, $
center_glat=75, center_glon=0, $
colorscalepos=[0.05, 0.65, 0.08, 0.95]

```

```

: MLTラベルを重ね描き (mltlabelキーワード)

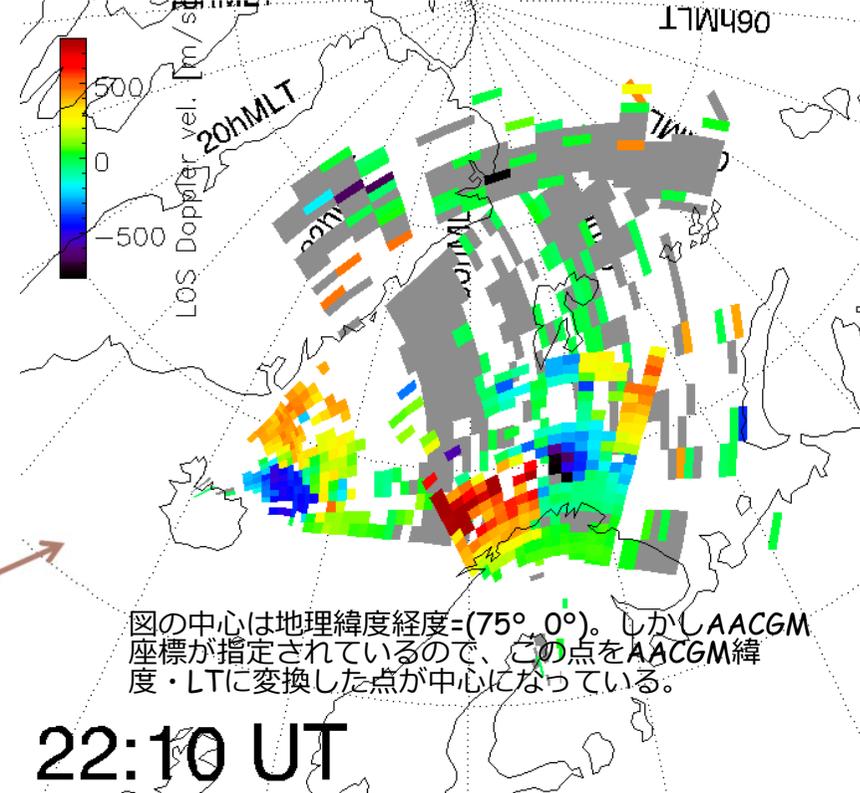
```

```

THEMIS> plot_map_sdfit, ['sd_pyk_vlos_2', $
'sd_han_vlos_2'], /coast, /clip, $
center_glat=75, center_glon=0, $
colorscalepos=[0.05, 0.65, 0.08, 0.95], $
/mltlabel

```

前ページのプロットと比べて、より拡大
(focus)されている！



図の中心は地理緯度経度=(75°, 0°)。しかしAACGM座標が指定されているので、この点をAACGM緯度・LTに変換した点が中心になっている。

22:10 UT

緯度線は80°, 70°, 60°...と10度刻み



2次元地図プロット

複数時刻プロット

```

THEMIS> map2d_time, 2155
THEMIS> plot_map_sdfit,
['sd_pyk_vlos_2', 'sd_han_vlos_2'],/clip, /coast,
center_glat=75, center_glon=0, position=[0.0,0.5,0.5,1.0],
/nocolorscale

