やさしい Python/pySALEPLotの つかいかた …

わきた しげる

Python 1/5

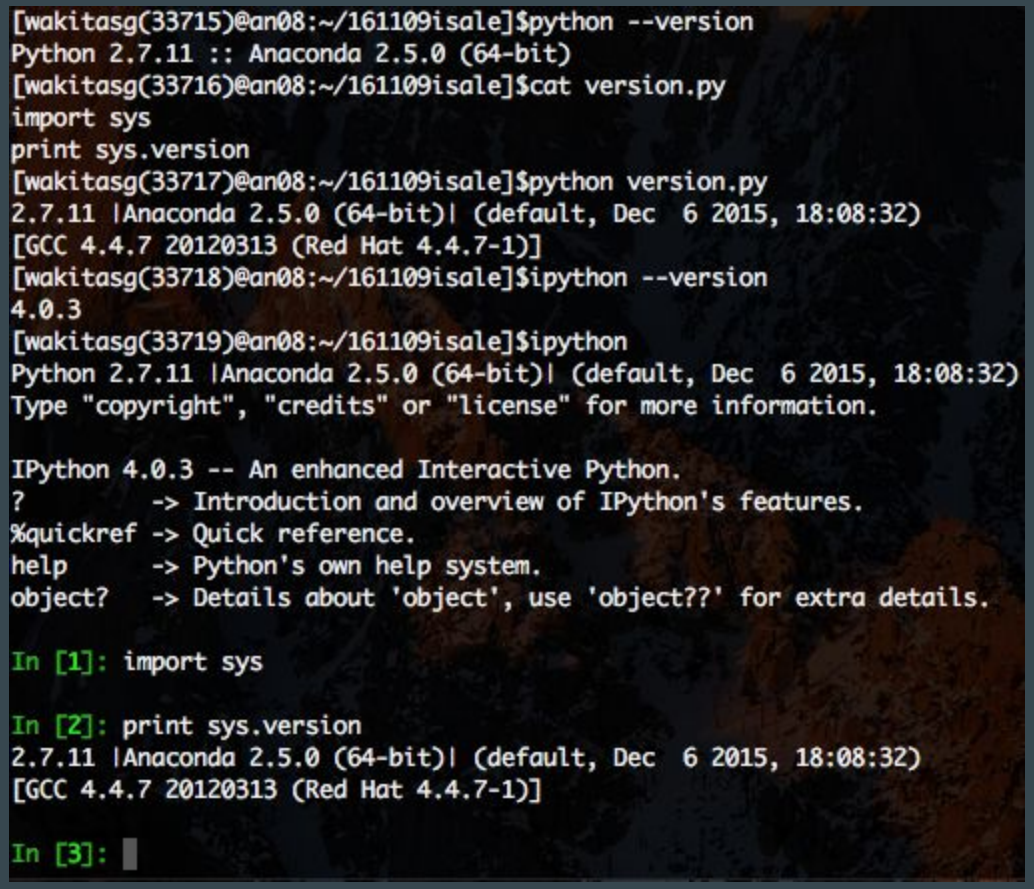
- Pythonとは？
  - 動的プログラミング言語（コンパイル不要）
- なぜPython？
  - 可読性・メンテナンス性に優れてる
  - すぐに書ける
  - Numpyなどのmodule(package)が抱負

Python 2/5

- 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
  - 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）

Python 3/5

- PythonとiPythonでversionを確認しよう



Python 4/5

- import *** で、必要なmoduleを読み込む

```
[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]:
```

Python 5/5

- print *** で、変数を出力する

```
[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.opendatafile(‘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
  - Jdata.datがある場所に移動して、ipythonを起動
    - 例えば、share/examples/demo2D
  - import pySALEPlot as psp
  - model = psp.opendatfile('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]
model.alemode          model.filepos          model.nmproj           model.objrad           model.skipStep         model.xext
model.cellVolumes      model.fmax             model.nmtarx           model.objvel           model.skipToInit       model.xhires
model.cinfo            model.fmin             model.nmtary           model.path             model.skipToStep       model.xx
model.closeFile        model.geometry         model.nmtarz           model.plotSetupInfo    model.surface          model.y
model.cppr             model.gravityAnomaly   model.nplots           model.plottype         model.surfaceProfile   model.yc
model.craterGrowth     model.inputDict        model.nsteps           model.position         model.tracerInfo       model.yext
model.dx               model.inputParams      model.nvar             model.quickPlot        model.tracer_num       model.yhires
model.dy               model.itracers         model.nx               model.readStep         model.tracer_numu      model.yy
model.fieldlist        model.laststep         model.nxny             model.readvariable     model.tru
model.fieldmax         model.laynum           model.nxp              model.refden           model.units
model.fieldmin         model.magic            model.ny               model.scale            model.verbose
model.fileid           model.modelInfo        model.nyp              model.scalelabel       model.x
model.filename         model.nmat             model.objnum           model.setScale         model.xc
```

pySALEPlot 3/5

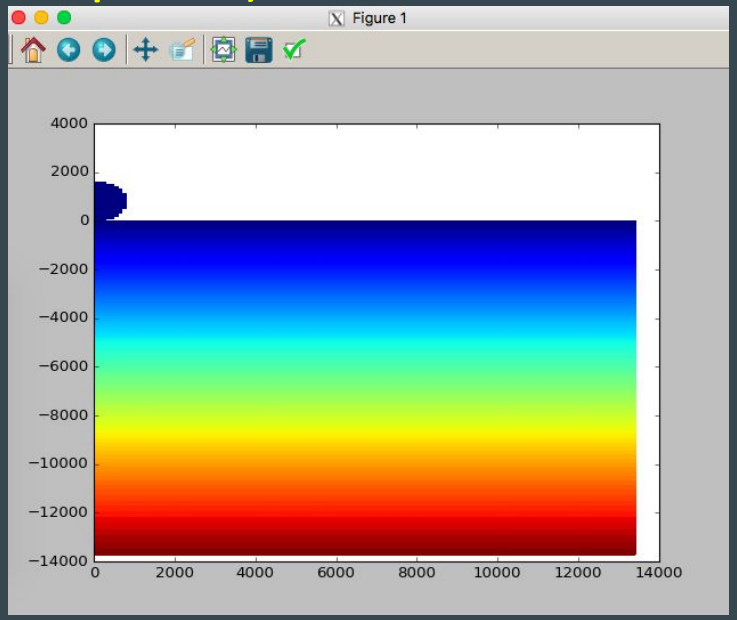
- 中身を少しだけみてみよう
  - step=model.readStep()
  - print step.Den

```
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>

In [20]: plt.show()
```



