Introduction to python

数値計算特論(1)

Why python?

First, take a look at “Why python?” in “Scipy Lecture Notes”:

(ぼくも「科学者」であったので、これがしっくりきた。この授業を始めた 2016年頃)

http://www.turbare.net/transl/scipy-lecture-notes/intro/intro.html

original page: http://www.turbare.net/transl/scipy-lecture-notes/intro/intro.html

今はそれに加えて、

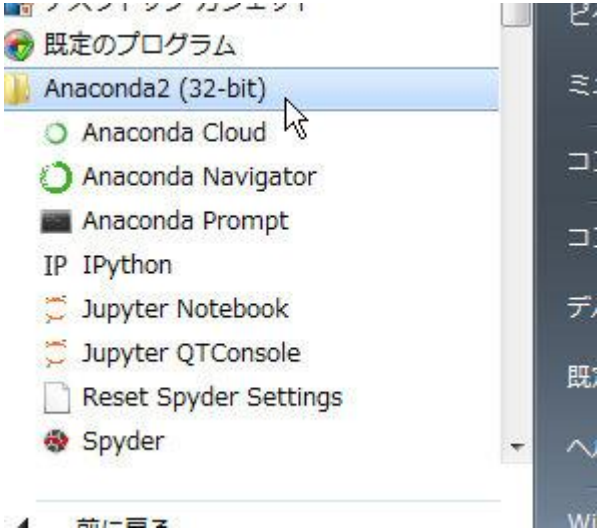
データサイエンス関係ライブラリ利用の最もアクティブな言語環境として急速に普及

My experiences

- FORTRAN, mid-1980s - 2000 for simulations
- Turbo Pascal, Turbo C, mid-1990's for classes
- Perl, mid-1990s - early 2000's for data analysis
- PHP, 2000 - for Web applications (+ MySQL)
- C on UNIX, 2000 - for simulations and classes
- JAVA, early-2000s - for simulations
- JavaScript, late-2000s for Web applications
- **python**, 2016 - for class, data analysis and controlling RasPi
- Others: HTML, shell script, awk, basic, ...

Trends: https://www.tiobe.com/tiobe-index/
https://spectrum.ieee.org/computing/software/the-2017-top-programming-languages
http://redmonk.com/sogrady/2017/06/08/language-rankings-6-17/

pythonの開発・実行環境(ツール)



1. 統合開発環境 **Spyder**
   a. エディタ、コンソール、デバッガなど
2. 対話型コンソール**IPython** + 外部エディタ
   a. **Jupyter QTConsole**も同様
3. ノートブック **Jupyter Notebook**
4. 外部エディタで編集 → python.exeのコマンドライン引数にプログラムを渡す
   （Windowsではファイルとアプリの関連付けができれば、ダブルクリックで実行可）