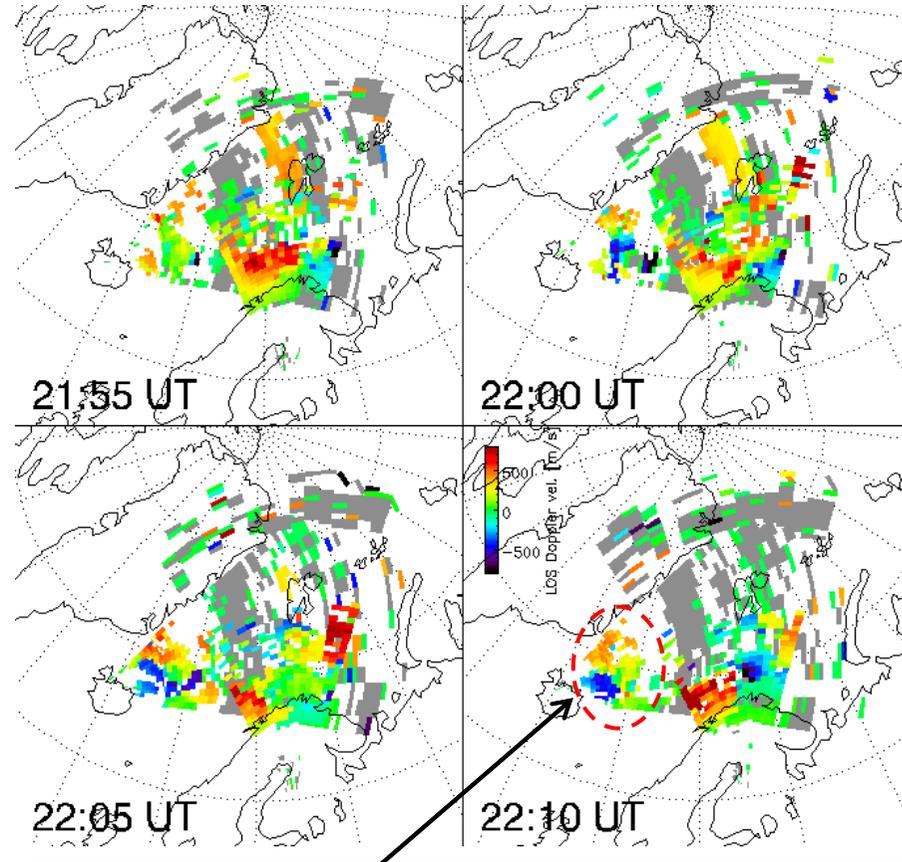
THEMIS> map2d_time, 2200
THEMIS> plot_map_sdfit,
['sd_pyk_vlos_2', 'sd_han_vlos_2'],/clip, /coast,
center_glat=75, center_glon=0, position=[0.5,0.5,1.0,1.0],
/noerase, /nocolorscale

THEMIS> map2d_time, 2205
THEMIS> plot_map_sdfit,
['sd_pyk_vlos_2', 'sd_han_vlos_2'],/clip, /coast,
center_glat=75, center_glon=0, position=[0.0,0.0,0.5,0.5],
/noerase, /nocolorscale

THEMIS> map2d_time, 2210
THEMIS> plot_map_sdfit,
['sd_pyk_vlos_2', 'sd_han_vlos_2'],/clip, /coast,
center_glat=75, center_glon=0, position=[0.5,0.0,1.0,0.5],
/noerase, colorscalepos=[0.05, 0.65, 0.08, 0.95]

```

- positionキーワードにnormal座標でのプロットの位置を与える ([x0, y0, x1, y1])
- plot_map_sdfit はデフォルトで描画毎にウィンドウをクリアしてしまうので、2つ目以降は /noerase を付ける
- 4番目のみカラースケールを描画する (/nocolorscale無し)



2D地図プロットにすれば、スライド22のRTIプロットで見えていたflow shear構造が見易い。



2次元地図プロット

複数時刻プロット

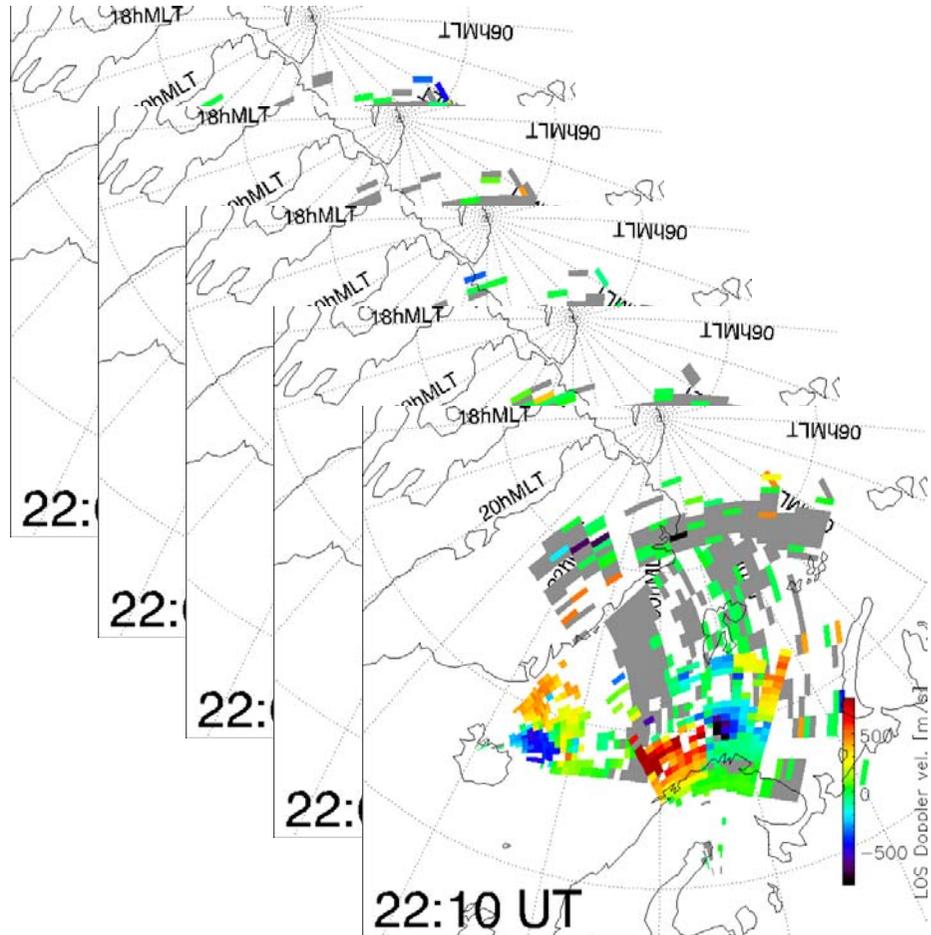


