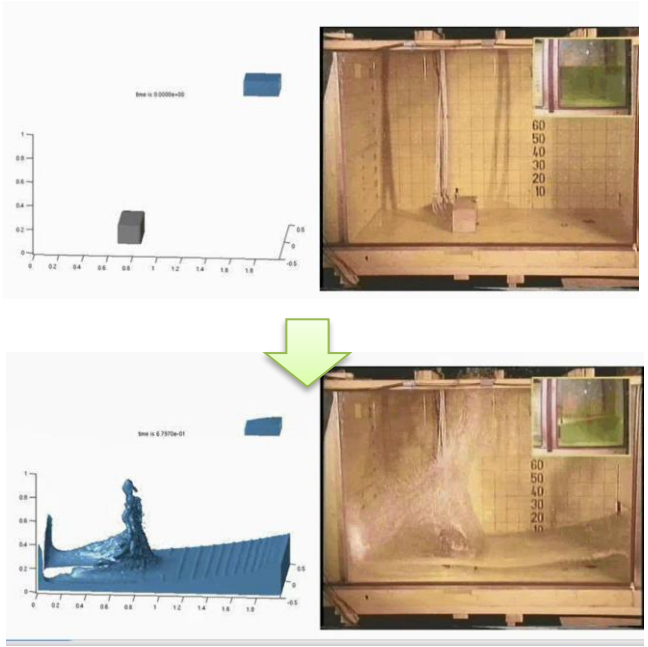
ParaView Python

2022年8月27日(土)

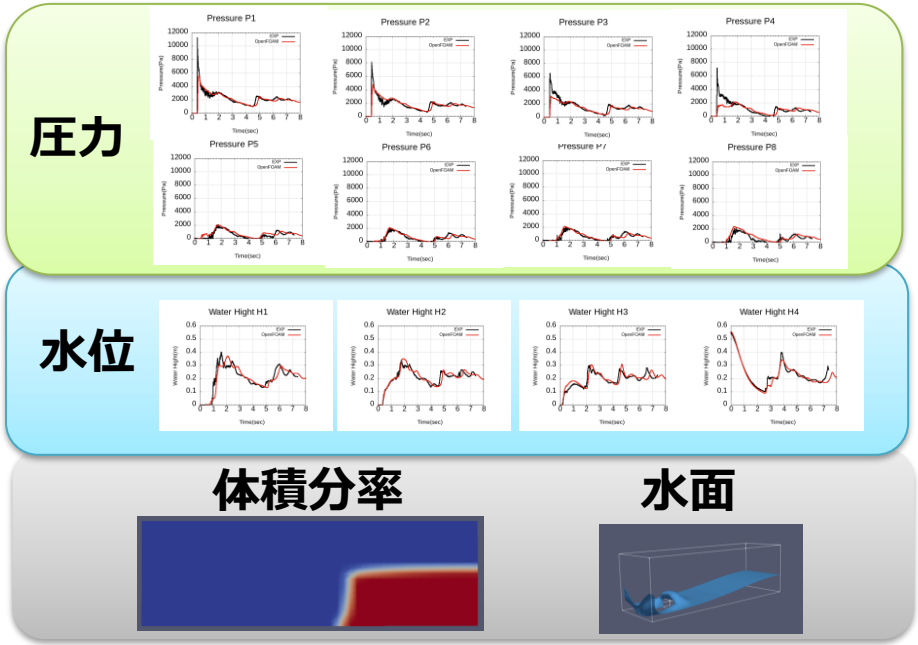
概要

- OpenFOAMによる混相流(気液二相流)の流体解析について学ぶ。
  解析対象は実験データとの結果比較のため「**3次元ダムブレイク**」を扱う。
- 結果処理の方法を学ぶ。

3次元ダムブレイク（実験）



OpenFOAMと実験データとの比較



環境

- DEXCS2021
- OpenFOAM v2106(ESI版)
- ParaView 5.7.0(pvpython python3.8.10)
- Python3.8.10

```
$コマンド
```

操作方法の説明

OpenFOAM Terminalの起動

(1)「of-v2106 Terminal」を起動

OpenFOAM Terminalが起動する

(2)フォルダの移動

cd /home/**ユーザー名**/Desktop/Track3/

```
OpenFOAMv2106 is enabled.
kamakiri@kamakiri-VirtualBox ~ $ cd '/home/kamakiri/Desktop/Track3'
```

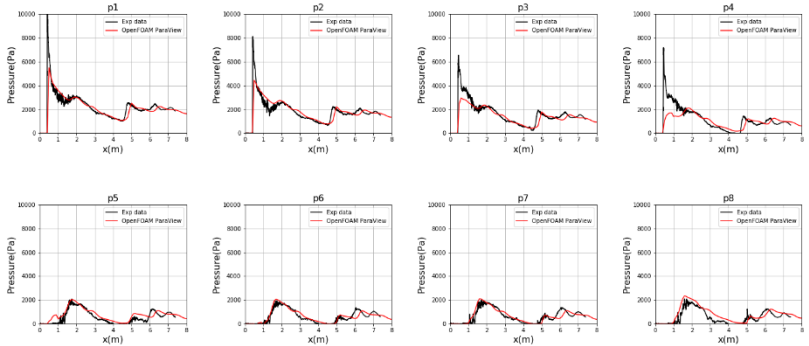
各時設定したユーザー名

圧力の時刻歴

ParaViewとPythonで自動化を行う

ParaView　指定座標の圧力のデータを出力(マクロ使用)

Python　出力したデータを結合しグラフ化する

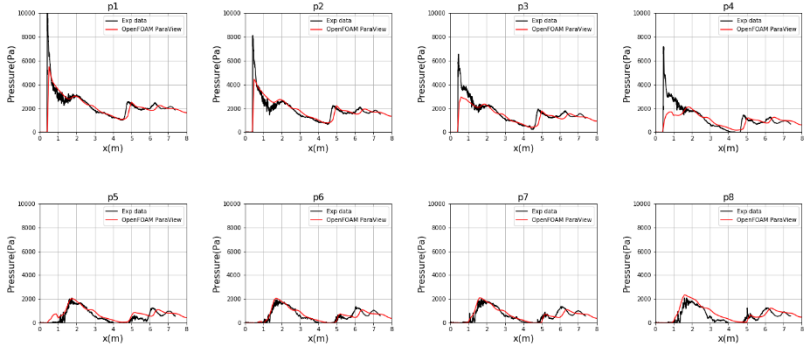


圧力の時刻歴

ParaViewとPythonで自動化を行う

ParaView　指定座標の圧力のデータを出力(マクロ使用)

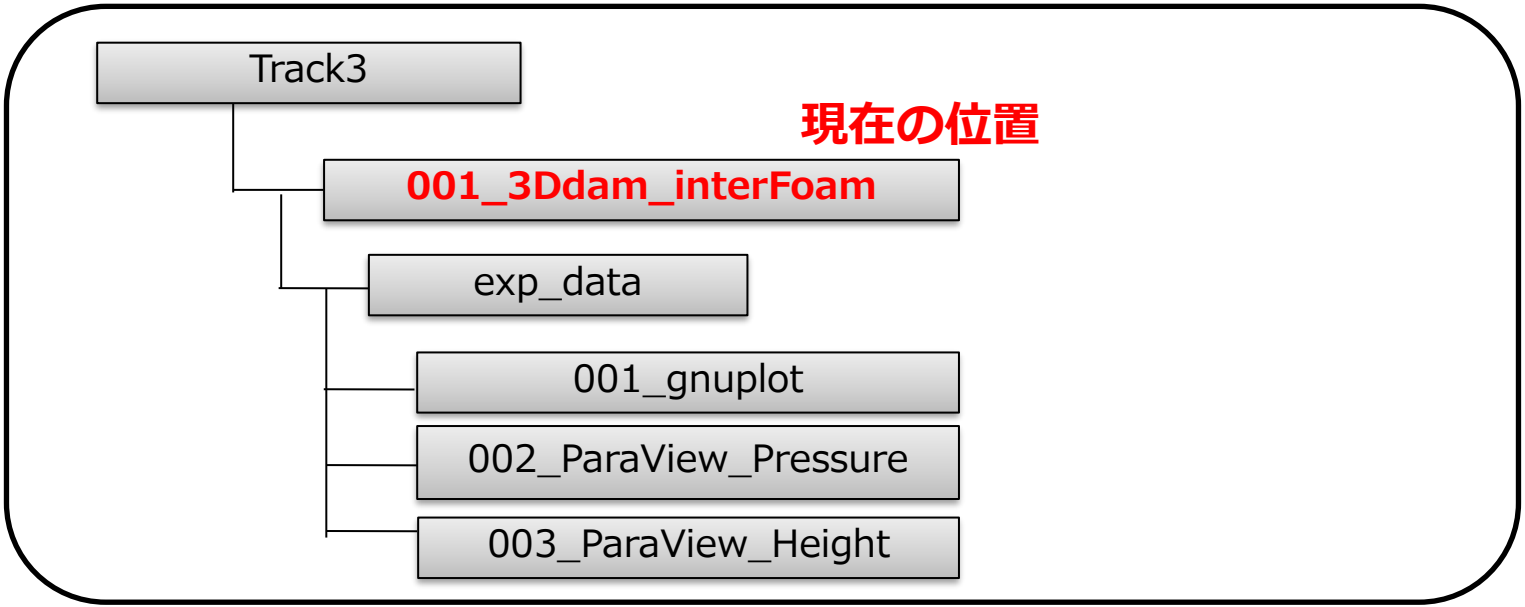
Python　出力したデータを結合しグラフ化する



以下のコマンドで「**001_3Ddam_interFoam**」フォルダに移動

フォルダを移動

```
$cd 001_3Ddam_interFoam
```

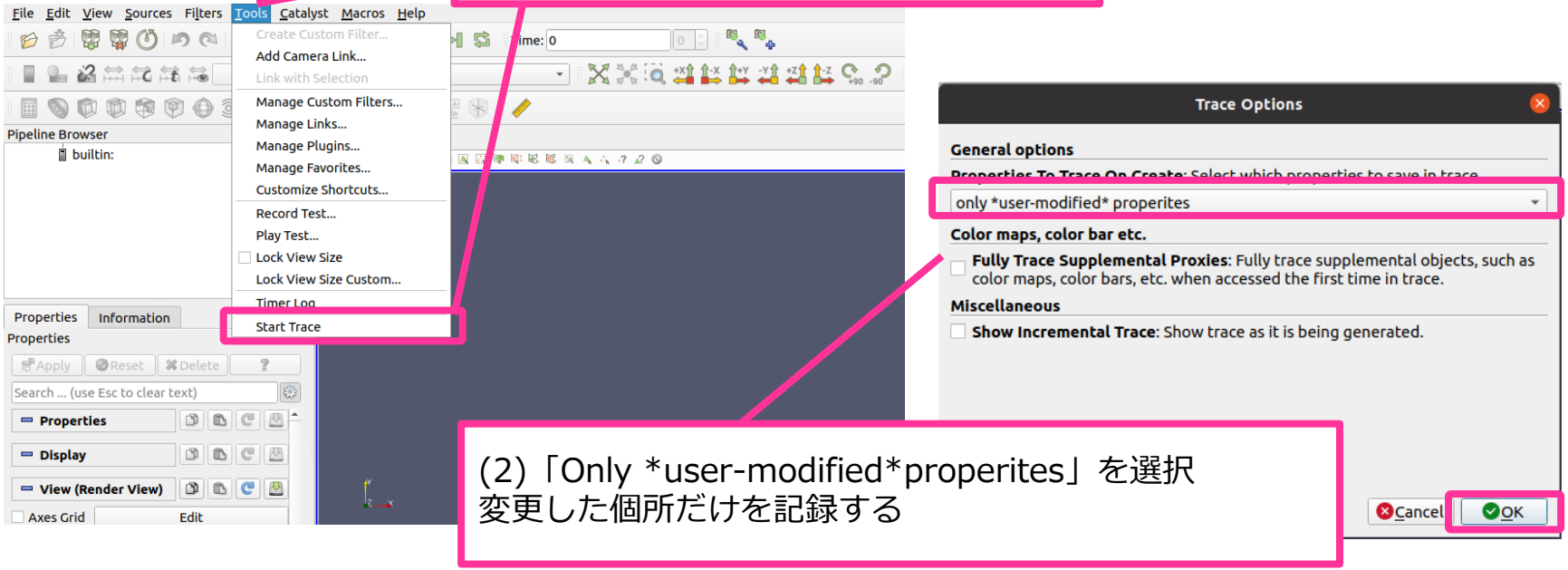


圧力の時刻歴

```
kamakiri@kamakiri-VirtualBox ~/Desktop/Track3/001_3Ddam_interFoam $ paraview
Invalid MIT-MAGIC-COOKIE-1 key
```

「paraview」コマンドで起動

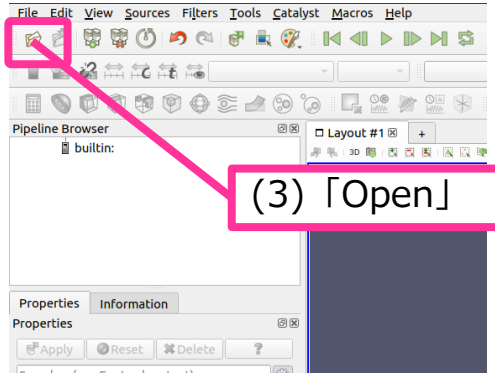
(1)「Tools」>「Start Trace」を選択



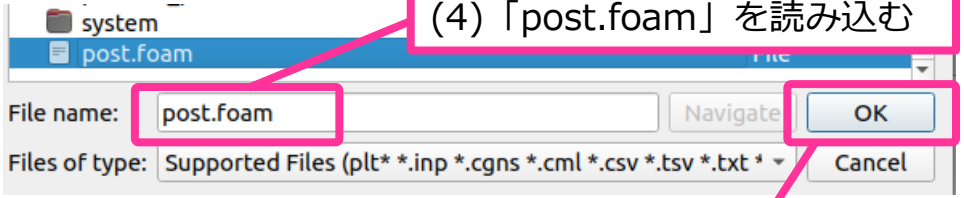
(2)「Only *user-modified*properites」を選択
変更した個所だけを記録する