Spyder

プログラムエディタ

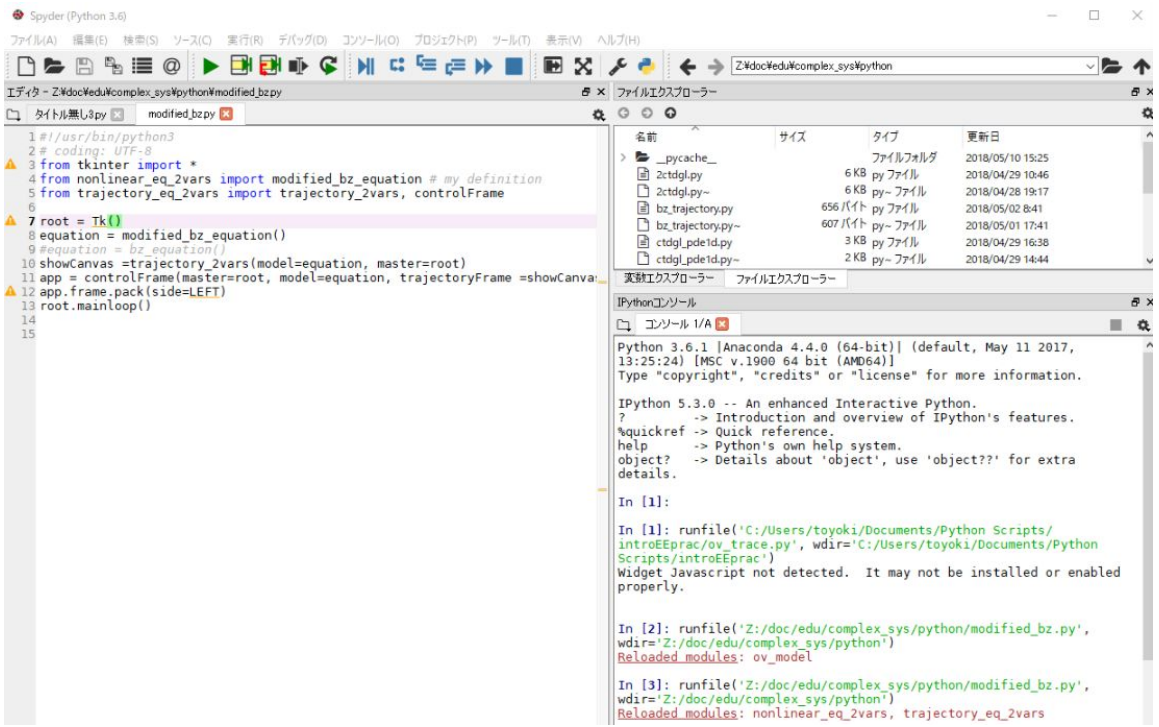
コンソール (IPython)

プロファイラーなどの様々な高機能がある

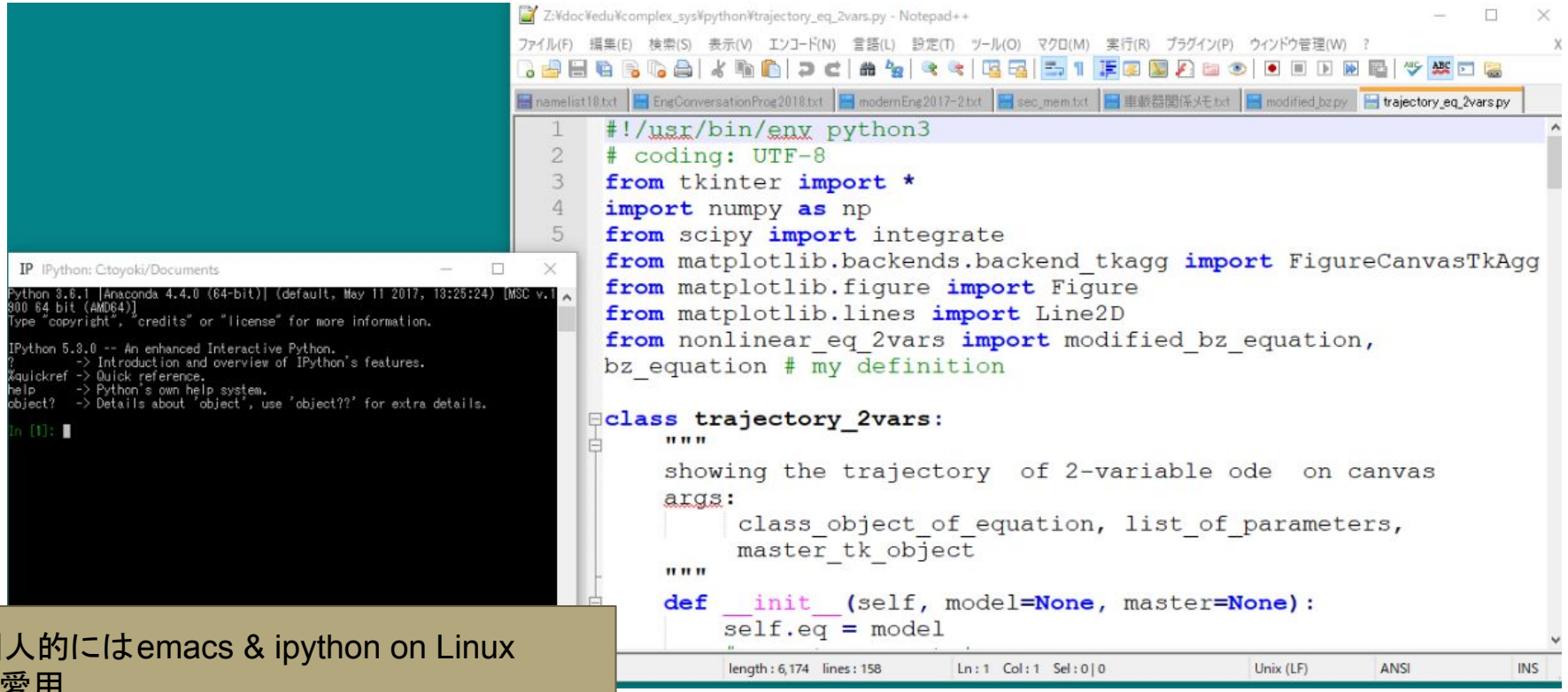
[短所]

