UDAS egg 操作マニュアル

for UDAS egg 1.02

2018年12月20日

IUGONETプロジェクトチーム

UDAS egg とは

UDAS egg は、SPEDAS未対応の科学データを簡単に読込み・解析するための

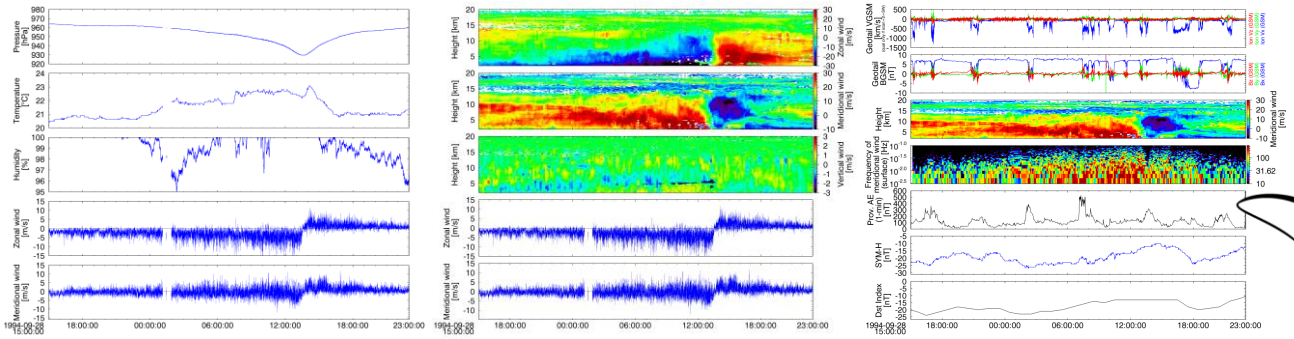
IDL/SPEDAS用プログラムテンプレート です。

このマニュアルに従って、

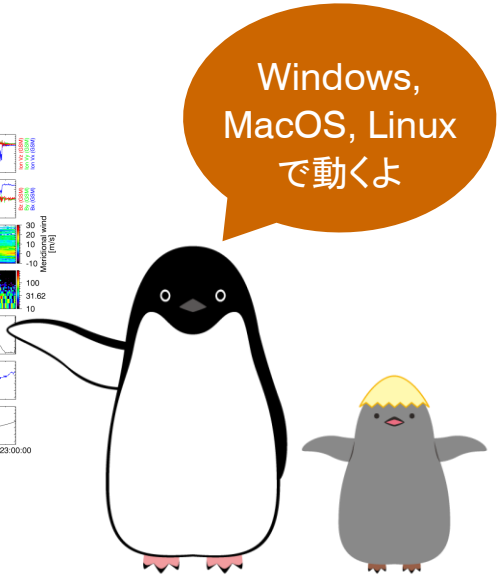
(1) プログラムコード上にマークされた箇所を書き換えるだけで、

(2) またはデータの読み込み方法を設定するだけで、

すぐに可視化・解析できます。



ラインプロット　　カラーコンター　　他のデータとの組合せ



IDL（Interactive Data Language）は、データの解析と可視化に特化した配列指向型のプログラミング言語です。
SPEDAS（Space Physics Environment Data Analysis Software）は、THEMIS, MMS, GOES, ERG など、様々な科学衛星と地上観測により得られた科学データを可視化・解析するための、IDLで記述されたパッケージソフトウェアです。詳しくは提供元のウェブサイトをご覧ください。IDL: https://www.harrisgeospatial.com/　SPEDAS: http://spedas.org/

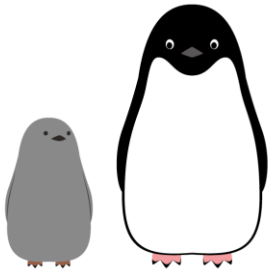
SPEDAS（UDAS含む）との違い

	SPEDAS	UDAS EGG CUI	UDAS EGG GUI
解析ルーチン（ロードプロシージャ）の作成	難解 科学データごとに作成する必要あり	簡単 テンプレートから10行程度変更すればOK	不要 汎用ルーチン組込済
可視化・解析したいデータファイルの取得元	オンラインサーバ インターネット接続されたサーバから	パソコンから直接も インターネット接続 または パソコンからファイルを選択	パソコンから直接 パソコンからファイルを選択 *インターネット取得にも今後対応予定

解析ルーチンの作成に困っている方、

プログラム作成よりも **研究に時間を使いたい方** に

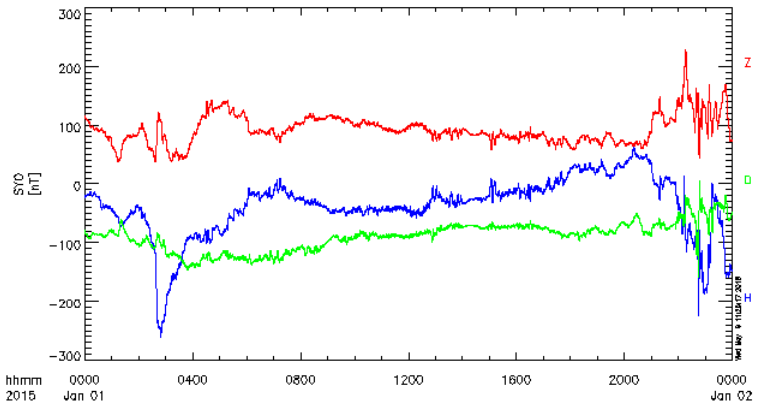
ぴったりです。ダウンロードして、SPEDASに組み込んで使います。



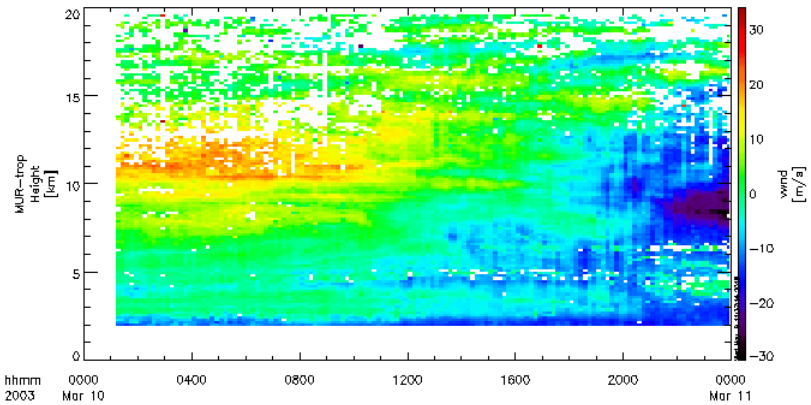
読み込めるファイルの種類と可視化の例

1. CDF (Common Data Format)

CDFファイル



ラインプロット



カラーコンター

CDFの場合、SPEDASが自動的に、CDFの中のvデータの有無を読み取り、それを解釈してラインプロットかカラーコンターを選択します。

解析プログラムを作成する際、あるいは可視化する際に、手動でラインプロットかカラーコンターを選択する必要はありません。

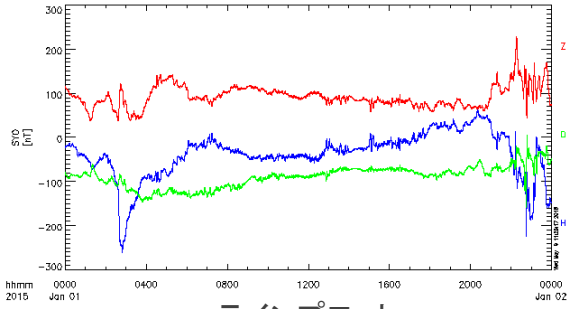
読み込めるファイルの種類と可視化の例

2. ASCII Format

0. 単純時系列データ

時刻	データ種 1	2	3	...
1994/09/28 15:00:07,	964.0,	20.4,	100.0,	...
1994/09/28 15:00:12,	964.0,	20.4,	100.0,	...
1994/09/28 15:00:17,	964.0,	20.4,	100.0,	...
1994/09/28 15:00:22,	964.0,	20.4,	100.0,	...
1994/09/28 15:00:27,	964.0,	20.4,	100.0,	...
1994/09/28 15:00:32,	964.0,	20.4,	100.0,	...
1994/09/28 15:00:37,	964.0,	20.4,	100.0,	...
1994/09/28 15:00:42,	964.0,	20.4,	100.0,	...
1994/09/28 15:00:47,	964.0,	20.4,	100.0,	...
1994/09/28 15:00:52,	964.0,	20.4,	100.0,	...
...				

