

PyVTK による VTK ファイルの作成 と ParaView による 3D 可視化

妹尾 賢 (SENOO, Ken)

✉ contact@senooken.jp

🗉 <https://social.senooken.jp/senooken>

2014-08-30

第 41 回オープン CAE 勉強会 @ 関東

<http://opencae-kanto.connpass.com/event/7835/>

URL: <https://senooken.jp/public/20140830/>



This work is licensed under the [Creative Commons Attribution 4.0 International License](https://creativecommons.org/licenses/by/4.0/).

内容

- 可視化の方法
- VTK ファイル
- PyVTK
- 簡単な例
- 応用：日本近海の地形データの 3D 可視化

データ（計算結果）の可視化の手段と目的

手段	目的	難易度
■ 表 ■ 1D 折れ線グラフ ■ 2D コンター（画像・動画） ■ 3D CG（画像・動画）	■ 統計値の確認 ■ 検証（観測との比較） ■ 結果のおおまかな確認 ■ PR	

データの次元の上昇

-> データ処理の難易度↑

-> インパクト大

3D アニメーションのインパクトが強い！

3D 可視化の課題

可視化ソフトやデータ形式, データの用意・変換など障壁大

ParaView が以下の点でベスト？

- ▶ OpenFOAM でも推奨
- ▶ 利用者が多い
- ▶ 自由・高機能

- 可視化ソフト -> ParaView
- 入力データ -> 計算結果・オープンデータ
- データ形式 -> VTK

■ OpenGL をバックに備えた 3D 可視化ライブラリ群

- * ParaView で表示可能
- * データタイプにより書式が異なる。
- * POLYDATA が汎用的な印象

```
# vtk DataFile Version 2.0           (1)
Really cool data                     (2)
ASCII | BINARY                       (3)
DATASET type                        (4)
...
POINT_DATA n                       (5)
...
CELL_DATA n
...
```

VTK ファイル作成が面倒
なんかライブラリとかないか？

Python ならある -> PyVTK

Part 1: Header

Part 2: Title (256 characters maximum, terminated with newline \n character)

Part 3: Data type, either ASCII or BINARY

Part 4: Geometry/topology. *Type* is one of:

STRUCTURED_POINTS
STRUCTURED_GRID
UNSTRUCTURED_GRID
POLYDATA
RECTILINEAR_GRID
FIELD

Part 5: Dataset attributes. The number of data items *n* of each type must match the number of points or cells in the dataset. (If *type* is FIELD, point and cell data should be omitted.)

Figure 1 Overview of five parts of VTK data file format.

PyVTK

- Python の VTK ファイルの読み書きモジュール
- 公式サイト： pyvtk - PyVTK - Tools for manipulating VTK files in Python - Google Project Hosting <https://code.google.com/p/pyvtk/>
- 最新バージョン： 0.4.85 (2014-07-23)
- 情報がほとんどない▷ソースコードやヘルプ・サンプルの解読必須
- 対応 VTK ファイルバージョン： 2.0 (ACSI , バイナリ)
 - ▶ いわゆるレガシー形式の VTK に対応
 - ▶ 最新のバージョン 4.2 や xml 形式には未対応
 - ▶ 時間データの取り扱いは連番ファイル

インストール

```
pip install pyvtk
```

PyVTK による VTK ファイルの作成手順

※：汎用性が最も高いと思われるポリゴンのみ扱う

1. モジュールのインポート

```
import pyvtk
```

ポリゴンの構成ノード：時計回りにつける

2. ファイル読み込み

1. numpy.genfromtxt や pandas.read_csv など

3. 座標値の格納

```
structure=pyvtk.PolyData(points=[( ノード 1 の座標 ), ( ノード 2 の座標 ), ..., ( ノード n の座標 ),  
polygons=[( ポリゴン 1 の構成ノード ), ( ポリゴン 2 の構成ノード ), ( ポリゴン 3 の構成ノード )])
```

