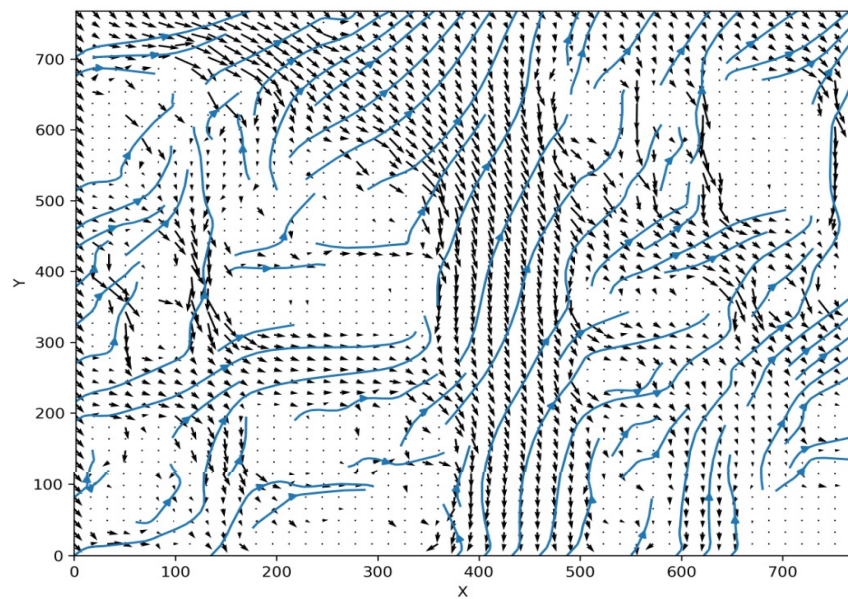


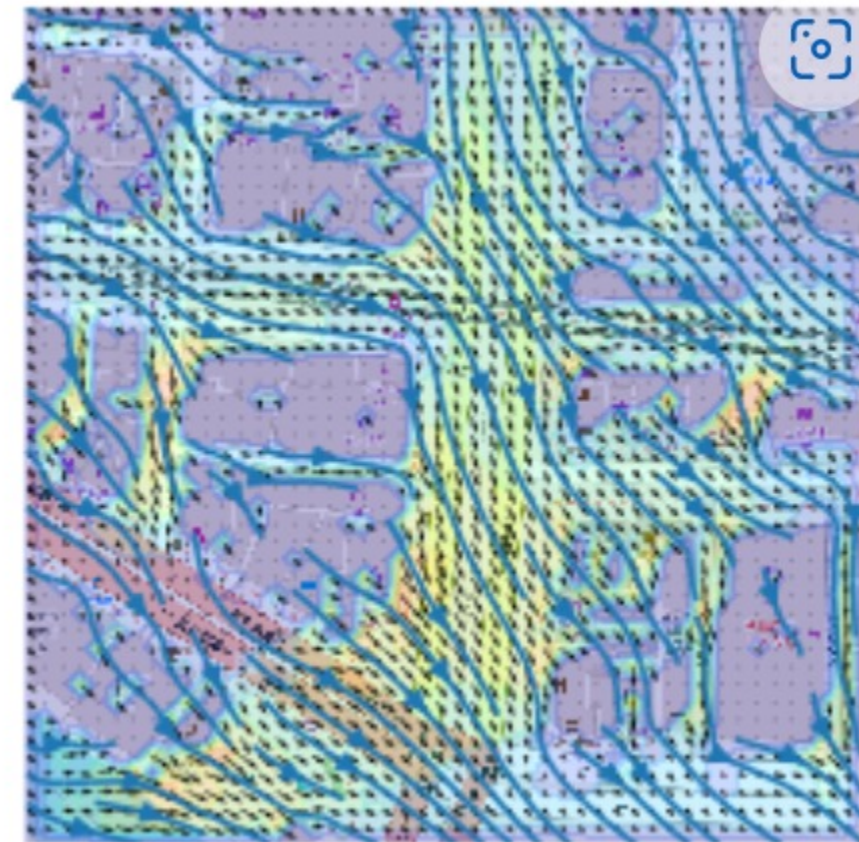
地図



気象
データ



合体

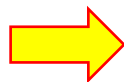
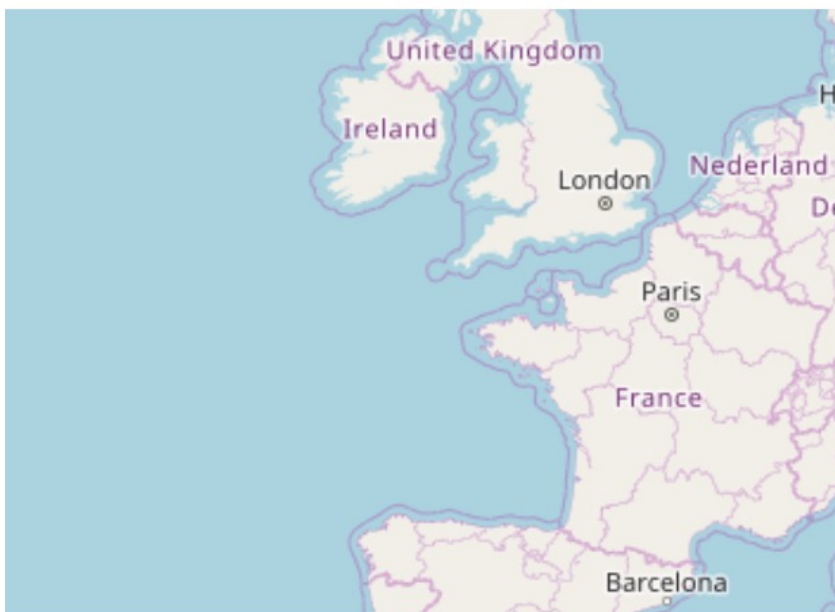


OpenStreetMap のタイル画像入手

pip install smopyでインストール
pip install smopy[ipython]

地理座標 (緯度/経度) とズームレベルのボックスを指定すると、Smopy は OpenStreetMap のタイル画像を入手

```
import smopy
map = smopy.Map((42., -1., 55., 3.), z=4)
map.show_ipython()
```



マップを matplotlib にインポートして、地理座標からピクセルに簡単に変換

```
x, y = map.to_pixels(48.86151, 2.33474)
ax = map.show_mpl(figsize=(8, 6))
ax.plot(x, y, 'or', ms=10, mew=2);
```



引用: https://github-com.translate.goog/rossant/smopy?_x_tr_sl=en&_x_tr_tl=ja&_x_tr_hl=ja&_x_tr_pto=sc

OpenWeatherMapの情報入手

OpenWeatherMap (<https://openweathermap.org/>) のAPI Keyを取得
無料のFreeプランだと5日後までの3時間毎の天気予報を取得



出典: <https://www.suzu6.net/posts/78-owm/>

```
In [1]: %matplotlib inline
```

```
In [20]: # we need to install pyowm. (Linux only)
# conda install -c auto pyowm
```

```
import pyowm
```

```
def getWind(lon, lat): # OWM 気象読み込み関数
```

```
    owm = pyowm.OWM('insert your OWM API key')
```

```
    weather = (owm.weather_at_coords(lat, lon)).get_weather()
```

```
    wind = weather.get_wind()
```

```
    return wind
```

OpenWeatherMapのAPI Key

緯度、経度

```
In [36]: # 地図情報読み込み関数 (画像行列を返す)
# we need to install smopy
# https://github.com/rossant/smopy
```

```
import smopy
```

```
import cv2
```

```
import numpy as np
```

```
import matplotlib.pyplot as plt
```

```
def getMapFromOSM(lon, lat, nx, ny):
```

```
    angle=0.000001
```

```
    map = smopy.Map((lon, lat, lon+angle, lat+angle), z=19)
```

```
    mapImg = map.to_numpy()
```

```
    return mapImg, cv2.resize(mapImg, (nx, ny))
```