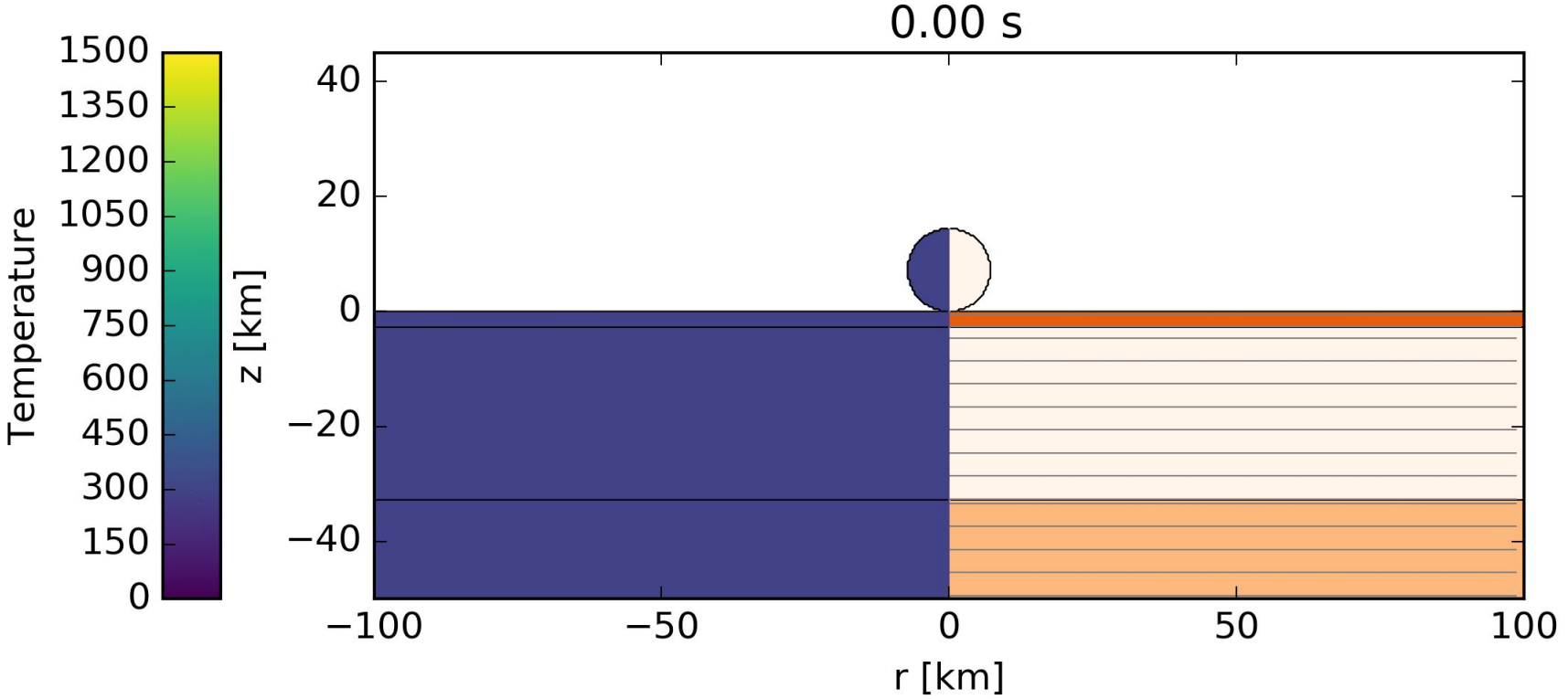
  - plt とは？
    - matplotlibのこと
    - pySALEPlot.py の最初の方で呼び出している
      - import matplotlib.pyplot as plt

pySALEPlot 5/5

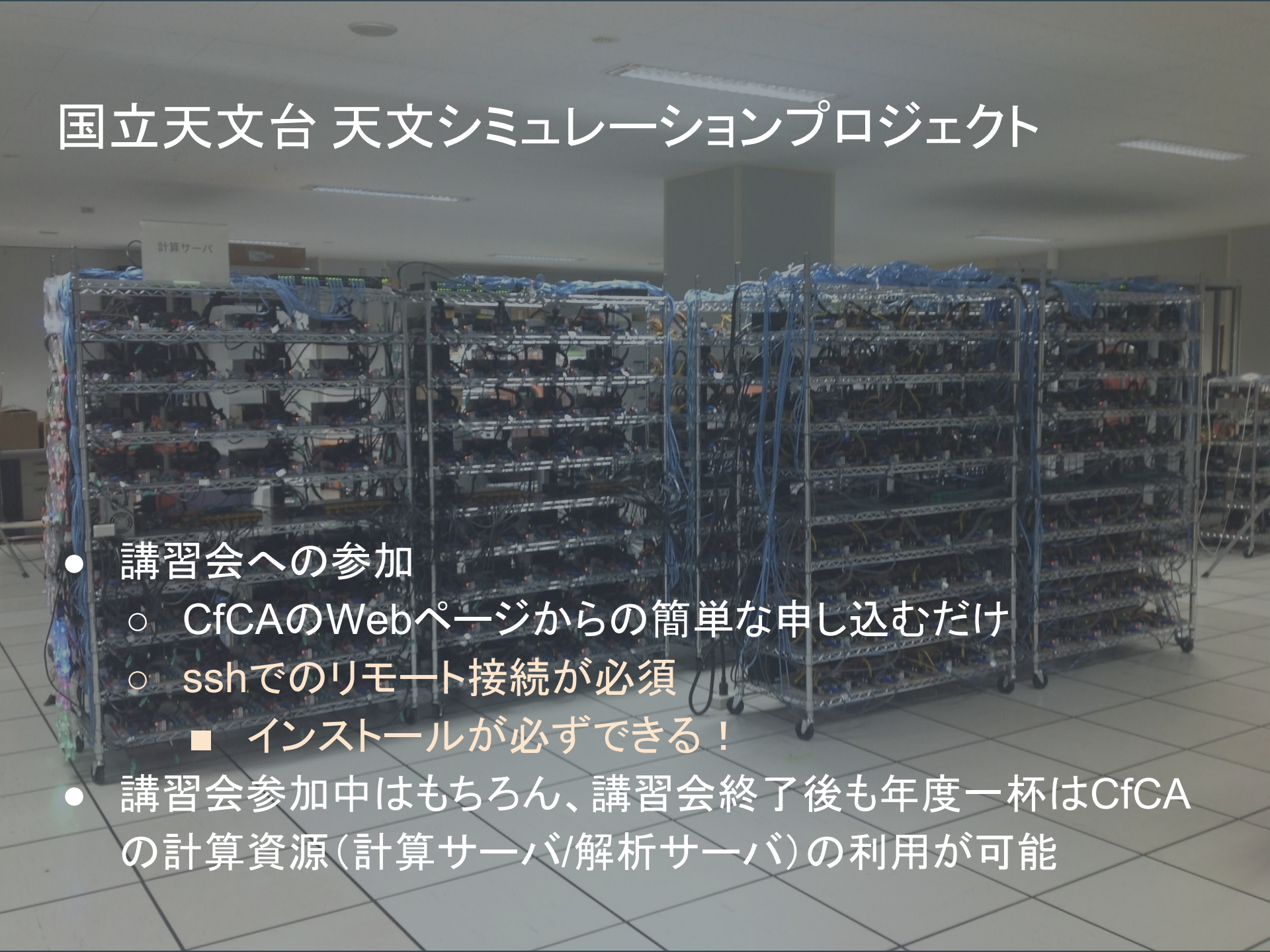
- 実際にプロットするときには、次を適当に組み合わせる
  - 時間
    - 動画を作るなら、繰り返し文(for)が必要
  - 変数
    - 密度[Den]、圧力[Pre]、温度[Tmp]、等々
  - プロットの種類
    - コンター（pcolormesh）
    - ライン（plot）
    - プロットの見栄えを整えるのは、一番最後
      - ラベル、カラーなど

ここまでのまとめ

- 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



国立天文台 天文シミュレーションプロジェクト



- 講習会への参加
  - CfCAのWebページからの簡単な申し込むだけ
  - sshでのリモート接続が必須
    - インストールが必ずできる！
- 講習会参加中はもちろん、講習会終了後も年度一杯はCfCAの計算資源（計算サーバ/解析サーバ）の利用が可能

pySALEPLot 実践編

...