これによりParaViewで行った操作がPythonスクリプトして記憶される

圧力の時刻歴

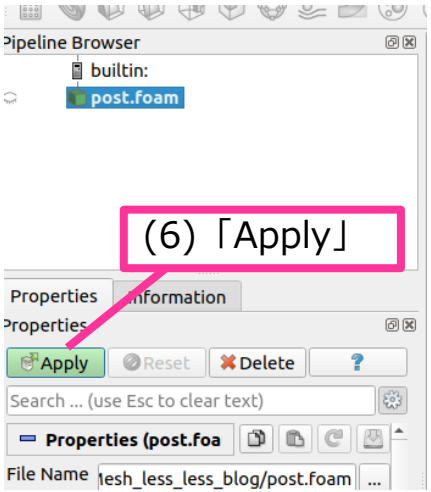


(3)「Open」

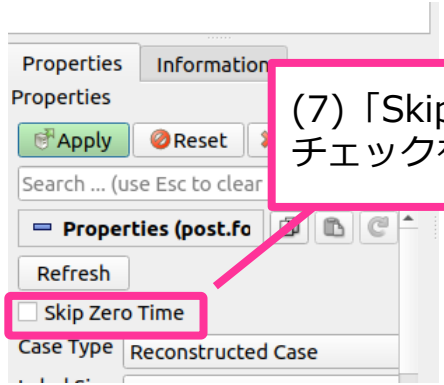


(4)「post.foam」を読み込む

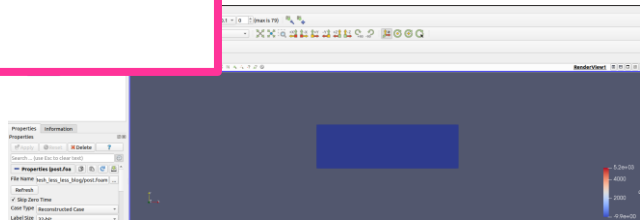
(5)「OK」



(6)「Apply」

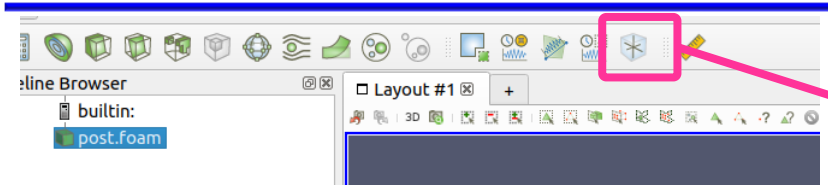


(7)「Skip Zero Time」のチェックを外す

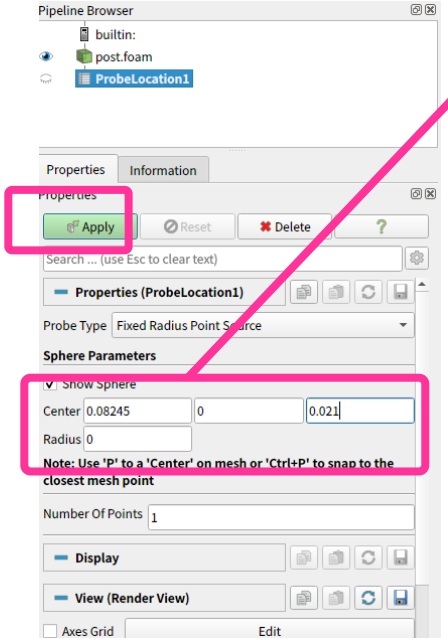


結果の読み込み完了

圧力の時刻歴



(8)「Probe Location」を選択。



(9)指定した座標(0.08245, 0 , 0.021)を入力
→「Apply」

実験での圧力測定位置

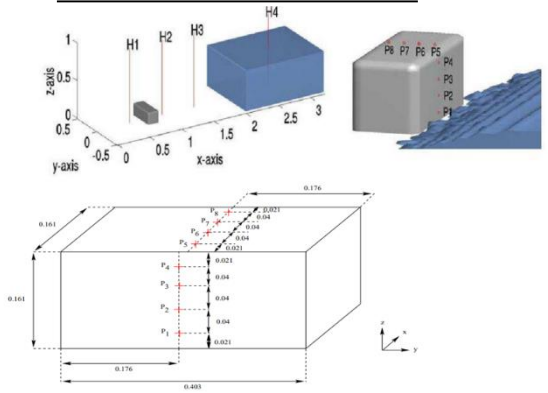
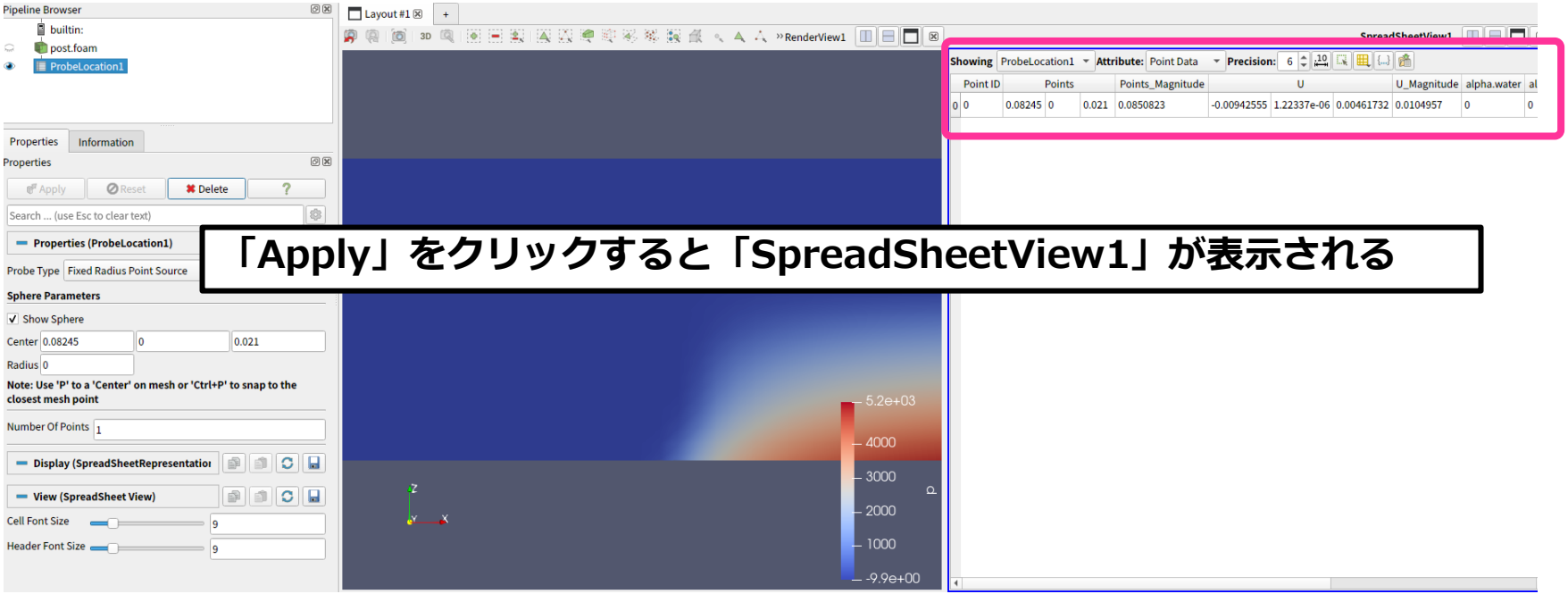


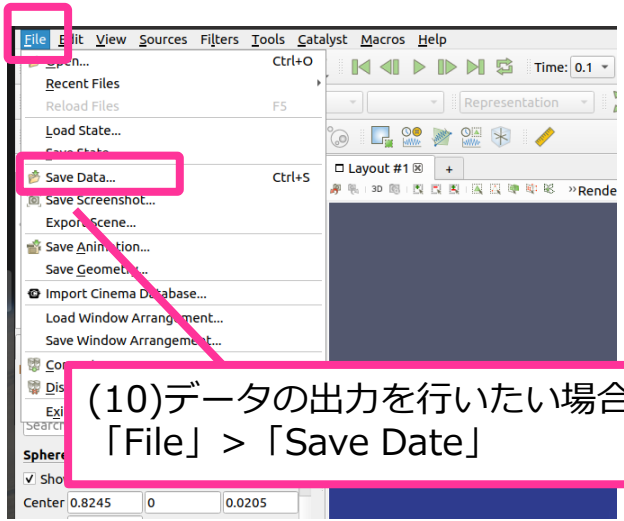
Figure 7: Description of the box.

圧力の時刻歴

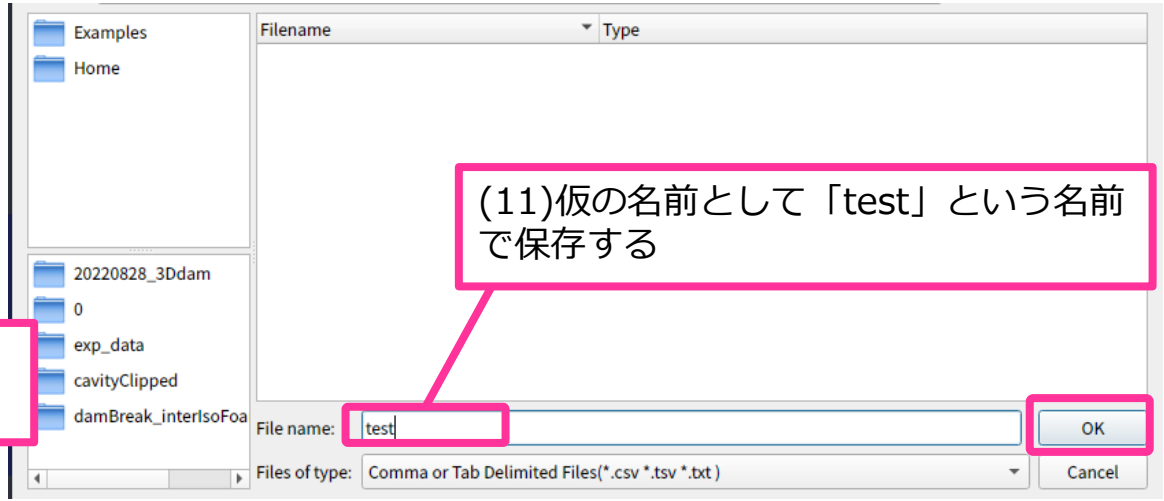

Pipeline Browser
builtin:
post.foam
ProbeLocation1
Properties
Information
Properties
Apply
Reset
Delete
Search ... (use Esc to clear text)
Properties (ProbeLocation1)
Probe Type
Fixed Radius Point Source
Sphere Parameters
Show Sphere
Center
0.08245
0
0.021
Radius
0
Note: Use 'P' to a 'Center' on mesh or 'Ctrl+P' to snap to the closest mesh point
Number Of Points
1
Display (SpreadSheetRepresentatio
View (SpreadSheet View)
Cell Font Size
9
Header Font Size
9
Layout #1
RenderView1
SpreadSheetView1
Showing
ProbeLocation1
Attribute:
Point Data
Precision:
6
Point ID
Points
Points_Magnitude
U
U_Magnitude
alpha.water
0
0.08245
0
0.021
0.0850823
-0.00942555
1.22337e-06
0.00461732
0.0104957
0
0
5.2e+03
4000
3000
2000
1000
-9.9e+00
「Apply」をクリックすると「SpreadSheetView1」が表示される

圧力の時刻歴

exp_data/002_ParaView_Pressure

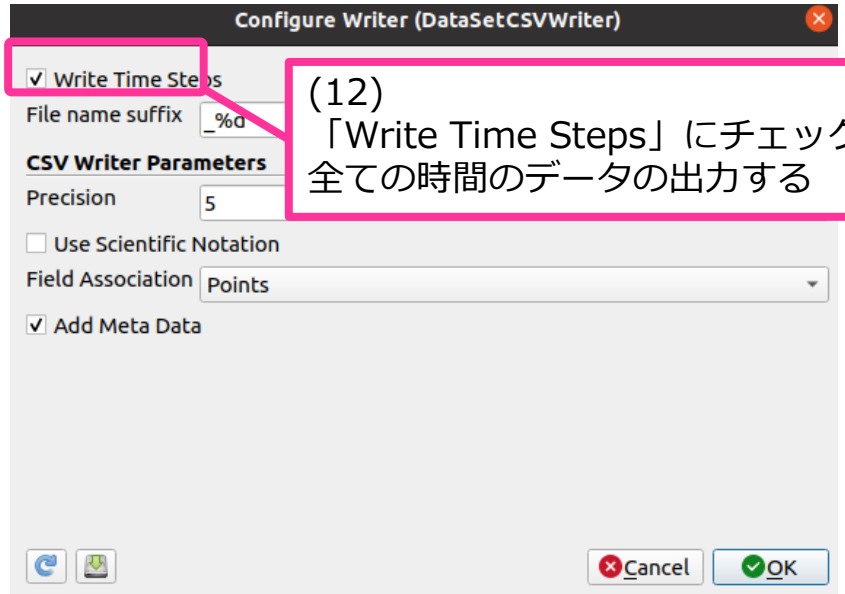


(10)データの出力を行いたい場合は「File」>「Save Date」

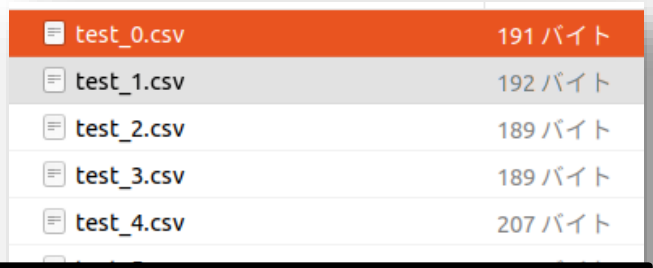


(11)仮の名前として「test」という名前で保存する

圧力の時刻歴



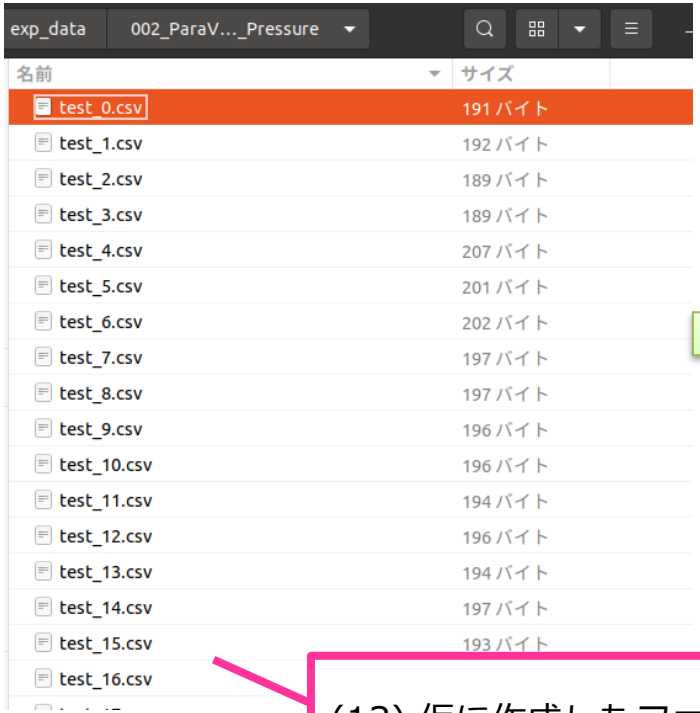
(12)
「Write Time Steps」にチェックを入れる。
全ての時間のデータの出力する