OpenStreetMapから地図情報ゲット

```
def getMaskFromMap(mapImg):
```

```
    lower = np.array([100, 19, 100])
```

```
    upper = np.array([150, 20, 250])
```

```
    hsv = cv2.cvtColor(mapImg, cv2.COLOR_BGR2HSV)
```

```
    mask = cv2.inRange(hsv, lower, upper) # building = 255, other = 0
```

```
    maskedMapImg = cv2.bitwise_and(mapImg, mapImg, mask)
```

```
    return maskedMapImg, mask
```

建物の色を抽出

In [22]: # 流体計算を行う関数定義

```
# Following code is customized version of "barbagroup/CFDPython"
# https://github.com/barbagroup/CFDPython
#
# Jun Hirabayashi added few small features(obstacles, export wind UV images)
# to original CFDPython.
#
# please check, original version and github URL.
# CFDPython: Copyright (c)Barba group, and it's license: BSD-3-Clause

import numpy
import math
import json
import datetime
from matplotlib import pyplot, cm
from mpl_toolkits.mplot3d import Axes3D
from __future__ import division # in Python 2.x envirointment
```

```
def loadVelocityMask(u, v, mask): #, mask_u, mask_v, u, v):
    for y in range(mask.shape[0]):
        for x in range(mask.shape[1]):
            if mask[y, x] == 255:
                u[y, x] = 0.0
                v[y, x] = 0.0
    return True

def pressure_poisson(p, dx, dy, b):
    pn = numpy.empty_like(p)
    pn = p.copy()
    for q in range(nit):
        pn = p.copy()
        p[1:-1, 1:-1] = (((pn[1:-1, 2:] + pn[1:-1, 0:-2]) * dy**2 +
                        (pn[2:, 1:-1] + pn[0:-2, 1:-1]) * dx**2) /
                        (2 * (dx**2 + dy**2)) -
                        dx**2 * dy**2 / (2 * (dx**2 + dy**2)) *
                        b[1:-1, 1:-1])
        p[:, -1] = p[:, -2] ##dp/dy = 0 at x = end
        p[0, :] = p[1, :] ##dp/dy = 0 at y = 0
        p[:, 0] = p[:, 1] ##dp/dx = 0 at x = 0
        p[-1, :] = p[-2, :] ##dp/dx = 0 at y = end
    return p
```

