

第5回OpenFOAM勉強会for beginner

テーマ進捗報告

id:oga_shin

小縣信也

ogata shinya

1. caseCの計算結果
2. caseCの並列計算実行方法
3. bashスクリプトの実行権限
4. Google sketchupを用いたSTLデータへの変換
5. wubiを用いたデュアルブート環境
6. ParaViewの補間方法
7. Python スクリプトを用いたParaviewの自動操作

caseC 計算モデル

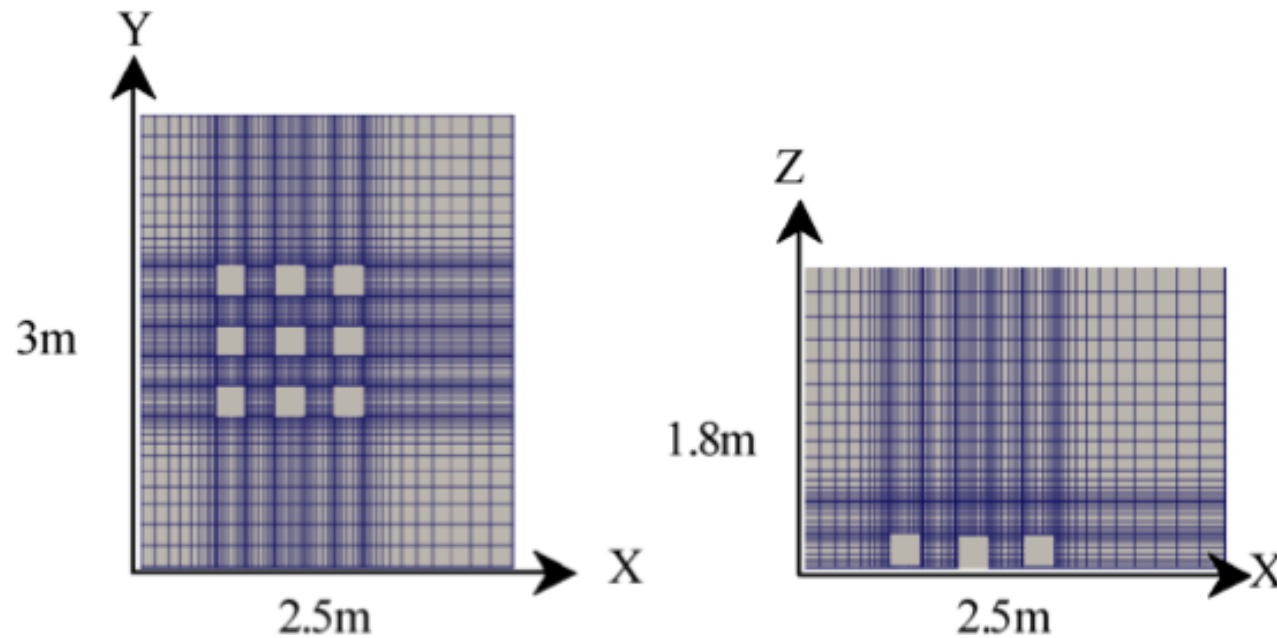


Fig.4-1 Calculation area

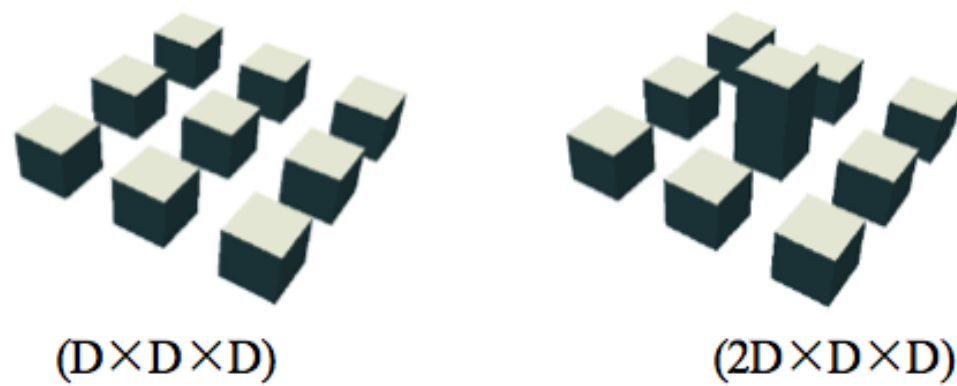


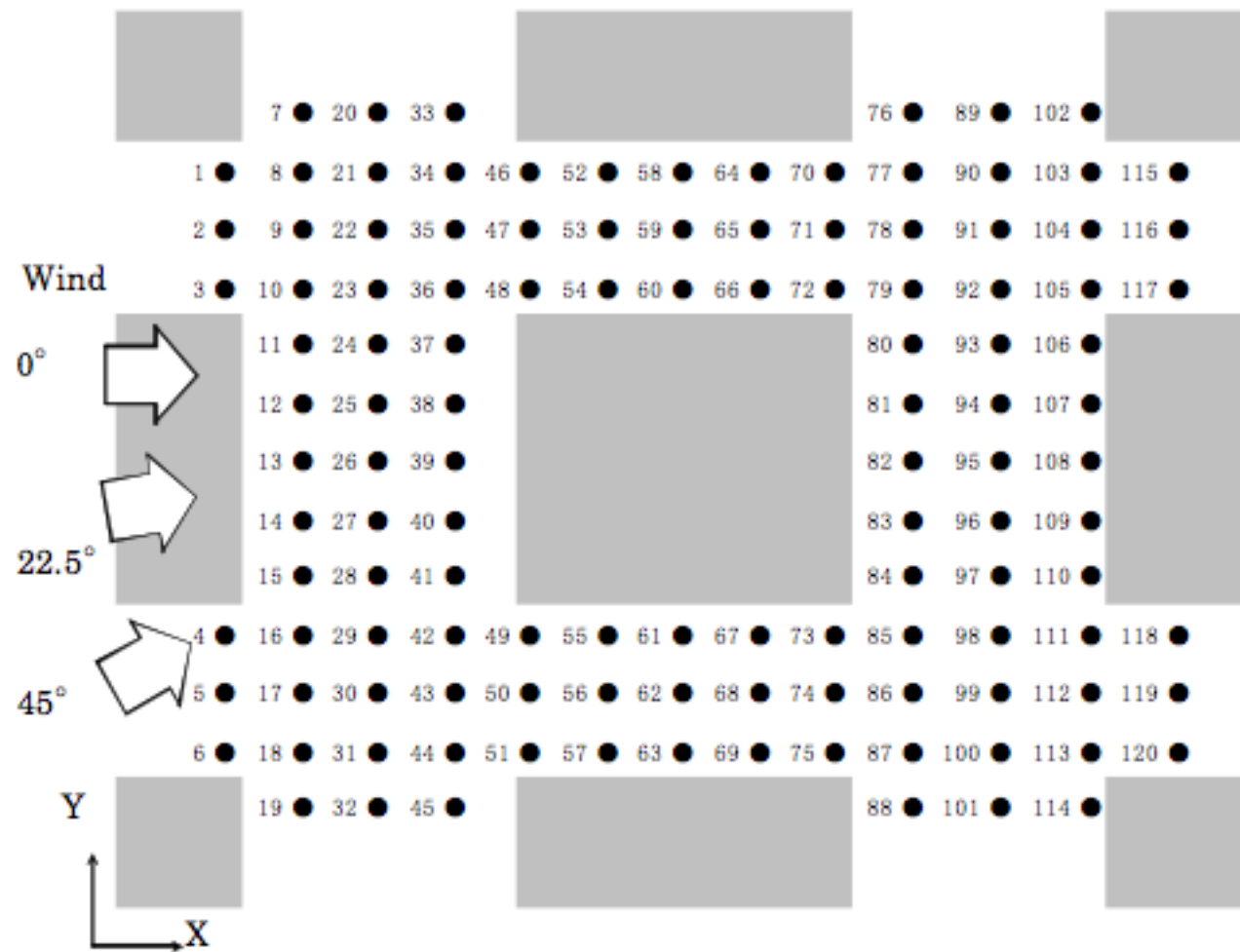