脇田茂（国立天文台 天文シミュレーションプロジェクト）

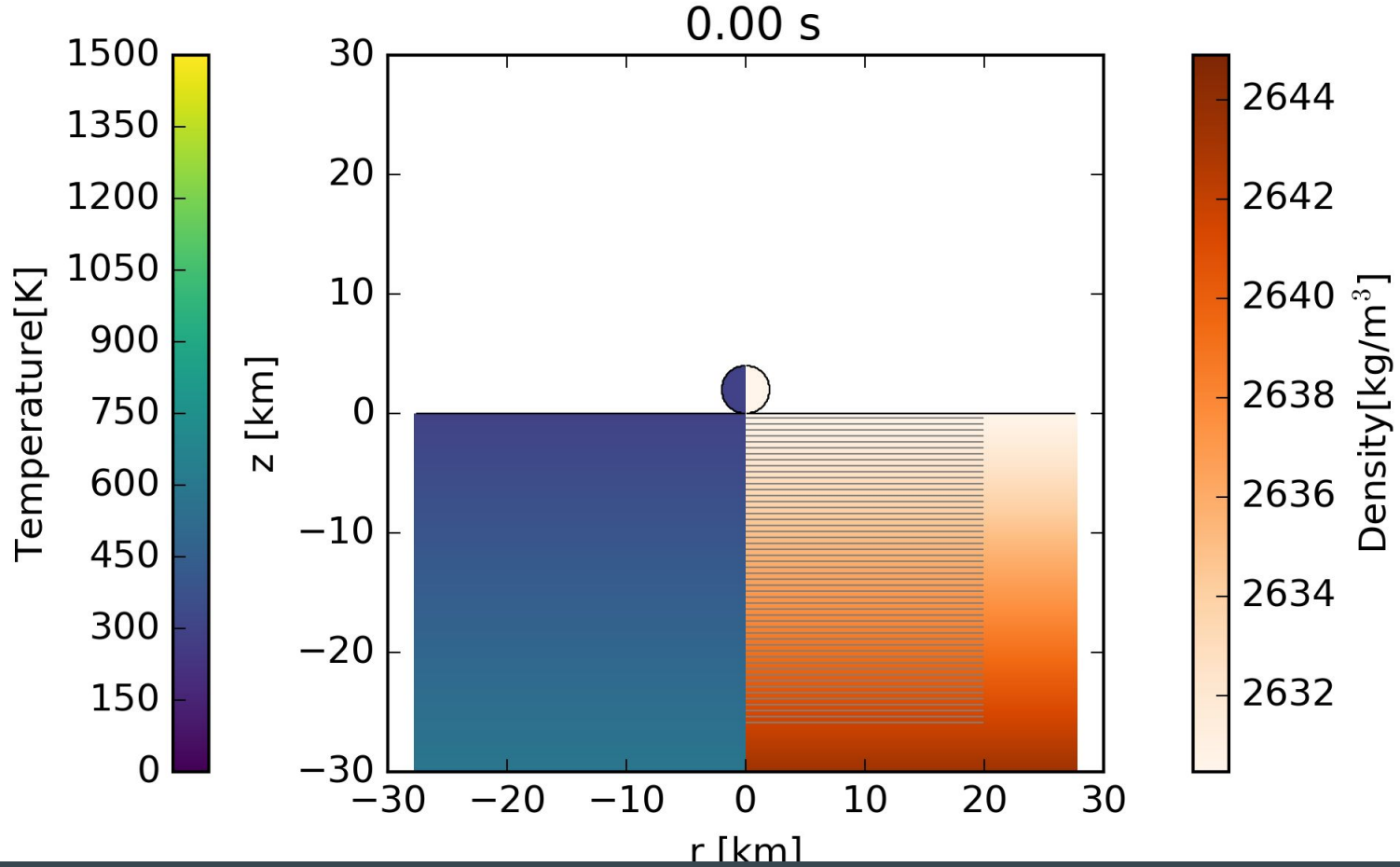
pySALEPlot 実践編

- [iSALE]/share/examples
  - Chicxulub
    - Plotting/MatTmp.py
    - profile_example.py
      - [iSALE]/share/pysaleplot/profile_example.py

pySALEPlot: MatTmp.py → DenTmp.py

- [iSALE]/share/examples
  - Chicxulub
    - Plotting/MatTmp.py

1. 読み込むデータの種類
   a. step=model.readStep(['Tmp', 'Den'],i)
2. プロットする変数
   a. p1=ax.pcolormesh(model.x,model.y,step.Den, cmap=””)
3. 出力先ディレクトリ名変更
   a. dirname=’DenTmp’


0.00 s
Temperature[K]
1500
1350
1200
1050
900
750
600
450
300
150
0
z [km]
30
20
10
0
−10
−20
−30
−30
−20
−10
0
10
20
30
r [km]
Density[kg/m^3]
2644
2642
2640
2638
2636
2634
2632

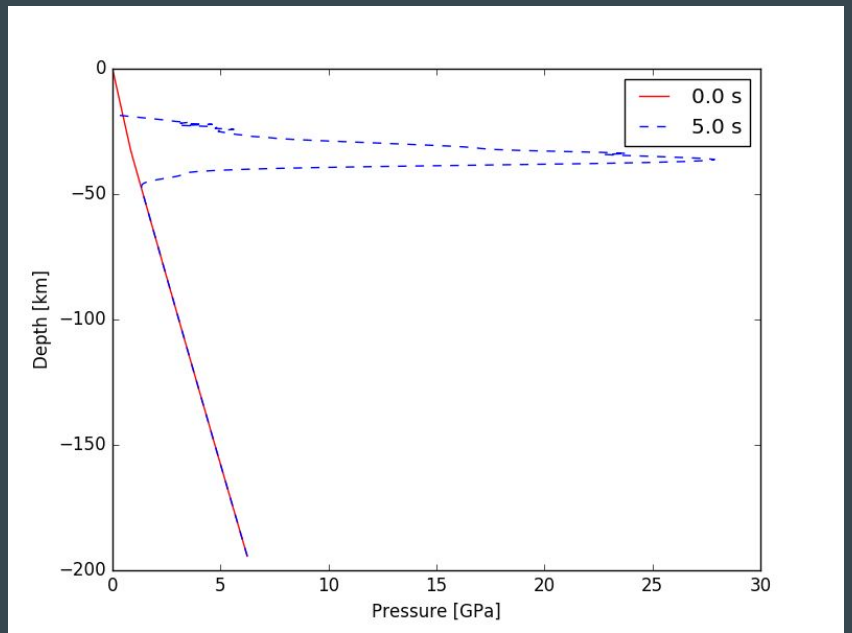
pySALEPlot: MatTmp.py → DenTmp.py

- [iSALE]/share/examples
  - Chicxulub
    - Plotting/MatTmp.py

4. カラーバー追加
   a. share/examples/Collision2D/Plotting/PrePor.py にある
      def make_colorbars(ax, p, f, units): 以下をコピー
   b. 次の文を置き換え
      if i == 0: make_colorbar(ax,p2,step.plottype[0])
      if i == 0:
          make_colorbars(ax, [p1, p2], [step.plottype[1], step.plottype[0]], [r'[kg/m^3]', '[K]'])

pySALEPlot: profile_example.py

- [iSALE]/share/examples
  - Chicxulub
    - profile_example.py
      - [iSALE]/share/pysaleplot/profile_example.py
      - x = 0.2 での 深さ方向(y 方向)の測線に沿った圧力プロファイル