関数定義1

```
def build_up_b(b, rho, dt, u, v, dx, dy):
    b[1:-1, 1:-1] = (rho * (1 / dt *
                        ((u[1:-1, 2:] - u[1:-1, 0:-2]) /
                        (2 * dx) + (v[2:, 1:-1] - v[0:-2, 1:-1]) / (2 * dy)) -
                        ((u[1:-1, 2:] - u[1:-1, 0:-2]) / (2 * dx))**2 -
                        2 * ((u[2:, 1:-1] - u[0:-2, 1:-1]) / (2 * dy) *
                        (v[1:-1, 2:] - v[1:-1, 0:-2]) / (2 * dx)) -
                        ((v[2:, 1:-1] - v[0:-2, 1:-1]) / (2 * dy))**2))

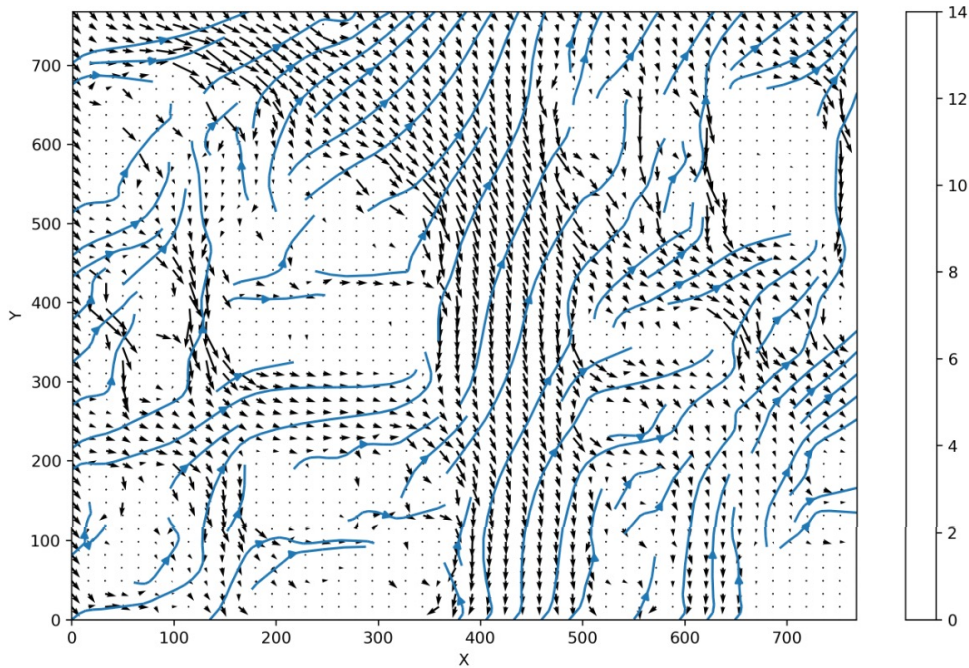
    return b
```

```
def cavity_flow(nt, u, v, dt, dx, dy, p, rho, nu):
    un = numpy.empty_like(u)
    vn = numpy.empty_like(v)
    b = numpy.zeros((ny, nx))
    for n in range(nt):
        un = u.copy()
        vn = v.copy()
        b = build_up_b(b, rho, dt, u, v, dx, dy)
        p = pressure_poisson(p, dx, dy, b)
        u[1:-1, 1:-1] = (un[1:-1, 1:-1] -
                        un[1:-1, 1:-1] * dt / dx *
                        (un[1:-1, 1:-1] - un[1:-1, 0:-2]) -
                        vn[1:-1, 1:-1] * dt / dy *
                        (un[1:-1, 1:-1] - un[0:-2, 1:-1]) -
                        dt / (2 * rho * dx) * (p[1:-1, 2:] - p[1:-1, 0:-2]) +
                        nu * (dt / dx**2 *
                        (un[1:-1, 2:] - 2 * un[1:-1, 1:-1] + un[1:-1, 0:-2]) +
                        dt / dy**2 *
                        (un[2:, 1:-1] - 2 * un[1:-1, 1:-1] + un[0:-2, 1:-1])))
```


In [23]: *# chart rendering*

```
fig = pyplot.figure(figsize=(11, 7), dpi=300)
#pyplot.imshow(mapImg)
plotX, plotY = numpy.meshgrid(numpy.linspace(0, nx*ratio, nx),
                               numpy.linspace(0, ny*ratio, ny))
pyplot.contourf(plotX, plotY, velocity, alpha=0.0, cmap=cm.jet)
# pyplot.contourf(plotX, plotY, mask, alpha=0.1, cmap=cm.jet)
pyplot.colorbar()
pyplot.quiver(plotX, plotY, u, v)
# pyplot.contour(plotX, plotY, mask, cmap=cm.viridis)
pyplot.streamplot(plotX, plotY, u, -v)
pyplot.xlabel('X'); pyplot.ylabel('Y')

pyplot.savefig('figure2.png')
```