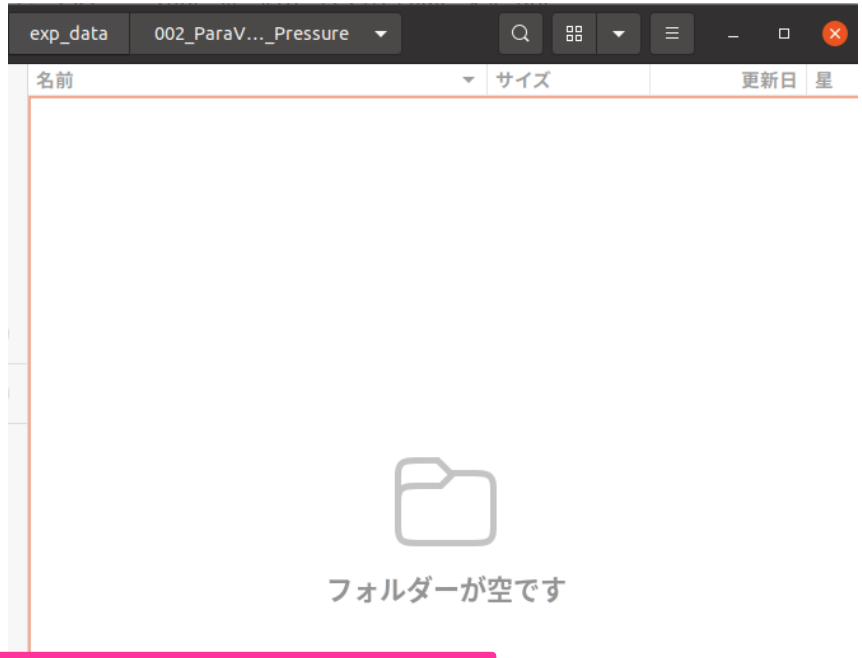


「test_%d.csv」という名前でファイルができる。
※%d：ステップ数
※時間ではなくステップ数なので注意
今は0.1秒ずつ出力しているので1/10すると時間になる。

圧力の時刻歴

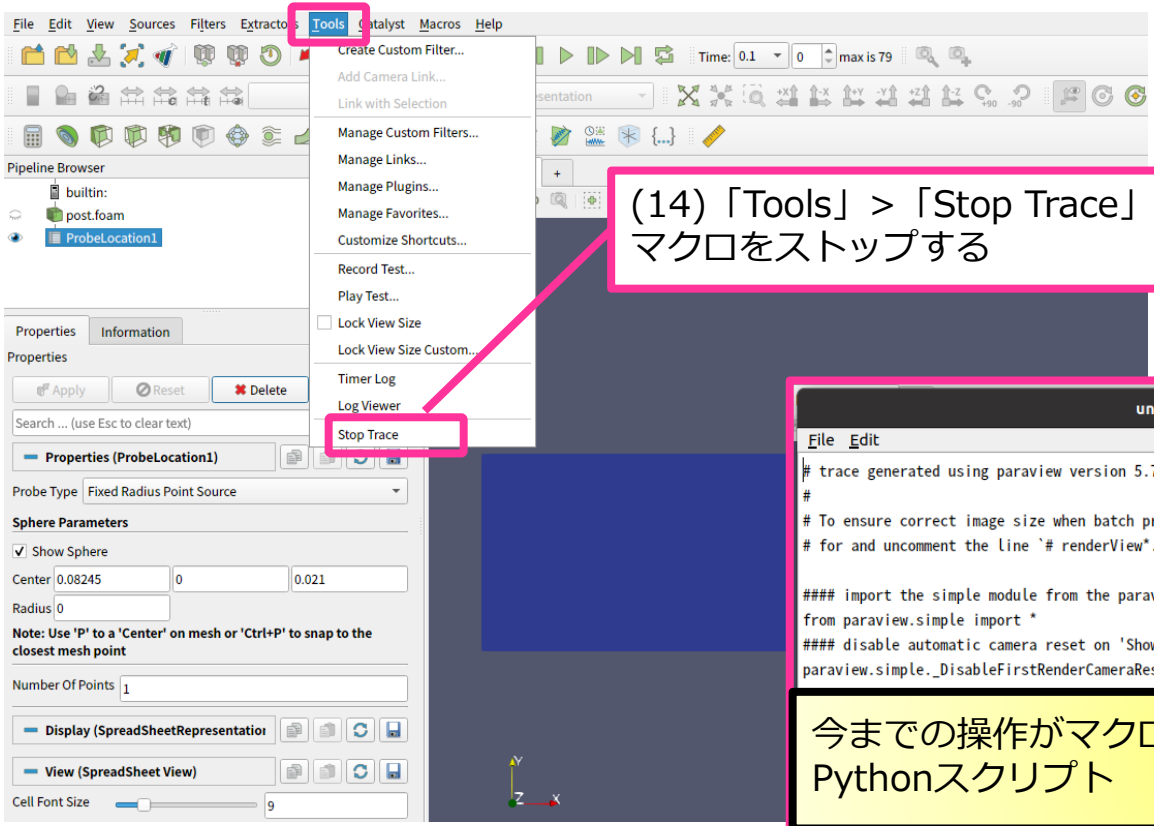
exp_data/002_ParaView_Pressure



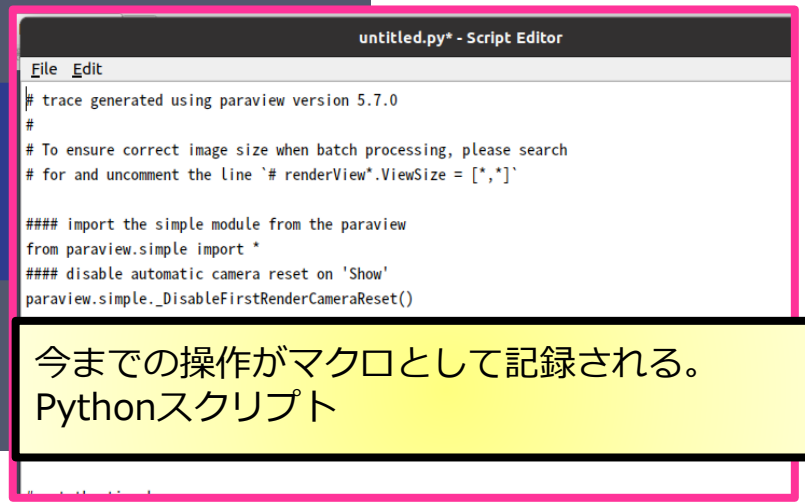


(13) 仮に作成したファイルなので全て削除する。

圧力の時刻歴



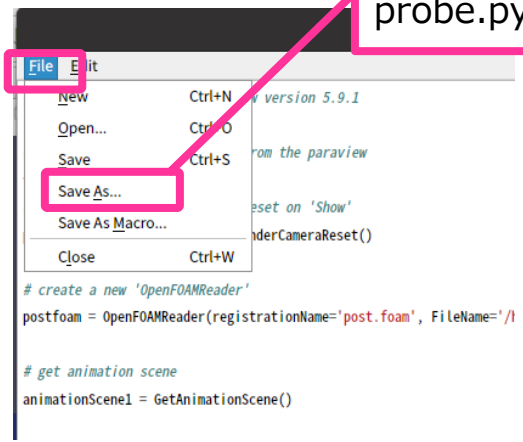
(14)「Tools」>「Stop Trace」でPython マクロをストップする



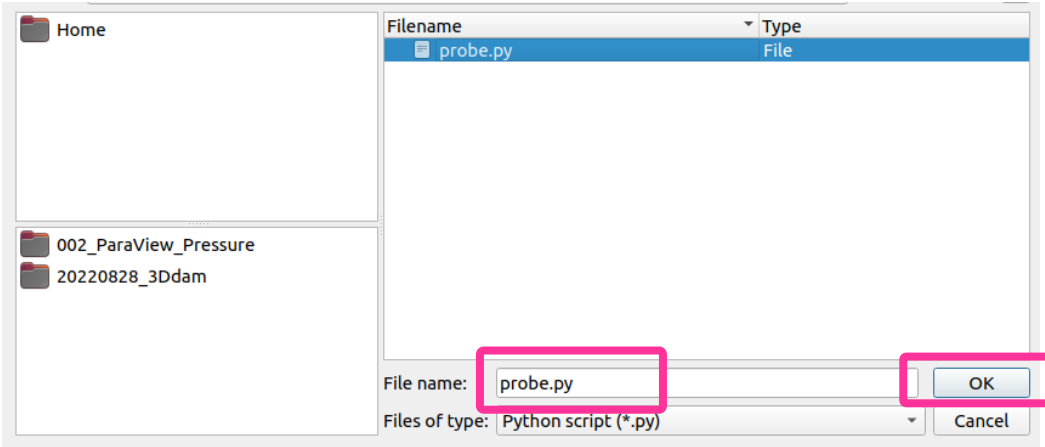
今までの操作がマクロとして記録される。
Pythonスクリプト

圧力の時刻歴

(15)「File」>「Save As...」
probe.pyという名前で保存



exp_data/002_ParaView_Pressure

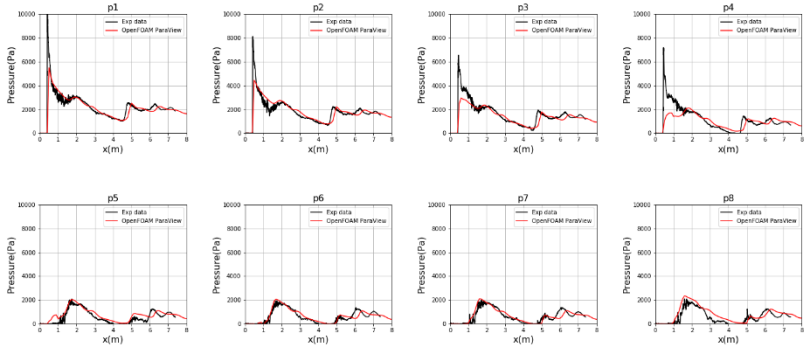


圧力の時刻歴

ParaViewとPythonで自動化を行う

ParaView　指定座標の圧力のデータを出力(マクロ使用)