```
THEMIS> !p.position = [0,0,1,1] ;画面分割設定を初期化  
THEMIS> make_fanplot_pictures, $  
['sd_pyk_vlos_2', 'sd_han_vlos_2'], $  
2155, 2210, /clip, /coast, center_glat=75, $  
center_glon=0, prefix='pngdir/sd_pykhan_'
```

2Dプロットの画像ファイル(png)をいっぺんに作成するコマンド。

第2,3引数の2155, 2210 の意味
→ 21:55-22:10 UTの間のプロットを全て作成。

出力先は prefix キーワードで指定する。上の例だと, pngdir というフォルダを作ってその中に sd_pykhan_hhmm.png というファイル名で出力 (hhmmは時刻)。



RTIプロットや2Dプロットの応用編



2次元地図プロット overlay_map_???コマンドでマニュアル

```

THEMIS> map2d_init
THEMIS> map2d_time, '2013-09-23/22:10'

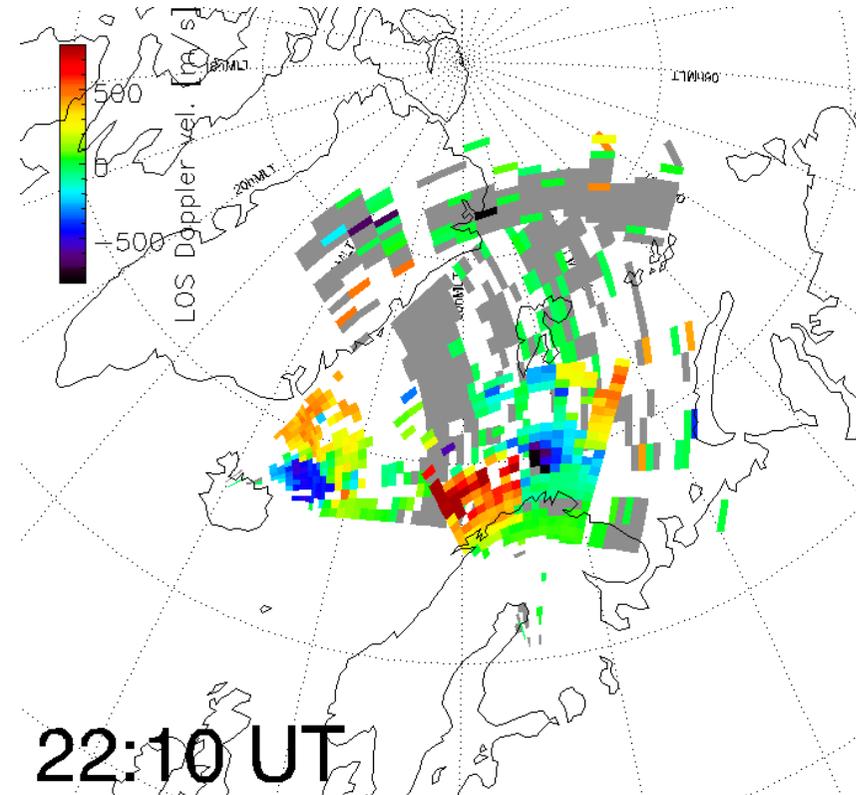
;緯度経度描画モードをオンにして、グリッドを描画する。
; aacgm: セットするとAACGM座標で描画
; glatc/glonc で描画する際の中心位置を指定。
; mllabel キーワードをセットするとMLTのラベルを描く
; erase キーワードをセットすると一度ウィンドウ内を消去
; scale: プロットのスケール

THEMIS> map2d_coord, 'aacgm'
THEMIS> map2d_set, $
      glatc=75, glonc=0, scale=35e+6, $
      /mllabel, /erase