pySALEPlot: profile_example.py → profile.py

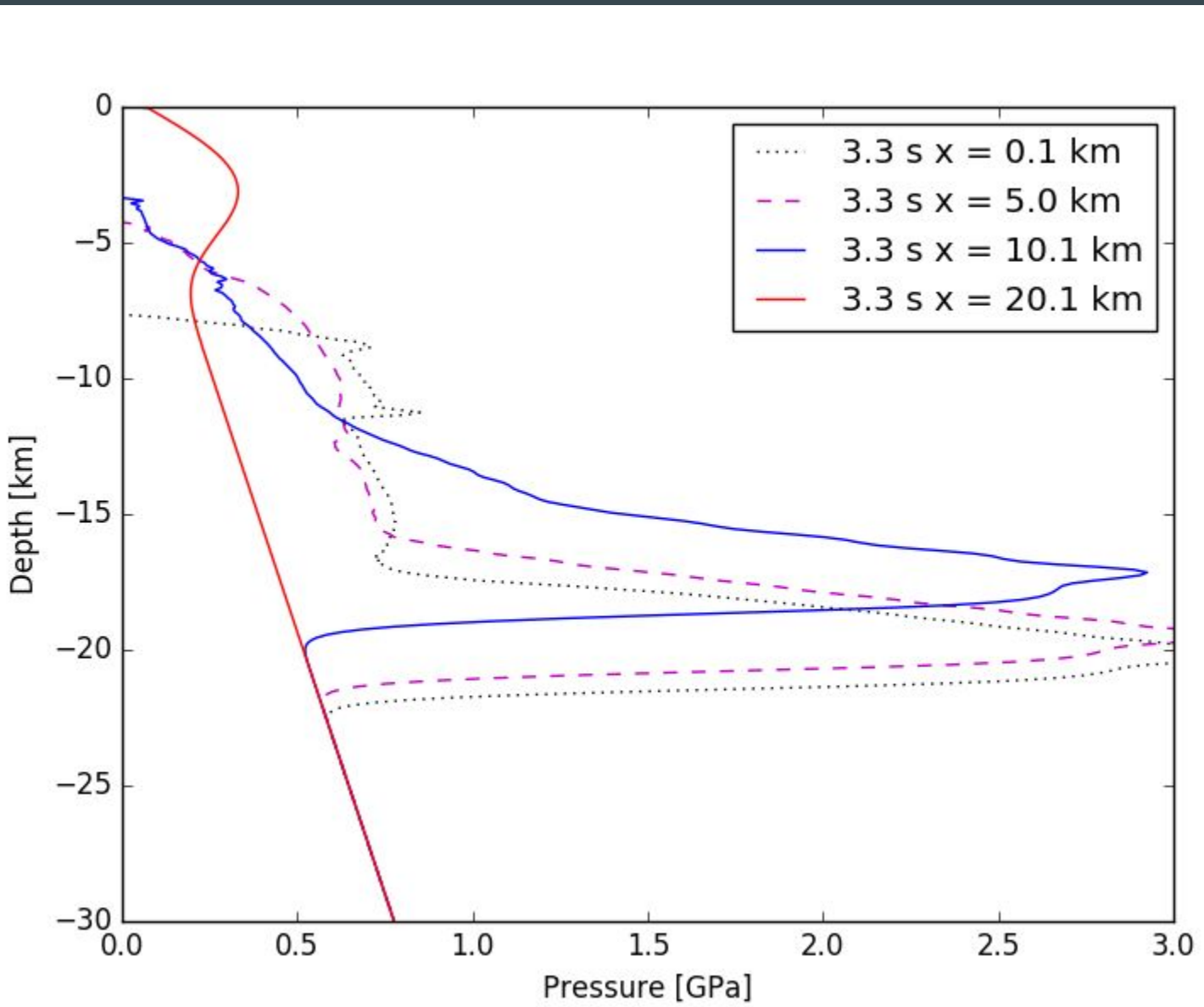
- [iSALE]/share/examples
  - Chicxulub
    - profile_example.py

1. プロットするxの場所の変更
   a. ax.plot(step0.Pre[0,:]*1e-9,model.yc[0,:])
      i. step0.Pre と model.yc を理解する
      ii. len(step1.Pre[0,:]) と len(model.yc[0,:]) を確認
      len(step1.Pre[:,0]) と len(model.yc[:,0]) を確認
      ※ len(???) で ??? のサイズを調べる

pySALEPlot: profile_example.py → profile.py

- [iSALE]/share/examples
  - Chicxulub
    - profile_example.py

1. プロットする x の場所の変更
   a. ax.plot(step0.Pre[0,:]*1e-9,model.yc[0,:])
      i. step0.Pre と model.yc を理解する
      ii. len(step1.Pre[0,:]) と len(model.yc[0,:]) は 360
          len(step1.Pre[:,0]) と len(model.yc[:,0]) は 240
      iii. asteroid.inp の GRIDV/GRIDH (の合計) に対応
      iv. step0.Pre[0,:] は model.xc[0,:] の場所でのPreに対応
      v. model.xc[0,:] は 全て 0.05 となっている


3.3 s x = 0.1 km
3.3 s x = 5.0 km
3.3 s x = 10.1 km
3.3 s x = 20.1 km
Depth [km]
Pressure [GPa]
0
−5
−10
−15
−20
−25
−30
0.0
0.5
1.0
1.5
2.0
2.5
3.0

pySALEPlot: profile_example.py → profile.py

- [iSALE]/share/examples
  - Chicxulub
    - profile_example.py

1. プロットする x の場所の変更
2. プロットされた x の表示
   a. ax.plot(step1.Pre[50,:]*1e-9,model.yc[0,:],'m--',
      label='{:3.1f} s x = {:3.1f}km'
      .format(step1.time,model.xc[50,:][0]))
3. y 方向ではなく、ある y での x 方向の圧力をプロットする（自習課題）

pySALEPlot: profile_example.py → tracer_profile.py

- [iSALE]/share/examples
  - Chicxulub
    - profile_example.py

1. トレーサー粒子の最大圧力をプロットする
   a. 読み込む変数を変える
      i. step0=model.readStep('TrP',0)