横方向にデータ種、縦方向に時系列



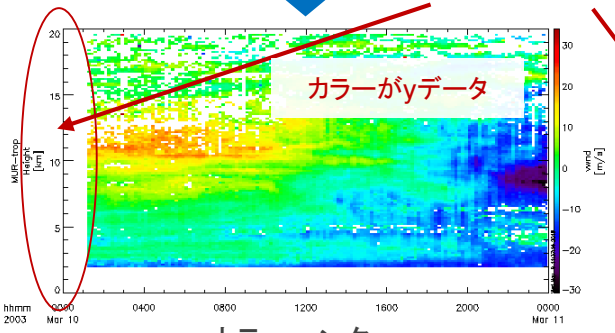
ラインプロット

1. vデータ(Y軸構成値)を含む縦方向データ

時刻	データ種 v	y1	y2	...
1994/09/28 15:05:00,	1.998,	-12.0,	...	
1994/09/28 15:05:00,	2.145,	-11.3,	...	
1994/09/28 15:05:00,	2.293,	-9.7,	...	
1994/09/28 15:05:00,	2.441,	-9.1,	...	
1994/09/28 15:05:00,	2.589,	-8.0,	...	
1994/09/28 15:05:00,	2.736,	-10.3,	...	
1994/09/28 15:05:00,	2.884,	-9.4,	...	
1994/09/28 15:05:00,	3.032,	-8.5,	...	
1994/09/28 15:05:00,	3.179,	-6.9,	...	
1994/09/28 15:05:00,	3.327,	-7.1,	...	
...				

vデータは、Y軸を構成する値。例えば、大気データにおける高度や、スペクトルデータにおける周波数、レーダーデータにおけるレンジ等にあたる。

縦軸がvデータ

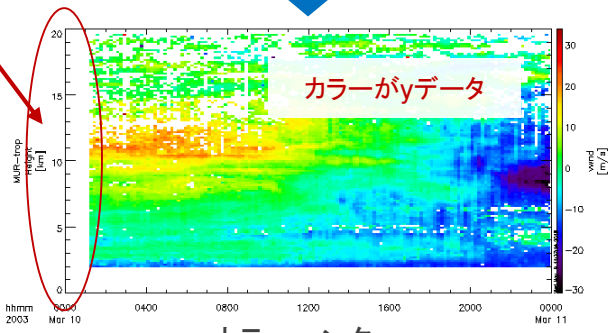


カラーコンター

2. vデータ(Y軸構成値)を含まない横方向データ

時刻	データ種 y	y	y	...
1994/09/29 00:05,	-12.0,	-11.3,	-9.7,	-9.1, ...
1994/09/29 00:15,	-12.2,	-12.1,	-11.7,	-11.5, ...
1994/09/29 00:25,	-12.8,	-11.5,	-11.1,	-10.9, ...
1994/09/29 00:35,	-10.5,	-11.1,	-10.6,	-10.8, ...
1994/09/29 00:45,	-12.9,	-12.7,	-10.9,	-12.4, ...
1994/09/29 00:55,	-13.6,	-13.9,	-12.6,	-12.2, ...
1994/09/29 01:05,	-11.2,	-11.1,	-11.1,	-10.4, ...
1994/09/29 01:15,	-10.5,	-10.2,	-11.4,	-10.8, ...
1994/09/29 01:25,	-11.5,	-11.1,	-11.6,	-10.9, ...
1994/09/29 01:35,	-12.6,	-12.5,	-11.8,	-12.0, ...
...				

パターンBのyデータが横方向に並ぶパターン。vデータ(縦軸に相当)のベクトルは可視化の際に与える。



カラーコンター

バイナリ形式やオリジナル形式のファイルも、上記の ASCII 形式に変換すれば UDAS egg で読むことができます。
トラフィックや変動・変遷などの社会データ、環境や人体情報などの測定データも、すぐに読むことができます。

はじめてみよう

1. 事前準備 → 2. 作成と読み込み → 3. 実行

IDLライセンスはありますか？

ある

CUI　テンプレートにアレンジを加えて自分のものに！

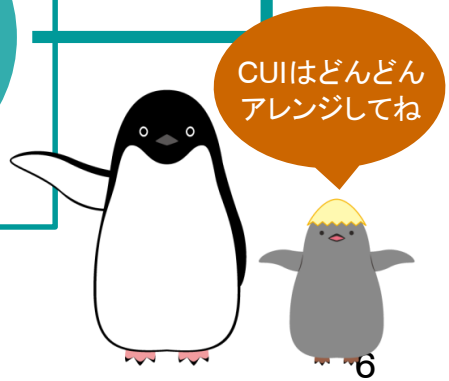
- インストールしてIDL環境に構築しよう p.7
- テンプレートを書き換えよう p.8
- プロットを表示しよう p.18

もっと解析してみよう p.23

ない

GUI　データファイルがあれば解析できます！

- インストールしよう p.19
- データファイルを読み込もう p.20
- プロットを表示しよう p.22



UDAS egg にはアレンジした例も入っています。

CUI 1. インストールしてIDL環境に構築しよう

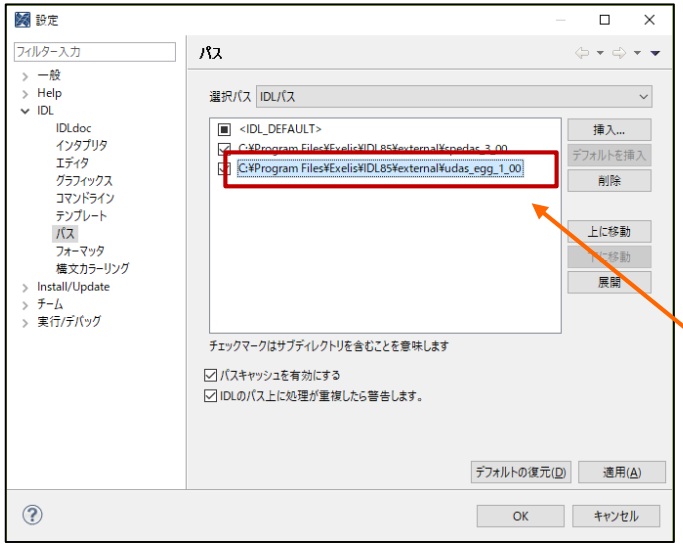
1. SPEDAS 3.1 ソースコード版をダウンロードして、開発環境を構築します。

   （SPEDAS環境がまだ構築されていない場合のみ）

   SPEDAS 3.0 ダウンロードと、開発環境構築の方法: http://themis.ssl.berkeley.edu/software.shtml

2. UDAS egg をダウンロードして、IDL開発環境に追加します。

   UDAS egg ダウンロード: http://www.iugonet.org/product/analysis.jsp



1. UDAS egg（zip形式）をダウンロードします。
2. 解凍してできたディレクトリを、開発環境にコピーします。

コピー先に困ったら、SPEDASと同じように、
C:¥Program Files¥Exelis¥IDL8x¥external¥ 配下にしましょう。

3. IDLを起動して、[ウィンドウ] タブから [設定] [IDL/パス] を選んで、SPEDAS と UDAS egg の両方にパスを通します。

3. IDLワークベンチで以下のコマンドを入力し、正常終了すればインストールは成功です。

```
IDL> .r loadproc_template
% Compiled module: LOADPROC_TEMPLATE.
IDL>
```

