地球流体データ解析・数値計算ワークショップ　2021/3/31 11:00～12:00

Pythonを用いた気象の理解と模倣

京都大学防災研究所
榎本剛

自己紹介

榎本剛

京大防災研 教授

気象力学，大気大循環モデル

MacPortsコミッタ

地球流体電脳倶楽部との関わり

- 学生の時はDCLを使っていた
- f2cでMacOSのMPWで使えるようにした
    - フリーのFortranコンパイラがなかった
- 電脳Ruby 2005 横浜

お品書き

1. Pythonの特徴
2. 教科書にある図を描いてみる。
    - 標準大気
3. その他のサンプル
    - https://github.com/tenomoto
4. 質疑応答

- 覚えやすい: キーワード36個
- 分かりやすい: 字下げ
- 強力: 標準ライブラリ，外部ライブラリ
- 人気: 人工知能や機械学習
- 枯れている: 最初の登場1991年2月 0.9.0
    - 0.9.1のリポジトリ
    - 最新版は3.9.2
    - Python 2系統は2020年4月の2.7.18が最終版（end-of-life）。

The 2020 Top Programming Languages

Language Ranking: IEEE Spectrum

Rank	Language	Type	Score
1	Python	web, desktop, embedded	100.0
2	Java	web, mobile, desktop	95.3
3	C	mobile, desktop, embedded	94.6
4	C++	mobile, desktop, embedded	87.0
5	JavaScript	web	79.5
6	R	desktop	78.6
7	Arduino	embedded	73.2
8	Go	web, desktop	73.1
9	Swift	mobile, desktop	70.5
10	Matlab	desktop	68.4
11	Ruby	web, desktop	66.8
12	Dart	web, mobile	65.6
13	SQL	desktop	64.6
14	PHP	web	63.8
15	Assembly	embedded	63.7
16	Scala	web, mobile, desktop	63.5
17	HTML	web	61.4
18	Kotlin	web, mobile	57.8
19	Julia	desktop	56.0
20	Rust	web, desktop, embedded	55.6
21	Shell	desktop	52.0
22	Processing	web, desktop	49.2
23	C#	web, mobile, desktop, embedded	48.1
24	SAS	desktop	45.2
25	Fortran	desktop	43.0
26	Cuda	desktop	41.0
27	Visual Basic	desktop	40.3
28	Objective-C	mobile	38.9
29	Delphi	web, mobile, desktop	38.6
30	Perl	web, desktop	38.2

https://spectrum.ieee.org/at-work/tech-careers/top-programming-language-2020

気象でよく使われる言語

- モデル，データ処理: Fortran
- レーダー等データの読み書き: C
- 描画，解析:
  - GrADS: Tcl風（コマンド 引数 引数 ...）の言語。対話型に向いている。
  - NCL: NCARGの上に作られたDSL（ドメイン固有言語）。Pythonに移行中 GeoCAT
  - GMT: ver. 4までコマンド群，ver. 5からgmt サブコマンド，ver. 6 API化 pyGMT
  - 電脳Ruby: 汎用言語Rubyを使ったことに先見の明

Takeshi Enomoto @takeshi_enomoto · 2015年1月13日

gmt4で検索するとGMT本家や使い方をまとめたサイトがヒットするが，gmt5を検索すると「地元に帰ろう」がヒットする。

9　3

今はGMT4で画像検索がGMT47の検索でないと「あまちゃん」は出てこない

データサイエンスでよく使われる言語

R: 統計言語，Sのオープンソース版・後継

- tidyverse: モダンな解析・描画（ggplot2）
- RStudio: IDE

Python: 汎用インタプリタ言語

- Numpy, Scipy: 行列，科学技術計算
- pandas:「表計算」，統計
- matplotlib: 描画

RとPython連携: Jupyter, r2py

どのPython

本家python.orgは最新。Christoph GohlkeさんのWindowsバイナリを使うと便利

MacPortsのPythonは最新についていっている。

外部ライブラリのインストールが楽なcondaがよく使われているが,
pipとの二重になって管理となることがある。

Windows 10ではMicrosoft Storeからも無料で取得できる。

クラウド pythonanywhere, Google Colaboratory, Azure Notebooksは終了

モジュールとパッケージそしてライブラリ

モジュール

- foo.pyはfooという名前のモジュール

パッケージ

- モジュールを集めたディレクトリ。
- __init__.pyというからファイルを置いて名前空間を定義する。

ライブラリ

- Pythonスリクプトや他の言語で書かれたものも含めて特定の機能を提供

モジュールとしての再利用

foo.py

```
def add_two(x, y):
    return x + y
if __name__ == "__main__":
    x = 1
    y = 2
    add_two(x, y)
```

```
import foo
x = 3
y = 6
foo.add_two(x, y)
```