4. 属性値の指定 (属性：通常は温度や圧力などの 3 次元データ)

1. ノードごと

```
pointdata=pyvtk.PointData(pyvtk.Scalars([ ノードの値 ], name= 値の名前 , lookup_table="default"))
```

2. セルごと

```
celldata=pyvtk.CellData(  
pyvtk.Scalars([ セル 1 の値 , セル 2 の値 , ..., セル n の値 ], name= 値の名前 , lookup_table="default"))
```

5. VTK オブジェクトの作成

```
vtk=pyvtk.VtkData(structure, ヘッダー, pointdata, celldata)
```

6. ファイルへの出力

```
vtk.tofile("name")
```

ノード番号： points への代入順

難点： ポリゴンの構成ノードの用意

簡単な例

x, y, z 座標の書かれたファイルを読み込んで VTK 形式で出力

```
#!/usr/bin/env python2.7
# coding: utf-8
# test.py
import pyvtk
import numpy as np

FR="./test.csv"
XYZ=np.genfromtxt(FR, delimiter=",", skip_header=1)

## X と Y の座標の数を取得
XMAX=len(np.unique(XYZ[:,0]))
YMAX=len(np.unique(XYZ[:,1]))

### polygon の構成ノードを格納
polygon= [[ XMAX*(i)+j, XMAX*(i+1)+j, XMAX*(i+1)+j+1, XMAX*(i)+j+1]
           # LL, LR, UR, UL の順番。 端を除外
           for i in range(YMAX-1) for j in range(XMAX-1) ]

structure=pyvtk.PolyData(points=XYZ, polygons=polygon)

pointdata = pyvtk.PointData(pyvtk.Scalars(XYZ[:,2], name='point-scalar', lookup_table='default'))

cellvalue=[np.mean([XYZ[node, 2] for node in nodelist], dtype=np.float32) for nodelist in
polygon]

celldata = pyvtk.CellData(pyvtk.Scalars(cellvalue, name='cell-scalar', lookup_table="default"))

vtk = pyvtk.VtkData( structure, "# test", celldata, pointdata)
vtk.tofile('test')
```