← インストールが成功しました。

2. テンプレートを書き換えよう

1. テンプレートファイルをコピーして、新しいファイルを作成します。

［コピー元ファイル］ loadproc_template.pro

ここでは例として、コピーしたファイルを loadproc_mag.pro という名前に変更して
ルーチンを作成します。

2. 新しく作成したファイルを編集します。

*印：必須項目

```
pro loadproc_mag, site=site, datatype=datatype, $
      trange=trange, verbose=verbose, downloadonly=downloadonly, $
      no_download=no_download
```

* **pro loadxxxxx：解析ルーチン名**
新しく作成したファイル名と同じ名称を指定します。この名称は、実行時のコマンド名に相当します。

2. テンプレートを書き換えよう

2. 新しく作成したファイルを編集します。(Step.1: 環境設定)

*印：必須項目

```
;*********************************;
;***** Step1: Set parameters *****;
;*********************************;
file_format='cdf'   ; Choose  'cdf' or 'ascii'
url='http://www.iugonet.org/data/SITE/DATATYPE/YYYY/mag_SITE_DATATYPE_YYYYMMDD_v??.cdf'
acknowledgement = 'You can write the data use policy here. ' + $
   'This description is displayed when you use this load procedure. '
site_list='sta1 sta2 sta3'
datatype_list='1sec 1min 1hr'
```

* file_format

cdf または ascii を入力します。

* url: 読込ファイルのURL

読み込みたいファイルのURLを記述します。以下の文字列を記述すると、後でプログラムが自動的に具体的な値に変換します。

SITE: 観測所名(site_list の要素値)

DATATYPE: データ種(datatype_list の要素値)

日時を表す変数

年4桁	年2桁	月(01-12)	日(01-31)	時2桁
YYYY	yy	MM	DD	hh

2. テンプレートを書き換えよう

2. 新しく作成したファイルを編集します。(Step.1: 環境設定)

```
;*********************************;
;***** Step1: Set parameters *****;
;*********************************;
file_format='cdf'   ; Choose  'cdf' or 'ascii'
url='http://www.iugonet.org/data/SITE/DATATYPE/YYYY/mag_SITE_DATATYPE_YYYYMMDD_v??.cdf'
acknowledgement = 'You can write the data use policy here. ' + $
   'This description is displayed when you use this load procedure. '
site_list='sta1 sta2 sta3'
datatype_list='1sec 1min 1hr'
```

【記述例】

例えば、URLを

http://www.iugonet.org/data/SITE/DATATYPE/YYYY/mag_SITE_DATATYPE_YYYYMMDD_v??.cdf

とした場合、プログラムは自動的に

http://www.iugonet.org/data/ath/1sec/2018/mag_ath_1sec_20181013_v01.cdf

http://www.iugonet.org/data/ath/1sec/2018/mag_ath_1sec_20181014_v01.cdf

……

のように置換します。

日時は解析実行時に与える timespan により決定します。**??** は SPEDAS における、ワイルドカードを示します。

2. テンプレートを書き換えよう

2. 新しく作成したファイルを編集します。(Step.1: 環境設定)

```
;*********************************;
;***** Step1: Set paramters *****;
;*********************************;
file_format='cdf'   ; Choose  'cdf' or 'ascii'
url='http://www.iugonet.org/data/SITE/DATATYPE/YYYY/mag_SITE_DATATYPE_YYYYMMDD_v??.cdf'
acknowledgement = 'You can write the data use policy here. ' + $
   'This description is displayed when you use this load procedure. '
site_list='sta1 sta2 sta3'
datatype_list='1sec 1min 1hr'
```

acknowledgement: データ利用ポリシー

ルーチンを実行した際に表示する、データの利用ポリシーを記述します。

site_list: 観測所リスト

観測地点をリストで指定します。複数存在する場合は、'stn1 stn2 stn3' のように、半角空白文字で区切って入力します。
何も存在しない場合はダミー値を1つ入力します。実行時のsiteキーワードとその値に相当します。

datatype_list: データ種リスト

サンプリングタイデータ種リストムや波長、観測モード、ファイルのバージョンなど、観測モードなどを指定します。
何も存在しない場合はダミー値を1つ入力します。実行時のdatatypeキーワードとその値に相当します。

2. テンプレートを書き換えよう

2. 新しく作成したファイルを編集します。(Step.2: 読込み設定)

CDFの場合

```
;***********************************************;
;**** Step2: Load data into tplot variables ****;
;***********************************************;
;----- For CDF format files -----;
 prefix='test_'+site[isite]+'_'+datatype[idt]+'_'
;          suffix='_'+site[isite]+'_'+datatype[idt]
```

prefix: tplot変数名接頭辞

読み込んだデータを格納する構造体(tplot変数)名の接頭辞を指定します。自分が分かりやすいものを付与します。

上図の例ではtest_stn_1min_ という接頭辞を付与しています。

suffix: tplot変数名接尾辞

tplot変数の名前の後ろに加える接尾辞を指定します。(指定しない場合が多数です。)

2. テンプレートを書き換えよう

2. 新しく作成したファイルを編集します。(Step.2: 読込み設定)

*印：必須項目

ASCIIの場合

```
;***********************************************;
;**** Step2: Load data into tplot variables ****;
;***********************************************;
;----- For ASCII format files -----;
format_type=0     ; 0:xy, 1:xyv_1, 2:xyv_2
tformat='YYYY-MM-DD hh:mm:ss.fff'
tvar_column=[1, 2, 3, 4]
tvarnames='test_'+site[isite]+'_'+datatype[idt]
delimiter=','
data_start=0
```

*** format_type：ファイルタイプ形式**
巻頭に示したファイルタイプのうち、読み込むデータファイルがどのタイプにあたるか、0～2の数値で指定します。

*** tformat：時刻形式**
データファイルに記載されている日時の形式を指定します。

*** tvar_column：読込み列**
日時列を除く列のうち、何列目を読み込むかを指定します。日時列を除いて0から数えます。

*** tvarnames：tplot変数名**
読み込んだ値を格納する構造体(tplot変数)の名前を指定します。自分が分かりやすい名前を付与します。

2. テンプレートを書き換えよう

2. 新しく作成したファイルを編集します。(Step.2: 読込み設定)

*印：必須項目

ASCIIの場合

```
;***********************************************;
;**** Step2: Load data into tplot variables ****;
;***********************************************;
;----- For ASCII format files -----;
format_type=0     ; 0:xy, 1:xyv_1, 2:xyv_2
tformat='YYYY-MM-DD hh:mm:ss.fff'
tvar_column=[1, 2, 3, 4]
tvarnames='test_'+site[isite]+'_'+datatype[idt]
delimiter=','
data_start=0
```

* **delimiter**：デリミタ形式
  データの区切り文字を指定します。

data_start：データ開始行
読み飛ばす行数を指定します。指定しない場合は0が適用されます。

2. テンプレートを書き換えよう

2. 新しく作成したファイルを編集します。(Step.2: 読込み設定)