Fig.4-2 Building Model

caseC 計算条件

移流項にTVD (van Leer limiter) スキームを, 地表面境界に滑面用一般化対数則を適用している以外は学会指定のベンチマーク条件に準じている.

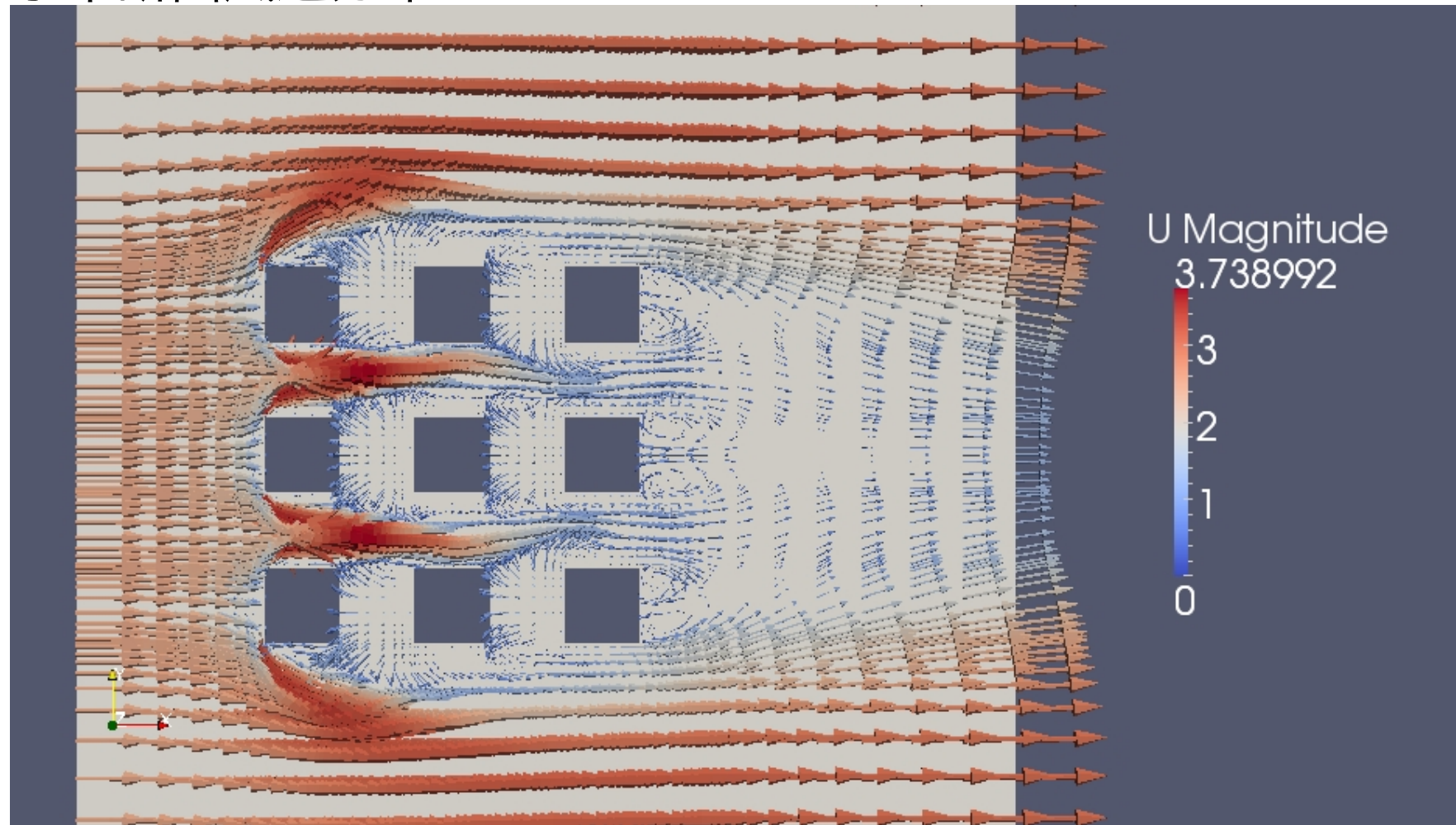
流入境界条件	$\langle u \rangle$, k , ε はベンチマーク指定値 ⁽²⁾ を使用 風向, 22.5° 及び 45° の Case は, 2 面を流入条件とした.
上面境界条件	slip 壁
側面境界条件	slip 壁
流出境界条件	勾配ゼロ. 風向, 22.5° 及び 45° の Case は, 2 面を流出条件とした.
固体壁面境界	滑面用の一般化対数則
乱流モデル	標準 k - ε モデル
計算格子分割法	ベンチマーク指定の格子分割 ⁽²⁾ を使用 (約 8 万要素)

