

やさしい  
**Python/pySALEPLot**の  
つかいかた

...

わきた しげる  
+  
すえつぐりょう

# Python

- Pythonとは？
  - 動的プログラミング言語  
(コンパイル不要)
- なぜPython？
  - 可読性・メンテナンス性に優れている
  - すぐに書ける
  - Numpyなどの  
module(package)が抱負



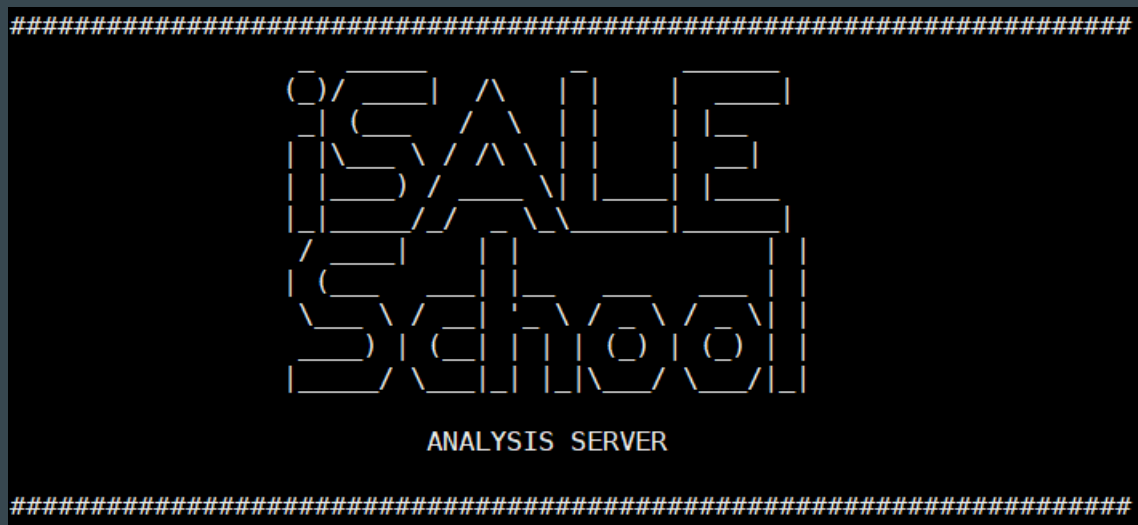
# Python

- Numpy
  - 配列や行列の基本タイプ、  
それらの数値計算のためのmodule
- Matplotlib
  - グラフ描画のためのmodule
- iPython (後半)
  - 対話型インターフェイスのシェル
    - タブ補完が便利
    - (`ipython --matplotlib` での起動も便利)

# 目標

- pySALEPlotを用いた描画
  - 前半
    - Sampleのpythonスクリプトを変更する
      - 出力したい変数に変える
  - 後半
    - pySALEPlot + python の仕組みを理解

- 今回は解析サーバでの作業なので以下でログイン
  - `ssh -Y isale??@an00.cfca.nao.ac.jp`
  - `ssh -Y isale??@an01.cfca.nao.ac.jp`
  - `ssh -Y isale??@an.cfca.nao.ac.jp` ←上のどちらにも接続できない場合



無事、ログインできると、この素晴らしいロゴがでてる

- ログイン後、以下を入力
  - `module load anaconda/2 intel`

# pySALEPlot: DenTmp.py

1. ログイン後、iSALE-Dellenまで移動

```
cd iSALE-Dellen
```

2. plot.pyをコピーする

```
cp plot.py DenTmp.py
```

3. DenTmp.py を編集する

```
emacs DenTmp.py -nw
```

もしくは

```
vi DenTmp.py
```

# pySALEPlot: DenTmp.py

4. DenTmp.pyの読み込むデータの種類を変更 (51行目)

```
step=model.readStep(['Tmp', 'Den'],i)
.....
```

5. 出力時間を短くする (39行目)

```
for i in np.arange(0,20,10):
    ....
```

6. 出力ファイル名変更 (82行目)

```
fig.savefig('{}'/DenTmp-{:05d}.eps'.format(dirname, i),
.....
format='eps', dpi=300)
```