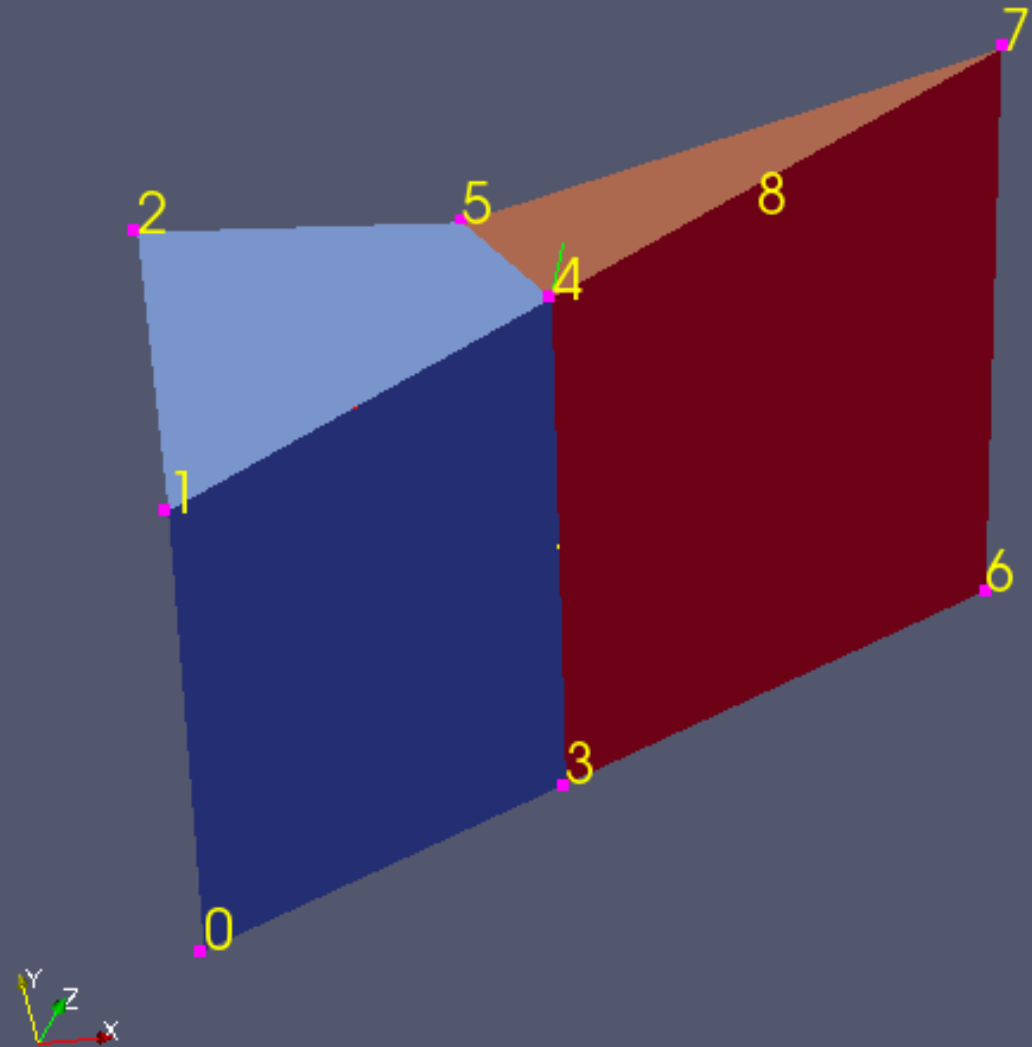
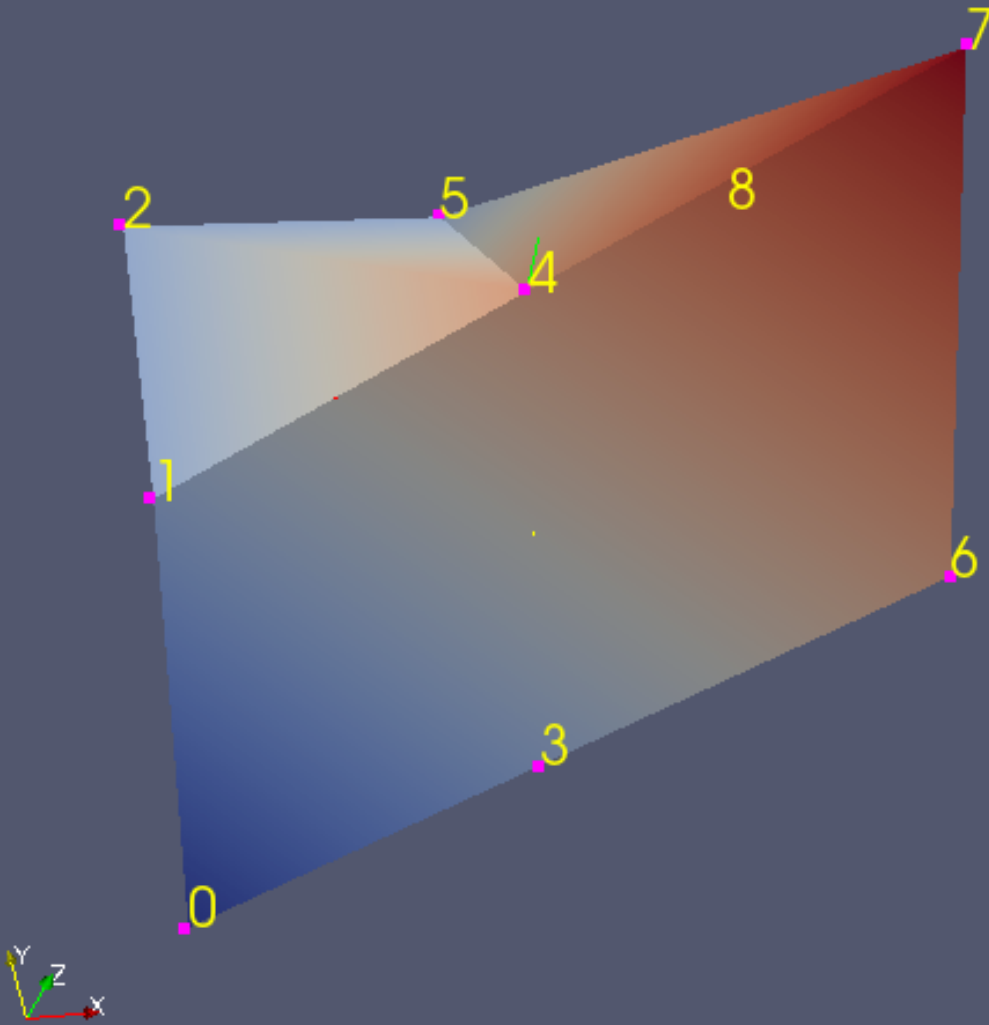
test.csv