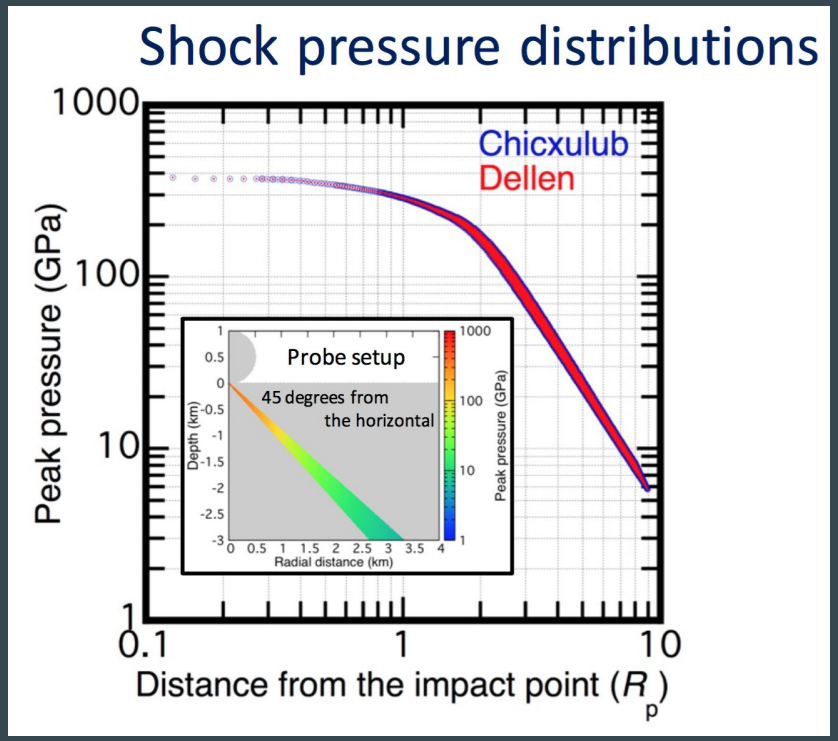


図:黒澤校長より寄贈

pySALEPlot: profile_example.py → tracer_profile.py

- [iSALE]/share/examples
  - Chicxulub
    - profile_example.py

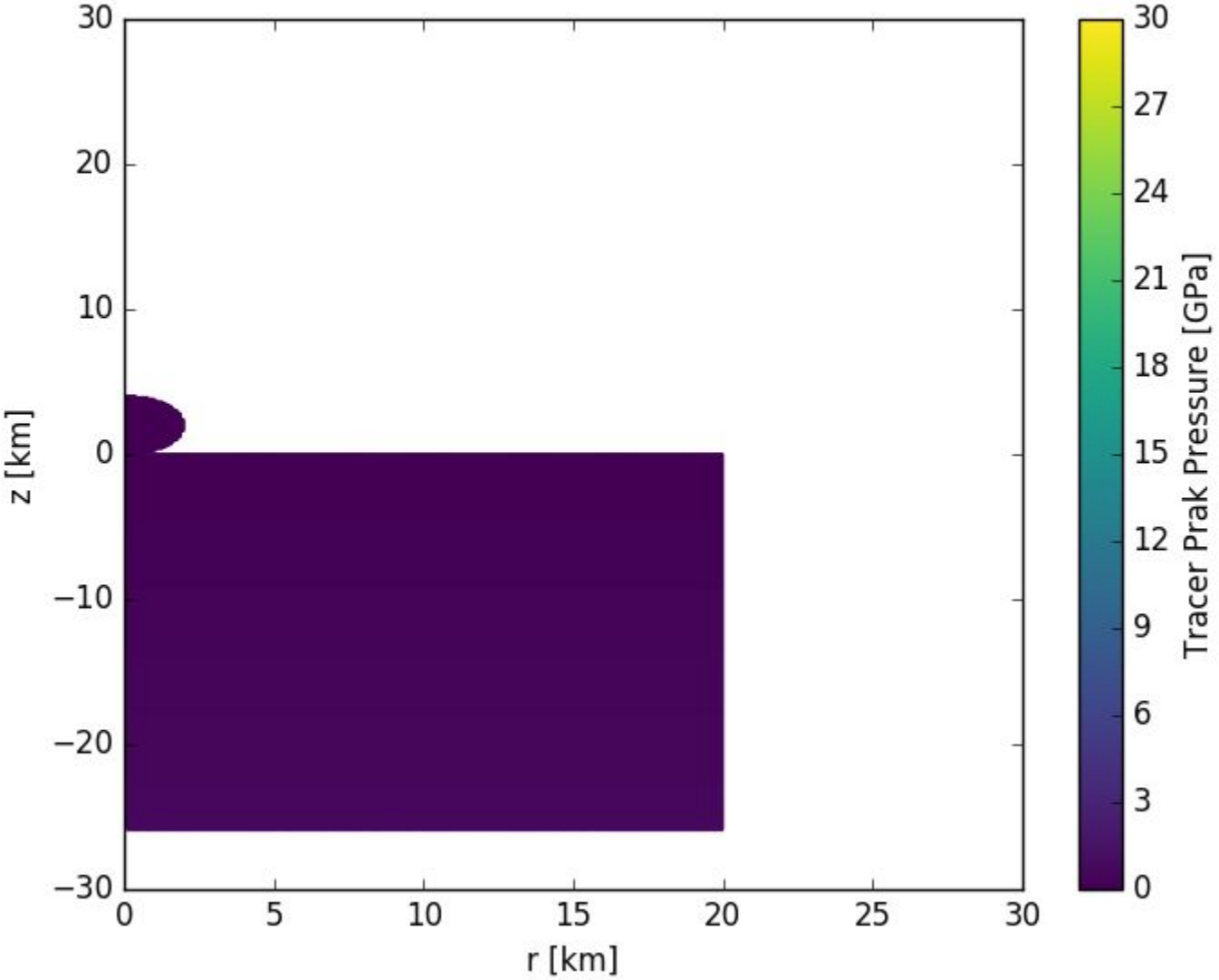
1. ある角度での最大圧力をプロットする
   a. 読み込む変数を変える
   b. トレーサー粒子の位置を得る
      i. XY座標: step0.xmark と step0.ymark
      ii. オブジェクトの数: model.tracer_numu
      iii. トレーサー粒子の番号:
      オブジェクトナンバーが0なら
      model.tru[0].start から model.tru[0].end まで

pySALEPlot: profile_example.py → tracer_profile.py

- [iSALE]/share/examples
  - Chicxulub
    - profile_example.py

1. トレーサー粒子の最大圧力をプロットする
   a. プロットする

```
for u in range(model.tracer_numu):
    tstart = model.tru[u].start
    tend = model.tru[u].end
    scat = ax.scatter(step0.xmark[tstart:tend],step0.ymark[tstart:tend],
                      c=step0.TrP[tstart:tend]*1e-9,vmin=0,vmax=30,
                      cmap='viridis',s=4,linewidths=0)
```


30
20
10
0
−10
−20
−30
z [km]
0
5
10
15
20
25
30
r [km]
30
27
24
21
18
15
12
9
6
3
0
Tracer Prak Pressure [GPa]

pySALEPlot: profile_example.py → tracer_profile.py

- [iSALE]/share/examples
  - Chicxulub
    - profile_example.py

1. ある角度での最大圧力をプロットする
   a. 角度を求める
      i. import numpy as np
      ii. angles = np.rad2deg(np.arctan(step0.xmark/step0.ymark))
      -90 度 から 90度まで

