

研究室インターンレポート

小山田研究室 1030-30-5061 青山望

1. 導入

宇宙分野で用いられる観測機器や計算手法などは、高額であったり一定の需要しかないことから、2000年以前に発明された物が未だ使われ続けていることもある。こうした機器は長く用いられていることもあり、ライブラリや拡張機能が充実している。しかし、近年の科学の発展に伴い、同様の機能を実現するより高性能な機器やプログラミング言語も登場していて、そうした新しい技術を取り入れようという動きもある。

衛星の観測データはcdfというデータ形式で取り扱われる。本インターンシップでは、このcdf宇宙データ取り扱いのための基礎的な処理プログラム作成と、その結果の可視化をPythonという言語を用いて行った。Pythonはコードがシンプルであり扱いやすいように設計されていて、また少ない行数で記述できるといった特徴があり、近年ではNASAにも用いられ、機械学習にも利用される汎用性の高い言語である。Pythonを用いて宇宙データの処理が可能になれば、機械学習など他の技術にも応用が容易になり、かつNASAなどの他の宇宙機関にも連携が取れるようになることが期待される。

2. 手法

Pythonでcdfデータを読み取るために、今回はSpacePy(<https://pythonhosted.org/SpacePy/>)というライブラリを用いた。このライブラリは宇宙科学を対象としたPythonのライブラリであり、cdfファイルを読み取ることが可能である。本章では、このライブラリを用いてcdfファイルを読み込み、データを可視化する手法について述べる。

2.1 SpacePyのインストール

本節ではSpacePyのインストール方法について説明する。尚今回はMacOSへのインストール方法について述べる。PythonのバージョンはPython2.6以降が推奨されている。Python3系でも動作はするが、Python2系を使用することが推奨されている。

2.1.1 依存関係のあるパッケージのインストール

SpacePyのインストールにあたっては、依存関係のあるパッケージをインストールすることが必要である。インストールする必要があるのは、numpy, scipy, matplotlib, networkx, ffnetの5つであり、どれもPythonのパッケージ管理システムであるpipを通してインストールできる。尚、これらの名称をrequirement.txtファイルに打ち込み、'pip install -r requirement.txt'を実行することで一括インストールが可能である。

2.1.2 SpacePyのダウンロードとインストール

<https://sourceforge.net/projects/spacepy/> から ~tar.gzをダウンロードする。ファイルを解凍後、そのディレクトリに移動し以下のコマンドを入力する。

- ・ 'python setup.py build'
- ・ 'python setup.py install --user'

これにより、SpacePyをインストールすることができる。

2.2 CDFファイルの読み込み

SpacePy中のpycdf.CDF()関数によりCDFデータを読み込むことができる。読み込んだデータは辞書型をしていて、いくつかのインデックスに分けられてデータが保存されている。

2.3 データの処理・可視化

本インターンシップでは、erg_pwe_wfc_e_l1_prime_65khz_2017041301_v05.cdf というcdfファイルに含まれる2つの電界データについて、その時間変化と周波数スペクトルを可視化した。合計の観測点中からユーザーが指定した興味のある範囲について電界の時間変化を抜き出し、fftによってそのスペクトルを計算した。

電界データは補数になっており、数値が32767を超える場合は65536を減算した。fftにはnumpyというライブラリのfft.fft()関数を用いていて、サンプリング周波数は65536とした。また、可視化にはmatplotlibというライブラリを使用している。

3. 結果

3.1 CDFデータ読み出し

pycdf.CDF() 関数を用いてerg_pwe_wfc_e_l1_prime_65khz_2017041301_v05.cdf を読みだした結果、以下のインデックスを確認した。

```
ap_id: CDF_UINT1 [3835]
attr_id: CDF_UINT1 [3835]
cat_id: CDF_UINT1 [3835]
ccsds_hdr: CDF_UINT1 [3835, 22]
ch1: CDF_UINT1 [3835]
ch2: CDF_UINT1 [3835]
cmp: CDF_UINT1 [0]
dr_id: CDF_UINT1 [3835]
e1_waveform: CDF_REAL4 [3835, 8192]
e2_waveform: CDF_REAL4 [3835, 8192]
epoch: CDF_TIME_TT2000 [3835]
fm_hdr: CDF_UINT1 [3835, 8]
head_id: CDF_UINT1 [3835]
input_mode: CDF_UINT1 [3835]
```

```
lofo_id: CDF_UINT1 [3835]
lpf: CDF_UINT1 [3835]
obs_cal: CDF_UINT1 [3835]
pga: CDF_INT1 [3835]
sampling_rate: CDF_REAL4 [3835]
source: CDF_UINT1 [3835]
swpia_cnt: CDF_UINT4 [3835]
ti_original: CDF_UINT4 [3835]
time_offsets: CDF_REAL4 [8192] NRV
wpt_ac_gain: CDF_INT1 [3835]
```

