

太陽圏サイエンスセンターデータ解析講習会 (PySPEDASコース番外編)

新堀淳樹

(名古屋大学宇宙地球環境研究所)

事前アンケート結果1

SPEEDASの講習でどういうことを知りたい・身につけたいですか？ (例: 粒子データの速度モーメント計算、地上観測データとの比較、...)	あらせデータ解析等で最近困っていることはありますか？ もしあれば、それについて教えてください。無ければ「無し」とお答えください。
PySPEEDASの使用感をざっくりと知りたい。	無し
PySpedasの環境設定など基本的な使い方	無し
粒子データの密度・温度モーメント計算、等	* FT/ET diagram等で「データがない部分」のplotが、Python/matplotだと一見つながってしまうこと(一般的な場合ですが、pyspedasでどうなのかは要確認) * 論文にでてくる図をどう作ったか、の事例紹介があると、割と身にしみるかも。
どういうデータがあるかを知りたい	無し
衛星・探査機の観測データの扱いの基本	無し
粒子データの表示	無し
IDLで作ったプロシージャを自動でpySPEEDASのプロシージャに変換できてほしい	なし



事前アンケート結果2

SPEEDASの講習でどういうことを知りたい・身につけたいですか？ (例: 粒子データの速度モーメント計算、地上観測データとの比較、...)	あらせデータ解析等で最近困っていることはありますか？ もしあれば、それについて教えてください。無ければ「無し」とお答えください。
あらせ衛星以外にどのようなデータをロード(取得)できるか	無し
講師補助	講師補助
地上観測データとの比較	無し
コードの書き方	あらせのデータに合わせて、pythonでのみ提供されているモデル計算のパッケージを利用したい
地上観測データとの比較	XEPのシンチレータ(GSO)のデータをpythonでもIDLでもspedasから落とせず、原因が自分のミスなのかそもそもデータが公開されていないのかわからず困っている。
風間さんの電子モーメント読み	無し
現在のPySPEEDASの使いやすさを知りたい	なし
波動の瞬時位相を出すとか... 描画のテクニックいろいろ	HEP L3 PA が pyspedas でダウンロードできない。
HEPと他衛星との比較	IDL Spedas の nn ってあるのか



PySPEDAS features and capabilities

(<https://drive.google.com/drive/u/0/folders/1RZgRtVowhdcMUHuDd1Mv9w3fWCEwYVWH>から抜粋)

- **Load data from many different providers and formats into a common environment**
 - Support for directly loading data from 30+ missions (including THEMIS, MMS, ERG/Arase, RBSP/Van Allen probes, Parker Solar Probe, FAST, WIND, many others)
 - Load via CDAWeb web service
 - Load via HAPI
- **Analysis and Modeling tools**
 - Interface to native Python geopack package
 - avg_data, deriv_data, FFT, wavelet
 - Field-aligned coordinates, minimum variance analysis, wave polarization, etc
- **Plotting tools**
 - Line plots
 - Spectrograms
 - Being worked on: interactive plots
(zoom in on specific time ranges, see time/data values at cursor, etc)
 - Also coming soon: 3-D particle distribution interactive visualization tool



PySPEDASの最近の状況

- ▶ 多くの衛星観測データが利用できるようになってきている
 - 現在では、30以上の衛星ミッションのデータをサポート(THEMIS, MMS, ERG/Araseなど)
 - 詳細は、<https://github.com/spedas/pyspedas/tree/master/pyspedas>を参照
 - 各衛星データなどをロードする仕方は
<https://github.com/spedas/pyspedas/tree/master/docs/source>
を参照
 - IDL/SPEDAS 6.0がリリースされているが、これとはpySPEDASは同期していない
- ▶ 今後の開発プラン
(<https://drive.google.com/drive/u/0/folders/1RZgRtVowhdcMUHuDd1Mv9w3fWCEwYVWH>から
抜粋)
 - More missions and datasets
 - Improved modeling tools (e.g. additional GEOPACK models, field line tracing)
 - More interactivity with plots (e.g. mousing over to get times and data values)
 - More wrappers for working with particle datasets for additional missions (e.g. THEMIS)