x, y, z
0, 0, 0
0, 1, 1
0, 2, 1
1, 0, 1
1, 1, 2
1, 2, 1
2, 0, 2
2, 1, 3
2, 2, 1

test.vtk の表示結果

point-scalar

cell-scalar



- * 頂点の値を使うと間の色は勝手に補完
- * セルごとの色のほうが好ましい？

応用

日本近海の地形データの 3D 可視化

使用データ：ETOPO1

NOAA NATIONAL GEOPHYSICAL DATA CENTER
NATIONAL OCEANIC AND ATMOSPHERIC ADMINISTRATION

NOAA > NESDIS > NGDC > Marine Geology & Geophysics > Bathymetry & Relief

All Bathy/Relief Coastal DEM Portal Fishing Global Lakes Multibeam privacy policy NOS

- 全球の陸地 + 海底地形データ
- 解像度：1' (約 1.8 km)
- 配布元：米国大気海洋庁 (NOAA: National Oceanic and Atmospheric Administration)
- URL: ETOPO1 Global Relief | [ngdc.noaa.gov](http://ngdc.noaa.gov/mgg/global/global.html)
<http://www.ngdc.noaa.gov/mgg/global/global.html>
- ライセンス：出典明記

ETOPO1 Development Report (PDF)
Download data files (zipped)
NEW! Frequently Asked Questions
Ocean Volumes
NEW! Bathymetry Data Sources
Download Color Image
Download KMZ Images File
Download "Posters"
Access Web Services

ETOPO1 Global Relief Model
ETOPO1 is the most complete global bathymetry and topography data set available. It includes numerous global and regional data sets, and is available in "Ice Surface" (top of Antarctic and Greenland ice sheets) and "Bedrock" (base of the ice sheets).
NEW! NGDC has utilized the ETOPO1 Global Relief Model to calculate the volume of Earth's oceans and to derive a hypsographic curve of Earth's surface.

DEM Discovery Portal
Earth Icosahedron
More Images & Animations...
Related Projects:
Related Data at NGDC:
Bathymetry & Relief
Marine Geology & Geophysics
Related External Links:

データ入手手順

フルセット (解凍後 6 GB) は重い。
日本近海のみ切り出す。

NOAA NATIONAL GEOPHYSICAL DATA CENTER
NATIONAL OCEANIC AND ATMOSPHERIC ADMINISTRATION

NOAA > NESDIS > NGDC > Marine Geology & Geophysics > Bathymetry & Relief

All Bath/Relief Coastal DEM Portal Fishing Global Lakes Multibeam NOS

ETOPO1 Global Relief Model

Extract Custom Grid

Interactive map interface to Bathymetry Data

ETOPO1 Development Report (PDF) (download data files at right)