7. 編集を終了してDenTmp.pyを実行  
python DenTmp.py

保存して終了のやり方

emacsの場合 ctrl+x, ctrl+sの後

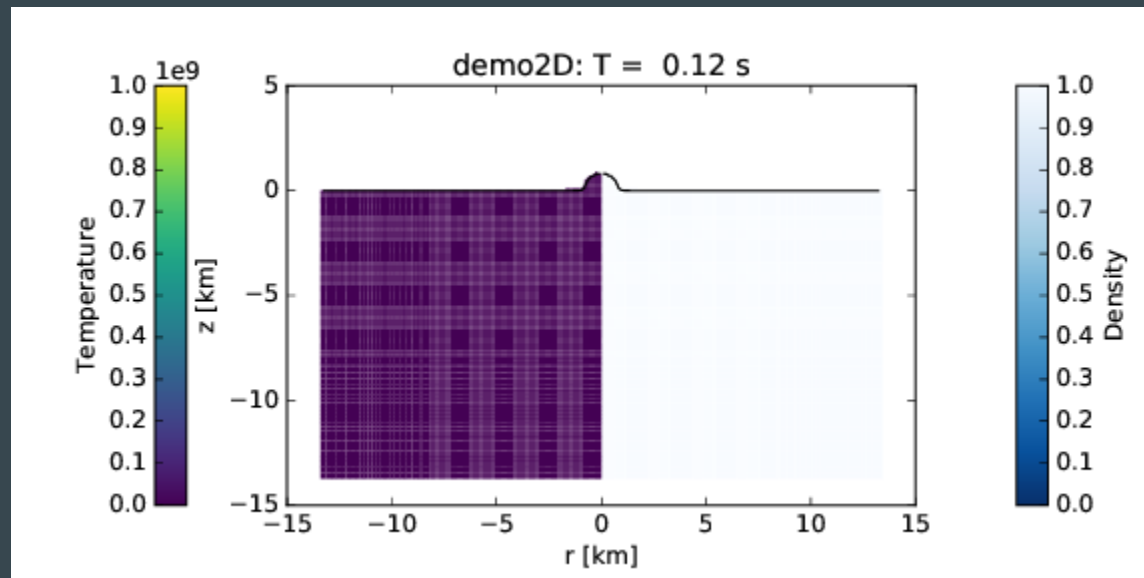
ctrl+x, ctrl+c

viの場合 Esc押した後 :wq

## 8. 画像を確認

display Plots/DenTmp-00010.eps

- plotされる値の範囲がおかしい





9. DenTmp.pyを編集しplotする値の範囲を変更(55-58行目)

```
p1=ax.pcolormesh(model.x, model.y, step.data[1],  
cmap='Blues_r', vmin=1700., vmax=2700.)
```

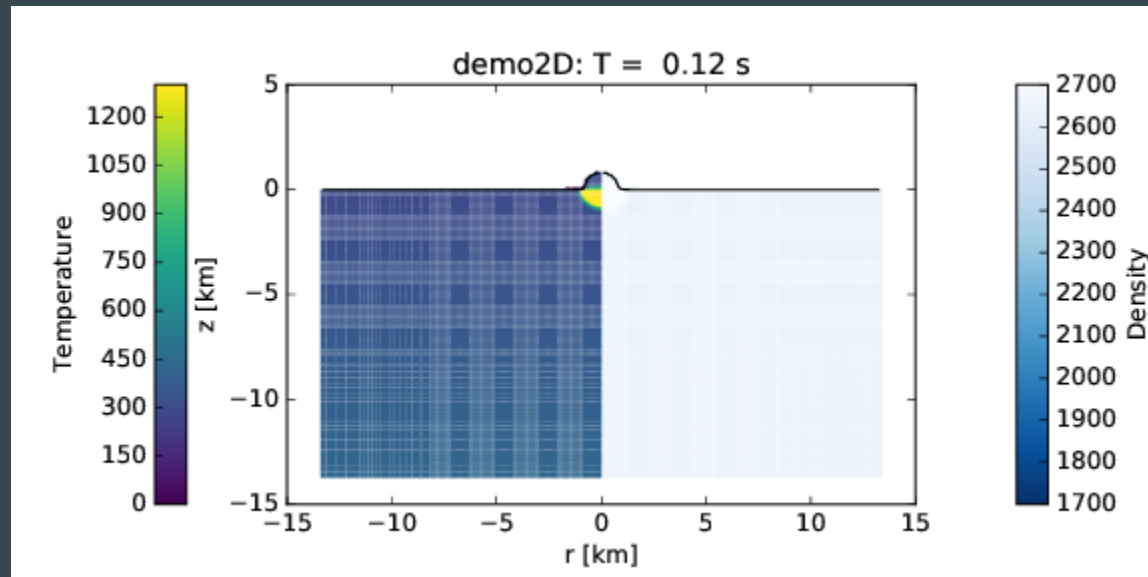
```
p2=ax.pcolormesh(-model.x, model.y, step.data[0],  
vmin=0., vmax=1300.)
```