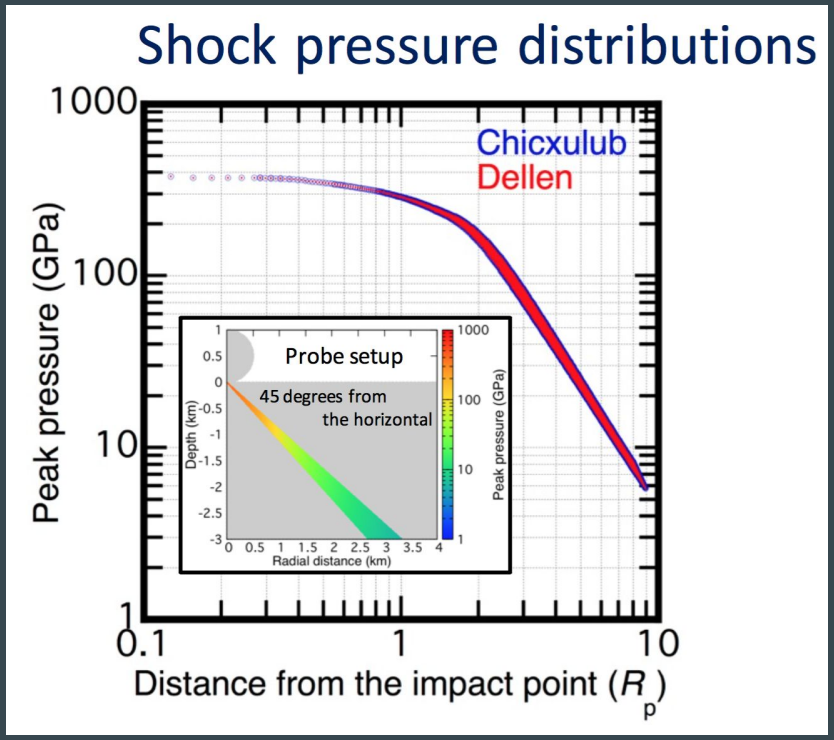


図:黒澤校長より寄贈

pySALEPlot: profile_example.py → tracer_profile.py

- [iSALE]/share/examples
  - Chicxulub
    - profile_example.py

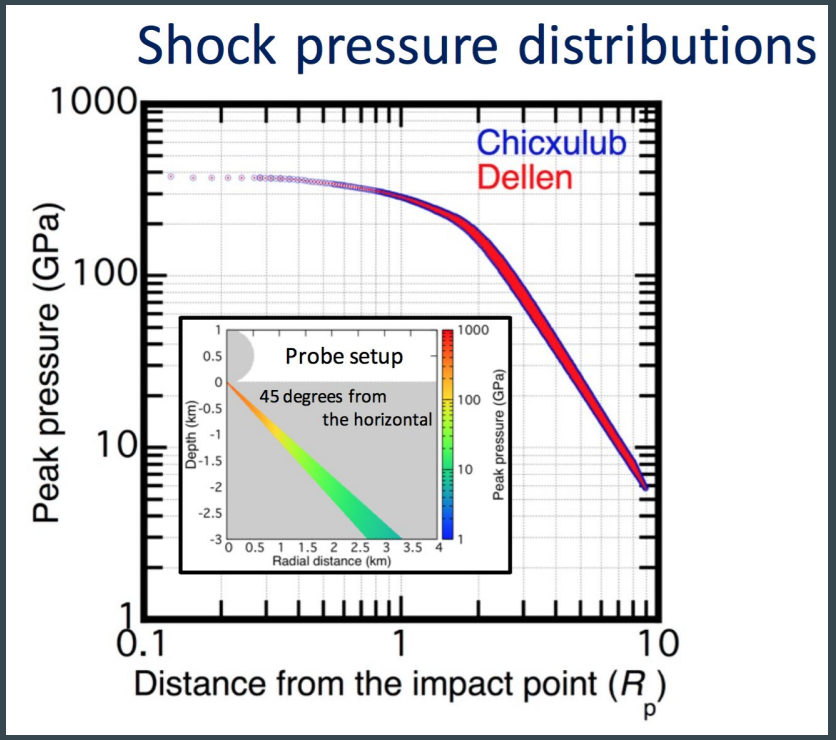
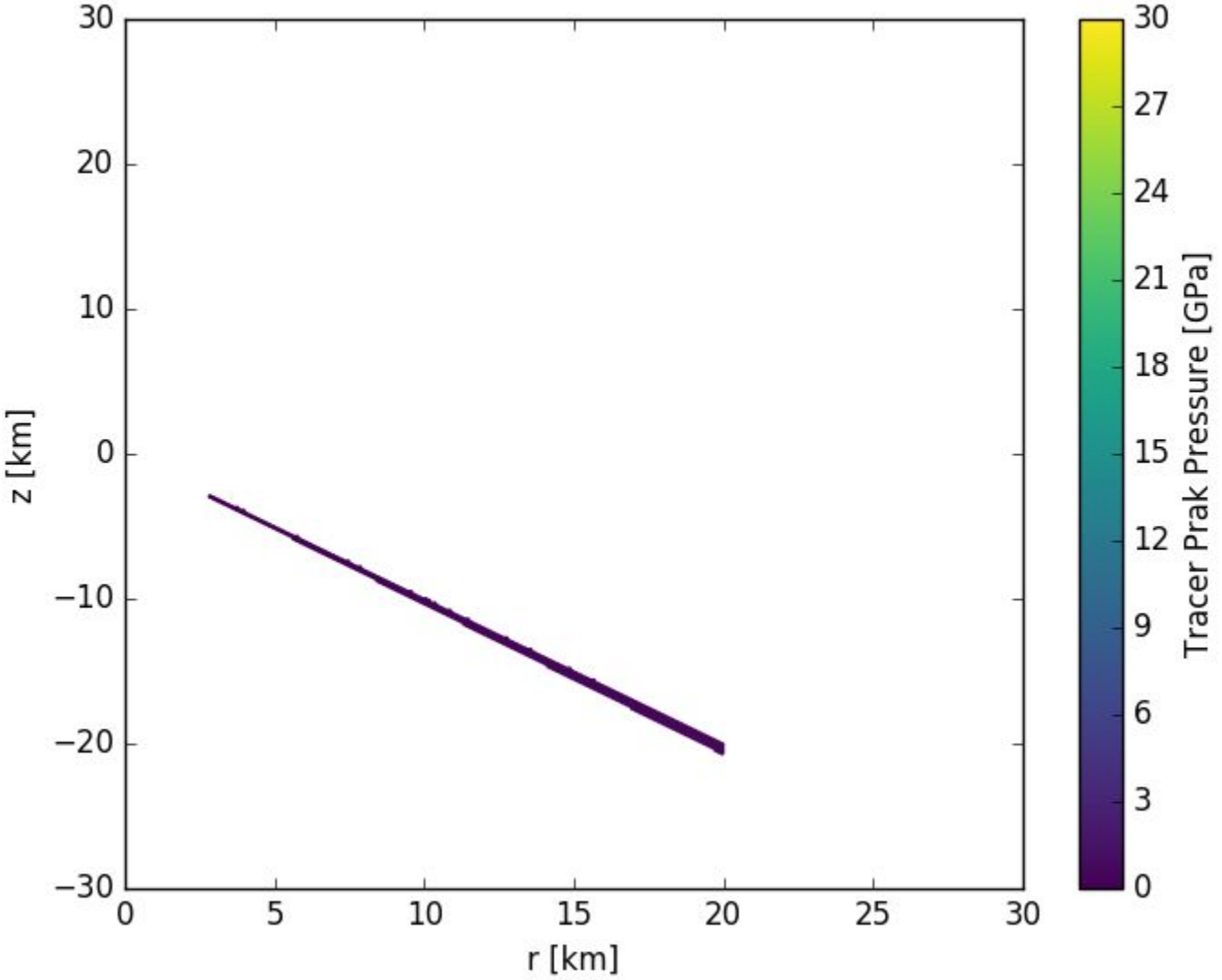


図:黒澤校長より寄贈

1. ある角度での最大圧力をプロットする
   a. 角度を求める
   b. ある角度を探す
      i. angle = -45.
      ii. angle_range = np.where((angles > angle)
      & (angles < angle + 1.))