立ち上がりが遅い

キーレイアウト（ショットカットキー）のカスタマイズが困難



外部エディタ & IPython: Notepad++の場合



個人的にはemacs & ipython on Linux
を愛用
VisualStudioとの連携も進んでいるようだ

jupyter notebook

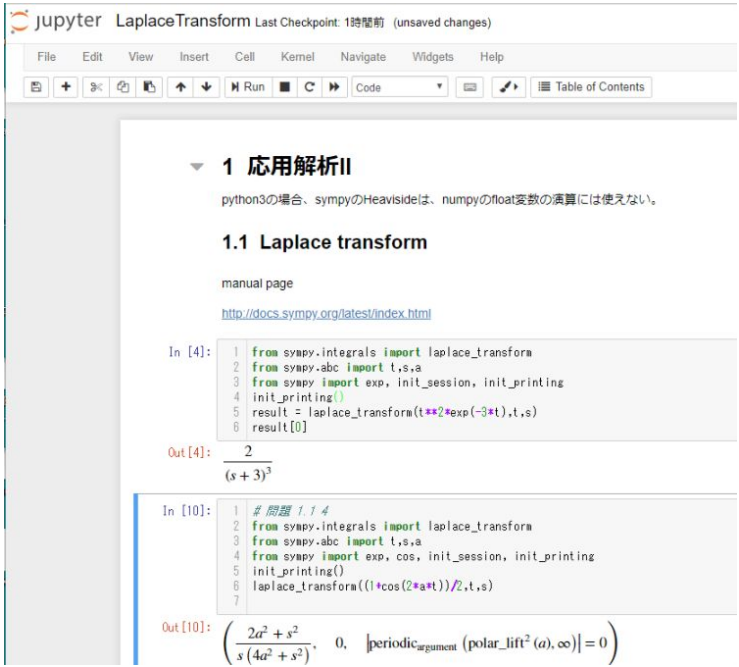
Mathematicaライクなノートとプログラムエディタ、実行環境がWebブラウザの中で動く