caseC 測定点



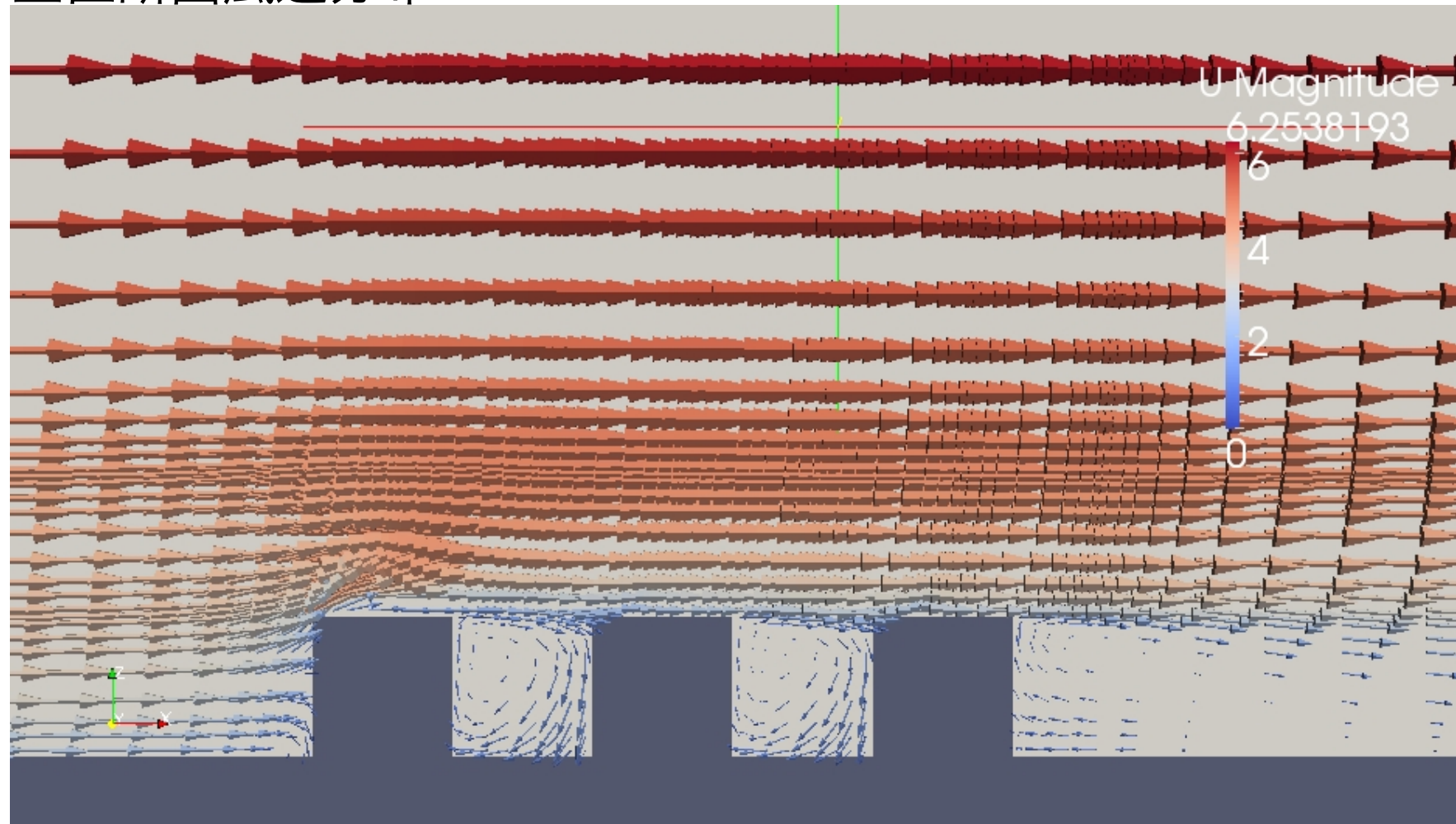
caseCの計算結果

水平断面風速分布



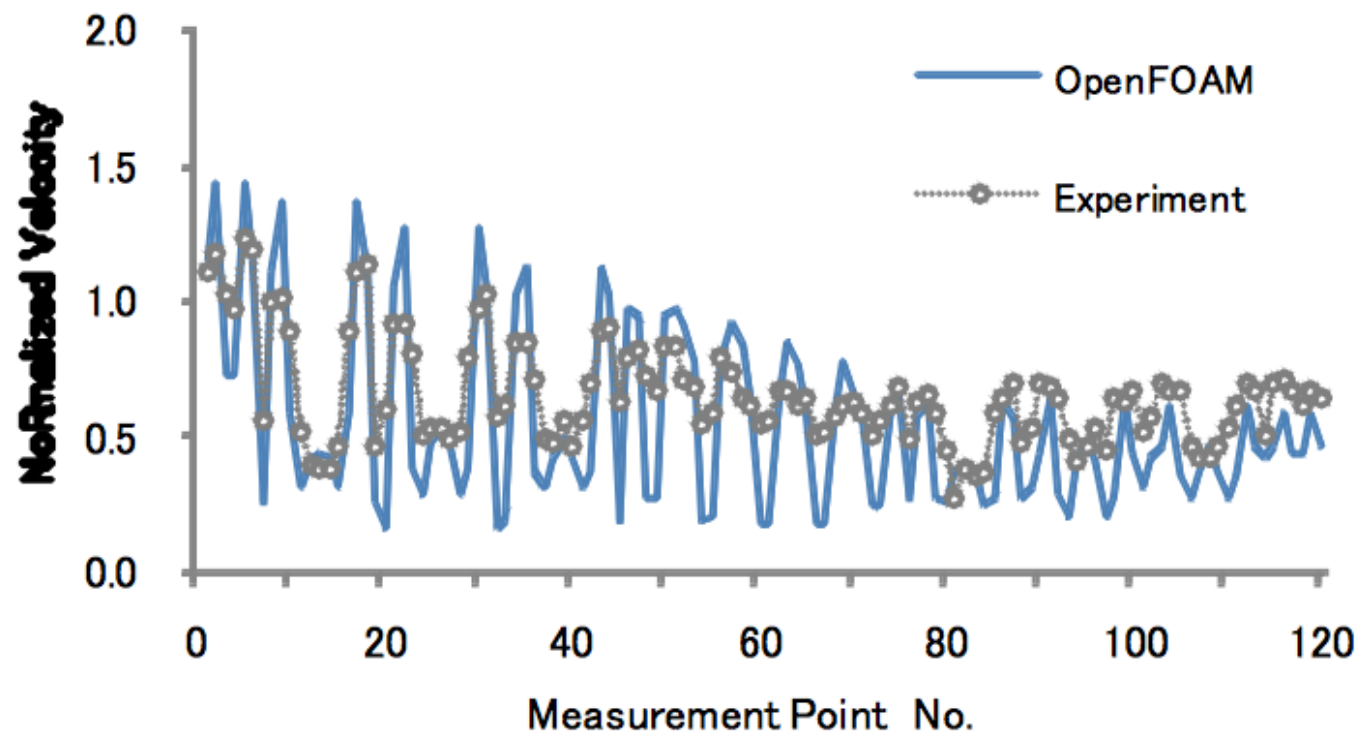
caseCの計算結果

垂直断面風速分布



caseC 計算結果と風洞実験の比較

各計測点における風速比を比較すると、全体としては概ね近い値を示しているが、対象建築物の側面および後流域において乖離が大きくなる傾向が見られる。この傾向はCase2～Case6も同様であった。



caseC 計算結果と風洞実験の比較

相関係数は概ね0.8～0.9の範囲であり、両者の対応は良好である。公開されている他のCFD結果と比較すると、相関係数に関しては同等であり、標準誤差に関しては他のCFD結果よりも小さくなるという結果になった。

Table 4-2 Correlation factor and standard error

Case	対象建築物形状	風向	相関係数	標準誤差
1	D×D×D	0°	0.81	0.19
2	D×D×D	22.5°	0.88	0.16
3	D×D×D	45°	0.85	0.19
4	2D×D×D	0°	0.88	0.12
5	2D×D×D	22.5°	0.85	0.16
6	2D×D×D	45°	0.92	0.13

ケースファイル保存場所



caseCの並列計算実行方法

■ Allrun (bashスクリプト)

```
if [ "x$1" = "x-parallel" ]  
then  
    runApplication decomposePar  
    runParallel $application 4  
    runApplication reconstructPar  
else  
    runApplication $application  
fi
```

caseCの並列計算実行方法

■Allrun (bashスクリプト)

```
if [ "x$1" = "x-parallel" ]