地上観測データとの比較

PySPEDASには、ERG-SCが開発しているいくつかの地上観測データをロードするモジュールが含まれているので、以下の地上観測データも利用可能です。

名大ISEEリオメーター、名大ISEE OMTIカメラ、名大ISEE VLFアンテナ、名大ISEE・フラックスゲート磁力計、名大ISEE・誘導磁力計、MAGDAS磁力計ネットワーク、210度磁気子午線磁力計ネットワーク、SuperDARNレーダー

IUGONETが開発しているプラグインツールをインストールする

ダウンロード元URL: <https://github.com/iugonet/pyudas>

pyUDASに含まれているロード関数:

No.	データの種類	ロード関数名
1	極地研・全天イメージャデータ	asi_nipr
2	極地研・全天イメージャ・ケオグラムデータ	ask_nipr
3	極地研・フラックスゲート磁力計データ	gmag_nipr
4	極地研・誘導磁力計データ	gmag_nipr_induction
5	EISCATレーダーデータ	eiscat
6	北大・誘導磁力計データ	elf_hokudai
7	京大WDC・地磁気・指数データ	gmag_wdc
8	九大・GCMシミュレーションデータ	kyushugcm

[IUGONETより]

※開発途中であり、まだバグを含んでいる可能性が高いことにご留意ください。

※2024年度中に複数のロード関数を追加していきます。

IDLとPythonの連携

- ▶ Python Bridge を用いることで、IDLでPythonのコードを実行 (IDL to Python Bridge)、PythonでIDLのコードを実行 (Python to IDL Bridge)し、データをやりとりすることができる

詳しくは、<https://www.nv5geospatialsoftware.com/docs/python.html>を参照

ただし、IDLとpythonのバージョンの相性があるので注意

← PySPEDASに準拠 →

	Python 2.7	3.4	3.5	3.6	3.7	3.8	3.9	3.10	3.11	3.12
IDL 9.0	No	No	No	No	No	Yes	Yes	Yes	Yes	No
IDL 8.9	No	No	No	No	No	Yes	Yes	Yes	No	No
IDL 8.8.2, 8.8.3	No	No	No	No	Yes	Yes	Yes	Yes	No	No
IDL 8.8.1	No	No	No	Yes	Yes	Yes	Yes	No	No	No
IDL 8.8	No	No	No	Yes	Yes	Yes	No	No	No	No
IDL 8.7.1-8.7.3	Yes	No	Yes	Yes	No	No	No	No	No	No
IDL 8.6.1, 8.7	Yes	Yes	Yes	Yes	No	No	No	No	No	No
IDL 8.5.2, 8.6	Yes	Yes	Yes	No	No	No	No	No	No	No
IDL 8.5, 8.5.1	Yes	Yes	No	No	No	No	No	No	No	No



IDLとPythonの連携

- ▶ PythonからIDLのコードを実行する (**Python to IDL Bridge**)
(<https://www.nv5geospatialsoftware.com/docs/pythontoidl.html>)
 - PythonからIDLのコードを実行するために必要な条件:
 1. Pythonがモジュールを検索するパスに idlpy.pyが置かれているパス (\$IDL_DIR/lib/bridges) が含まれていること
 2. 【IDL 8.8.3以前のみ】
Python がモジュールを検索するパスに pythonidlXX.so (XXはPythonのバージョン) が置かれているパス(\$IDL_DIR/bin/bin.linux.x86_64)が含まれていること
共有ライブラリの探索パスに pythonidlXX.so (XXはPythonのバージョン) が置かれているパス (\$IDL_DIR/bin/bin.linux.x86_64)が含まれていること
 3. Python で NumPy モジュールが利用可能であること
 4. Python がモジュールを検索するパスを追加するには、Python 起動前ならPYTHONPATH環境変数で、Python 起動後なら sys.path.append を使う。
- ▶ 共有ライブラリの探索パスを追加するには LD_LIBRARY_PATH環境変数を用いる。



そのほか

- ▶ `ascii_dump`の仕方

IDL版にあった`tplot_ascii`みたいなモジュールがないので、PySPEDAS側へ要求

- ▶ `pySPEDAS`の更新に伴って正常な動作が期待できなくなったものもある

例えば、`pytplot.split.vec`

→ 実行後、生成された`tplot`変数にプロットに必要な情報が引き継がれない

- ▶ あらせフットプリントのプロットの仕方

軌道データ (1v.3) に電離圏にマップした緯度、経度データがあるので、それを利用する

```
orb(trange=trng, level = '13', model = 't89')
```