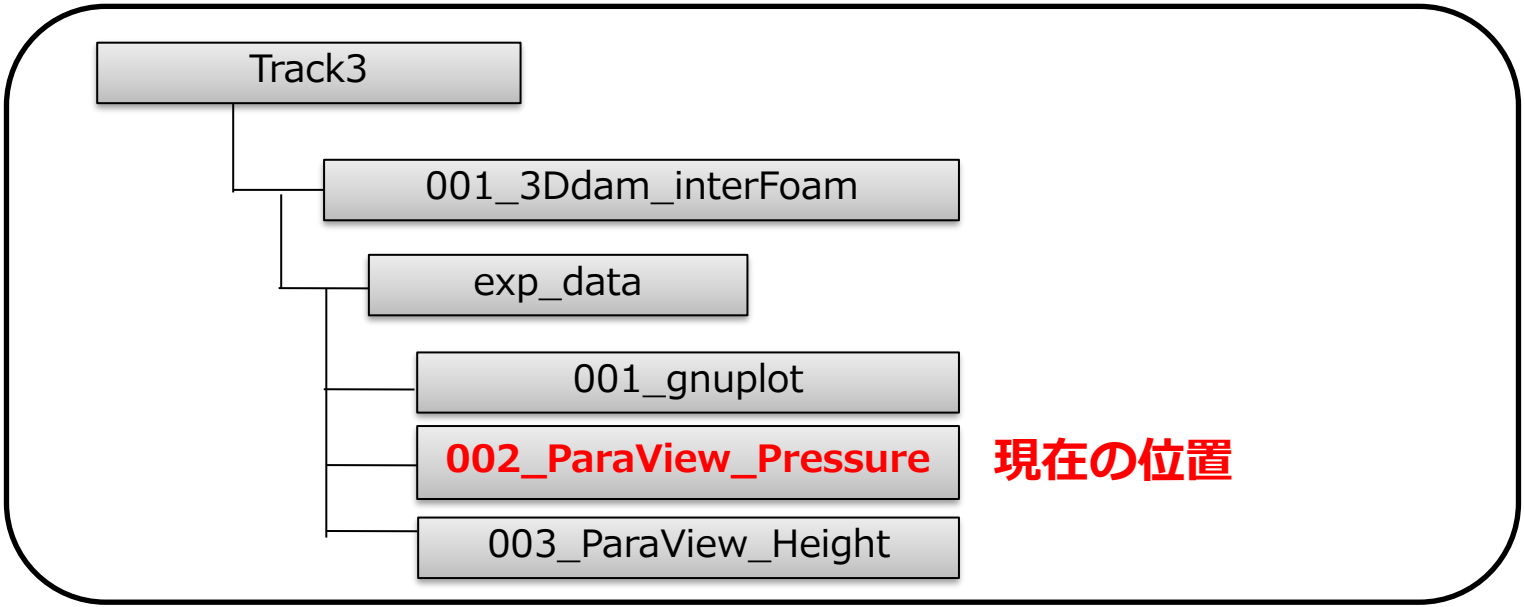
Python　出力したデータを結合しグラフ化する



以下のコマンドで「**002_ParaView_Pressure**」フォルダに移動

フォルダを移動

```
$cd exp_data/002_ParaView_Pressure
```



pandasライブラリをインストール

$pip3 install pandas

```
kamakiri@kamakiri-VirtualBox ~/Desktop/Track3/001_3Ddam_interFoam/exp_data/003_ParaView_Height $ pip3 install pandas
Defaulting to user installation because normal site-packages is not writeable
Collecting pandas
  Using cached pandas-1.4.3-cp38-cp38-manylinux_2_17_x86_64.manylinux2014_x86_64.whl (11.7 MB)
Collecting numpy>=1.18.5
  Downloading numpy-1.23.2-cp38-cp38-manylinux_2_17_x86_64.manylinux2014_x86_64.whl (17.1 MB)
     ━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━ 17.1/17.1 MB 3.2 MB/s eta 0:00:00
Requirement already satisfied: pytz>=2020.1 in /home/kamakiri/.local/lib/python3.8/site-packages (from pandas) (2022.1)
Requirement already satisfied: python-dateutil>=2.8.1 in /home/kamakiri/.local/lib/python3.8/site-packages (from pandas) (2.8.2)
Requirement already satisfied: six>=1.5 in /usr/lib/python3/dist-packages (from python-dateutil>=2.8.1->pandas) (1.14.0)
Installing collected packages: numpy, pandas

Successfully installed numpy-1.23.2 pandas-1.4.3
```

https://pypi.org/project/pandas/

圧力の時刻歴

exp_data/002_ParaView_Pressure/probe.py

```
46 # get opacity transfer function/opacity map for 'p'
47 pPWF = GetOpacityTransferFunction('p')
48
49 # create a new 'Probe Location'
50 probeLocation1 = ProbeLocation(Input=postfoam,
51     ProbeType='Fixed Radius Point Source')
52
53 # Properties modified on probeLocation1.ProbeType
54 probeLocation1.ProbeType.Center = [0.08245, 0, 0.021]
55
56 # create a new 'SpreadSheet View'
57 spreadSheetView1 = CreateView('SpreadSheetView')
58 spreadSheetView1.ColumnToSort = ''
59 spreadSheetView1.BlockSize = 1024
60 # uncomment following to set a specific view size
61 # spreadSheetView1.ViewSize = [400, 400]
62
63 # show data in view
64 probeLocation1Display = Show(probeLocation1, spreadSheetView1)
```

コピー用

コピーして使ってください

```
# Properties modified on probeLocation1.ProbeType
probeLocation1.ProbeType.Center = [0.08245, 0, 0.021]
```

(1)圧力計測座標を指定
Pythonのリスト型として定義

圧力の時刻歴

(2)以下のコマンドでprobe.pyを実行

```
$pvpython probe.py
```

もしくは

```
$paraview --script=probe.py
```

exp_data/002_ParaView_Pressure

自動でcsvファイルが作成されていれば成功

圧力の時刻歴

pythonスクリプトを変更してP1~P8のデータを作成する

(1) probe.pyをprobe_Alltime.pyという名前でコピーする

exp_data/002_ParaView_Pressure/probe.py

exp_data/002_ParaView_Pressure/**probe_Alltime.py**

圧力の時刻歴

(2)ファイル読み込みを変数にする

変更前

```
10
11 # create a new 'OpenFOAMReader'
12 postfoam = OpenFOAMReader(FileName='/home/kamakiri/Desktop/Track3/001_3Ddam_interFoam/post.foam')
13
```

変更後

```
 6 #### import the simple module from the paraview
 7 from paraview.simple import *
 9 import os
10
11 # 現在のパス
12 PWD = os.getcwd()
13
14 paraview.simple._DisableFirstRenderCameraReset()
15
16 # create a new 'OpenFOAMReader'
17 postfoam = OpenFOAMReader(FileName=f'{PWD}/../../post.foam')
```

os操作のライブラリをインポート（標準ライブラリ）
現在のパスをPWD変数にする

f(フォーマット文字列)を付けてとすることで変数を文字列内に埋め込む

コピー用

コピーして使ってください

```
import os

# 現在のパス
PWD = os.getcwd()

paraview.simple._DisableFirstRenderCameraReset()

# create a new 'OpenFOAMReader'
postfoam = OpenFOAMReader(FileName=f'{PWD}/../../post.foam')
```

圧力の時刻歴

exp_data/002_ParaView_Pressure/probe_Alltime.py

```
 8 #### disable automatic camera reset on 'Show'
 9 import os
10
11 # 現在のパス
12 PWD = os.getcwd()
13
14 paraview.simple._DisableFirstRenderCameraReset()
15
16 # create a new 'OpenFOAMReader'
17 postfoam = OpenFOAMReader(FileName=f'{PWD}/../../post.foam')
18
19 probe_points = {
20         'p1':[0.8245001, 0.0, 0.0205],
21         'p2':[0.8245001, 0.0, 0.0605],
22         'p3':[0.8245001, 0.0, 0.1005],
23         'p4':[0.8245001, 0.0, 0.1405],
24         'p5':[0.8040   , 0.0, 0.161],
25         'p6':[0.7640   , 0.0, 0.161],
26         'p7':[0.7240   , 0.0, 0.161],
27         'p8':[0.6840, 0.0, 0.161]
28        }
```

(3)圧力計測点
辞書型で定義

コピー用

コピーして使ってください

```
probe_points = {
                'p1':[0.8245, 0.0, 0.021],
                'p2':[0.8245, 0.0, 0.061],
                'p3':[0.8245, 0.0, 0.101],
                'p4':[0.8245, 0.0, 0.141],
                'p5':[0.8035, 0.0, 0.162],
                'p6':[0.7635, 0.0, 0.162],
                'p7':[0.7235, 0.0, 0.162],
                'p8':[0.6835, 0.0, 0.162]
                }
```

圧力の時刻歴

(4)繰り返し制御構文に変更

```
for point in probe_points:

    #set probe points
    probe = probe_points[point]

    # get animation scene
    animationScene1 = GetAnimationScene()

    # get the time-keeper
    timeKeeper1 = GetTimeKeeper()

    # update animation scene based on data timesteps
    animationScene1.UpdateAnimationUsingDataTimeSteps()

    # get active view
    renderView1 = GetActiveViewOrCreate('RenderView')
    # uncomment following to set a specific view size
    # renderView1.ViewSize = [1546, 682]

    # show data in view
    postfoamDisplay = Show(postfoam, renderView1)

    # trace defaults for the display properties.
    postfoamDisplay.Representation = 'Surface'

    # reset view to fit data
    renderView1.ResetCamera()
```

インデント（タブ）

以下全てインデントをつける

圧力の時刻歴

(5)圧力計測点を変数probeに変更

変更前

```
63        # create a new 'Probe Location'
64        probeLocation1 = ProbeLocation(Input=postfoam,
65            ProbeType='Fixed Radius Point Source')
66
67        # Properties modified on probeLocation1.ProbeType
68        probeLocation1.ProbeType.Center = [0.08245, 0, 0.021]
69
```

変更後

```
68
69        # create a new 'Probe Location'
70        probeLocation1 = ProbeLocation(Input=postfoam,
71            ProbeType='Fixed Radius Point Source')
72
73        # Properties modified on probeLocation1.ProbeType
74        probeLocation1.ProbeType.Center = probe
75
76        # Properties modified on postfoam
77        postfoam.CellArrays = ['U', 'alpha.water', 'alpha.water_
78
```

コピー用

コピーして使ってください

```
# Properties modified on probeLocation1.ProbeType
probeLocation1.ProbeType.Center = probe
```

圧力の時刻歴

(6)ファイル名の変更

変更前

```
80
81 # save data
82 SaveData('/home/kamakiri/Desktop/Track3/001_3Ddam_interFoam/exp_data/002_ParaView_Pressure/test.csv', proxy=probeLocation1, WriteTimeSteps=1)
83
```

変更後

```
100        AssignViewToLayout(view=spreadSheetView1, layout=layout1, hint=0)
101
102        # save data
103        SaveData(f'{PWD}/{point}.csv', proxy=probeLocation1, WriteTimeSteps=1)
104
105        #### saving camera placements for all active views
106
```

コピー用

コピーして使ってください

```
# save data
SaveData(f'{PWD}/{point}.csv', proxy=probeLocation1, WriteTimeSteps=1)
```

圧力の時刻歴

(7)以下のコマンドでprobe_Alltime.pyを実行

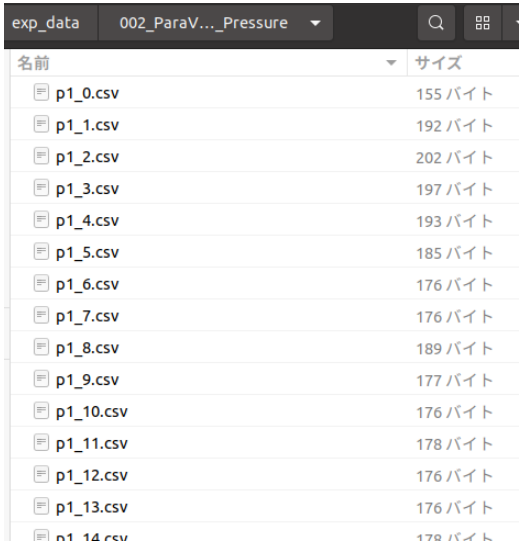
```
$pvpython probe_Alltime.py
```

exp_data/002_ParaView_Pressure

自動でcsvファイルが作成されていれば成功

圧力の時刻歴

exp_data/002_ParaView_Pressure



exp_data/002_ParaView_Pressure/p1_0.csv

A	B	C	D	E	F	G	H	I	J	K	L
U:0	U:1	U:2	alpha.water	k	nut	omega	p_rgh	vtkValidPointMask	Points:0	Points:1	Points:2
0	0	0	0	0.00015	5E-07	2	0	1	0.8245	0	0.0205

0ステップ目には圧力pのデータがない

exp_data/002_ParaView_Pressure/p1_1.csv

	A	B	C	D	E	F	G	H	I	J
1	U:0	U:1	U:2	alpha.water	p	p_rgh	vtkValidPointMask	Points:0	Points:1	Points:2
2	-0.038444	-1.331E-05	0.034272	-4.6243E-44	0.94821	1.235	1	0.8245	0	0.0205
3										
4										

ファイル名が「測定ポイント_ステップ数.csv」となっている

圧力の時刻歴

Pythonでデータをつなげる
1. 各ステップの圧力データをつなげる
2. 各測定箇所で「1」を実行
3. 「2」までをグラフ化する
4. グラフを画像データとして保存

(8)以下のコマンドでmain.pyを実行

$python3 mian.py

```
kamakiri@kamakiri-VirtualBox ~/Desktop/Track3/001_3Ddam_interFoam/exp_data/002_ParaView_Pressure $ python3 main.py
```