10. 編集を終了してDenTmp.pyを実行

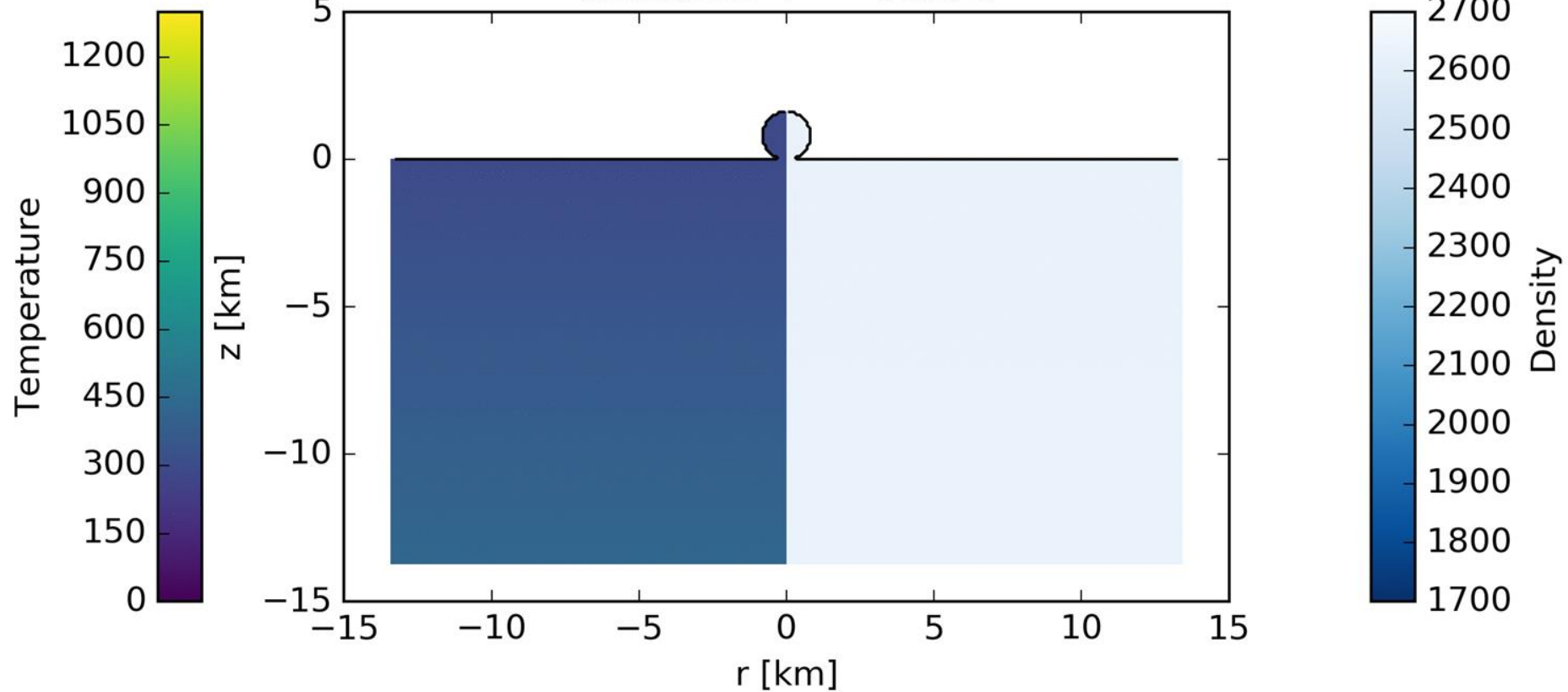
```
python DenTmp.py
```

11. 画像を確認

```
display Plots/DenTmp-00010.eps
```



demo2D: T = 0.00 s



## pySALEPlot: DenTmp.py

12. DenTmp.pyを編集し出力時間を元に戻す(39行目)

```
for i in np.arange(0,210,10):  
    ...
```

13. 実行して連番ファイルを作成

```
python DenTmp.py
```