さまざまなプラグイン開発がアクティブ

この講義で使用

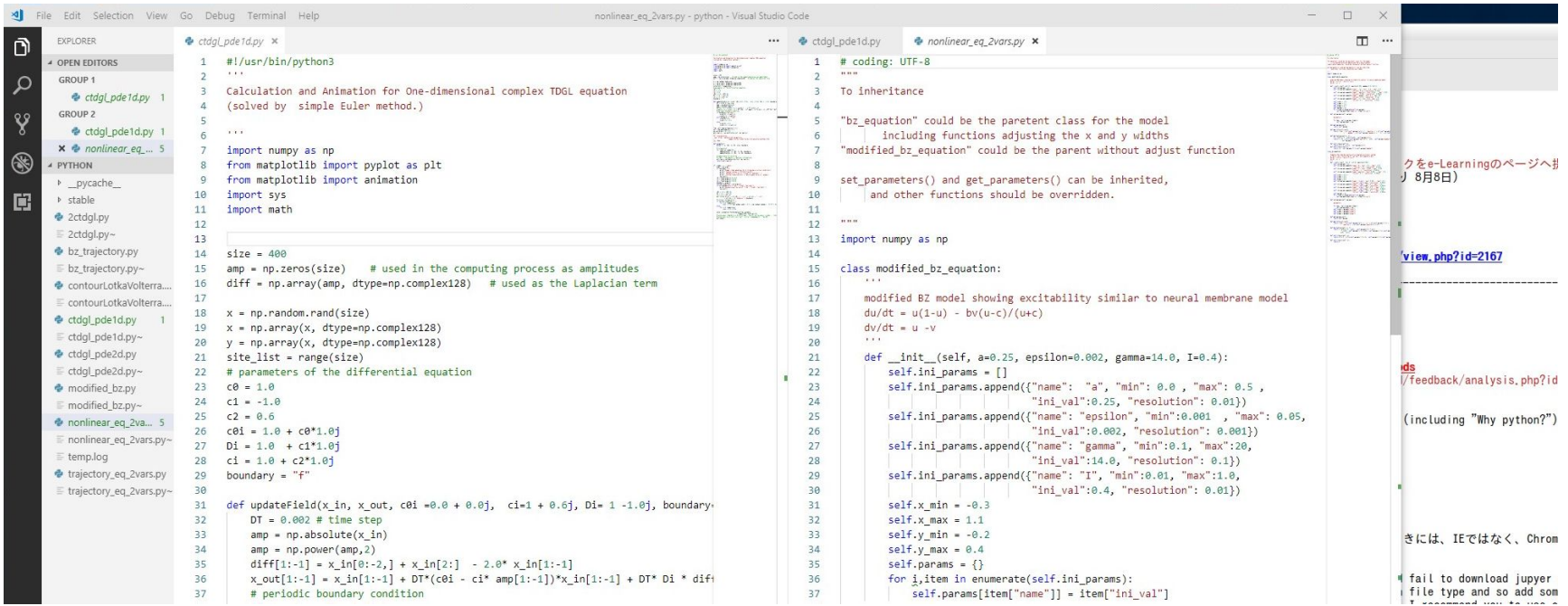
個人的には、
・授業用資料作成
・短いコードの実行 (右図は数式処理利用の例)
・各種コードのメモ用
に利用
簡単なGUIを作りたいときも便利

[欠点]
今のところインラインのアニメーションが作れない



VScodeもお薦め

様々なプログラム開発環境として、人気急上昇中



この授業の教材及び実習について

- 教材はWebページで行う
- スクリプトを含む教材は基本的にはjupyter notebookを用いて作成
  - 演示もjupyter notebookにて示す
- スクリプトの実行により体験的に学ぶことを基本とする
  - 積極的な質問を歓迎する
- 実習（演示プログラムの実行や独自のプログラム作成）はどの環境で行ってもよい
  - 統合環境spyderを立ち上げ、notebookにある個々のスクリプトを spyder内に新しいファイルを作成し、そこにコピー＆ペーストして実行してみるのがよい かもしれない
  - その際、notebookのスクリプト中
    #matplotlib inline
    のような行頭の#で始まる文はnotebook独自のものであるので、除くこと
- python3環境を作って、試してみてほしい。だめなら情報処理教室の端末で（python3とpython2.7それぞれのAnacondaパッケージが入っている）

python文法をざっと

少なくとも1つのプログラム言語を習ったことがあることを想定

ここから後の説明はpython notebookで

1. 変数
    a. 文字列
    b. 配列（リストと辞書）
2. 制御構造 (if, for, while, ...)
3. 関数定義

以下は後で

4. 複数ファイルのコード利用
5. オブジェクト指向