In [24]: *# chart rendering*

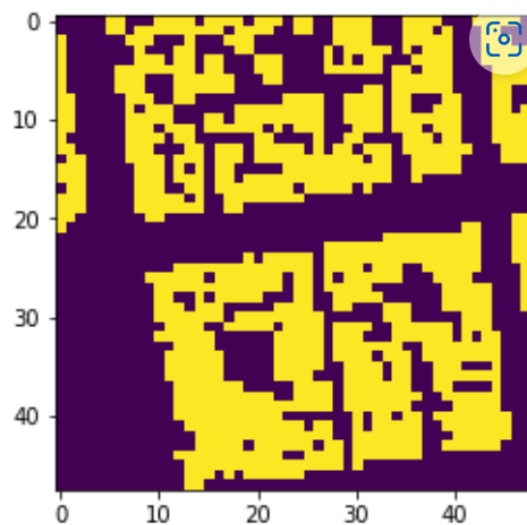
```
fig = pyplot.figure(figsize=(11, 7), dpi=300)

pyplot.imshow(mapImg)
plotX, plotY = numpy.meshgrid(numpy.linspace(0, nx*ratio, nx),
                               numpy.linspace(0, ny*ratio, ny))
pyplot.contourf(plotX, plotY, velocity, alpha=0.0, cmap=cm.jet)
# pyplot.contourf(plotX, plotY, mask, alpha=0.1, cmap=cm.jet)
pyplot.colorbar()
# pyplot.quiver(plotX, plotY, u, v)
# pyplot.contour(plotX, plotY, mask, cmap=cm.viridis)
# pyplot.streamplot(plotX, plotY, u, -v)
pyplot.xlabel('X'); pyplot.ylabel('Y')

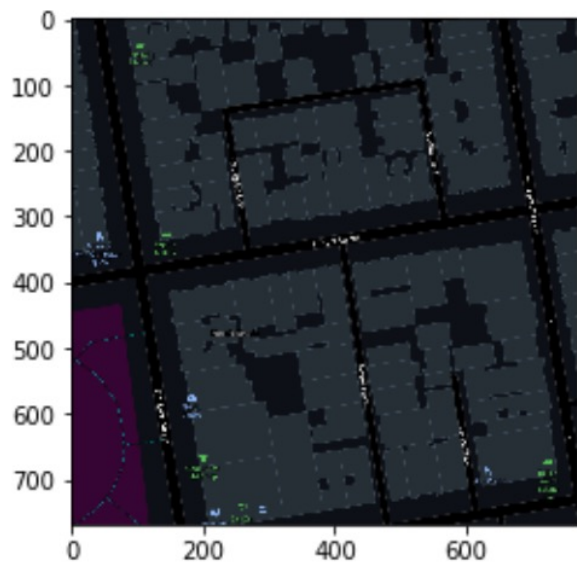
pyplot.savefig('figure3.png')
```




```
In [40]: plt.imshow(lowMask)
```



```
In [41]: cv2.imwrite('negaMap.png', negaMap)
plt.imshow(negaMap)
```



```
In [42]: # 流体計算
```

```
nit = 100
```

```
u, v, velocity = doSimulation(nx, ny, wind, lowMask, 'withoutUVMa')
```

```
#u, v, velocity, uvmeta, uvImage = doSimulation(nx, ny, wind, lowmask, 'withUVMa')
```

u:X方向、v:Y方向

```
In [44]: fig = pyplot.figure(figsize=(11, 7), dpi=300)
```

```
pyplot.imshow(mapImg)
```

```
plotX, plotY = numpy.meshgrid(numpy.linspace(0, nx*ratio, nx),
                               numpy.linspace(0, ny*ratio, ny))
```

```
pyplot.contourf(plotX, plotY, velocity, alpha=0.2, cmap=cm.jet)
```

```
# pyplot.contourf(plotX, plotY, mask, alpha=0.1, cmap=cm.jet)
```

```
pyplot.colorbar()
```

```
pyplot.quiver(plotX, plotY, u, v)
```

```
# pyplot.contour(plotX, plotY, mask, cmap=cm.viridis)
```

```
pyplot.streamplot(plotX, plotY, u, -v)
```

```
pyplot.xlabel('X'); pyplot.ylabel('Y')
```

```
pyplot.savefig('figure1.png')
```