;指定時刻のLOS velocityデータを重ね描きする
THEMIS> overlay_map_sdfit, $
      ['sd_pyk_vlos_2', 'sd_han_vlos_2'], $
      colorscalepos=[0.05, 0.65, 0.08, 0.95]

;世界地図を重ね描きする
THEMIS> overlay_map_coast

```



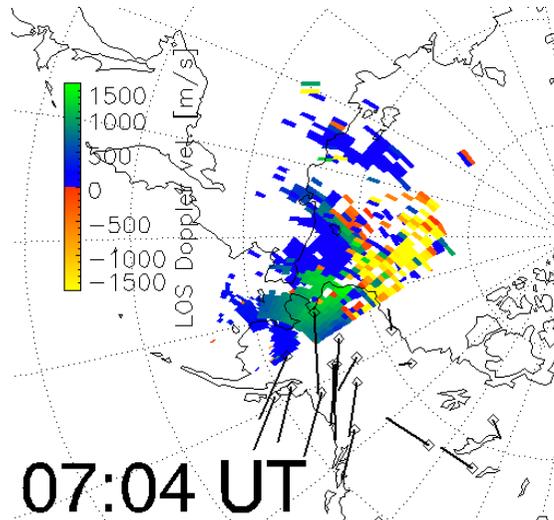
実はplot_map_sdfit は 内部で

- map2d_set
- overlay_map_sdfit
- overlay_map_coast (/coastの場合)

を順に実行している。

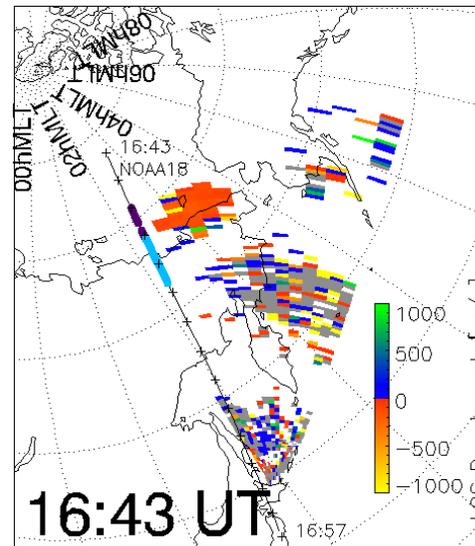


2次元地図プロット 応用編



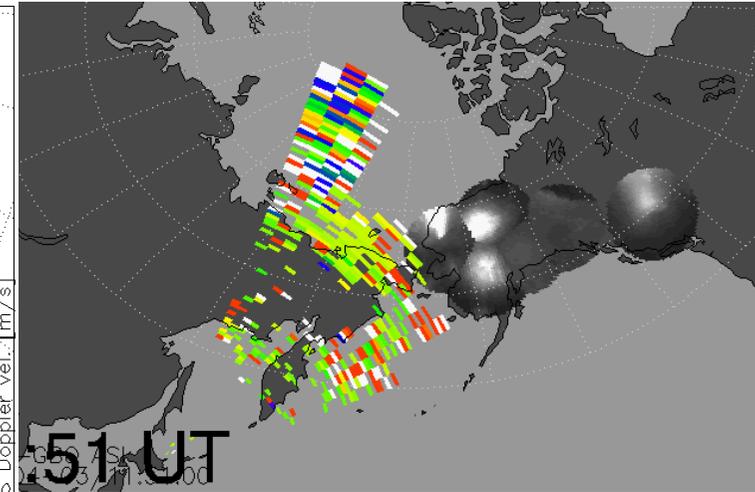
07:04 UT

地磁気の水平2成分の矢印と



16:43 UT

衛星のfootpointと
(この例はNOAA/POES-18)



15:51 UT

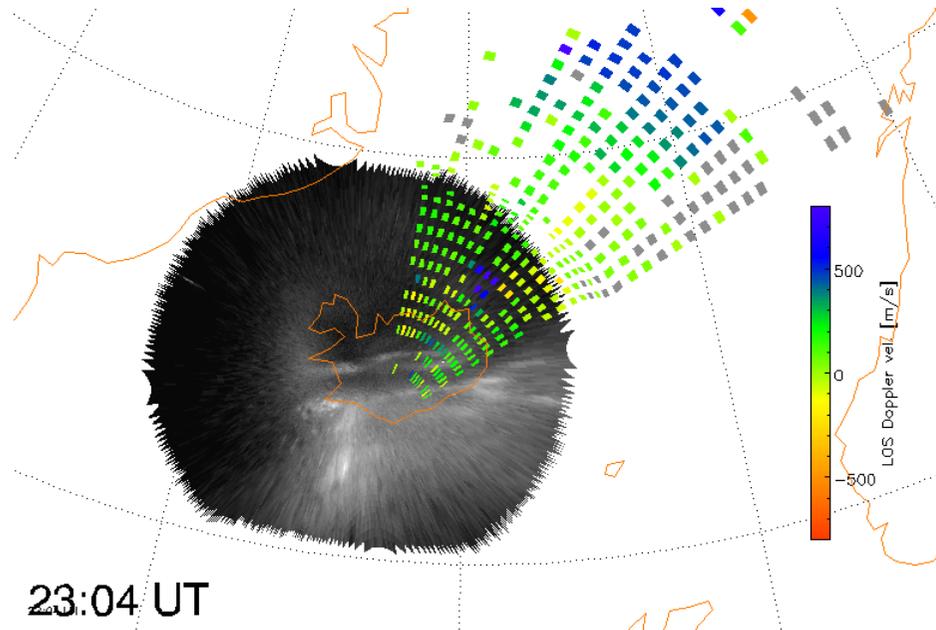
地上全天カメラのデータと

- ▶ map2d_set は内部でIDL標準のmap_set を呼んでいる.
- ▶ map_set を宣言すると, plot, 経度, 緯度, ... のようにして地図上に描画できるようになる.
- ▶ map_set上に描画するような他の(自作・他作)プロットルーチンと組み合わせることで, 複数種のデータを地図上に描画できる.



2次元地図プロット

応用編 - 全天カメラ像と重ねる



!! 注意 !!

データの読込でかなりメモリーを消費するので、
`erg_load_sdfit`,
`iug_load_asl_nipr`
 を実行したところで `out of memory` エラーが出る可能性があります。その場合、もっとメモリーに余裕があるPCでやってみて下さい。