30
20
10
0
−10
−20
−30
z [km]
0
5
10
15
20
25
30
r [km]
30
27
24
21
18
15
12
9
6
3
0
Tracer Prak Pressure [GPa]

pySALEPlot: profile_example.py → tracer_profile.py

- [iSALE]/share/examples
  - Chicxulub
    - profile_example.py

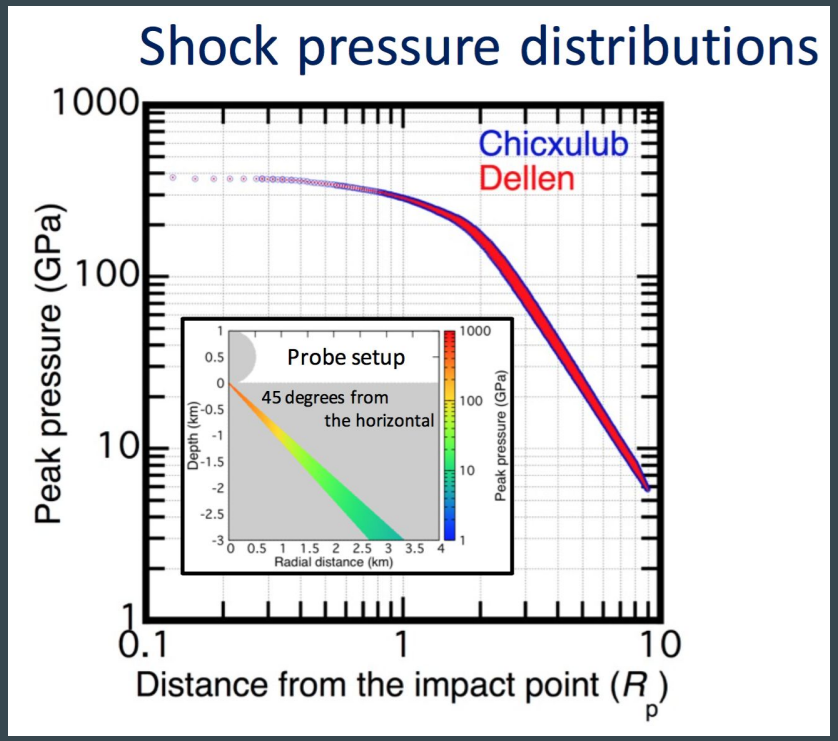
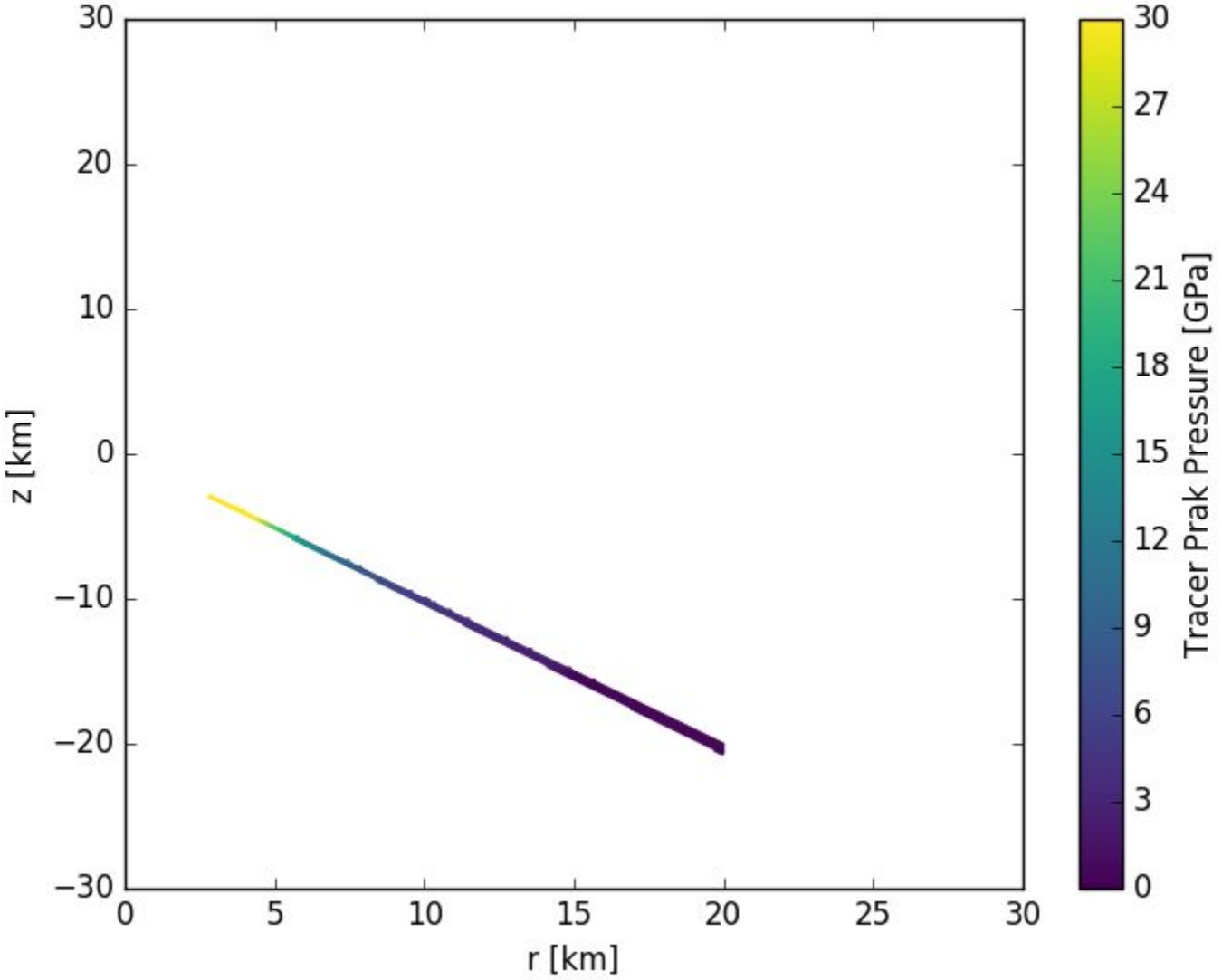


図:黒澤校長より寄贈

1. ある角度での最大圧力をプロットする
   a. 角度を求める
   b. ある角度を探す
   c. ある角度だけプロットする
      i. scat = ax.scatter(step0.xmark[angle_range],step0.ymark[angle_range],
         c=step0.TrP[angle_range]*1e-9,


30
20
10
0
−10
−20
−30
z [km]
0
5
10
15
20
25
30
r [km]
30
27
24
21
18
15
12
9
6
3
0
Tracer Prak Pressure [GPa]

pySALEPlot: profile_example.py → tracer_profile.py

- [iSALE]/share/examples
  - Chicxulub
    - profile_example.py

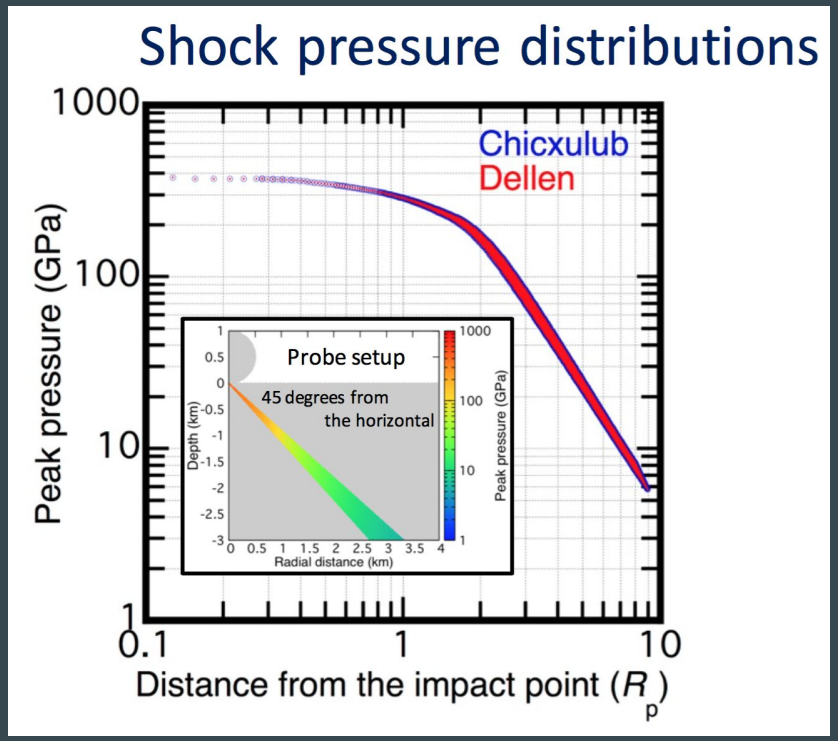


図:黒澤校長より寄贈

1. ある角度での最大圧力をプロットする
   a. 角度を求める
   b. ある角度を探す
   c. ある角度だけプロットする
   d. ある時間での最大圧力をプロットする(自習課題)
   e. 距離と圧力の図を描く