# pySALEPlot: DenTmp.py

## 14. 動画にする

`convert Plots/DenTmp*.eps DenTmp.gif`

もしくは

```
ffmpeg -y -r 10 -i ./DenTmp/DenTmp-%04d0.png -s 800x400  
DenTmp.mp4
```

ただし、epsではなくpngで出力する必要がある

```
mpl.use('Agg') #for png
```

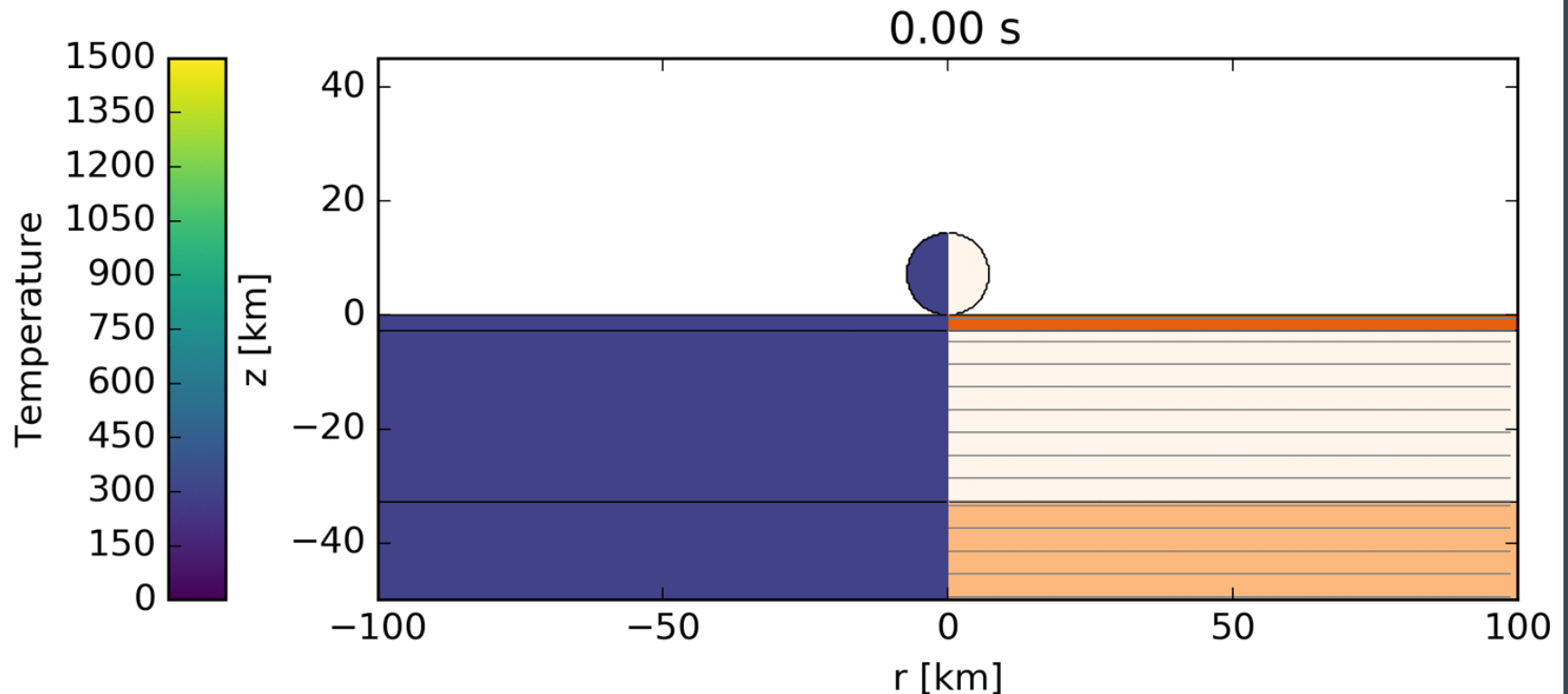
```
fig.savefig('{} /DenTmp-{:05d}.png'.format(dirname, i),  
format='png', dpi=300)
```