```

引数が
指定さ
れた時

1
番
目
の
引
数

引数が
指定さ
れた時

1番目の
引数の値

■端末での入力

```
$ ./Allrun -parallel
```

caseCの並列計算 2coreで計算する場合

■ Allrun

```
if [ "x$1" = "x-parallel" ]
then
    runApplication decomposePar
    runParallel $application 2
    runApplication reconstructPar
else
    runApplication $application
fi
```

■ decomposeParDict

```
numberOfSubdomains 2;

// preservePatches (inlet);
// preserveFaceZones (heater solid1 solid3);

method            simple;
// method          hierarchical;
// method          metis;
// method          manual;

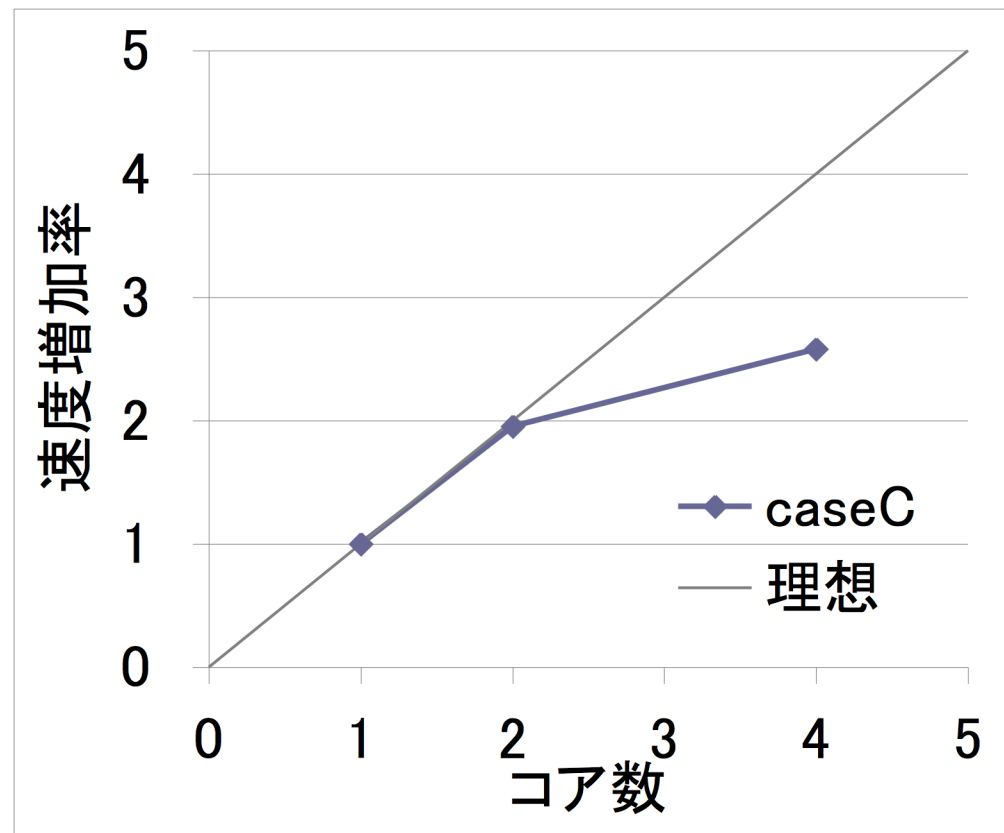
simpleCoeffs
{
    n              (2 1 1);
    delta          0.001;
}
```

caseCの計算時間比較

■ 並列計算効率

core数	計算時間[s]	速度増加率
1	80	1.0
2	41	2.0
4	31	2.6

Intel core2Quad2.3GHz



bashスクリプトの実行権限

■bashを実行できないというエラーメッセージ

```
ogashin@ogashin-desktop:~/Dropbox/caseB$ ./Allclean  
bash: ./Allclean: No such file or directory
```