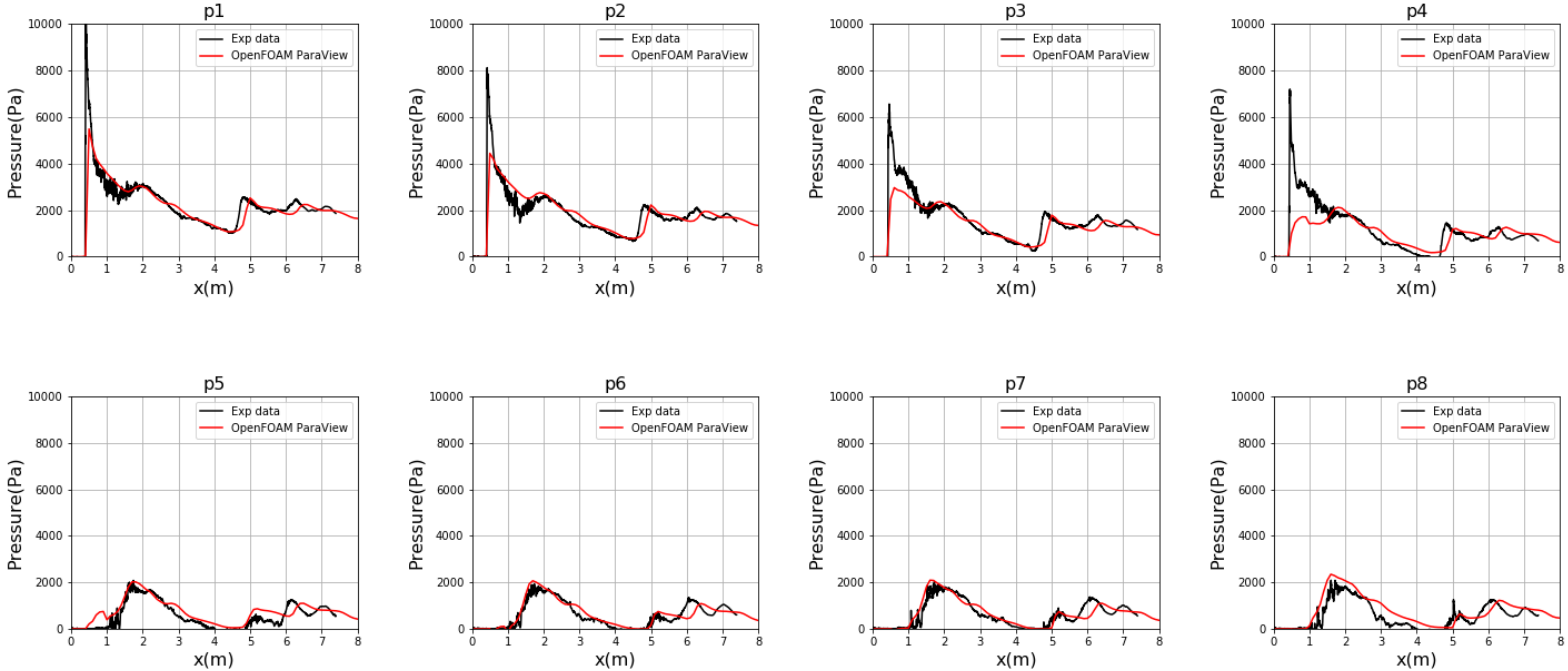
exp_data/002_ParaView_Pressure/main.py

```
import pandas as pd
import glob
import re
import matplotlib.pyplot as plt
import subprocess


# ファイル名の昇順
def atoi(text):
    return int(text) if text.isdigit() else text

def natural_keys(text):
    return [ atoi(c) for c in re.split(r'(¥d+)', text) ]

# P1~P8のグラフ化
def probe_func(i, point):
    list_csv = glob.glob(f'{point}*.csv')
    list_csv_sort = sorted(list_csv, key=natural_keys)

    df = pd.DataFrame()
    for csv_file in list_csv_sort:
        df_ = pd.read_csv(csv_file)
        df_['time'] = float(re.findall(r"¥d+", csv_file)[1])/10.0
        df = pd.concat([df,df_])

    df['p'] = df['p'].fillna(0.0)

    ax = fig.add_subplot(2, 4, i+1)
    ax.plot(df_exp['Time (s)'].to_numpy(), df_exp[f'{point.upper()} (Pa)'].to_numpy(), color='black',label='Exp data')
    ax.plot(df['time'].to_numpy(), df['p'].to_numpy(), color='red', label='OpenFOAM ParaView')
    ax.set_xlabel('x(m)',fontsize=16)
    ax.set_ylabel('Pressure(Pa)',fontsize=16)
    ax.set_title(f'{point}',fontsize=16)
    ax.set_xlim(0, 8)
    ax.set_ylim(0,10000)
    ax.grid()
    ax.legend()

if __name__ == '__main__':

    # 指定した座標
    probe_points = {
    'p1':[0.8245001, 0.0, 0.0205],
    'p2':[0.8245001, 0.0, 0.0605],
    'p3':[0.8245001, 0.0, 0.1005],
    'p4':[0.8245001, 0.0, 0.1405],
    'p5':[0.8040   , 0.0, 0.161],
    'p6':[0.7640   , 0.0, 0.161],
    'p7':[0.7240   , 0.0, 0.161],
    'p8':[0.6840, 0.0, 0.161]
    }

    df_exp = pd.read_csv('../../exp_data/test_case_2_exp_data.csv',sep='¥t' )

    #subprocess.run(['pvpython','probe_Alltime.py'])
    fig = plt.figure(figsize=(24,10))
    plt.subplots_adjust(wspace=0.4, hspace=0.6)

    for i, point in enumerate(probe_points):
        probe_func(i, point)

    fig.savefig("point.png")
```

圧力の時刻歴

実験データとOpenFOAMの比較

exp_data/002_ParaView_Pressure/point.png

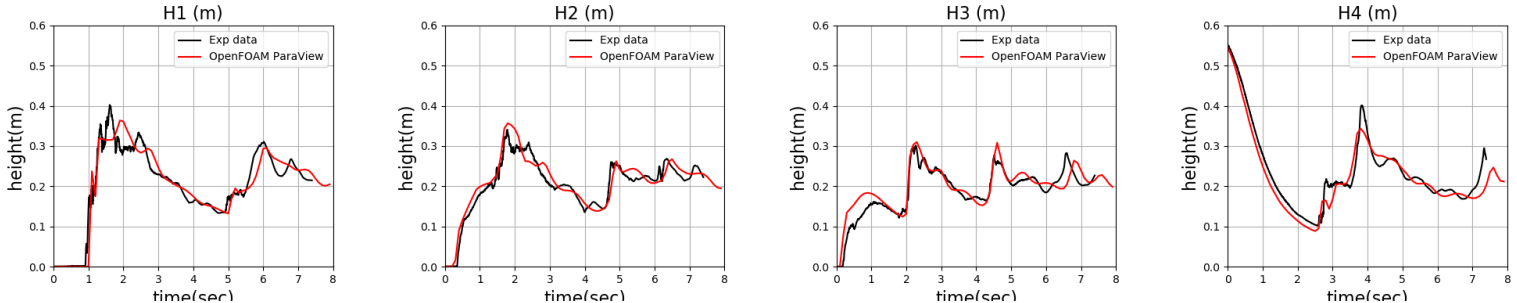


水位の時刻歴

ParaViewとPythonで自動化を行う

ParaView　指定座標の圧力のデータを出力(マクロ使用)

Python　出力したデータを結合しグラフ化する

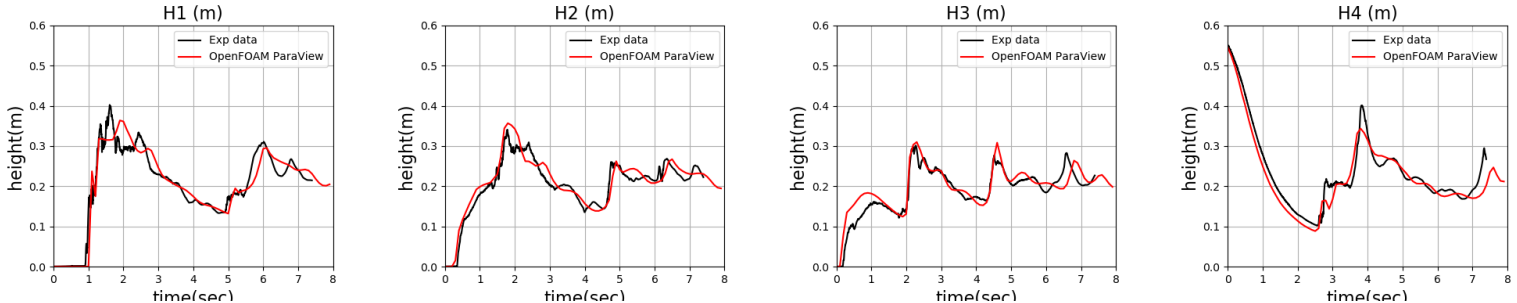


水位の時刻歴

ParaViewとPythonで自動化を行う

ParaView　指定座標の圧力のデータを出力(マクロ使用)

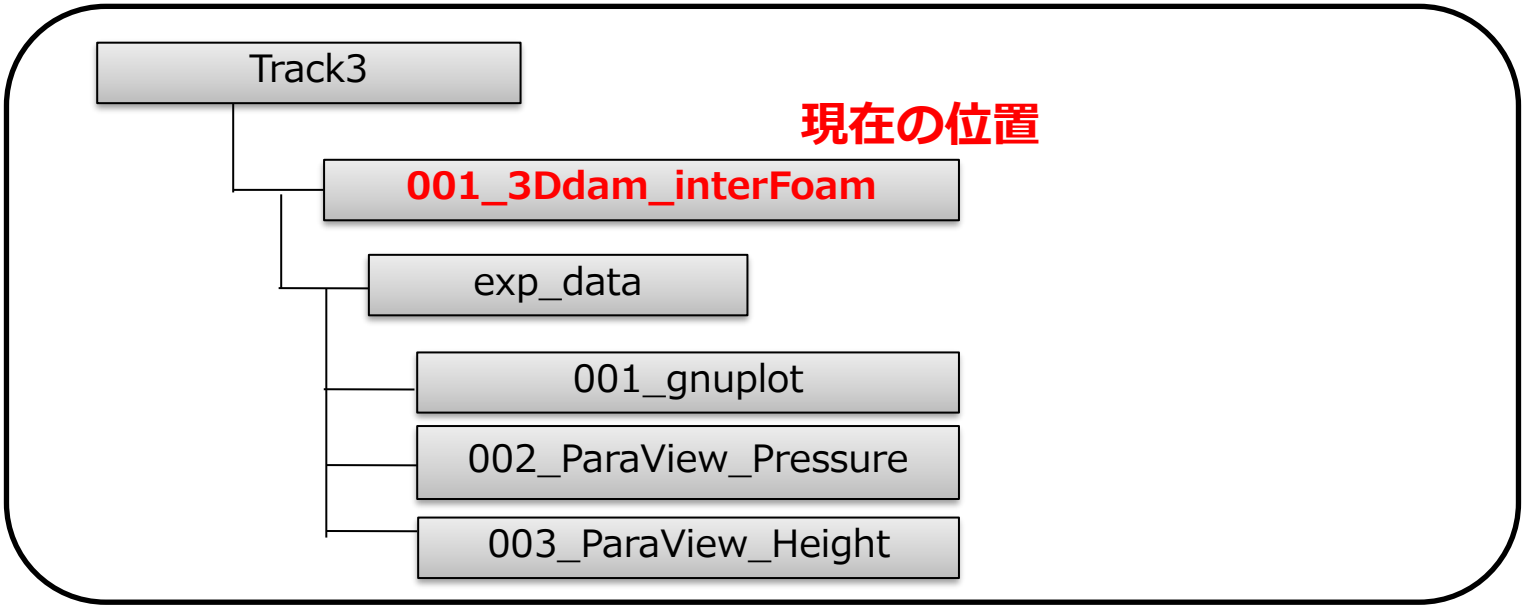
Python　出力したデータを結合しグラフ化する



以下のコマンドで「**001_3Ddam_interFoam**」フォルダに移動

フォルダを移動

```
$cd ../../
```

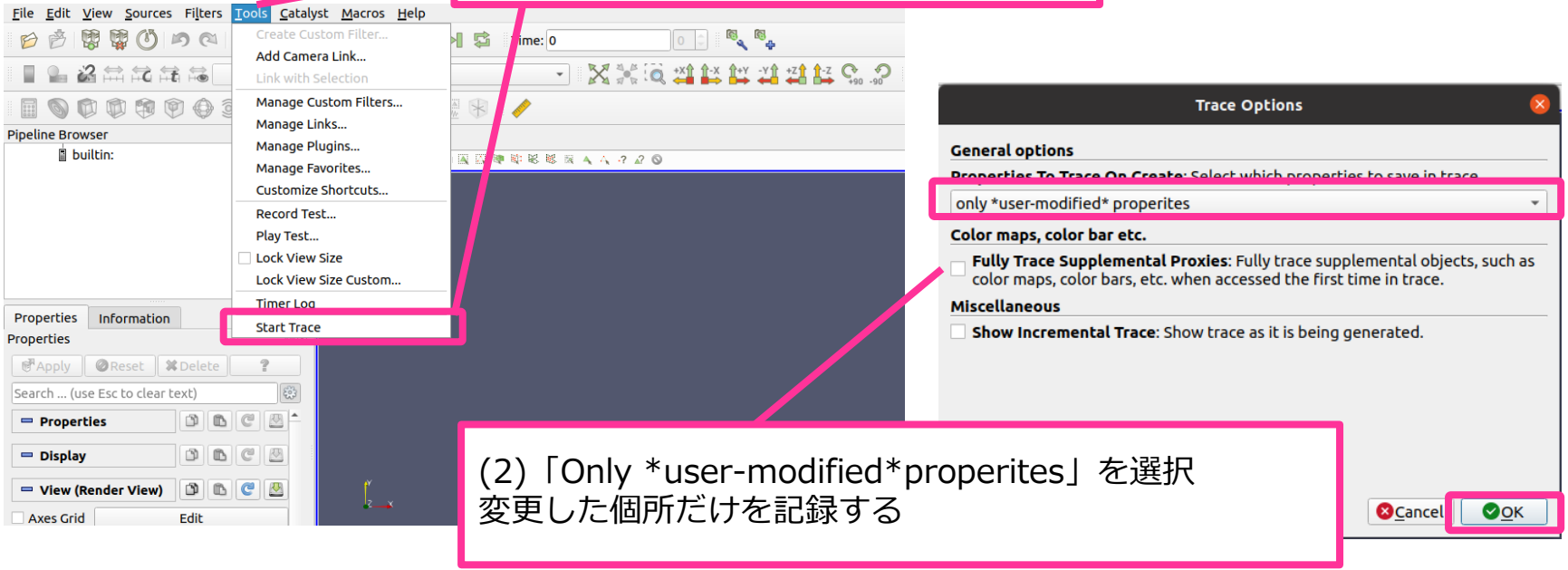


水位の時刻歴

```
kamakiri@kamakiri-VirtualBox ~/Desktop/Track3/001_3Ddam_interFoam $ paraview
Invalid MIT-MAGIC-COOKIE-1 key
```

「paraview」コマンドで起動

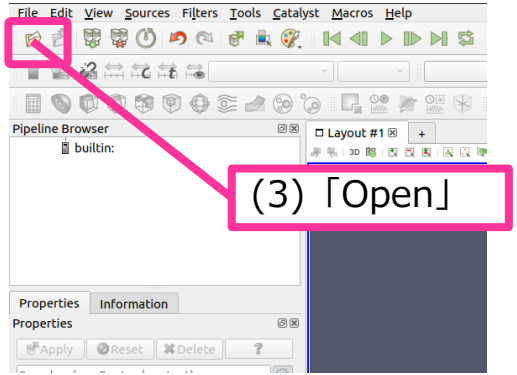
(1)「Tools」>「Start Trace」を選択



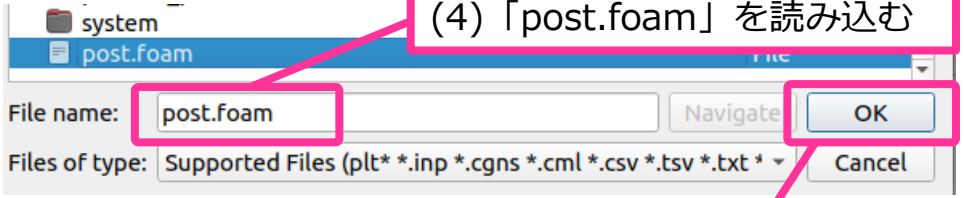
(2)「Only *user-modified*properites」を選択
変更した個所だけを記録する