ASCIIの場合

```
;***********************************************;
;**** Step2: Load data into tplot variables ****;
;***********************************************;
;----- For ASCII format files -----;

comment_symbol=''
;         v_column=0
;         vvec=[60., 70., 80., 90., 100., 110., 120.]
;         time_column=[0, 0, 0, 1, 1, 1]
;         input_time=[2017, 1, 1, 0, 0, 0]
```

comment_symbol:コメント行

コメントアウト文字列(読み込まないレコード)を指定します。指定しない場合はすべて読込みの対象となります。

v_column:vカラム番号指定

format_type=1 の場合のみ使います。日時を除く列のうち、何列目をvdata(縦軸データ)に使うかを指定します。

vvec:読込み列

format_type=2 の場合のみ使います。vdata(縦軸データ)の値(ベクトル)を入れます。

2. テンプレートを書き換えよう

2. 新しく作成したファイルを編集します。(Step.2: 読込み設定)

ASCIIの場合

```
;***********************************************;
;**** Step2: Load data into tplot variables ****;
;***********************************************;
;----- For ASCII format files -----;

comment_symbol=''
;         v_column=0
;         vvec=[60., 70., 80., 90., 100., 110., 120.]
;         time_column=[0, 0, 0, 1, 1, 1]
;         input_time=[2017, 1, 1, 0, 0, 0]
```

time_column:日時情報追加フラグ / **input_time**:日時情報追加

読み込みたいファイルの中の日時情報が [年, 月, 日, 時, 分, 秒] の6つの要素を持たない場合、
(一部あるいは全てが欠落している場合)に、日時情報を補います。time_column と input_time をペアで設定します。
上記の順番で、time_column には、ファイルに書かれていない要素を0、書かれている要素を1、
input_time には、time_column 項目で0とした要素に対し、具体的な日時の値を入力します。

例えば、ファイルに hh,、mm, ss のみ記載されている場合、下記のようになります。

time_column = [0, 0, 0, 1, 1, 1]
time_column = [2018, 10, 13, 0, 0, 0]

2. テンプレートを書き換えよう

2. 新しく作成したファイルを編集します。(Step.3: その他のカスタマイズ)

```
;***********************************************;
;***** Step3: Options for tplot variables *****;
;***********************************************;
;----- Missing data --> NaN -----;
;       tclip, tvarnames, -1e+5, 1e+5, /overwrite
;----- Ylim -----;
;       ylim, tvarnames, -1000, 1000
;----- Labels -----;
;       options, /def, tvarnames, labels=['ch1','ch2',' ch3'], $
;          ytitle = 'Tatejiku', $
;          ysubtitle = '[Unit]', labflag=1, colors=[2,4,6]
```

tclip行の数値: NaN置換範囲

この範囲を超えた場合、値をNaNに置き換えます。NaNに置換すると、SPEDASでは可視化対象外と扱われ、ラインプロットでの突出した異常線、カラーコンターでの異常な色付けなどを防ぐことができます。

ylim行の数値: Y軸範囲

Y軸の最小値と最大値を設定します。後でSPEDASの機能を使って設定することも可能です。

options行

labels	: ラベル　プロットに付与するラベルを指定します。改行は!Cで指定します。
ytitle	: Y軸タイトル　Y軸のタイトルを指定します。改行は!Cで指定します。
ysubtitle	: Y軸サブタイトル　Y軸のサブタイトルを指定します。改行は!Cで指定します。
labflag	: ラベル有無　0: 表示なし、1: 表示あり を指定します。
colors	: プロットの色　0: 黒、1: マゼンダ、2: 青、3: シアン、4: 緑、5: 黄、6: 赤 を指定します。

3. プロットを表示しよう

1. IDLを起動して、下記のコマンドを実行しましょう。

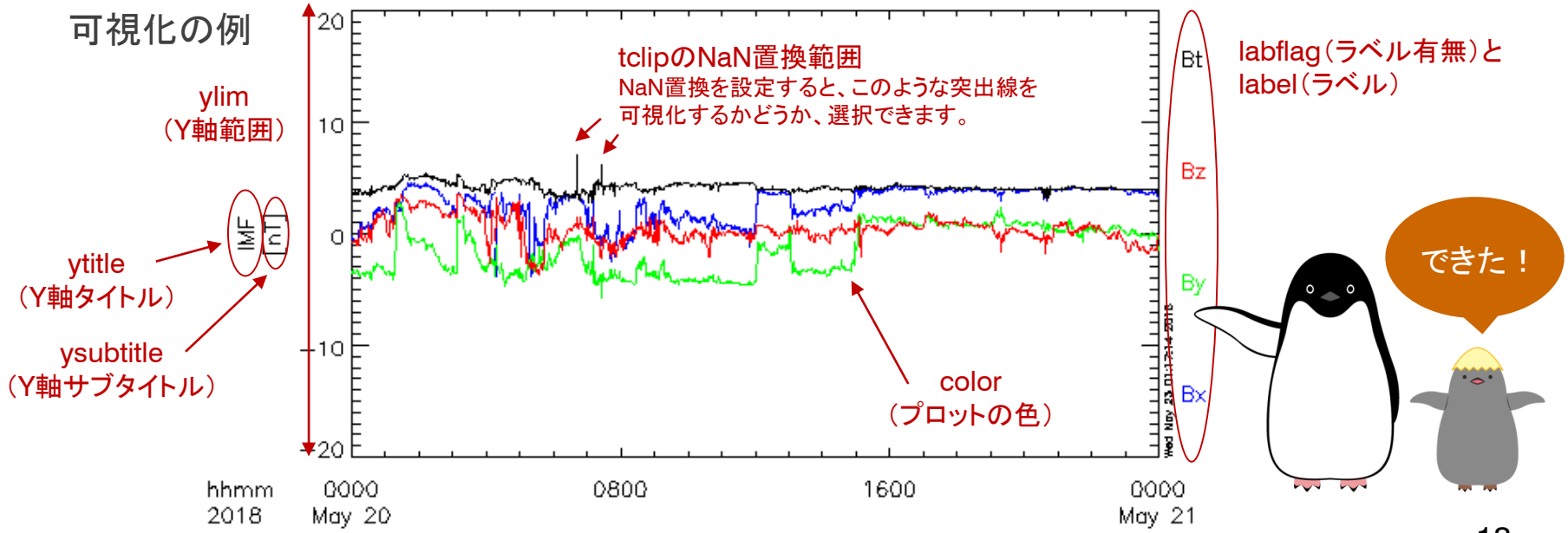
```
IDL> thm_init
THEMIS> timespan, '2018-05-20'
THEMIS> .r loadproc_mag
THEMIS> loadproc_mag, site= 'stn', datatype='1min'
THEMIS> tplot_names
  1 mag_stn_1min
THEMIS> tplot, 'mag_stn_1min'
```