- ▶ `pyspedas.erg`でロードできないデータがある

データによってはCDFデータが生成・公開されていない場合

(例えば、

XEPのシンチレータ(GSO)のデータ

HFA-L3データ(必要なデータセットがそろっていない場合(MGF+PWE-HFAとか))

データ構造が読めない形式になっている・・・LEP電子データ

Python言語のニーズと有効性

転職で企業からニーズが高い言語ランキング(2020年、2022年比較)



2022年順位 カッコ内は2020年の順位	言語	2020年との比較	言語別求人数比率
1位(1)	JavaScript	→ 0	15.6%
2位(2)	Java	→ 0	14.0%
3位(3)	PHP	→ 0	13.1%
4位(5)	Python	↑ 1	8.0%
5位(4)	C#	↓ -1	7.8%
6位(11)	TypeScript	↑ 5	6.7%
7位(6)	Ruby	↓ -1	5.1%
8位(12)	Kotlin	↑ 4	4.9%
9位(10)	Swift	↑ 1	4.8%
10位(7)	C++	↓ -3	3.8%
11位(13)	Go	↑ 2	3.4%
12位(8)	C	↓ -4	3.3%
13位(9)	Objective-C	↓ -4	3.2%
14位(14)	Visual Basic(VB.NET)	→ 0	2.6%
15位(16)	Sass	↑ 1	1.3%
16位(17)	Scala	↑ 1	1.2%
17位(15)	Perl	↓ -2	1.2%

●Pythonでできること

- Webサイトの制作(Instagram, YouTube)
- データ収集
- 人工知能の開発
- データ分析(Numpy, SciPy)

●Pythonが苦手なこと

- スマホアプリ開発 → Java, Swift, Kotlin, C#
- 高速な処理が求められるもの → C, IDL, Fortran
- デスクトップアプリ開発 → Java, C#

<https://www.sejuku.net/blog/wp-content/uploads/2023/05/image-76.png>



Installation of PySPEDAS

PySPEDASの動作環境

Windows、macOS、Linuxをサポートしています。

Python **3.8以上**のバージョンが必要です。

pySPEDASの公式ウェブサイトでは、Pythonの利用環境として、Anacondaを推奨していますが、利用に際し有償ライセンスが必要になる方、うまく動作しない方はPythonにJupyterLabのインストールをお勧めします。

Pythonのインストール

Pythonをインストールしていない場合は、以下のサイトを参考に、Pythonをインストールします。

- macOS: <https://www.python.jp/install/macos/index.html>
- Windows: <https://www.python.jp/install/windows/index.html>
- Linux(Ubuntu): <https://www.python.jp/install/ubuntu/index.html>

仮想環境を作成する

他のPythonパッケージとの依存関係の問題を避けるために、PySPEDAS用の仮想環境を作成します。

各OSのターミナルで以下を実行します。

Windows, macOS, Linux共通:

```
python -m venv pyspedas
```



Installation of PySPEDAS

仮想環境を開始する

ターミナルでactivateスクリプトを実行します。

Windows:

コマンドプロンプトの場合

```
¥pyspedas¥Scripts¥activate.bat
```

PowerShellの場合

```
¥pyspedas¥Scripts¥Activate.ps1
```

macOS, Linux:

```
source pyspedas/bin/activate
```

(参考)仮想環境を終了する

ターミナルでactivateスクリプトを実行します。

Windows:

コマンドプロンプトの場合

```
deactivate.bat
```

Windows: PowerShellの場合, macOS, Linux:

```
deactivate
```

仮想環境が有効になると、プロンプトの先頭に(pyspedas)と表示されます

```
PowerShell 7.3.8
PS C:\Users\abeshu> python -m venv pyspedas
PS C:\Users\abeshu> .\pyspedas\Scripts\Activate.ps1
(pyspedas) PS C:\Users\abeshu>
```