これによりParaViewで行った操作がPythonスクリプトして記憶される

水位の時刻歴

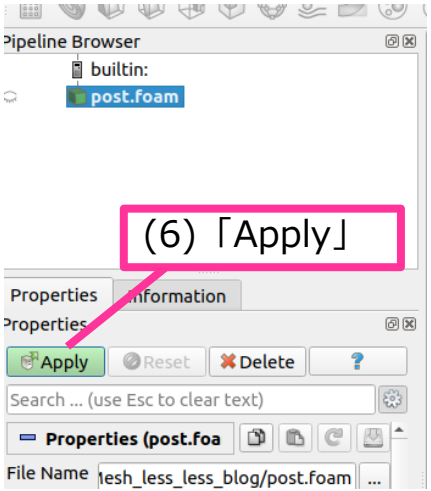


(3)「Open」

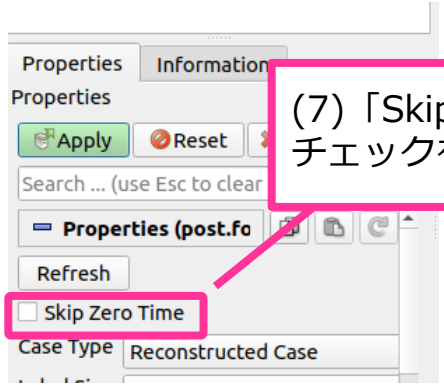


(4)「post.foam」を読み込む

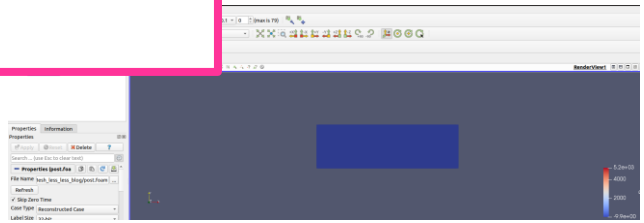
(5)「OK」



(6)「Apply」

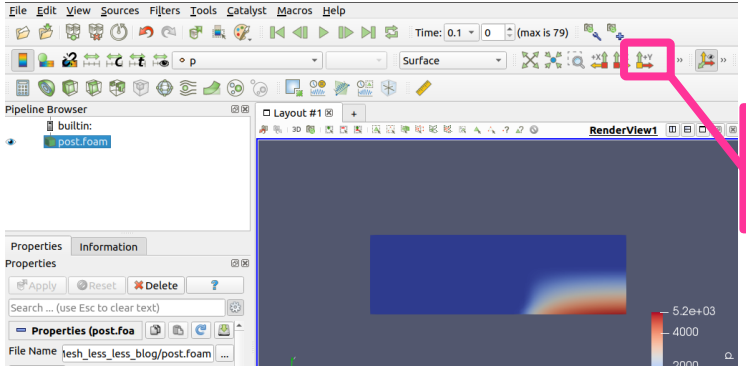


(7)「Skip Zero Time」のチェックを外す

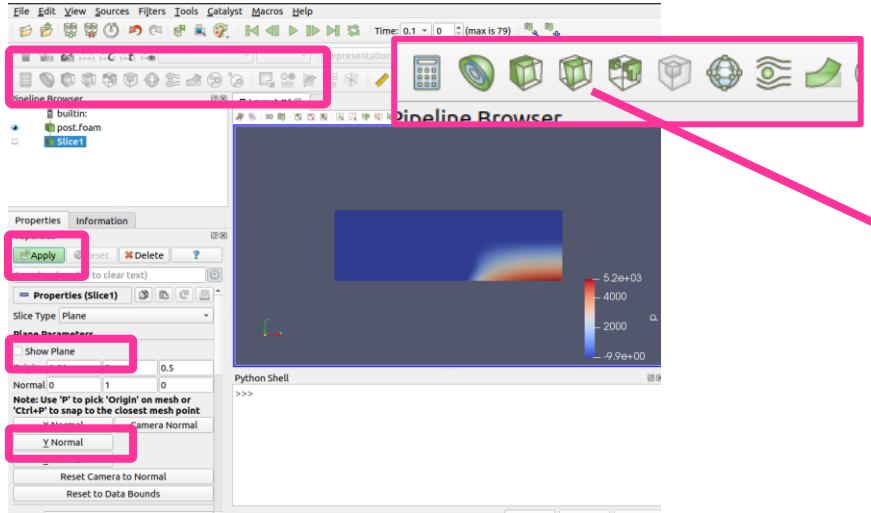


結果の読み込み完了

水位の時刻歴



(8)「Set view direction to +Y」を選択。
X-z面からのviewに変更

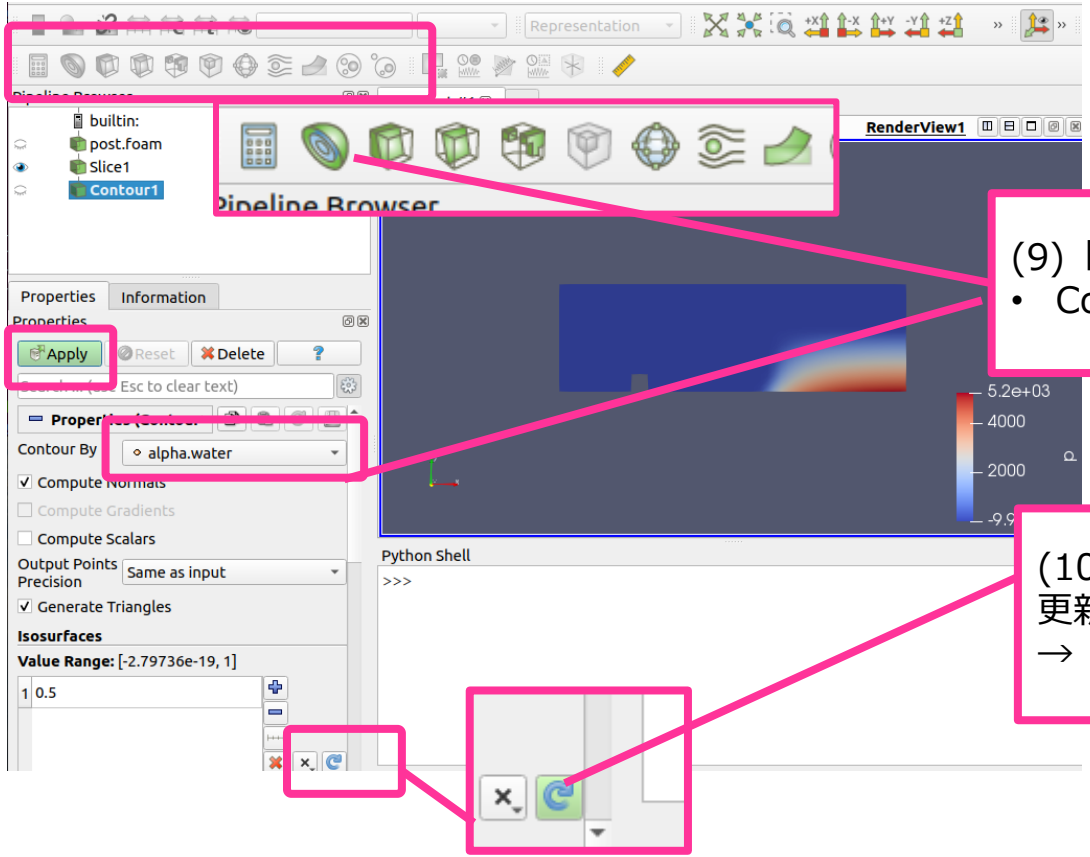


x-z面でy=0の断面を表示

(9)「Slice」を選択
- Show Planeのチェックを外す
- Y Normalを選択

→「Apply」

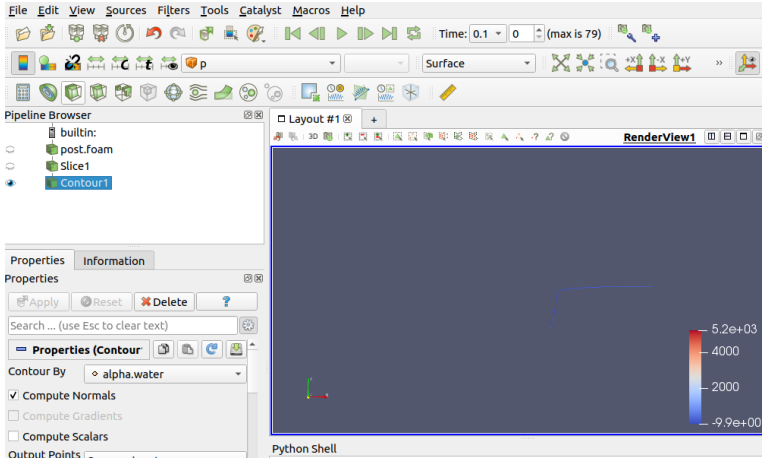
水位の時刻歴



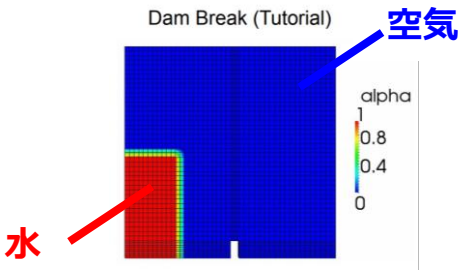
(9)「Counter」を選択
- Contour By：「alpha.water」を選択

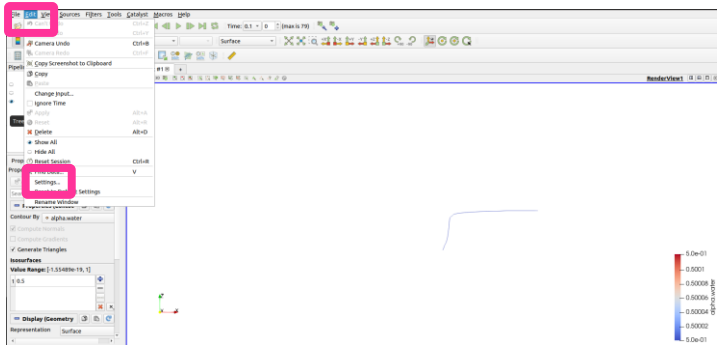
(10)Value Rangeを更新する必要があるので更新ボタンをクリック。
→「Apply」をクリック