作成したルーチンをコンパイルします。

作成したルーチンを実行します。
site_list またはdatatype_list を定義した場合は、ここに、キーワードとして、読込みたい値を値を与えます。

Step.3で定義したtplot変数が作成されます。

可視化の例

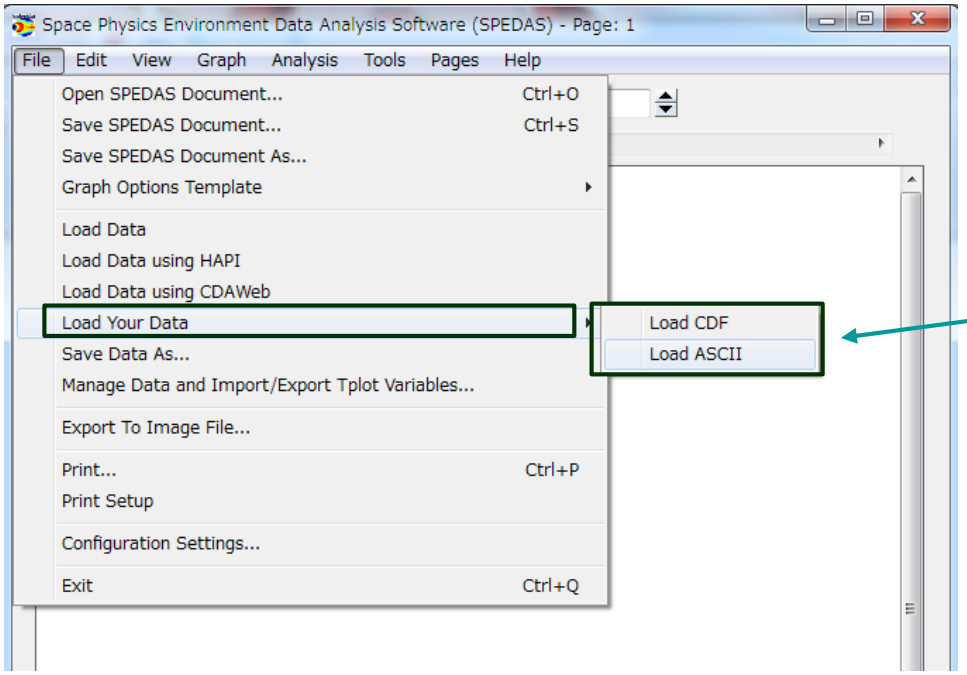


1. インストールしよう

1. SPEDAS Executables版（GUI）を入手します。

   IUGONET開発員から、UDAS egg に対応したExecutable版を入手します。

2. SPEDASの [File] タブに [Load Your Data] メニューがあれば使えます。



UDAS egg を使える
バージョンです。

2. データファイルを読み込もう

1. [File] タブから [Load Your Data] を選択します。
2. データファイルをどのように読み込むかを指定します。

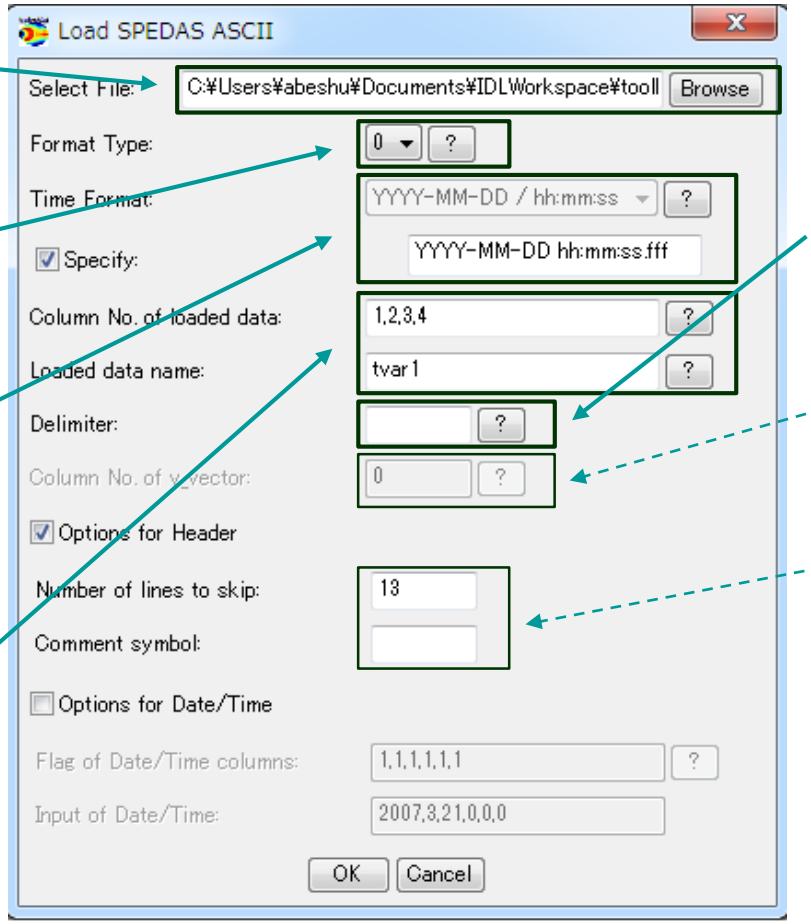
CDFの場合は、読み込みたいファイルを選択して、以降はすべて [OK] を選択します。
ASCIIの場合は、下図に従います。

読み込みたいデータファイルを指定します。

ファイルの形式を指定します。(0～1、巻頭を参照)

日時の形式を指定します。リストにない場合は、Specify欄で入力します。

読み込む列の番号を入力します。
読み込んだデータに付与する名前を入力します。



デリミタ文字(データの区切り)を入力します。

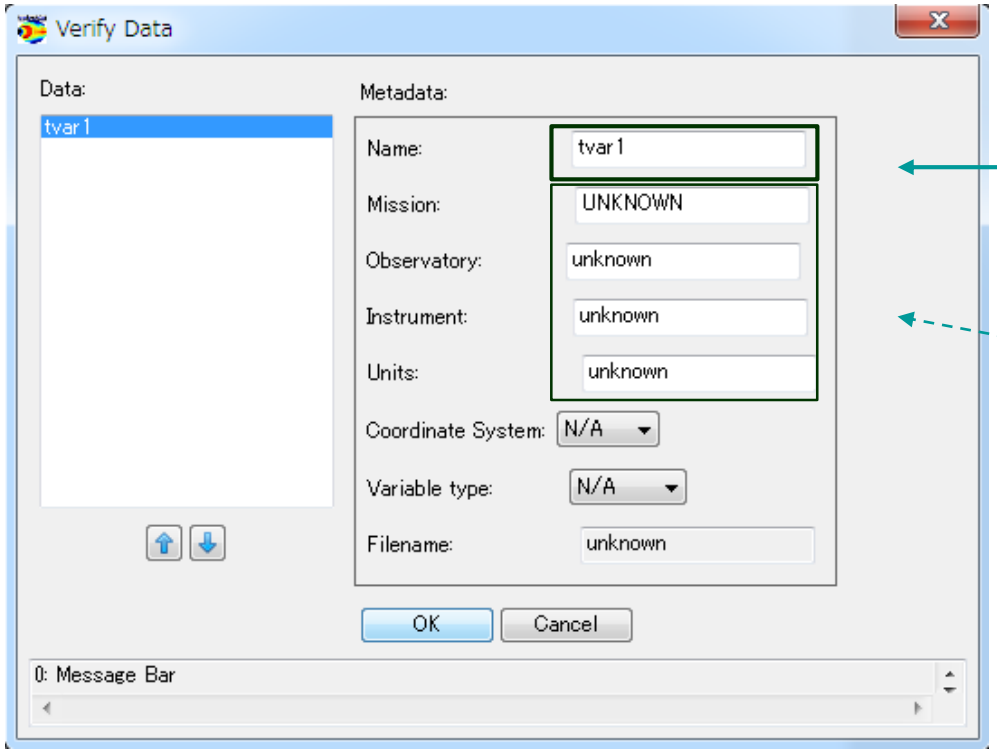
Vデータの列番号(日時列を含まずに0から数えた列位置)を指定します。(Format Type =1 の場合のみ)