New! Frequently Asked Questions

ETOPO1 is a 1-arc-minute global relief model of Earth's surface that integrates land topography and ocean bathymetry. It was built from numerous global and regional data sets, and is available in "Ice Surface"

DEM Discovery Portal

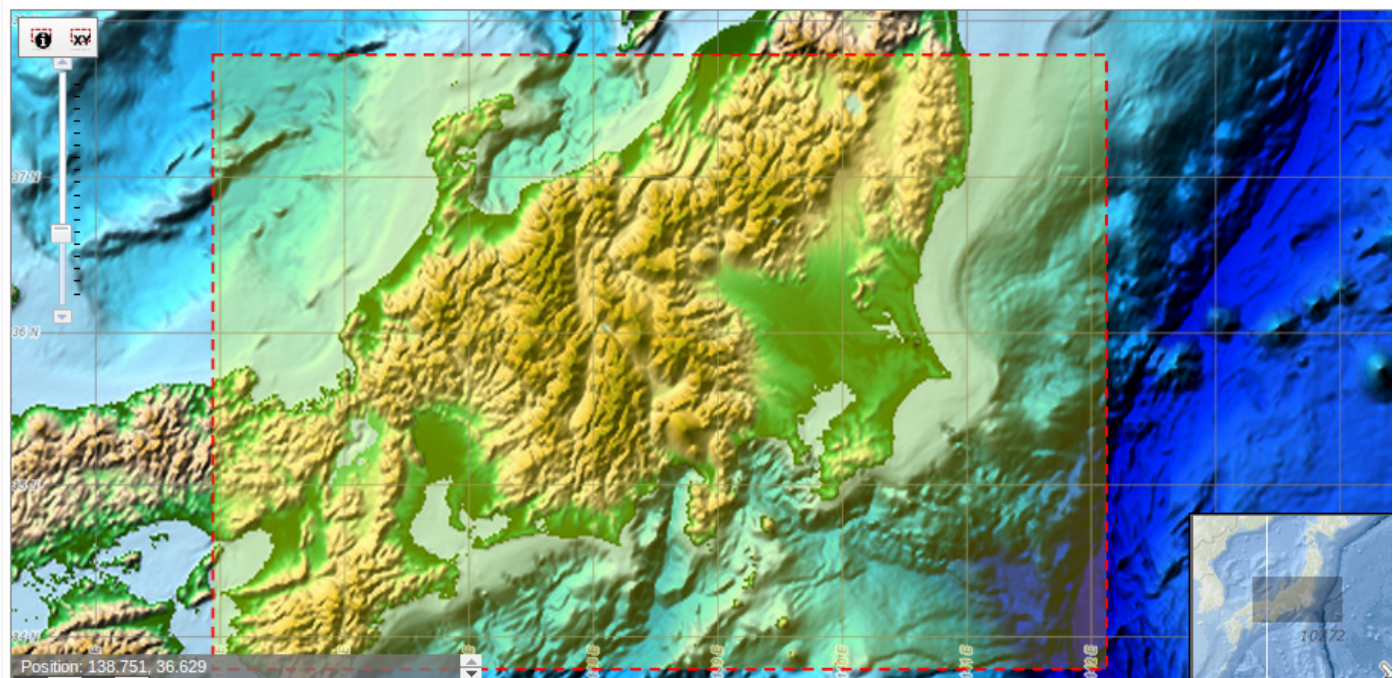
Earth Icosahedron

More Images & Animations...