bashスクリプトの実行権限

■実行権限の確認方法

端末で以下を実行

ls -l

<結果>

ogashin@ogashin-desktop:~/OpenFOAM/ogashin-1.7.0/run/caseB\$ ls -l

合計 128

drwxr-xr-x 2 ogashin ogashin 4096 2010-10-29 01:02 0

-rwxr-xr-x 1 ogashin ogashin 161 2010-09-21 16:00 Allclean

-rwxr-xr-x 1 ogashin ogashin 578 2010-09-21 13:30 Allrun

-rw-r--r-- 1 ogashin ogashin 1142 2010-09-21 16:32 README

-rw-r--r-- 1 ogashin ogashin 25 2008-08-05 14:53 box.dat

bashスクリプトの実行権限

■実行権限の違い

実行出来る時 -rwxr-xr-x

実行できない時 -rw-r--r--

■実行できなくなる現状が生じる時

ディレクトリを圧縮せずに他のパソコンに移動

①vmware/Ubuntu→windows→ファイルサーバー→
widows→ vmware/Ubuntu

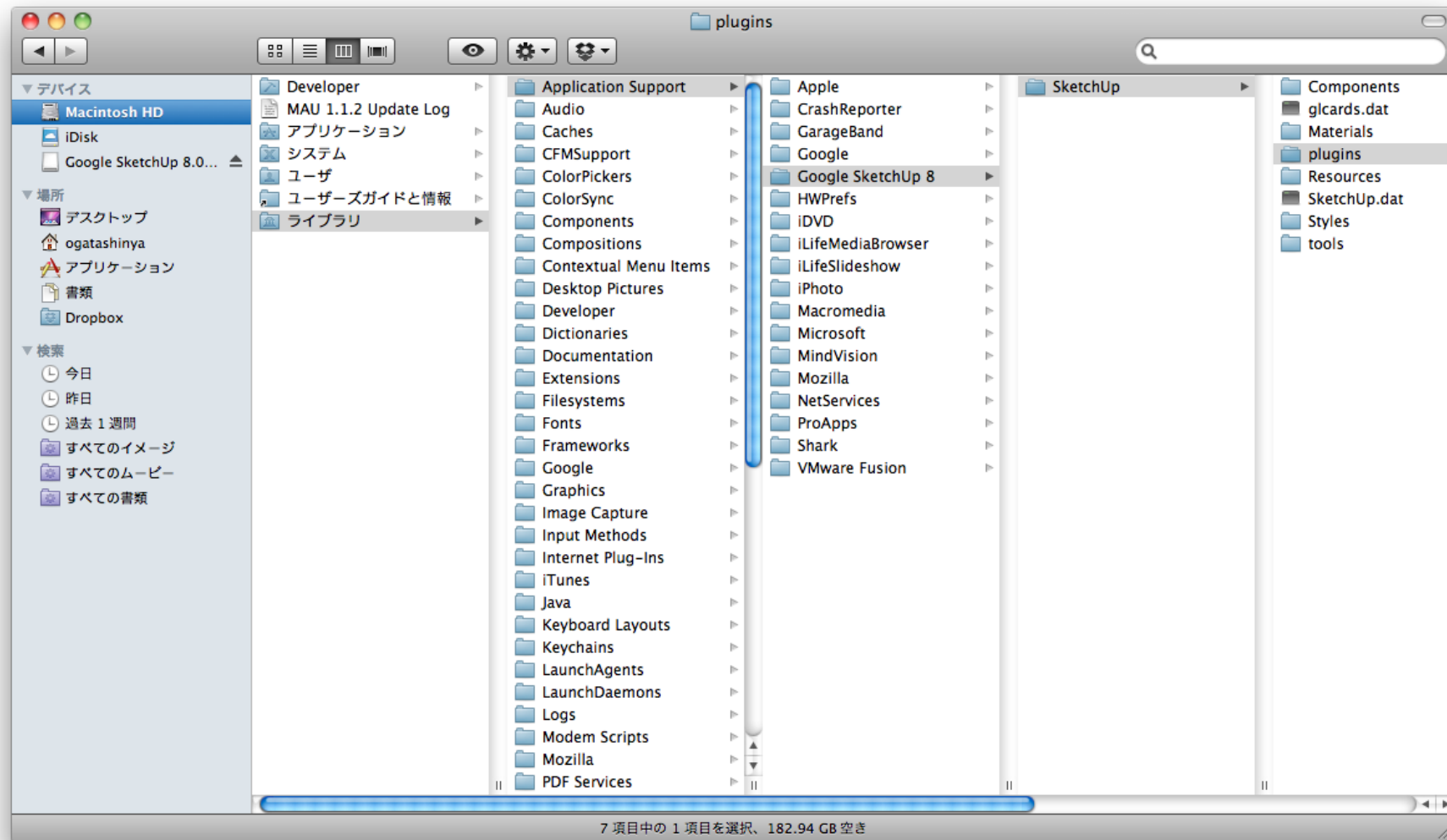
② vmware/Ubuntu→Dropbox→ vmware/Ubuntu

③メールで送られたbashファイル

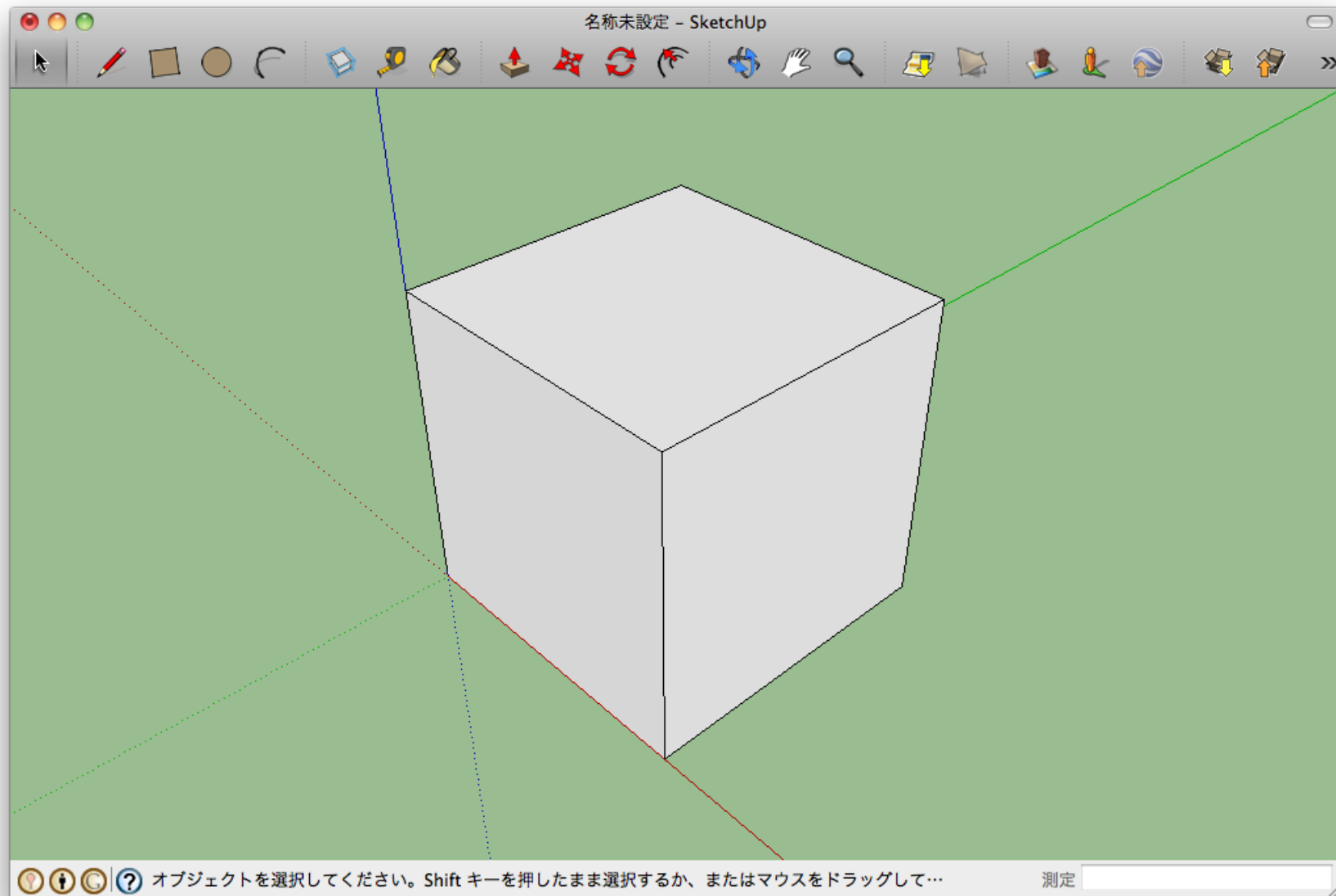
■対策

PC間を移動させる時は圧縮するようにする

Google sketchupを用いたSTLデータへの変換



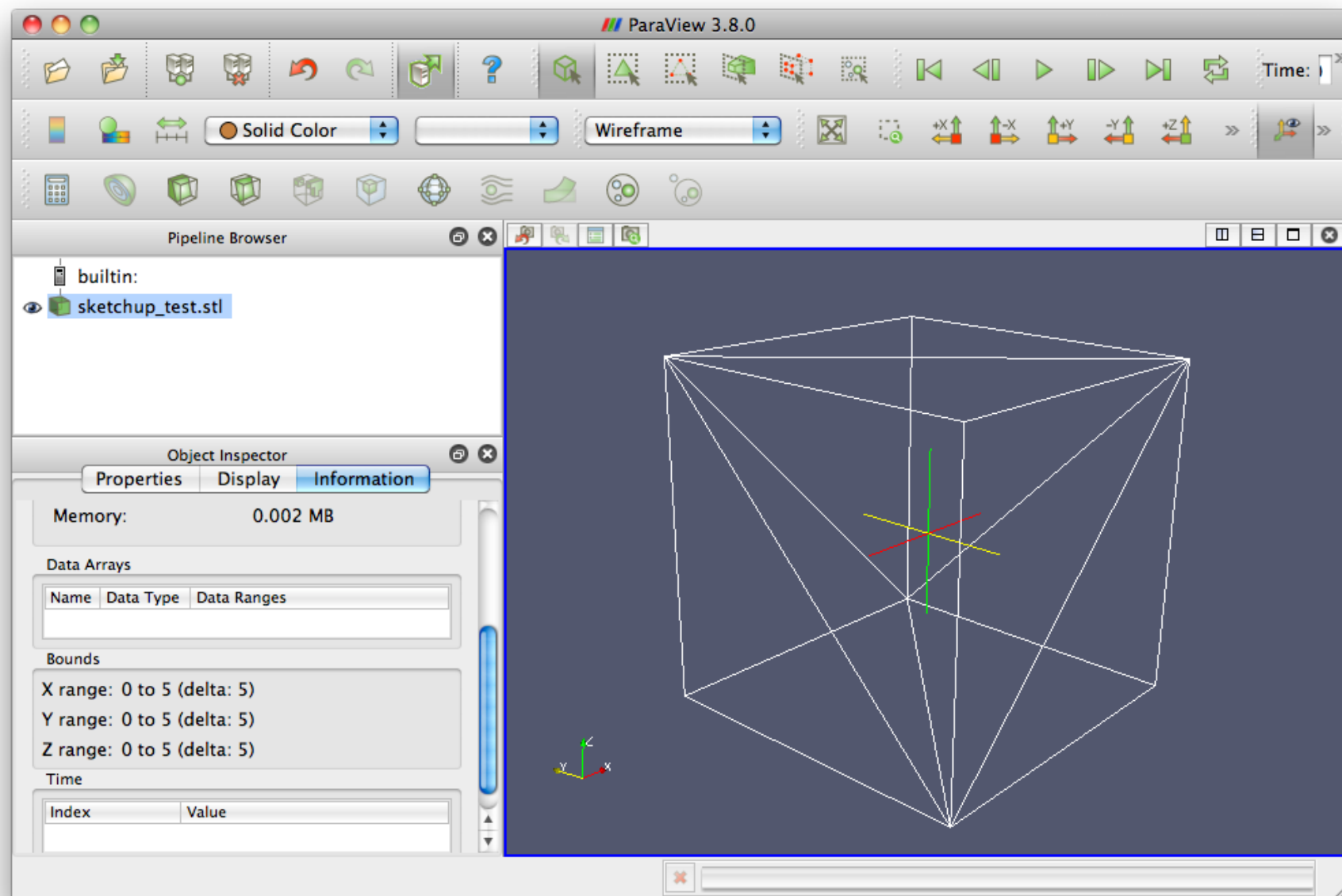
Google sketchupを用いたSTLデータへの変換



Google sketchupを用いたSTLデータへの変換

```
solid model
facet normal 0.0 0.0 -1.0
outer loop
vertex 5.0 5.0 0.0
vertex 0.0 0.0 0.0
vertex 0.0 5.0 0.0
endloop
endfacet
facet normal 0.0 0.0 -1.0
outer loop
vertex 0.0 0.0 0.0
vertex 5.0 5.0 0.0