読み飛ばす行数と、コメントアウト文字列(読み飛ばすレコード)を指定します。

OKボタンを押します。

2. データファイルを読み込もう

3. tplot変数の階層に付与する名称を入力します。



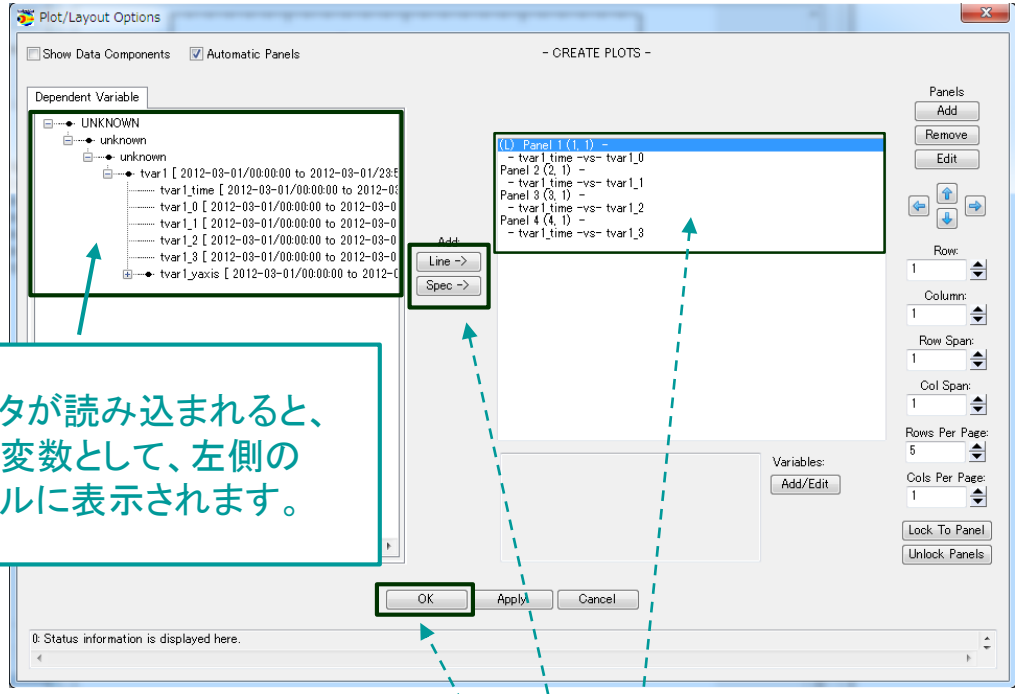
読み込んだデータに付与する名前を入力または確認します。

データがSPEDASに読み込まれてtplot変数が生成された際の、ツリー構造の名称を入力します。
他のデータと組み合わせたい場合に、分かりやすくなります。

OKボタンを押します。

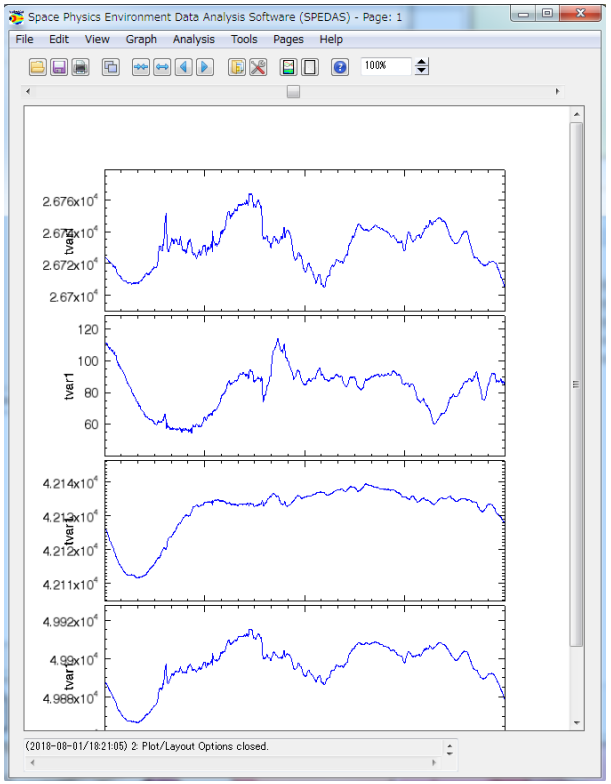
3. プロットを表示しよう

1. SPEDASに戻り、[Graph] タブから [Plot/Layout Options] を選択します。



データが読み込まれると、tplot変数として、左側のパネルに表示されます。

SPEDASの通常の操作と同じく、可視化したいtplot変数を選んで、OKを押します。



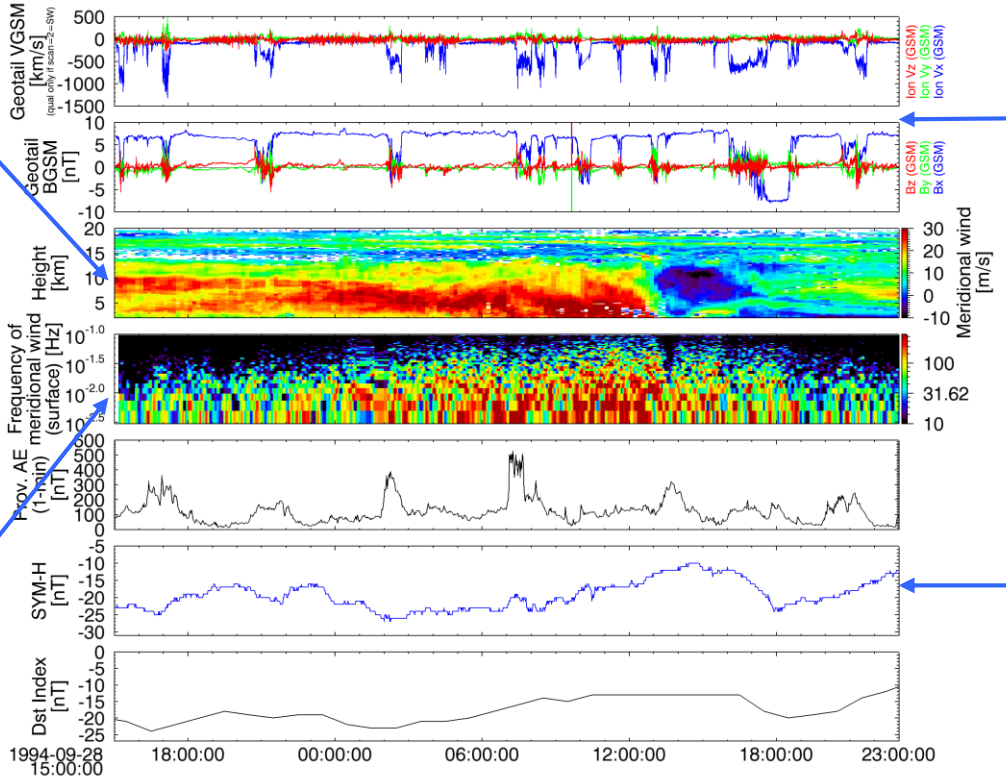
もっと解析してみよう

講習テキスト（共通編）を見ながら、読み込んだデータに解析を加えてみましょう。

可視化の例

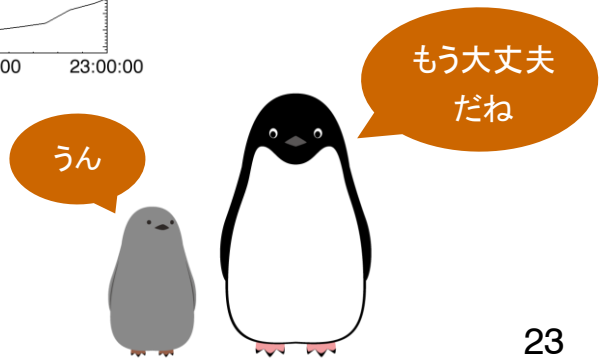
信楽MUレーダー対流圏・成層圏風速データを
1. Shellでファイル加工して、
2. UDAS egg で読み込んで、
3. v成分値をFFTしました。

地上風速データ（ASCII）を
1. UDAS egg で読み込んで、
2. 南北方向成分値をFFTしました。



Geotail衛星の太陽風データをSPEDASで追加読込みしました。

地磁気指数をSPEDASで追加読込みしました。



動作環境と注意事項

動作環境（2018年9月現在）

OS	Windows / MacOS / Linux
IDL	8.0以上　※GUIはIDLライセンス不要
SPEDAS	2.0以上（3.0以上を推奨）
対応フォーマット	CDF（Common Data Format） アスキーフォーマット