## 15. 動画の実行

`animate DenTmp.gif`

# 前半のまとめ

- PythonやpySALEPlotってかんたんそう
- iSALEで計算したデータを簡単にpySALEPlotでプロットができそう



# 後半

- pySALEPlotのスク립トの中身を簡単に理解する
- 解析サーバにログイン
  - `ssh -Y isale??@an00.cfca.nao.ac.jp`
  - `ssh -Y isale??@an01.cfca.nao.ac.jp`
  - `ssh -Y isale??@an.cfca.nao.ac.jp` ←どちらも接続できない場合
- ログイン後、以下を入力
  - `module load anaconda/2 intel`

- iPython 起動
  - pySALEPlot.py がある場所に移動してから
    - cd iSALE-Dellen
    - **ipython --matplotlib**

- PythonとiPythonでversionを確認しよう

```
[wakitasg(33715)@an08:~/161109isale]$python --version
Python 2.7.11 :: Anaconda 2.5.0 (64-bit)
[wakitasg(33716)@an08:~/161109isale]$cat version.py
import sys
print sys.version
[wakitasg(33717)@an08:~/161109isale]$python version.py
2.7.11 |Anaconda 2.5.0 (64-bit)| (default, Dec 6 2015, 18:08:32)
[GCC 4.4.7 20120313 (Red Hat 4.4.7-1)]
[wakitasg(33718)@an08:~/161109isale]$ipython --version
4.0.3
[wakitasg(33719)@an08:~/161109isale]$ipython
Python 2.7.11 |Anaconda 2.5.0 (64-bit)| (default, Dec 6 2015, 18:08:32)
Type "copyright", "credits" or "license" for more information.

IPython 4.0.3 -- An enhanced Interactive Python.
?          -> Introduction and overview of IPython's features.
%quickref  -> Quick reference.
help       -> Python's own help system.
object?    -> Details about 'object', use 'object??' for extra details.

In [1]: import sys

In [2]: print sys.version
2.7.11 |Anaconda 2.5.0 (64-bit)| (default, Dec 6 2015, 18:08:32)
[GCC 4.4.7 20120313 (Red Hat 4.4.7-1)]

In [3]: █
```



- `import ***` で、必要なmoduleを読み込む  
`import sys`

```
[wakitasg(33715)@an08:~/161109isale]$python --version
Python 2.7.11 :: Anaconda 2.5.0 (64-bit)
[wakitasg(33716)@an08:~/161109isale]$cat version.py
import sys
print sys.version
[wakitasg(33717)@an08:~/161109isale]$python version.py
2.7.11 |Anaconda 2.5.0 (64-bit)| (default, Dec 6 2015, 18:08:32)
[GCC 4.4.7 20120313 (Red Hat 4.4.7-1)]
[wakitasg(33718)@an08:~/161109isale]$ipython --version
4.0.3
[wakitasg(33719)@an08:~/161109isale]$ipython
Python 2.7.11 |Anaconda 2.5.0 (64-bit)| (default, Dec 6 2015, 18:08:32)
Type "copyright", "credits" or "license" for more information.

IPython 4.0.3 -- An enhanced Interactive Python.
?          -> Introduction and overview of IPython's features.
%quickref  -> Quick reference.
help       -> Python's own help system.
object?    -> Details about 'object', use 'object??' for extra details.

In [1]: import sys

In [2]: print sys.version
2.7.11 |Anaconda 2.5.0 (64-bit)| (default, Dec 6 2015, 18:08:32)
[GCC 4.4.7 20120313 (Red Hat 4.4.7-1)]

In [3]:
```