pySALEPlot: profile_example.py → tracer_profile.py

- [iSALE]/share/examples
  - Chicxulub
    - profile_example.py

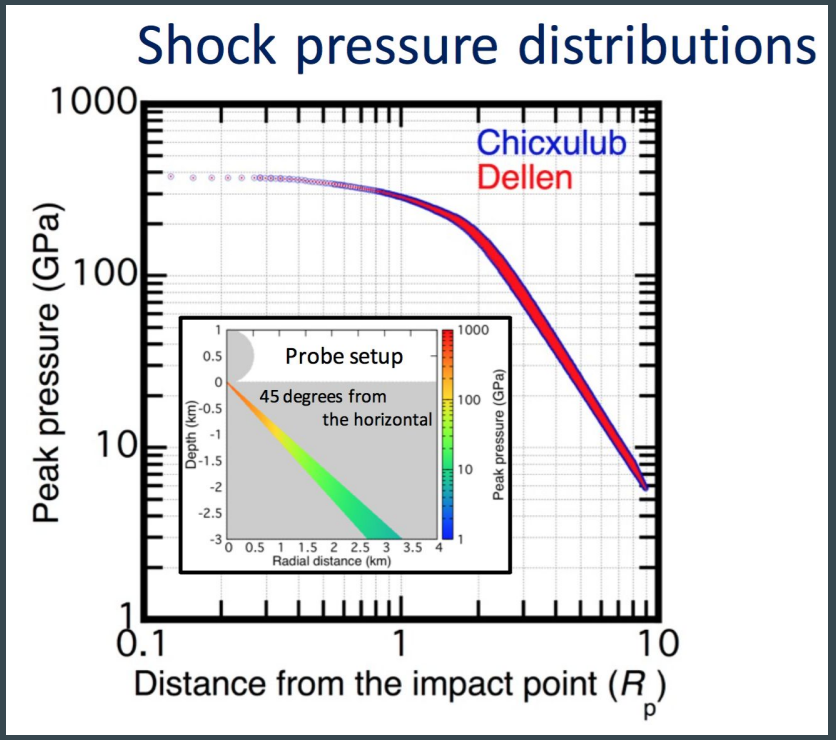
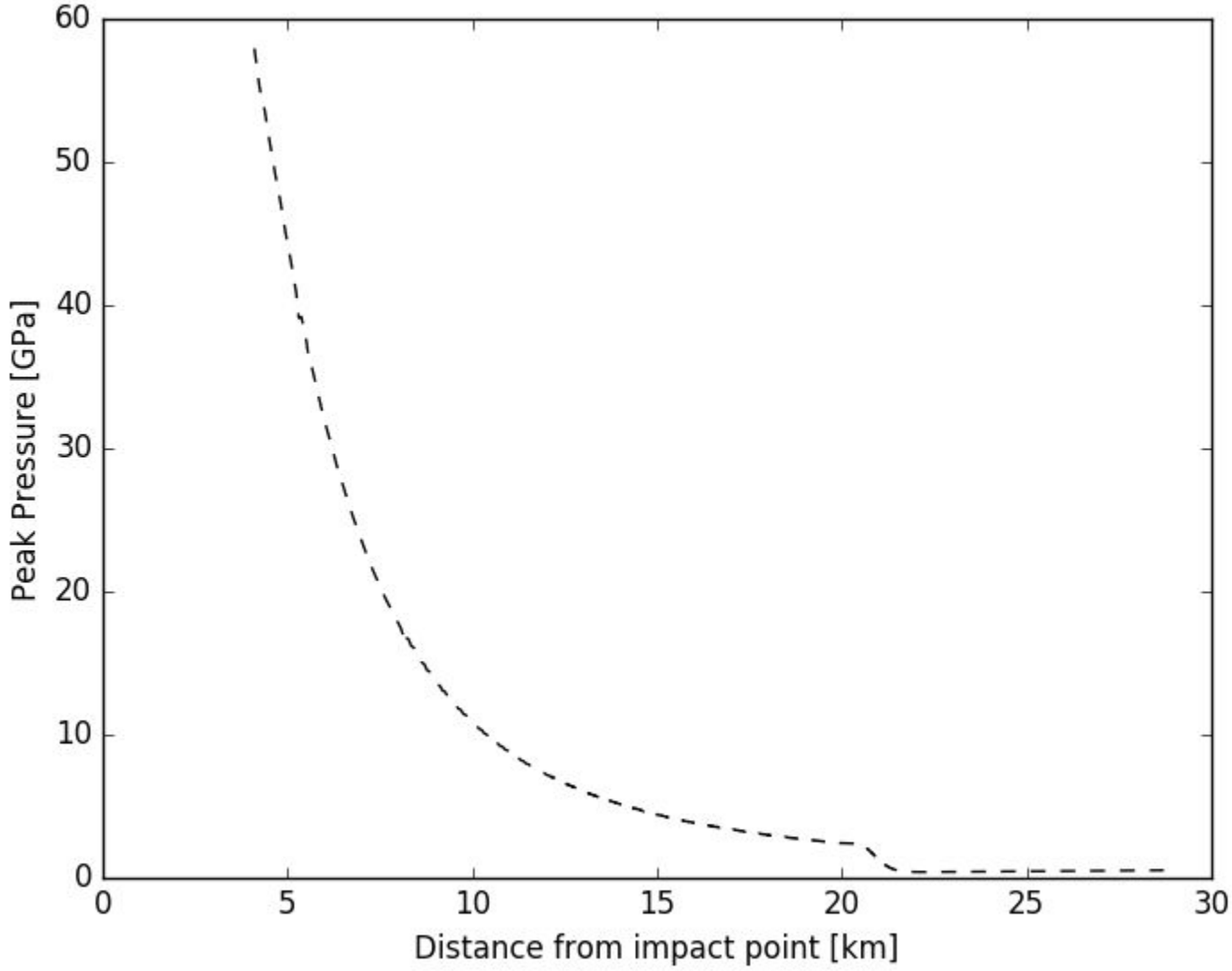


図:黒澤校長より寄贈

1. ある角度での最大圧力をプロットする
   e. 距離と圧力の図を描く
      i. 距離を求めて、並べる

```
r = np.sqrt(step0.xmark[angle_range]**2+step0.ymark[angle_range]**2)
trp = step1.TrP[angle_range]*1e-9
ax.plot(np.sort(r),trp[np.argsort(r)],'k-')
```


Peak Pressure [GPa]
Distance from impact point [km]
0
10
20
30
40
50
60
0
5
10
15
20
25
30

まとめ

- pySALEPlotを用いた描画
  - かんたん編
    - pySALEPlot + python の仕組みを理解
    - 一変数でのカラコンター図を描く
  - 実践編
    - Sampleのpythonスクリプトを変更する
    - 描きたい変数の形を知る
      - matplotlib と関係する
    - numpy の機能をちょっと使ってみる
      - 三角関数とか