注意事項

1. UDAS egg を利用する場合は、IUGONETプロジェクトの利用規則に従ってください。
   利用規約　http://www.iugonet.org/rules/
2. UDAS egg を使用の際、ご自身のパソコンのハードウェアおよびソフトウェア、資産、そのほかに損害が生じてもIUGONETでは責任を負いかねます。ご了承ください。

付録1. プログラム解説

```
1   ;+
2   ; PROCEDURE:
3   ;   loadproc_template, site = site, $
4   ;                      datatype=datatype, $
5   ;                      trange=trange, $
6   ;                      verbose=verbose, $
7   ;                      downloadonly=downloadonly, $
8   ;                      no_download=no_download
9   ;
10  ; PURPOSE:
11  ;   This is an example showing how to create a load procedure for
12  ;   CDF or ASCII files.
13  ;
14  ; KEYWORDS:
15  ;   site  : site
16  ;   datatype : datatype
17  ;   trange : (Optional) Time range of interest  (2 element array).
18  ;   /verbose: set to output some useful info
19  ;   /downloadonly: if set, then only download the data, do not load it
20  ;           into variables.
21  ;   /no_download: use only files which are online locally.
22  ;
23  ; EXAMPLE:
24  ;   loadproc_template_cdf, site = 'aaa', datatype = 'bbb', $
25  ;             trange=['2003-11-20/00:00:00','2003-11-21/00:00:00']
26  ;
27  ; Written by Y.-M. Tanaka, March 13, 2018
28  ; Modified by Y.-M. Tanaka, September 5, 2018
29  ;-
30
```

付録1では、テンプレートプログラムが実際にどのような処理をしているかについて、解説します。

Line 1-29 プログラムヘッダー（コメントアウト＝セミコロン ; で記された行）

記述は必須ではありませんが、プログラムに関する概要を記します。

他人とプログラムを共有する場合や、プログラムをパッケージに組み込んだり、大掛かりなシステムの一部に組み込んで動かしたりする場合は、記述しておくと親切です。

付録1. プログラム解説

```
31  pro loadproc_template, site=site, datatype=datatype, $
32        trange=trange, verbose=verbose, downloadonly=downloadonly, $
33              no_download=no_download
34
35  ;*********************************;
36  ;***** Step1: Set paramters *****;
37  ;*********************************;
38  file_format='cdf'   ; Choose 'cdf' or 'ascii'
39  url='http://www.iugonet.org/data/SITE/DATATYPE/YYYY/mag_SITE_DATATYPE_YYYYMMDD_v??.cdf'
40  acknowledgement = 'You can write the data use policy here.' + $
41     'This description is displayed when you use this load procedure.'
42  site_list='sta1 sta2 sta3'
43  datatype_list='1sec 1min 1hr'
44
45  ;===== Split URL =====;
46  split_url, url=url, remote_data_dir=remote_data_dir, $
47     pathname_base=pathname_base, filename_base=filename_base
48  ipos_local=strpos(remote_data_dir, '://')+3
49  local_data_dir  = root_data_dir() + $
50     strmid(remote_data_dir, ipos_local, strlen(remote_data_dir)-ipos_local)   ; Base local directory
51  ; remote_data_dir='http://www.iugonet.org/data/'
52  ; local_data_dir  = root_data_dir() + 'tmp/'
53
```

--- Line 45～53: 読込みたいファイルのURLと、保存先パスの構築 ---

Line 46-47, split_url: url変数から各種URI, PATHの作成　*SPEDAS予約語

39行目で記述したURL文字列を、1. remote_data_dir: 読み込みたいファイルが置いてあるリモートサーバのURL（ファイル名部分は除く）、2. pathname_base: 手元で保存先する先のディレクトリのサフィックス、3. filenames_base: ファイル名 に分割します。1～3はすべてSPEDASに引き渡す変数です。変数名を変えてはいけません。

Line 49-52, local_data_dir: 手元のパソコンにおけるデータ保存先　*SPEDAS予約語

SPEDASによるデータ保存先ディレクトリのサフィックス標準値　root_data_dir() = windows の場合はc:¥data　を使って保存先の絶対パスを構成します。

Line 51, remote_data_dir: 自動作成でなく自分でカスタマイズしたい場合に、ここに記述します。

Line 52, local_data_dir: 自動作成でなく自分でカスタマイズしたい場合に、ここに記述します。

付録1. プログラム解説

```
54  ;===== Keyword check =====;
55  ;----- default -----;
56  if ~keyword_set(verbose) then verbose=0
57  if ~keyword_set(downloadonly) then downloadonly=0
58  if ~keyword_set(no_download) then no_download=0
59
60  ;----- remote_data_dir -----;
61  if remote_data_dir eq '' then remote_data_dir=' '
62
63  ;----- site -----;
64  if n_elements(site_list) eq 0 then site_list='sta'
65  site_all = strsplit(site_list, /extract)
66  if(not keyword_set(site)) then site='all'
67  site = ssl_check_valid_name(site, site_all, /ignore_case, /include_all)
68  if site[0] eq '' then return
69
70  ;----- datatype -----;
71  if n_elements(datatype_list) eq 0 then datatype_list='dt'
72  datatype_all=strsplit(datatype_list, /extract)
73  if(not keyword_set(datatype)) then datatype='all'
74  datatype=ssl_check_valid_name(datatype, datatype_all, /ignore_case, /include_all)
75  if datatype[0] eq '' then return
76
```

--- Line 54～76: 実行キーワード・実行パラメータ、内部変数の入力状態確認とそのエラー処理 ---

Line 56-58, ～keyword_set(キーワード名): キーワード有無確認 *IDL関数

実行時にキーワードが入力されているかを確認し、入力されていない場合はどのような初期値をセットするか指定します。～記号は否定を示します。解析用プログラムでは、作法として書くのが通例です。

Line 61, [リモートサーバURL] eq '': リモートサーバURL確認 *IDL演算子

保存先情報が指定されているかを eq 演算子を使って確認し、指定されていない場合は適切な値を補います。

Line 64, n_elements(配列名) eq 0: 配列状態の確認 *IDL関数

引数に与えた配列の要素数を調べて、0個の場合は適切な値を補います。

Line 65, 配列名 = strsplit(文字列, /extract): 配列への格納 *IDL関数

文字列をセパレーターで分割してIDLの配列（型）に変換します。/extract を付与すると、インデックス値（出現位置）の数値でなく、実際の値を返します。

Line 67, 値 = ssl_check_valid_name(値, 配列名, /ignore_case, /include_all): 配列値確認 *SPEDAS関数

第一引数に与えた値が第二引数に与えた配列に含まれているかを確認します。/ignore_caseを付与すると、大文字と小文字を区別しません。inclide_allは、配列に'all'という文字列が含まれていることが分かっている場合に指定します。

付録1. プログラム解説

```
77  ;----- Make date & time string array -----;
78  ipos=strpos(url, 'hh')
79  if ipos lt 0 then hres=0 else hres=1
80  yyyy = file_dailynames(file_format='YYYY', trange=trange, hour_res=hres)
81  yy = file_dailynames(file_format='yy', trange=trange, hour_res=hres)
82  mm = file_dailynames(file_format='MM', trange=trange, hour_res=hres)
83  dd = file_dailynames(file_format='DD', trange=trange, hour_res=hres)
84  hh = file_dailynames(file_format='hh', trange=trange, hour_res=hres)
85  yyyymm = file_dailynames(file_format='YYYYMM', trange=trange, hour_res=hres)
86  yyyymmdd = file_dailynames(file_format='YYYYMMDD', trange=trange, hour_res=hres)
87  yyyymmddhh = file_dailynames(file_format='YYYYMMDDhh', trange=trange, hour_res=hres)
88
```