- `print ***` で、変数を出力する  
`print sys.version`

```
[wakitasg(33715)@an08:~/161109isale]$python --version
Python 2.7.11 :: Anaconda 2.5.0 (64-bit)
[wakitasg(33716)@an08:~/161109isale]$cat version.py
import sys
print sys.version
[wakitasg(33717)@an08:~/161109isale]$python version.py
2.7.11 |Anaconda 2.5.0 (64-bit)| (default, Dec 6 2015, 18:08:32)
[GCC 4.4.7 20120313 (Red Hat 4.4.7-1)]
[wakitasg(33718)@an08:~/161109isale]$ipython --version
4.0.3
[wakitasg(33719)@an08:~/161109isale]$ipython
Python 2.7.11 |Anaconda 2.5.0 (64-bit)| (default, Dec 6 2015, 18:08:32)
Type "copyright", "credits" or "license" for more information.

IPython 4.0.3 -- An enhanced Interactive Python.
?                -> Introduction and overview of IPython's features.
%quickref        -> Quick reference.
help             -> Python's own help system.
object?         -> Details about 'object', use 'object??' for extra details.

In [1]: import sys

In [2]: print sys.version
2.7.11 |Anaconda 2.5.0 (64-bit)| (default, Dec 6 2015, 18:08:32)
[GCC 4.4.7 20120313 (Red Hat 4.4.7-1)]

In [3]:
```

# pySALEPlot 1/5

- iSALEのデータ('jdata.dat')をプロットするためのツール
  - Python の module
- `import pySALEPlot as psp`
  - `import *** as ###` とすることで、以降は `###` で `***` を呼び出せる
- `model = psp.opendatfile('demo2D/jdata.dat')`
  - `model` という変数に、`jdata.dat`の中身を入れる

# pySALEPlot 2/5

- ファイルを読み込んでみよう
  - `import pySALEPlot as psp`
  - `model = psp.opendatfile('demo2D/jdata.dat')`

```
In [7]: import pySALEPlot as psp
```

```
In [8]: model=psp.opendatfile('jdata.dat')  
Opened iSALE data file 'jdata.dat', with 121 time steps
```

```
In [9]: step=model.タブ[tab]
```

<code>model.alemode</code>	<code>model.filepos</code>	<code>model.nmproj</code>	<code>model.objrad</code>	<code>model.skipStep</code>	<code>model.xext</code>
<code>model.cellVolumes</code>	<code>model.fmax</code>	<code>model.nmtarx</code>	<code>model.objvel</code>	<code>model.skipToInit</code>	<code>model.x hires</code>
<code>model.cinfo</code>	<code>model.fmin</code>	<code>model.nmtary</code>	<code>model.path</code>	<code>model.skipToStep</code>	<code>model.xx</code>
<code>model.closeFile</code>	<code>model.geometry</code>	<code>model.nmtarz</code>	<code>model.plotSetupInfo</code>	<code>model.surface</code>	<code>model.y</code>
<code>model.cppr</code>	<code>model.gravityAnomaly</code>	<code>model.nplots</code>	<code>model.plottype</code>	<code>model.surfaceProfile</code>	<u><code>model.yc</code></u>
<code>model.craterGrowth</code>	<code>model.inputDict</code>	<code>model.nsteps</code>	<code>model.position</code>	<code>model.tracerInfo</code>	<code>model.yext</code>
<code>model.dx</code>	<code>model.inputParams</code>	<code>model.nvar</code>	<code>model.quickPlot</code>	<code>model.tracer_num</code>	<code>model.y hires</code>
<code>model.dy</code>	<code>model.itracer</code>	<code>model.nx</code>	<u><code>model.readStep</code></u>	<code>model.tracer_numu</code>	<code>model.yy</code>
<code>model.fieldlist</code>	<code>model.laststep</code>	<code>model.nxn</code>	<u><code>model.readvariable</code></u>	<code>model.tru</code>	
<code>model.fieldmax</code>	<code>model.laynum</code>	<code>model.nxp</code>	<code>model.refden</code>	<code>model.units</code>	
<code>model.fieldmin</code>	<code>model.magic</code>	<code>model.ny</code>	<code>model.scale</code>	<code>model.verbose</code>	
<code>model.fileid</code>	<code>model.modelInfo</code>	<code>model.nyp</code>	<code>model.scalelabel</code>	<u><code>model.x</code></u>	
<code>model.filename</code>	<code>model.nmat</code>	<code>model.objnum</code>	<code>model.setScale</code>	<u><code>model.xc</code></u>	