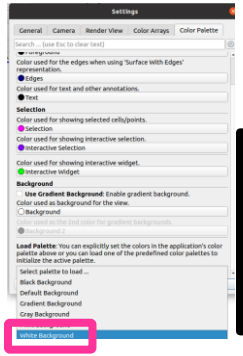
水位の時刻歴



体積分率α=0.5**（空気と水の境界面）**のコンターを描くことができた。
※背景の色と被って見えにくいが・・・

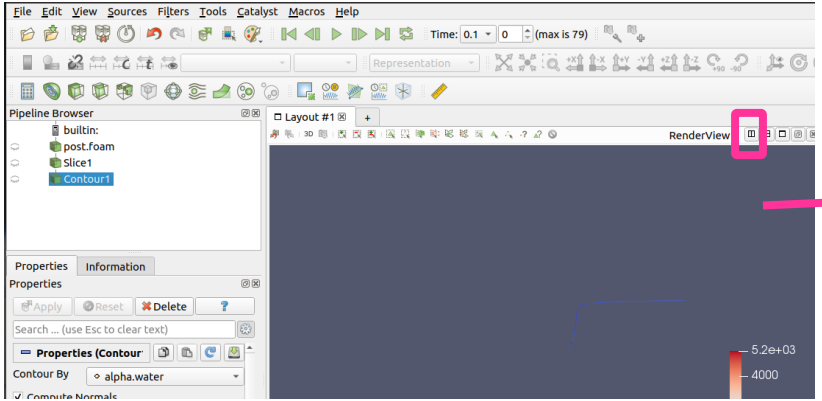




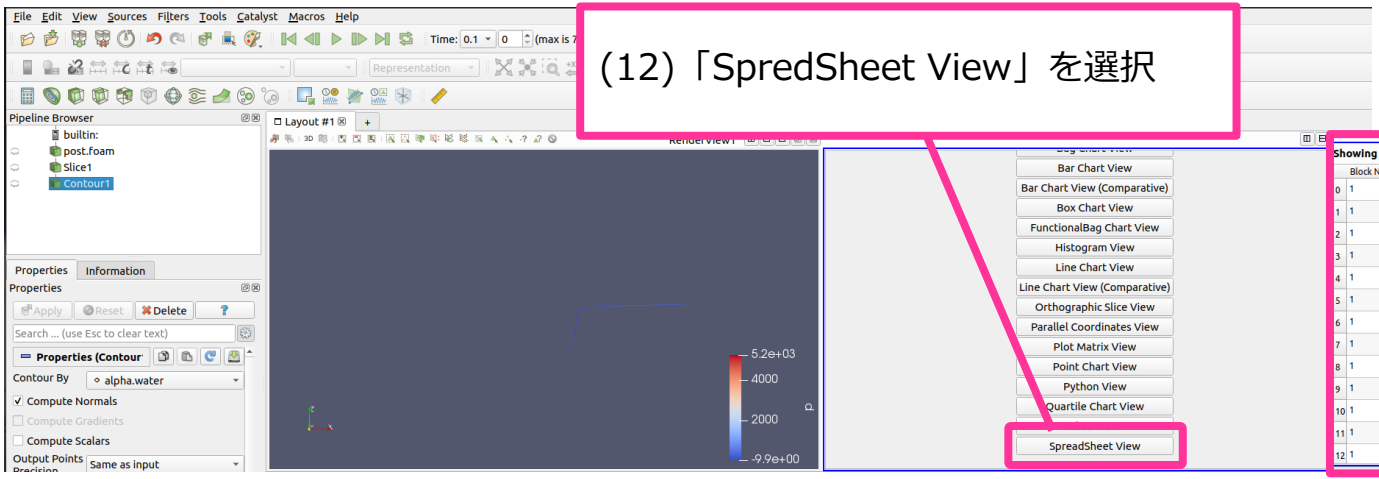


色の調整のため「Edit」>「Settings...」を開き、背景色を「Wthite Background」（白色ベース）にしても良い

水位の時刻歴

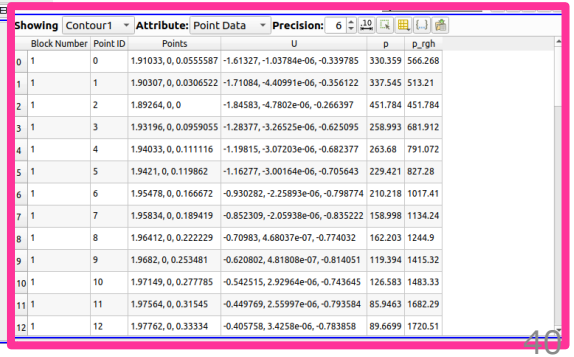


(11)「Split Horizontal」を選択
画面を2画面にする



(12)「SpredSheet View」を選択

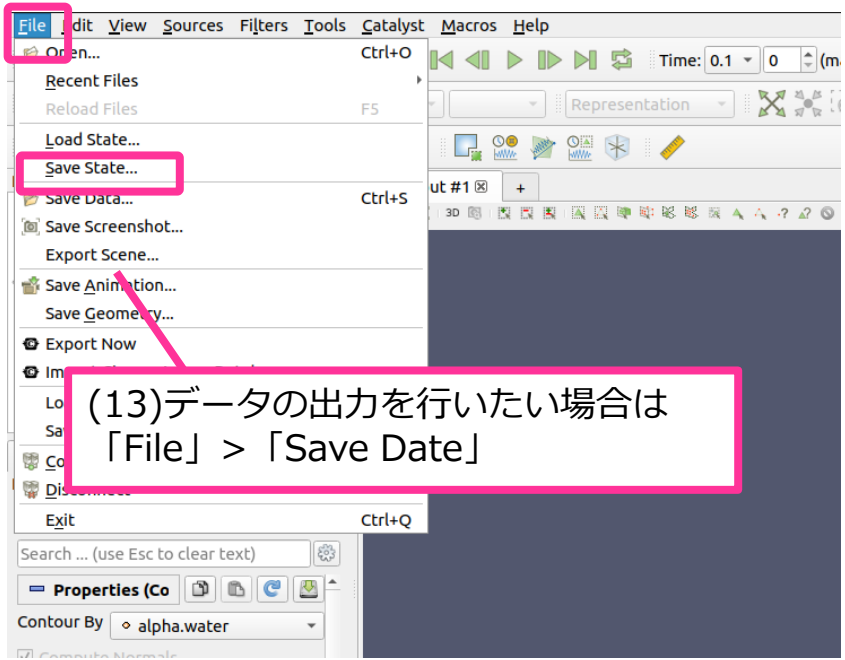
SpreadSheetView1に
データが表示される

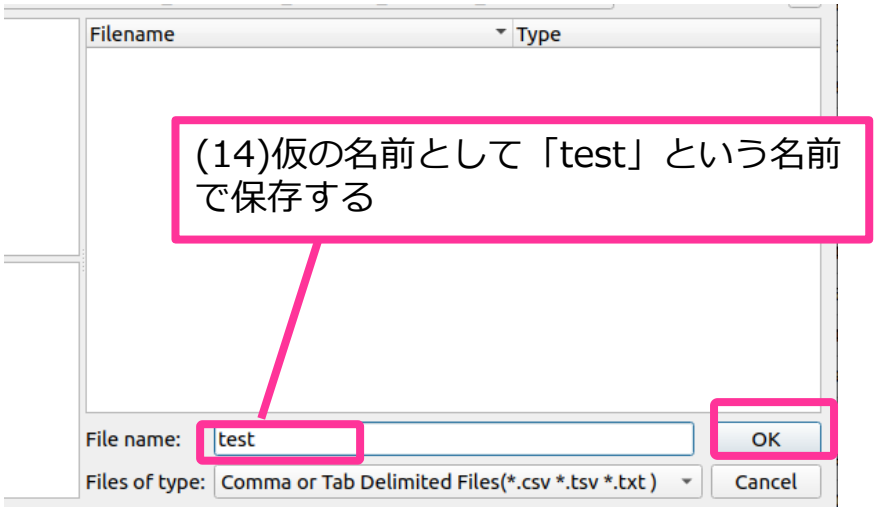


	Block Number	Point ID	Points	U	p	p_rgh
0	1	0	1.91033, 0, 0.0555587	-1.61327, -1.03784e-06, -0.339785	330.359	566.268
1	1	1	1.90307, 0, 0.0306522	-1.71084, -4.40991e-06, -0.356122	337.545	513.21
2	1	2	1.89264, 0, 0	-1.84583, -4.7802e-06, -0.266397	451.784	451.784
3	1	3	1.93196, 0, 0.0959055	-1.28377, -3.26525e-06, -0.625095	258.993	681.912
4	1	4	1.94033, 0, 0.111116	-1.19815, -3.07203e-06, -0.682377	263.68	791.072
5	1	5	1.9421, 0, 0.119862	-1.16277, -3.00164e-06, -0.705643	229.421	827.28
6	1	6	1.95478, 0, 0.166672	-0.930282, -2.25893e-06, -0.798774	210.218	1017.41
7	1	7	1.95834, 0, 0.189419	-0.852309, -2.05938e-06, -0.835222	158.998	1134.24
8	1	8	1.96412, 0, 0.222229	-0.70983, 4.68037e-07, -0.774032	162.203	1244.9
9	1	9	1.9682, 0, 0.253481	-0.620802, 4.81808e-07, -0.814051	119.394	1415.32
10	1	10	1.97149, 0, 0.277785	-0.542515, 2.92964e-06, -0.743645	126.583	1483.33
11	1	11	1.97564, 0, 0.31545	-0.449769, 2.55997e-06, -0.793584	85.9463	1682.29
12	1	12	1.97762, 0, 0.33334	-0.405758, 3.4258e-06, -0.783858	89.6699	1720.51

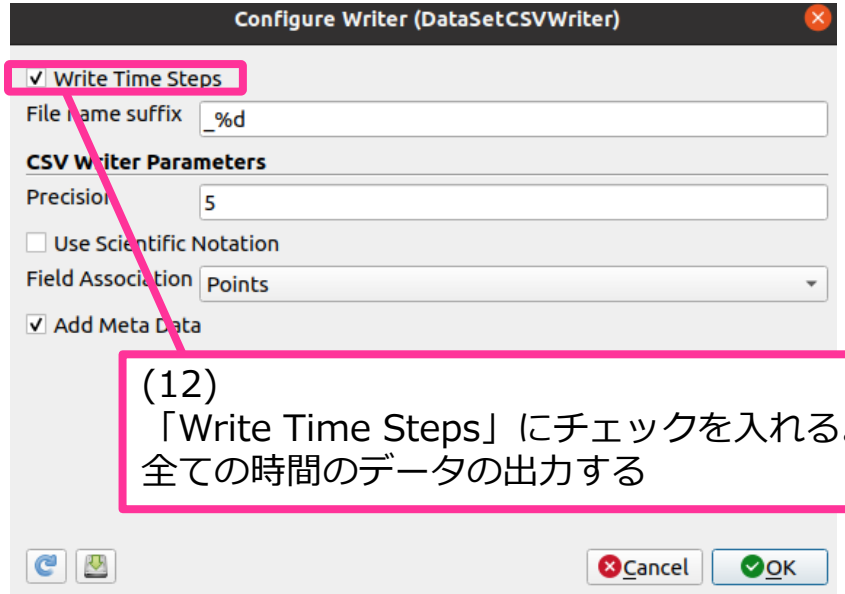
水位の時刻歴

exp_data/003_ParaView_Height

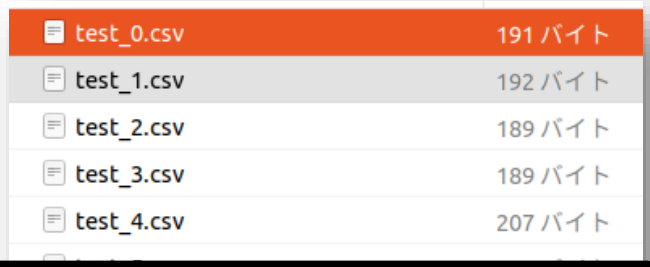




水位の時刻歴



(12)
「Write Time Steps」にチェックを入れる。
全ての時間のデータの出力する



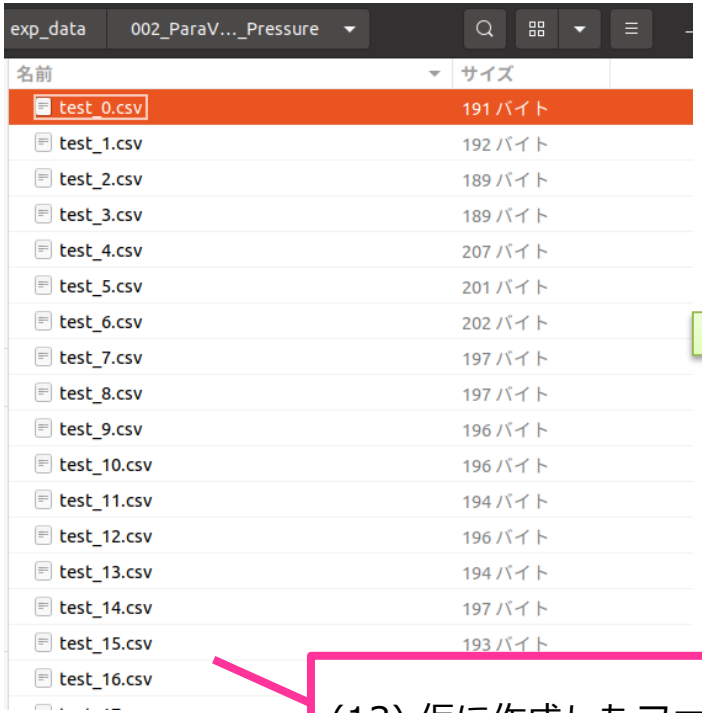
「test_%d.csv」という名前でファイルができる。
※%d：ステップ数
※時間ではなくステップ数なので注意
今は0.1秒ずつ出力しているので1/10すると時間になる。

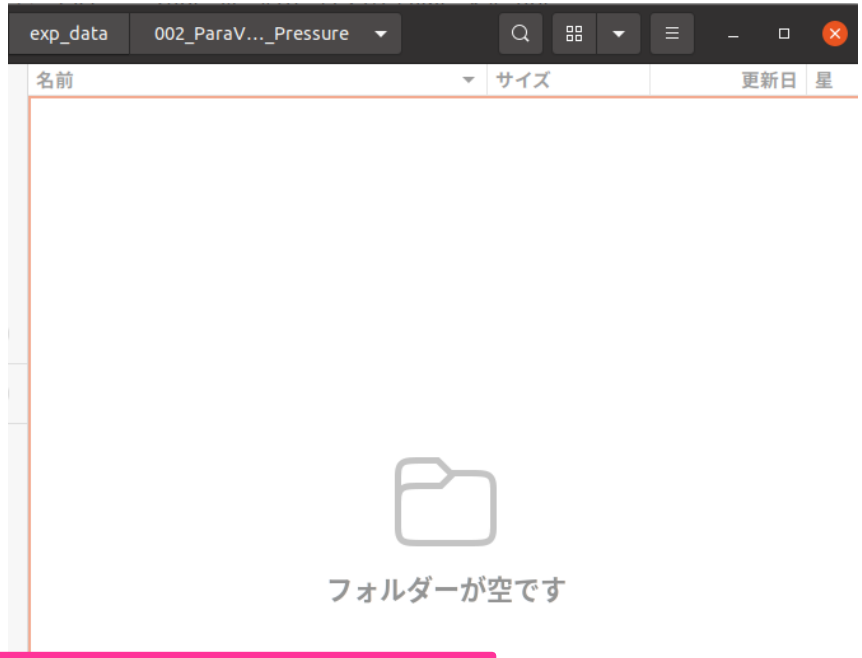
exp_data/002_ParaView_Height/test_0.csv

	A	B	C	D	E	F	G	H	I	J
1	Points:0	Points:1	Points:2	U:0	U:1	U:2	alpha.water	alpha.water_0	p	p_rgh
2	1.9103	0	0.055559	-1.6132	9.243E-06	-0.33978	0.5	0.40564	330.38	566.29
3	1.9031	0	0.030653	-1.7108	-1.4151E-07	-0.35613	0.5	0.38931	337.56	513.23
4	1.8926	0	0	1.8458	6.1526E-08	0.2664	0.5	0.37919	451.8	451.8