--- Line 77～88: 時刻の取り出し（UDAS egg 固有処理） ---

Line 78, strops(文字列1, ’文字列2’): 文字列出現位置の確認 *IDL関数

第一引数に与えた文字列1に文字列2が含まれるか判定し、含まれる場合は最初のインデックス値（出現位置）、含まれない場合は-1を返します。ここでは出現確認をしています。

Line 80-87, file_dailynames(日時定義文字列, 値, 時間フラグ) *SPEDAS関数

SPEDASにセットされたYYYYなどの予約語から、具体的な値をyyyyなどに入力します。UDAS egg 固有処理のため、通常は変更する必要はありません。

付録1. プログラム解説

```
89   ;===== Download files, read data, and create tplot vars at each site =====
90   ;----- Loop -----
91   for isite=0, n_elements(site)-1 do begin
92     for idt=0, n_elements(datatype)-1 do begin
93
94       ;----- Set parameters for spd_download -----;
95       source = file_retrieve(/struct)
96       source.verbose = verbose
97       source.local_data_dir  = local_data_dir
98       source.remote_data_dir = remote_data_dir
99       if keyword_set(no_download) then source.no_download = 1
100      if keyword_set(downloadonly) then source.downloadonly = 1
101
 …       …

191    endfor
192  endfor
```

--- Line. 89-192: データの読込み、保存、tplot変数（可視化構造体）への値の格納 ---

Line 91, for [観測所配列] do begin ***IDL記述**
観測所リストとして定義した配列に対して、ループを実行します。

Line 92, for [データ種配列] do begin ***IDL記述**
データ種として定義した配列に対して、ループを実行します。

2重ループにすることで、UDAS egg は、url変数に記述したSITEとDATATYPEを、具体的な観測所とデータ種の値に自動的に置換していきます。

Line 95, source = file_retrieve(/struct) ***SPEDAS関数**
データファイル取得のための通信オブジェクトを作成します。

Line 96, source.verbose = verbose ***SPEDASオブジェクト**
通信のためのパラメータを初期化します。verboseはIDLやSPEDASで初期値を示す予約変数や予約オブジェクトとして使われます。

Line 97, source.local_data_dir = local_data_dir ***SPEDASオブジェクト**
データの保存先情報をオブジェクトに格納します。

Line 98, source.remote_data_dir = remote_data_dir ***SPEDASオブジェクト**
読込みたいファイルのURL情報（ファイル名は含まない）をオブジェクトに格納します。

Line 99, source.no_download = 1 ***SPEDASオブジェクト**
通信によるファイルのダウンロードを省略し、パソコンに保存したファイルを読込みたい場合に '1' を指定します。

Line 99, source.downloadlonly = 1 ***SPEDASオブジェクト**
通信によるファイルのダウンロードのみを行い、データを読み込まない場合に '1' を指定します。

付録1. プログラム解説

```
102   ;----- Make relpathnames -----;
103     replace_strings, pathname_base, 'SITE', site[isite], pathnames
104     replace_strings, pathnames, 'DATATYPE', datatype[idt], pathnames
105     replace_strings, pathnames, 'YYYYMMDDhh', yyyymmddhh, pathnames
106     replace_strings, pathnames, 'YYYYMMDD', yyyymmdd, pathnames
107     replace_strings, pathnames, 'YYYYMM', yyyymm, pathnames
108     replace_strings, pathnames, 'YYYY', yyyy, pathnames
109     replace_strings, pathnames, 'yy', yy, pathnames
110     replace_strings, pathnames, 'MM', mm, pathnames
111     replace_strings, pathnames, 'DD', dd, pathnames
112     replace_strings, pathnames, 'hh', hh, pathnames
113     replace_strings, filename_base, 'SITE', site[isite], filenames
114     replace_strings, filenames, 'DATATYPE', datatype[idt], filenames
115     replace_strings, filenames, 'YYYYMMDDhh', yyyymmddhh, filenames
116     replace_strings, filenames, 'YYYYMMDD', yyyymmdd, filenames
117     replace_strings, filenames, 'YYYYMM', yyyymm, filenames
118     replace_strings, filenames, 'YYYY', yyyy, filenames
119     replace_strings, filenames, 'yy', yy, filenames
120     replace_strings, filenames, 'MM', mm, filenames
121     replace_strings, filenames, 'DD', dd, filenames
122     replace_strings, filenames, 'hh', hh, filenames
123     relpathnames = pathnames + filenames
124
```

--- Line. 102-123: URLの組み立て（UDAS egg 固有処理）---

Line 103-122, URLの組み立て（パス）

読込みたいデータのURL（ファイル名は含まない）を組み立てます。UDAS egg 固有処理のため、通常は変更する必要はありません。

Line 123, URLの組み立て（ファイル名含む）

読込みたいデータのURL（ファイル名含む）を組み立てます。UDAS egg 固有処理のため、通常は変更する必要はありません。

付録1. プログラム解説

```
125   ;----- Download data files -----;
126     files = spd_download(remote_file=relpathnames, $
127               remote_path=source.remote_data_dir, $
128               local_path=source.local_data_dir, $
129               no_server=no_server, $
130               no_download=no_download, $
131               _extra=source, /last_version)
132
133     filestest=file_test(files)
134     if total(filestest) ge 1 then begin
135        files=files(where(filestest eq 1))
136     endif
137
```

--- Line. 125-137: ファイルの取得 ---

Line 126-131, spd_download(.., ..) *SPEDAS関数

SPEDASの関数を使って、ファイルの取得を試みます。

Line 133-137, ファイル取得結果の確認 *IDL記述

IDLのfile_test関数を使ってファイル群（複数）が実在するかを確認します。file_test関数の戻り値は1（true）または0（false）です。
ファイル群が実在した場合は、その中で実際に存在したもののみ、files変数にセットします。

付録1. プログラム解説

```
138   ;----- Load data into tplot variables -----;
139     if(downloadonly eq 0) then begin
140       ;************************************************;
141       ;***** Step2: Load data into tplot variables *****;
142       ;************************************************;
143       case file_format of
144         'cdf': begin
145           ;----- For CDF format files -----;
 …          …  中略 …
148           cdf2tplot, files=files, verbose=source.verbose, $
149             prefix=prefix, suffix=suffix
150         end
151         'ascii': begin
152           ;----- For ASCII format files -----;
 …          … 中略 …
164           ascii2tplot, files, files=files2, format_type=format_type, $
165             tformat=tformat, tvar_column=tvar_column, $
166             tvarnames=tvarnames, delimiter=delimiter, $
167             data_start=data_start, comment_symbol=comment_symbol, $
168             v_column=v_column, vvec=vvec, $
169             time_column=time_column, input_time=input_time
170         end
171         else: begin
172           print, 'Not support the file format: '+file_format
173           return
174         end
175       endcase
 …       … 中略 …
190     endif
```

--- Line. 138-190: 読み込んだファイルのtplot変数への格納 ---

Line 148-149, cdf2tplot *SPEDAS関数

SPEDASのcdf2tplot関数を使って、読み込んだCDFファイルをtplot変数（可視化・解析するための構造体）に格納します。

Line 164-169, ascii2tplot *UDAS egg 関数

UDAS egg が準備したascii2tplot関数を使って、読み込んだアスキーファイルをtplot変数に格納します。

付録1. プログラム解説

```
194  ;----- Display data policy -----;
195  print_str_maxlet, acknowledgement
196
197  end
```

--- Line. 194-195: データ利用ポリシーの表示---

Line 194-145, print_str_maxlet, acknowledgement *SPEDAS関数

指定したacknowledgementの文字列値を標準出力します。

以上でプログラム処理は終了です。
読み込んだデータをtplot変数に格納することができれば、可視化や解析の準備が整っている状態にあります。