有用なライブラリ

Numpy/Scipy: 配列・科学技術計算

matplotlib: 描画の標準ライブラリ

ipython/jupyter: 対話型シェル，ウェブインターフェース

Cartopy: 地図の描画。地図投影。basemapは開発停止

MetPy: 気象データの

Magics: ECMWF製のC++ライブラリ。Pythonインターフェースを提供

netcdf4/xarray: NetCDFデータの入出力

pandas: PythonでRのような統計

PyArma

C++で書かれたArmadilloのPythonインターフェース

MATLAB寄りの文法で使いやすさを重視

Numpy/Scipyに代わる選択肢

Armadilloのベクトルはなく1行または1列のmatを使う。

疎行列のインターフェースはまだ実装されていない。

MATLAB/Octave	PyArmadillo	Numpy
A(1, 1)	A[0, 0]	A[0, 0]
size(A, 1)	A.n_rows	A.shape[0]
A'	A.t()	A.transpose()
A = zeros(3)	A = zeros(3, 3)	A = np.zeros((3, 3))
A * B	A * B	A @ B
A .* B	A @ B	A * B
A \ B	pa.solve(A, B)	np.linalg.solve(A, B)
[A B]	pa.join_horiz(A, B)	np.hstack((A, B))
save -ascii 'A.txt' A	A.save("A.txt", raw_ascii)	np.savetxt("A.txt", A)

Syntax conversion table

Numpy for MATLAB users

Pythonはなぜ0始まりを使っているのか

Why Python uses 0-based indexingでGuido van Rossumが説明している。

前身のABCは1始まりだった。

Pythonの開発言語Cは0，Guidoが最初に学んだAlgol, Fortran, Pascalは1始まり。

スライスは「最初からn要素を取り出す」という使い方が考えられる。

0始まりの半開区間ならa[:n]やa[i:i+n]と書けてきれい。

Jupyter

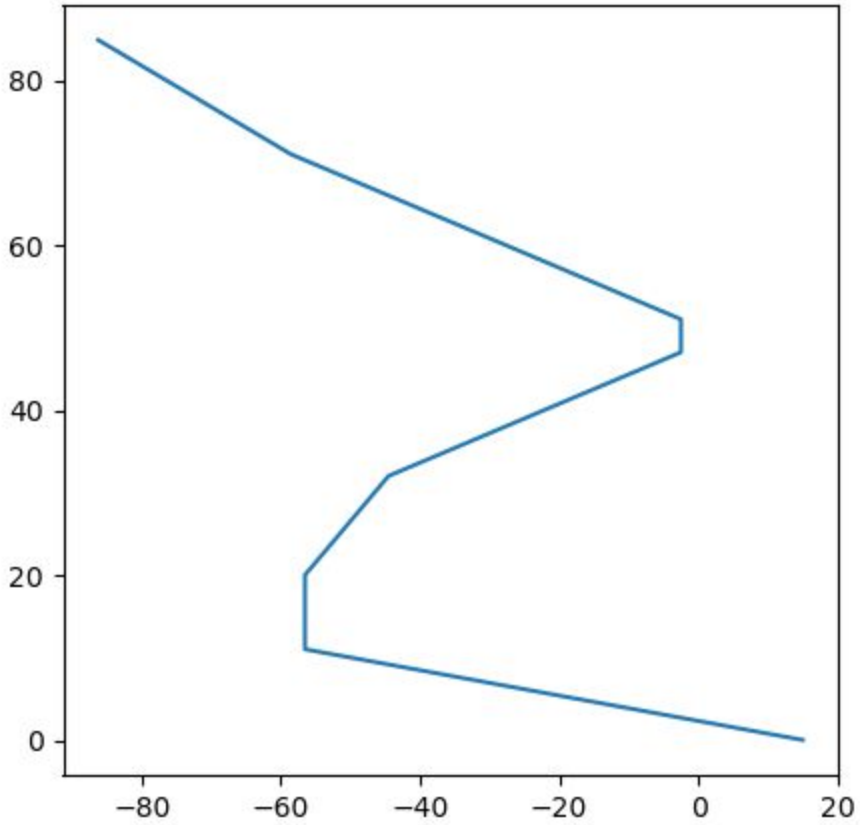
- シェルのpythonやipythonの代わりにウェブブラウザでPythonを実行
  - Python以外にもJulia, Rなども
- Mathematicaのようにセルで入力と出力
- Markdownでメモもできる。
- RISEでスライドにもできる。
- Visual Studio Codeで編集，実行できる。
- モジュール化が進まない。
- 自動保存のためタイムスタンプが更新されてしまう。

Pythonの一習得法

- 最新で自分にとって一番楽そうなPythonをインストールする。
- チュートリアルをやってみる。
  - 文字列フォーマットはよく使うことになる。
  - Introduction to Pythonの部分訳
  - Numpy quickstart
- ドキュメントを参照する。
  - ipythonのドキュメント: import numpy as npなどを自動実行する設定
- 教科書や論文を再現してみる。
  - Scipy Lecture Notes

標準大気

```
import matplotlib.pyplot as plt

fig, ax = plt.subplots(figsize=(5, 5)) # figsizeは必須ではない

T = [15.0, -56.5, -56.5, -44.5, -2.5, -2.5, -58.5, -86.2]

h = [0., 11, 20., 32., 47, 51., 71, 84.852]

ax.plot(T, h)

fig.savefig("standard.png") # ファイルに保存

plt.show() # 表示
```