水位の時刻歴

exp_data/002_ParaView_Height

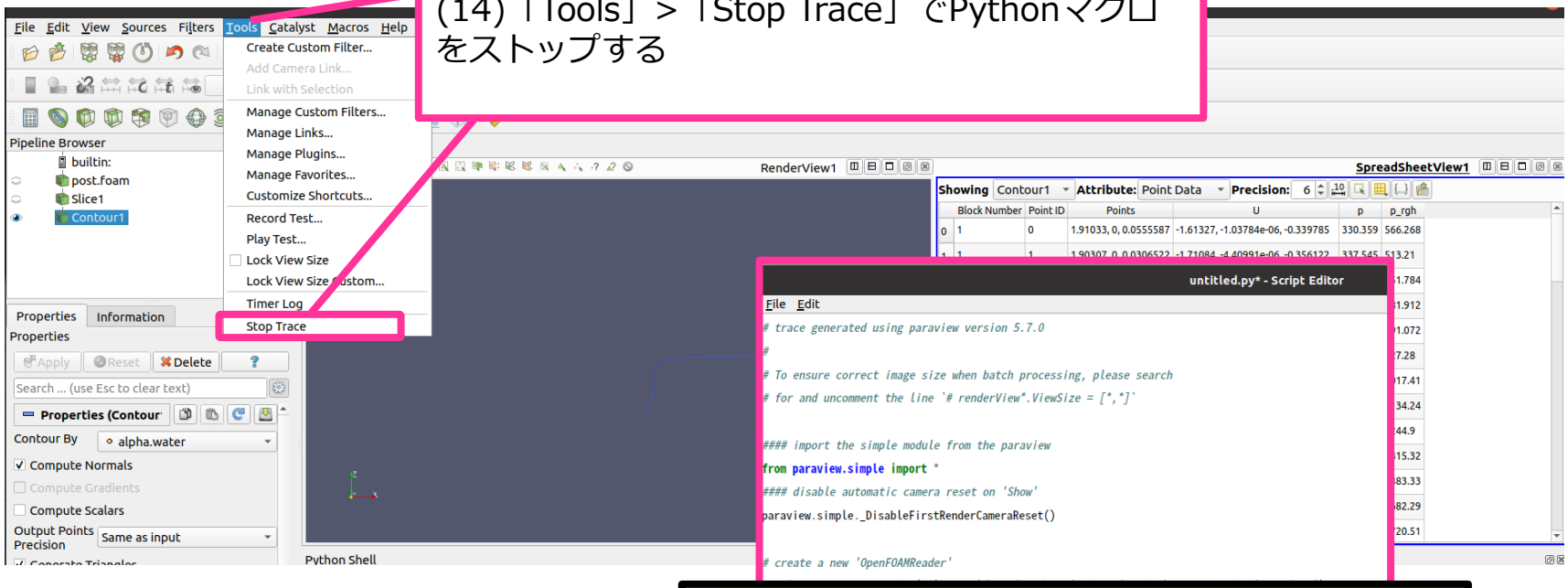




(13) 仮に作成したファイルなので全て削除する。

水位の時刻歴

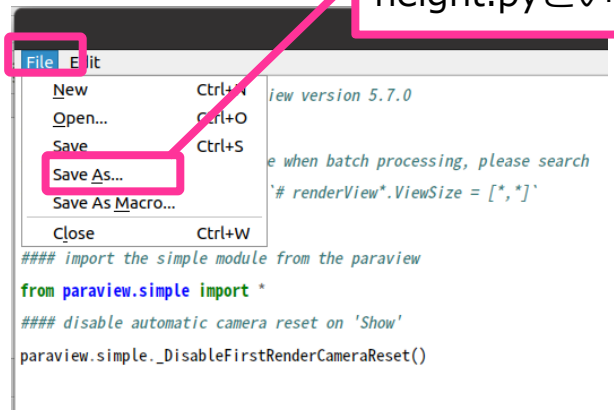
(14)「Tools」>「Stop Trace」でPythonマクロをストップする



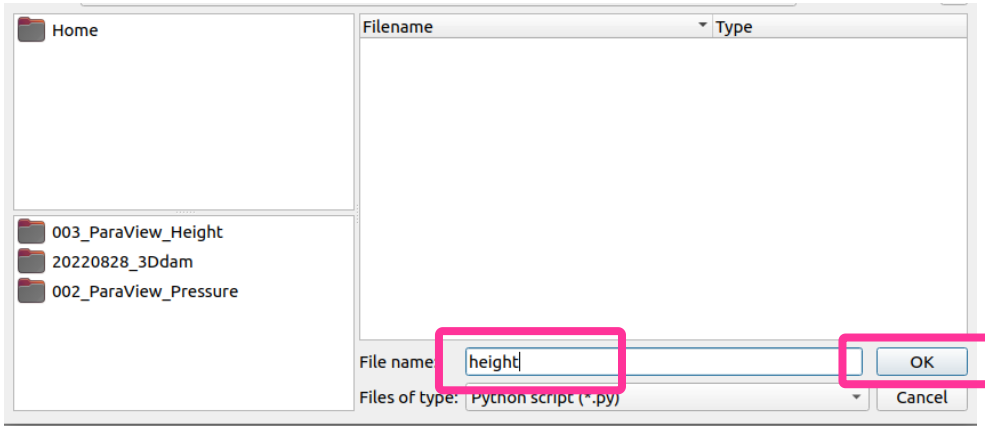
今までの操作がマクロとして記録される。
Pythonスクリプト

水位の時刻歴

(15)「File」>「Save As...」
height.pyという名前で保存



exp_data/002_ParaView_Height

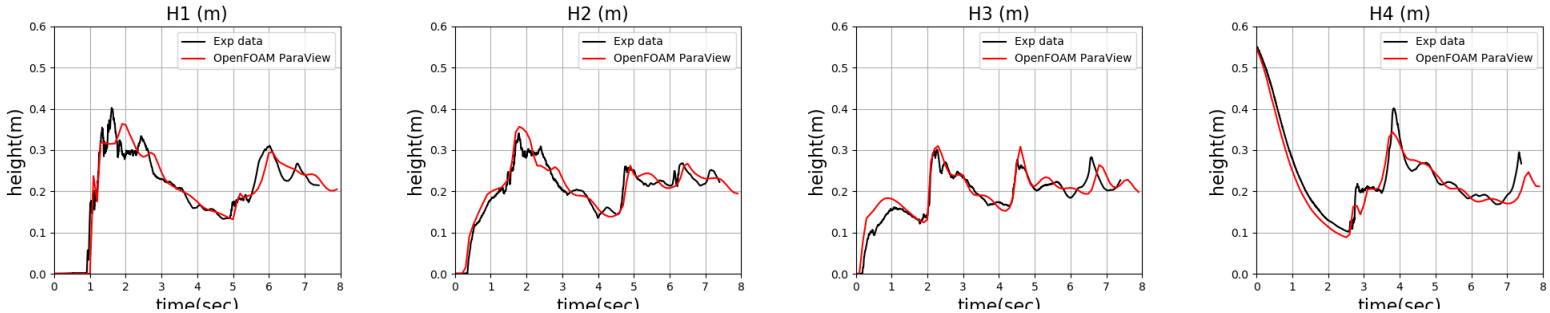


水位の時刻歴

ParaViewとPythonで自動化を行う

ParaView　指定座標の圧力のデータを出力(マクロ使用)

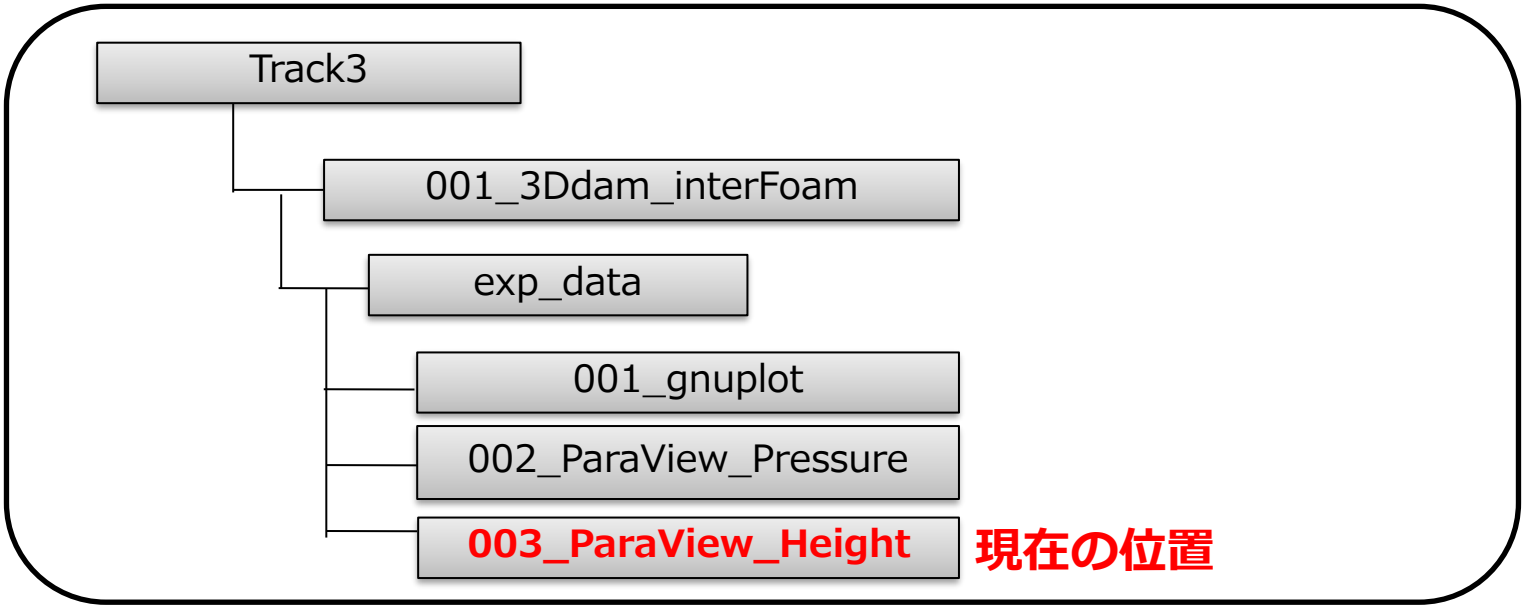
Python　出力したデータを結合しグラフ化する



以下のコマンドで「**003_ParaView_Height**」フォルダに移動

フォルダを移動

```
$cd exp_data/003_ParaView_Pressure
```



水位の時刻歴

(1)height.pyのファイル内を変更

exp_data/002_ParaView_Pressure/height.py

```
 9 import os
10
11 # 現在のパス
12 PWD = os.getcwd()
13
14 paraview.simple._DisableFirstRenderCameraReset()
15
16 # create a new 'OpenFOAMReader'
17 postfoam = OpenFOAMReader(FileNam=f'{PWD}/../../post.foam')
18
19 # get animation scene
```

追加

変更

```
134
135 # save data
136 SaveData(f'{PWD}/height.csv', proxy=contour1, WriteTimeSteps=1)
137
```

変更

(2)以下のコマンドを実行

```
$paraview --script=height.py
```

height.py	4.1 kB
height_0.csv	3.7 kB
height_1.csv	3.4 kB
height_2.csv	3.9 kB
height_3.csv	4.7 kB
height_4.csv	5.0 kB
height_5.csv	5.4 kB
height_6.csv	5.9 kB
height_7.csv	6.2 kB
height_8.csv	7.8 kB
height_9.csv	8.5 kB
height_10.csv	9.0 kB
height_11.csv	9.7 kB

水位の時刻歴

Pythonでデータをつなげる
1. 各ステップの圧力データをつなげる
2. 各測定箇所で「1」を実行
3. 「2」までをグラフ化する
4. グラフを画像データとして保存

(3)以下のコマンドでmain.pyを実行

```
$python3 mian.py
```

exp_data/002_ParaView_Hiehgt/main.py

```
import matplotlib.pyplot as plt
import math
import re
import pandas as pd
import glob

# read csv file
def df_csv(csvfile):
    df = pd.read_csv(csvfile)
    return df

# csvファイルのファイル名のソート
def atoi(text):
    return int(text) if text.isdigit() else text

def natural_keys(text):
    return [ atoi(c) for c in re.split(r'(¥d+)', text) ]

# 時刻歴データをつなげる
def df_height_data(csvfile_sort):
    delta = 0.03
    height_list = []

    for csvfile in csvfile_sort:
        height_dict = {}
        height_dict['time(sec)'] = float(re.findall(r"¥d+", csvfile)[0])/10
        df_h = pd.DataFrame()

        df = df_csv(csvfile)
        for point in Hxpoints:
            Hz_mean = df[(df['Points:0']>=Hxpoints[point]-delta) & (df['Points:0']<=Hxpoints[point]+delta)]['Points:2'].mean()
            if math.isnan(Hz_mean) == True: # nan
                height_dict[point] = 0.0
            else:
                height_dict[point] = Hz_mean

        height_list.append(height_dict)

        df_h_ = pd.DataFrame(height_list)
        df_h = pd.concat([df_h, df_h_])

    return df_h

# グラフ化
def graph_func(Hxpoints,df_openfoam, df_exp):
    for i, point in enumerate(Hxpoints):
        ax = fig.add_subplot(1, 4, i+1)
        ax.plot(df_exp['Time (s)'].to_numpy(), df_exp[f'{point}'].to_numpy(), color='black',label='Exp data')
        ax.plot(df_openfoam['time(sec)'].to_numpy(), df_openfoam[point].to_numpy(), color='red', label='OpenFOAM ParaView')
        ax.set_xlabel('time(sec)',fontsize=16)
        ax.set_ylabel('height(m)',fontsize=16)
        ax.set_title(f'{point}',fontsize=16)
        ax.set_xlim(0, 8.0)
        ax.set_ylim(0,0.6)
        ax.grid()
        ax.legend()

if __name__ == '__main__':
    Hxpoints = {
        'H1 (m)':0.496,
        'H2 (m)':0.992,
        'H3 (m)':1.488,
        'H4 (m)':2.638
        }

    # csvファイルのリスト化
    csv_file = glob.glob('*.csv')
    csvfile_sort = sorted(csv_file, key=natural_keys)
    df_openfoam = df_height_data(csvfile_sort)

    # csvファイルのリスト化
    df_exp = pd.read_csv('../../exp_data/test_case_2_exp_data.csv',sep='¥t')

    # グラフ化（OpenFOAMと実験の比較）
    fig = plt.figure(figsize=(24,4))
    plt.subplots_adjust(wspace=0.4, hspace=0.6)


    graph_func(Hxpoints,df_openfoam,df_exp)

    fig.savefig("height.png")
```

水位の時刻歴

実験データとOpenFOAMの比較

exp_data/002_ParaView_Hiehgt/height.png