vertex 5.0 0.0 0.0
endloop
endfacet
facet normal -0.0 -0.0 1.0
outer loop
vertex 5.0 0.0 5.0
vertex 0.0 5.0 5.0
vertex 0.0 0.0 5.0
endloop
endfacet
facet normal -0.0 -0.0 1.0
outer loop
vertex 0.0 5.0 5.0
vertex 5.0 0.0 5.0
```

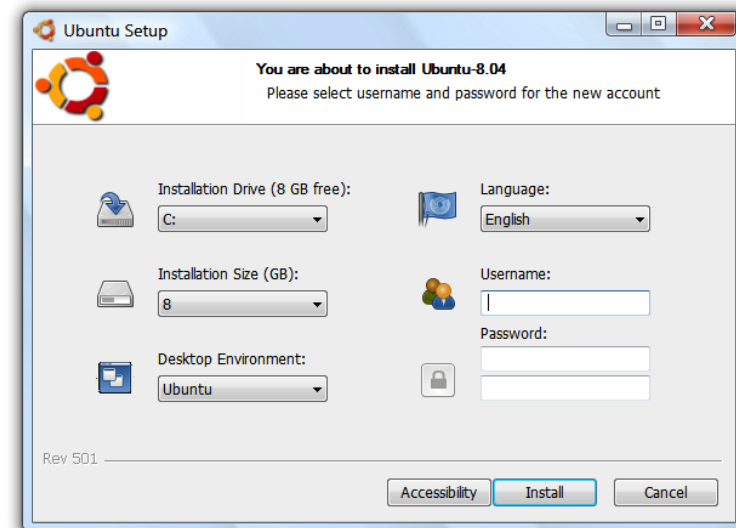
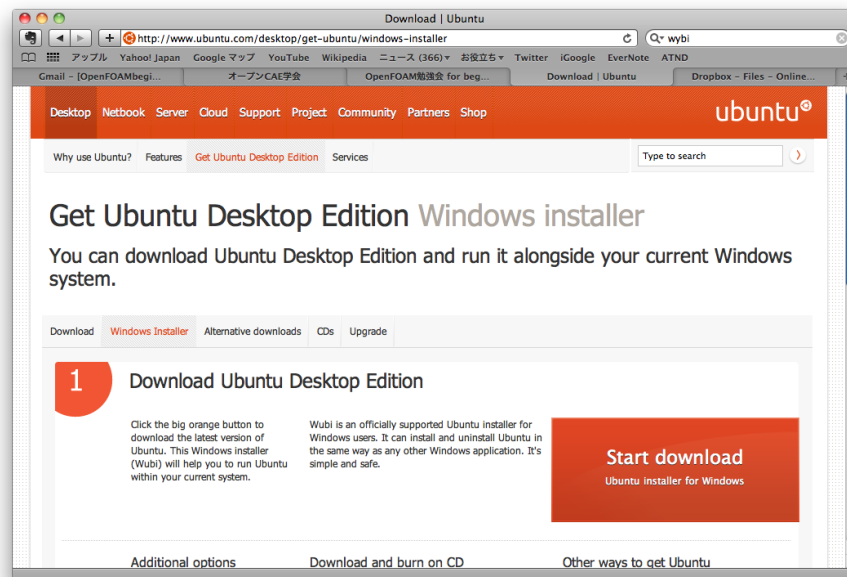
Google sketchupを用いたSTLデータへの変換



wubiを用いたデュアルブート環境

■wubiとは？

Wubi (Windows-based Ubuntu Installer) はWindows上で動作するUbuntu公式のフリーソフトウェアインストーラー。UbuntuをWindowsのファイルシステム上にインストールし、Ubuntuを利用することができる。GPLで配布される[1]。Wikipediaより