# pySALEPlot 3/5

- 中身を少しだけみてみよう
  - `step=model.readStep()`
  - `print step.Den`

```
In [9]: step=model.readStep()
Read in ['Den'] for timestep -1 (0.000e+00 s)
```

```
In [10]: print step
<pySALEPlot.modelStep instance at 0x7f58ef5223f8>
```

```
In [12]: print step.タブ[tab]
```

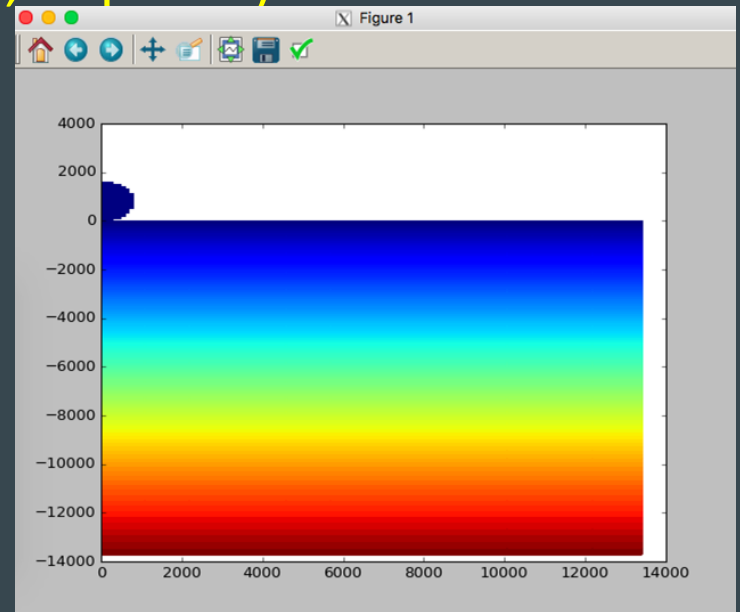
step.Den	step.cyc	step.eof	step.filepos	step.mat	step.stepInfo	step.xmark
step.cmc	step.data	step.fileid	step.findTracer	step.plottype	step.time	step.ymark

```
In [12]: print step.Den
[[3461.76611328125 3456.147216796875 3450.804443359375 ..., -- -- --]
 [3461.76611328125 3456.147216796875 3450.804443359375 ..., -- -- --]
 [3461.76611328125 3456.147216796875 3450.804443359375 ..., -- -- --]
 ...,
 [3461.76611328125 3456.147216796875 3450.804443359375 ..., -- -- --]
 [3461.76611328125 3456.147216796875 3450.804443359375 ..., -- -- --]
 [3461.76611328125 3456.147216796875 3450.804443359375 ..., -- -- --]]
```

# pySALEPlot 4/5

- せっかくだからプロット
  - `plt.pcolormesh(model.x,model.y,step.Den)`
  - `plt.show()`

```
In [19]: plt.pcolormesh(model.x,model.y,step.Den)
Out[19]: <matplotlib.collections.QuadMesh at 0x7f58eb9075d0>
```



## ■ plt とは？

- matplotlibのこと
- pySALEPlot.py の最初の方で呼び出している
  - `import matplotlib.pyplot as plt`

ipythonの終了 `exit()`

# pySALEPlot 5/5

- 実際にプロットするときには、次を適当に組み合わせる
  - 時間
    - 動画を作るなら、繰り返し文(**for**)が必要
  - 変数
    - 密度[**Den**]、圧力[**Pre**]、温度[**Tmp**]、等々
  - プロットの種類
    - コンター (**pcolormesh**)
    - ライン (**plot**)
    - プロットの見栄えを整えるのは、一番最後
      - ラベル、カラーなど