仮想環境が終了すると、プロンプトの先頭の(pyspedas)が表示されなくなります



Installation of PySPEDAS

JupyterLabのインストール

ブラウザ上で動作する対話型プログラム実行環境のJupyterLabを使う場合、ターミナル(仮想環境有効状態)で以下を実行してください。

Windows, macOS, Linux:

```
python -m pip install jupyterlab
```

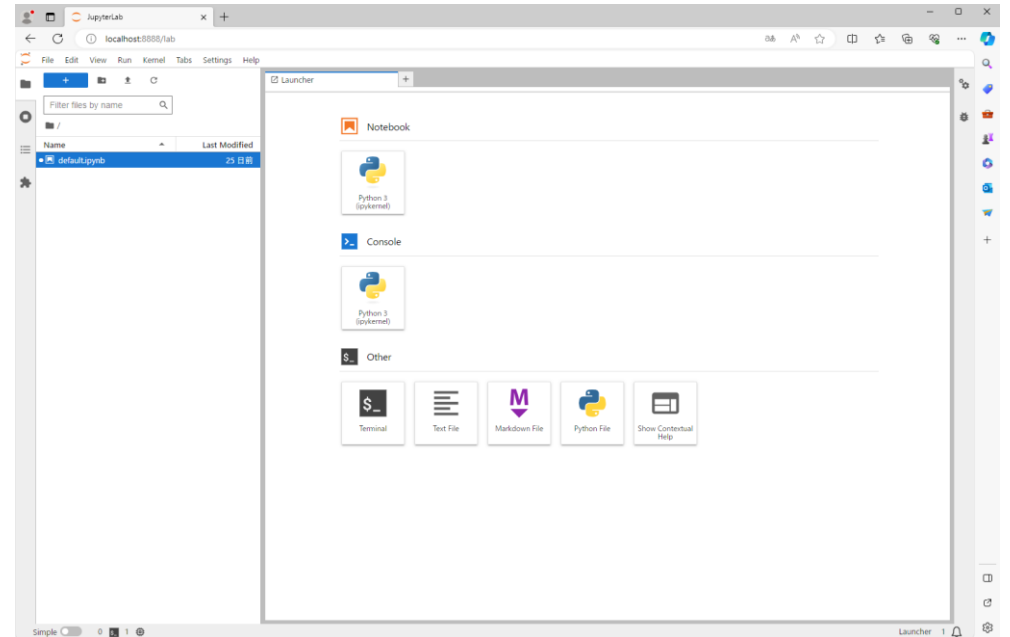
JupyterLabの起動

ターミナル(仮想環境有効状態)で以下を実行してください。

Windows, macOS, Linux:

```
jupyter lab
```

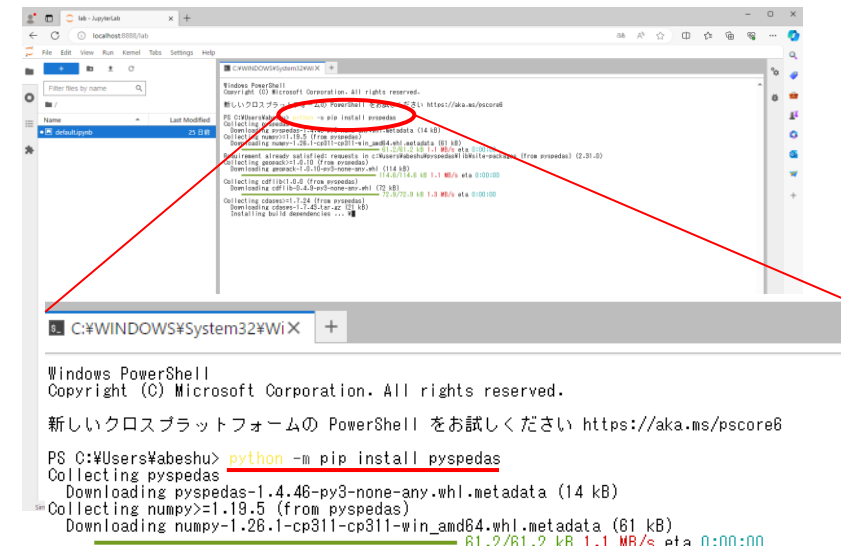
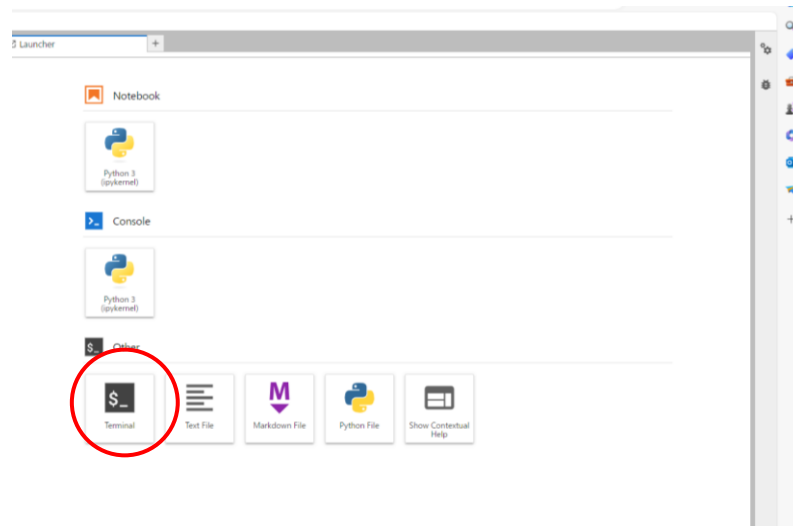
ブラウザが自動で開き、右図のようなJupyterLabのウィンドウが現れます。