Related Projects:

NOAA NATIONAL GEOPHYSICAL DATA CENTER
NATIONAL OCEANIC AND ATMOSPHERIC ADMINISTRATION

Grid Extract



1. Choose a Layer 2. Select an Area from the map 3. Choose Output Format 4. Download the Data

ETOPO1 (ice)

Grid of Earth's surface depicting the top of the Antarctic and Greenland ice sheets (1-minute resolution)

134.94, 33.784, 142.11, 37.784

XYZ

[click here to download](#)

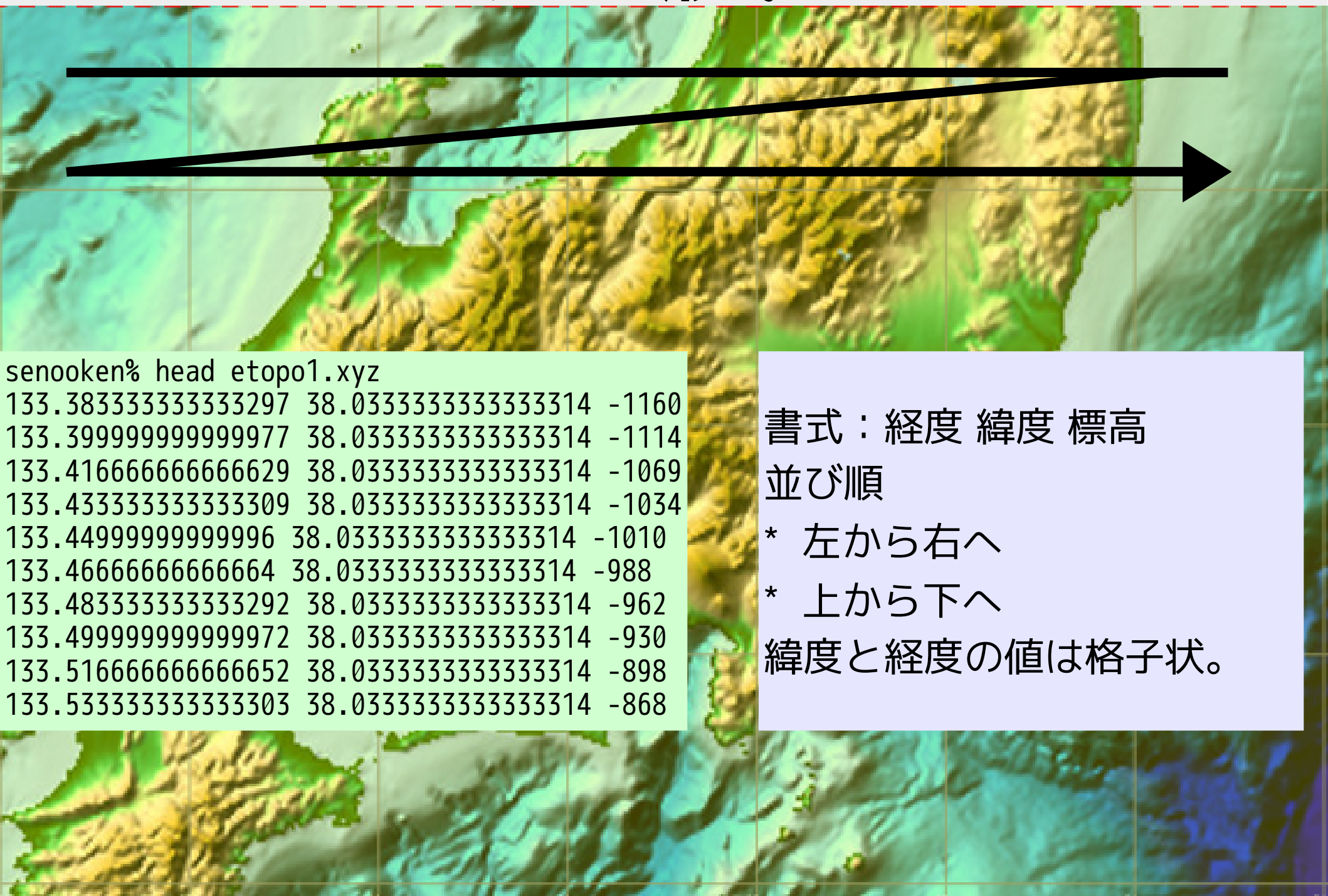
左上の "i" マークでドラッグで
範囲指定

書式は以下を選択

- * ETOPO1(ice)
- * データ形式： XYZ

[click here to donwload](#)