80
60
40
20
0
−80
−60
−40
−20
0
20

Magics

ECMWFで使われている描画ライブラリ

解析ツールMetviewの描画レイヤ

C++で書かれており，C++，Fortranなどと共にPythonやJupyterから使える。

ecCodesを通してGRIBを扱える。

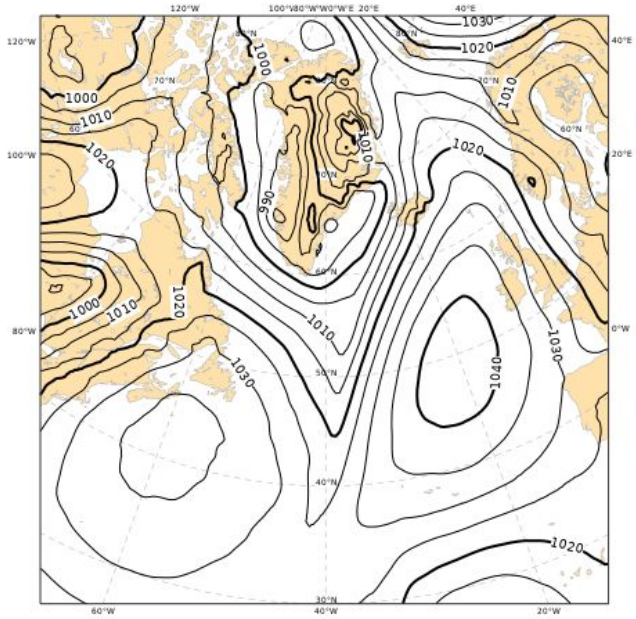
Tutorial

Jupyter notebooks examples

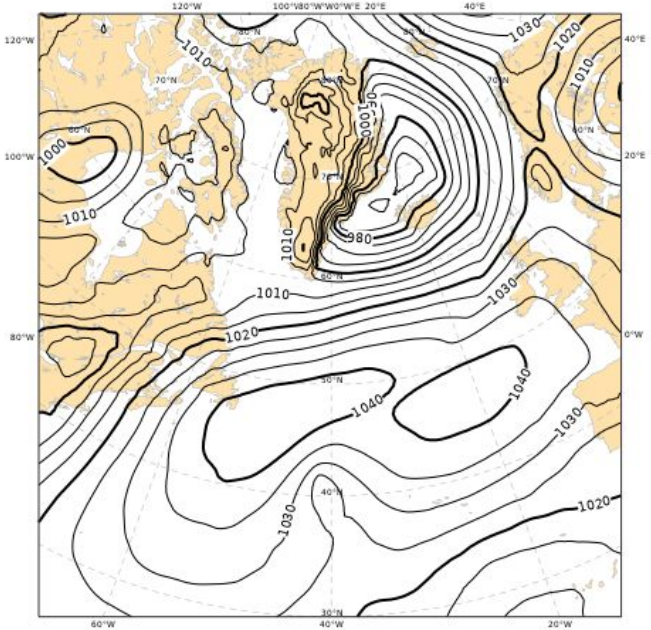
```
import Magics.macro as mg

output = mg.output(
   output_name = "map",
   output_formats = ["png"],
   output_name_first_page_number = "off")

mgmap = mg.mmap(
   subpage_map_area_definition = "corners",
   subpage_lower_left_longitude = -80.0,
   subpage_lower_left_latitude = 0.0,
   subpage_upper_right_longitude = 160.0,
   subpage_upper_right_latitude = 70.0,
   subpage_map_projection = "lambert"
)

coast = mg.mcoast(
   map_coastline_colour = "grey",
   map_coastline_land_shade = "on",
   map_coastline_land_shade_colour =
"cream",
   map_coastline_sea_shade = "on",
   map_coastline_sea_shade_colour =
"white",
   map_label = "off")

mg.plot(output, mgmap, coast)
```

気象庁全球解析値

init 2020122412
valid 2020122412



init 2020122512
valid 2020122512



利用事例

- 日射の年周期（Hartmann 2016）
- スタッガード格子（Durran 2010）
- 分散関係（Arakawa and Konor 2009）
- 台風中心の追跡
- VTKレガシーフォーマット（Rubyの移植）
- ガウス緯度の計算
- RBFを用いた移流モデル（Enomoto, submitted to SOLA），浅水波モデル
- 2Dキャビティ流れ
- 黄金分割探索
- 最適化（Zhang et al. 2000），共軛勾配法（Navon and Legler 1987）
- 衛星放射輝度のリトリーバル標準サンプル（Rogers 2000）
- 最尤法アンサンブルフィルタ（Bowler et al. 2013）

まとめ

- Pythonは親しみやすく，豊富なライブラリが利用できる。
- 気象・地球科学系の可視化・解析ツールのPython化が進行中。
- コードを書くハードルが低いので，気軽に教科書や論文の追試をして理解を深めることに使える。
- 数値解法などでは，ライブラリで実装がどうなっているかに注意が必要なこともある。
- Cythonやnumbaなどで高速化したり，f2pyやpybind11を用いてFortran/C++の資産を利用したりできる。