このうち、e1_waveform: CDF_REAL4 [3835, 8192] と e2_waveform: CDF_REAL4 [3835, 8192] が今回可視化したデータである。これらは2次元配列の形で格納されているが、n+1行目が時間的にn行目の後に来る1次元配列として扱った。また2.3節で述べたように電界データは補数になっていたためにこれを補正した。

3.2 可視化

3.1節で得られた二つの電界データE1, E2に対して、その時間変化とスペクトルを可視化したものを図1に示す。この際、スペクトルは両対数グラフにて図示した。この際表示範囲は0-20000としてある。

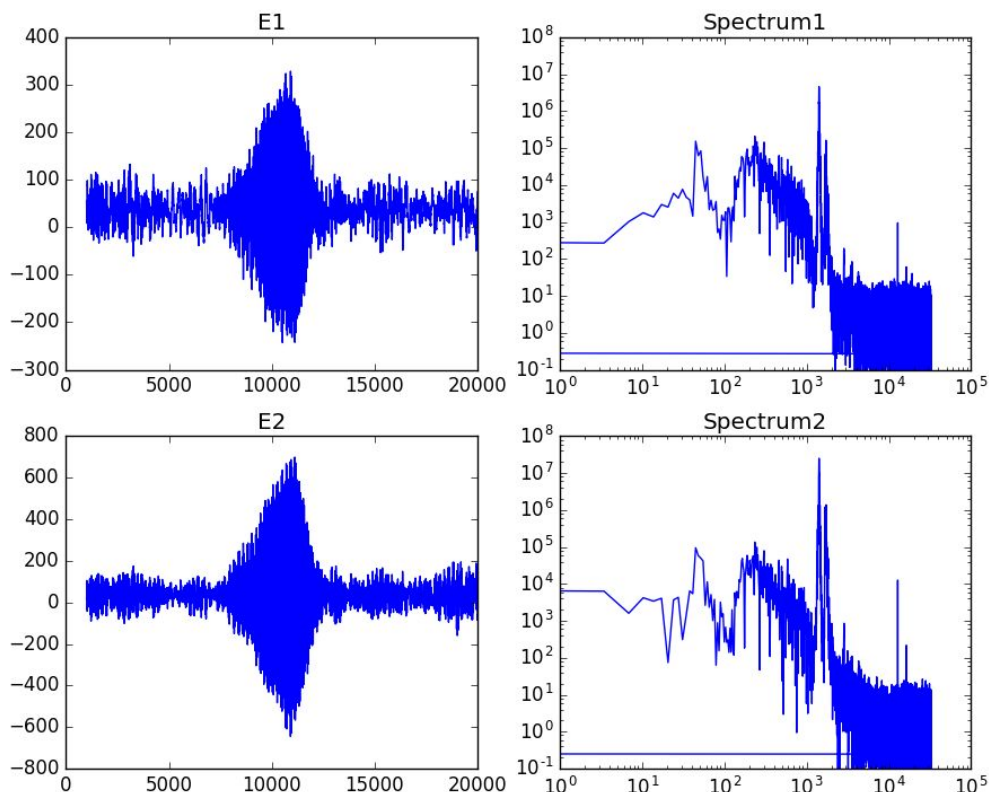


図1: 電界データE1, E2の時間・周波数特性

4. プログラムの評価

SpacePyを用いることで、cdfファイルをPythonで読み込むことができた。今回は電界の波形データのみを用いているが、それ以外のデータも有効に活用することができるであろう。

また、Pythonの他のライブラリを組み合わせることで、信号波形にfftなどの処理を加えた上、可視化することができた。同様の方法で、機械学習などといったPythonがよく用いられる技術との親和性を持たせることが可能であろう。