ETOPO1 XYZ のデータ形式



```
senooken% head etopo1.xyz
133.383333333333297 38.033333333333314 -1160
133.399999999999977 38.033333333333314 -1114
133.416666666666629 38.033333333333314 -1069
133.433333333333309 38.033333333333314 -1034
133.449999999999996 38.033333333333314 -1010
133.466666666666664 38.033333333333314 -988
133.483333333333292 38.033333333333314 -962
133.499999999999972 38.033333333333314 -930
133.516666666666652 38.033333333333314 -898
133.533333333333303 38.033333333333314 -868
```

書式：経度 緯度 標高
並び順

* 左から右へ

* 上から下へ

緯度と経度の値は格子状。

VTK への変換プログラム

```
#!/usr/bin/env python2.7
# coding: utf-8
# etopo2vtk.py
import pyvtk
import numpy as np
import matplotlib.pyplot as plt
import pyproj
```

```
## ファイルの読み込み
FR="./etopo1.xyz"
XYZ=np.genfromtxt(FR)
```

```
## X と Y の座標の数を取得
XMAX=len(np.unique(XYZ[:,0]))
YMAX=len(np.unique(XYZ[:,1]))
X, Y=np.meshgrid(np.unique(XYZ[:,0]), np.unique(XYZ[:,1]))
Y=Y[::-1] # データが降順なので
```

```
## 座標変換
proj=pyproj.Proj(proj="utm", zone=54, ellps="WGS84")
X, Y=proj(X, Y)
```

```
## データをうまくよみとれたか確認
plt.clf()
plt.contourf(X,Y, XYZ[:,2].reshape(YMAX, XMAX))
plt.savefig("./check.png", bbox_inches="tight")
```

```
### polygon の構成ノードを格納
polygon= [[ XMAX*y+x, XMAX*y+x+1, XMAX*(y+1)+x+1, XMAX*(y+1)+x]
           # UL, UR, LR, LL の順番。 端を除外 1
           for y in range(YMAX-1) for x in range(XMAX-1) ]
```

```
# pointdata = pyvtk.PointData(pyvtk.Scalars(XYZ[:,2], name='point-scalar', lookup_table='default'))
XYZ=np.dstack([X,Y,XYZ[:,2].reshape(YMAX, XMAX)])
XYZ=XYZ.reshape(-1,3).tolist()
```

```
structure=pyvtk.PolyData(points=XYZ, polygons=polygon)
cellvalue=[np.mean([XYZ[node][2] for node in nodelist], dtype=np.float32) for nodelist in polygon]
celldata = pyvtk.CellData(pyvtk.Scalars(cellvalue, name='cell-scalar', lookup_table="default"))
```