Installation of PySPEDAS

pySPEDASのインストール

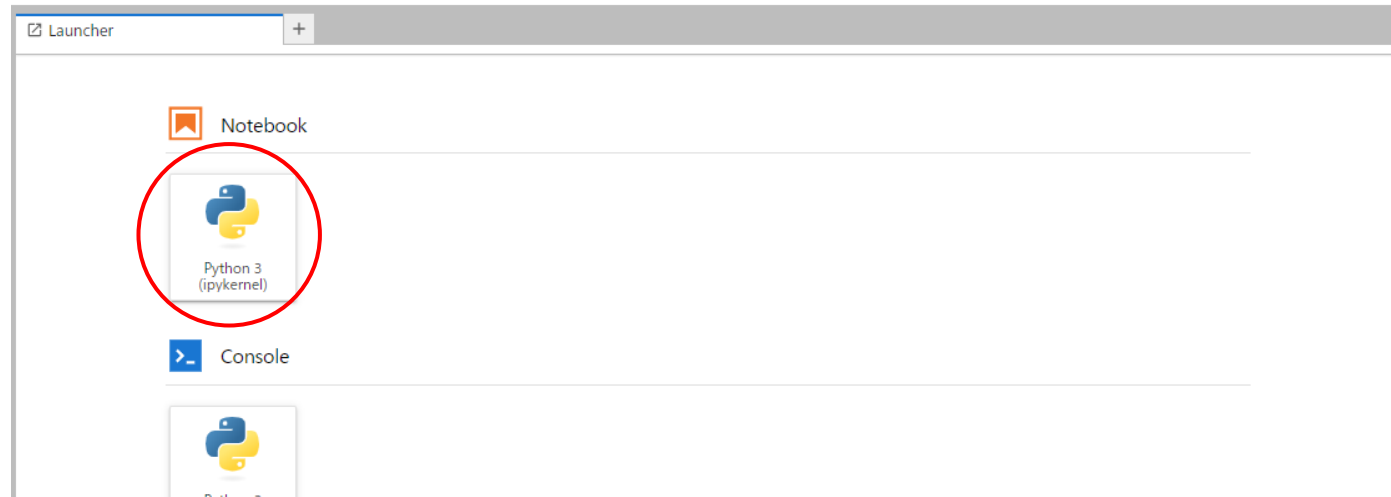
1. Launcher > Othersの「Terminal」をクリックします。
2. 新しいタブでターミナルが開くので、以下を実行します。
`python -m pip install pypedas`
3. タブの×ボタンを押して、ターミナルを閉じます。



Installation of PySPEDAS

Notebookの起動

1. Launcher>NoteBookの「Python3(ipynkernel)」をクリックします。



以下のようなウィンドウが開けば、準備OKです。