```

THEMIS> timespan, '2012-03-27/23:00:00',1,/hour
THEMIS> erg_load_sdfit, site='pyk', /get
THEMIS> iug_load_asl_nipr, site='hus'
THEMIS> map2_time, 2304 & map2d_coord, 'geo'
THEMIS> map2d_set, /erase, scale=17e+6, glatc=70, glonc=346
THEMIS> loadct, 0 0番のカラーテーブル(白黒)をセットする
THEMIS> overlay_map_asl_nipr, 'nipr_asl_hus_0000', /nocolor, colorr=[20,140]
THEMIS> loadct_sd, 44 & overlay_map_sdfit, 'sd_pyk_vlos_2', pixel=0.5
THEMIS> overlay_map_coast, col=40
  
```

Iceland全天カメラのデータの読込および2D地図プロットコマンド。
 詳細は中級編-2の後半で。

SDのピクセルを0.5倍にして
 ASIも見通せるようにする。

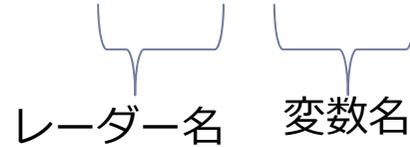
地図を40番の色(茶色)で描画

補足資料



tplot変数名のnotation

sd_hok_vlos_1_azim03



Range gate (RG) suffix

- 0: 75 RGデータ
- 1: 110 RGデータ
- 2: 70 RGデータ
- ...

この添字が異なるとRange gate数やビーム数が異なる。レーダーによってこれらの数は異なる。

Beam番号
ほとんどのレーダーで beam0
が一番西端のbeam (例外有り)

▶ 主な変数名と中身

- pwr**: エコー強度
- vlos**: Line-of-sightドップラー速度(LOSv)
- spec_width**: スペクトル幅
- vnorth**: LOSドップラー速度の地理緯度成分(北向き)
- veast**: LOSドップラー速度の地理経度成分(東向き)
- (vlos|vnorth|veast)_iscat**: 電離圏エコーのみのデータ
- (vlos|vnorth|veast)_gscat**: 地上エコーのみのデータ
- (vlos|vnorth|veast)_bothscat**: 電離圏・地上エコー両方のデータ
- elev_angle**: elevation angle値
- echo_flag**: 電離圏エコーか地上エコーかの判定フラグ

- quality**: データのqualityについての情報(0: good, 1以上: poor)
- quality_flag**: quality判定の内訳 (詳細は担当者へ)
- position_tbl**: 各pixelの四隅の緯度、経度値テーブル
- positioncnt_tbl**: 各pixelの中心の緯度、経度値テーブル
- cpid**: beam毎の観測モード
- tfreq**: beam毎の周波数
- noise**: beam毎のノイズレベル