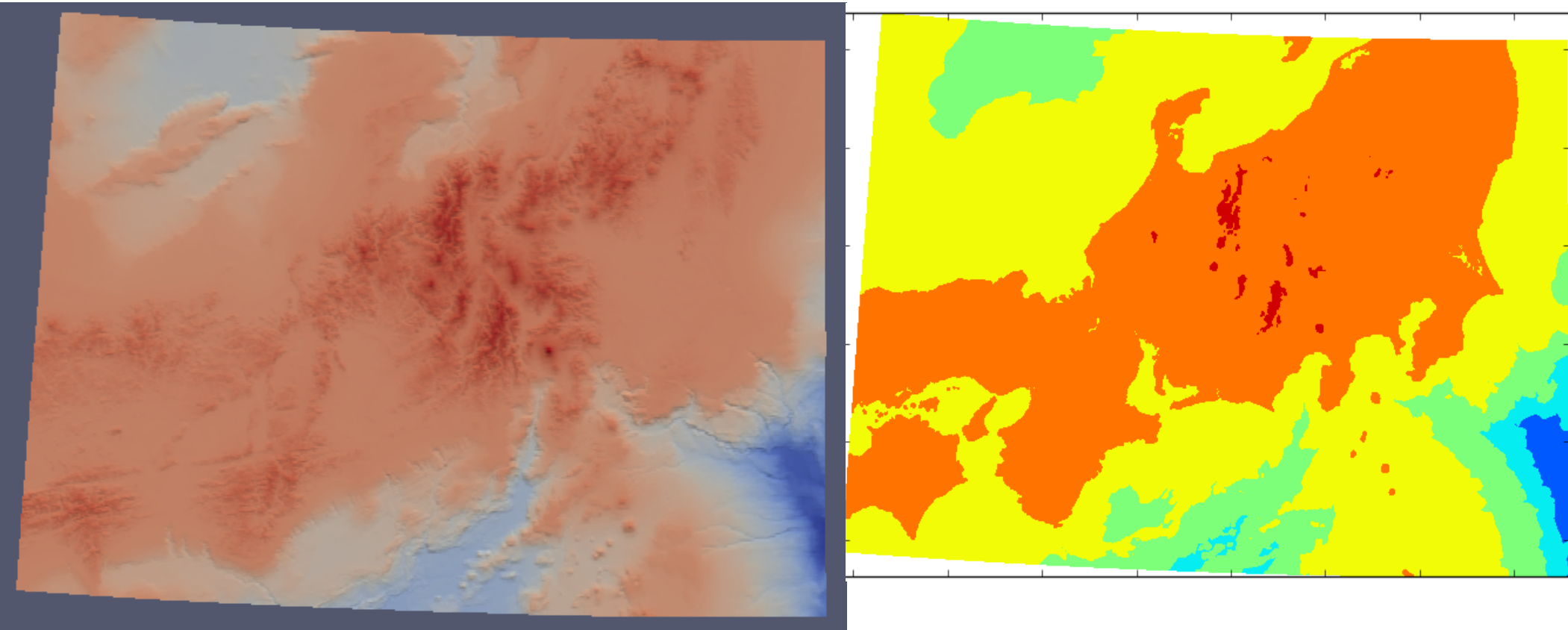
```
# vtk = pyvtk.VtkData( structure, "# etopo", pointdata, celldata)
vtk = pyvtk.VtkData( structure, "# etopo", celldata)
vtk.tofile('etopo')
```

ポイント

XY と Z のスケール差が大
-> 3D 表現困難 (2D は可)
-> 座標変換

データ読み取り確認
-> 2D コンター図の出力

日本付近の ETOPO1 の可視化結果



後は ParaView で煮るなり焼くなり自由に表現

まとめ

- Python の VTK ファイルの読み書きモジュール PyVTK を紹介
- 簡単な例で VTK ファイルの可視化例を紹介
- 応用としてオープンデータ ETOPO1 で日本近海を 3D 表示

課題

- ポリゴン以外の VTK のデータタイプの利用
- 複雑データのポリゴンの構成ノード番号の対応関係の取得
- メモリに入りきらないような大規模データの処理
- プログラムの高速化 (ループ部分での Cython の利用)

質疑応答（発表後に追記）

- 複雑地形のポリゴンの幾何情報にはデローニ（ドロネー）三角分割が使える。
- 本当に複雑なら別ファイルで輪郭線の情報を用意して参照するとか，欠損はダミーとしてあつかうとかして対処するのあり。
- VTK を経由せずに ParaView の Python filter 書いて，直接ファイルを読み込んだほうがデータの変換忘れがなくてよい。