<http://www.ubuntu.com/desktop/get-ubuntu/windows-installer>

wubiを用いたデュアルブート環境

■動作確認

windowsXP→問題なし

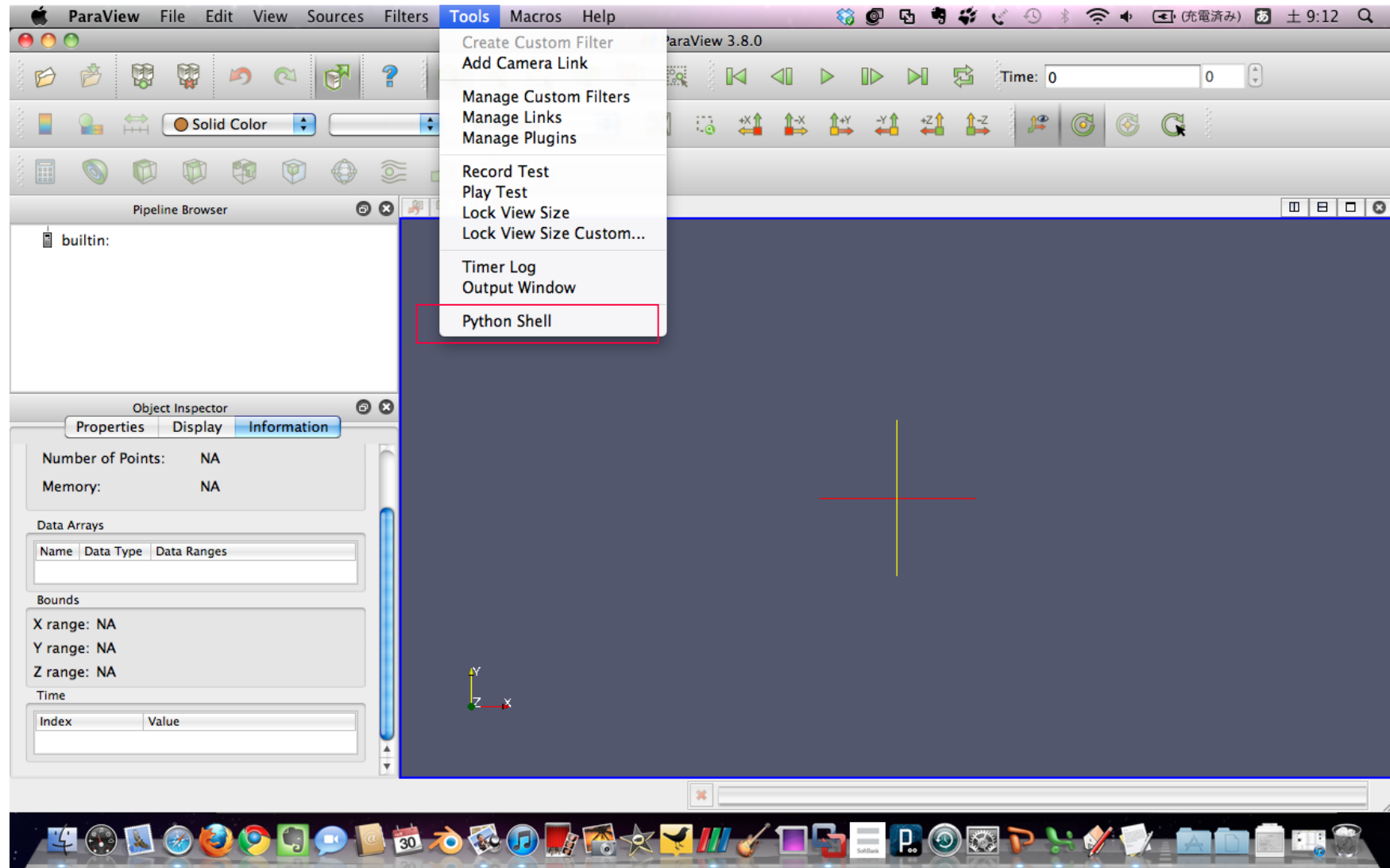
Windows7→問題なし

※64ビットマシンで実行すると、自動的に64ビット版のUbuntuをインストールする仕様になっているらしい。

※公式ubuntuしかインストールできないので、
はじめから入っているソフトが少なく不便。
gnuplotなど自分でインストールしなければならない。

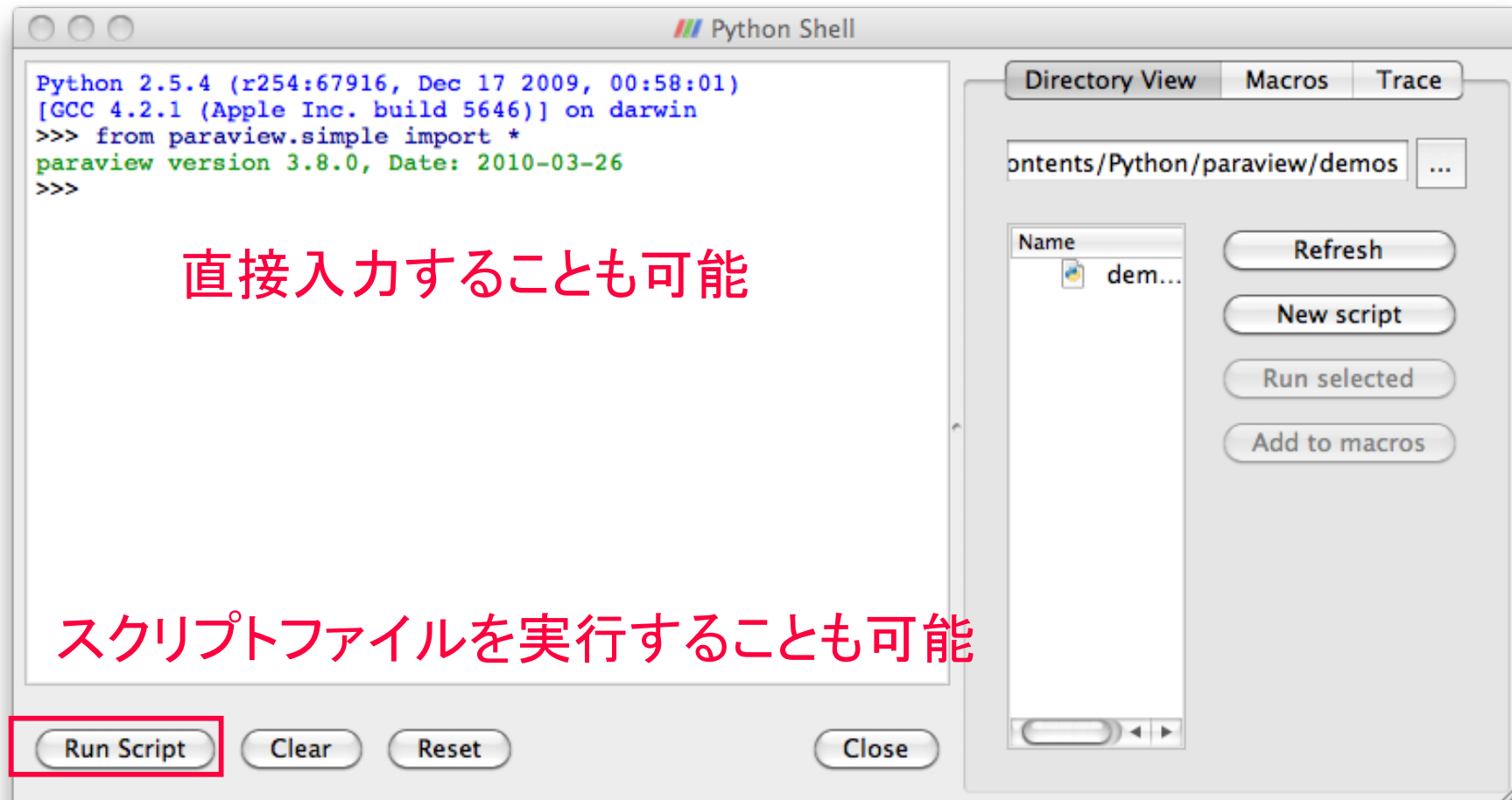
pythonスクリプトを用いたParaviewの操作

■Python Shell



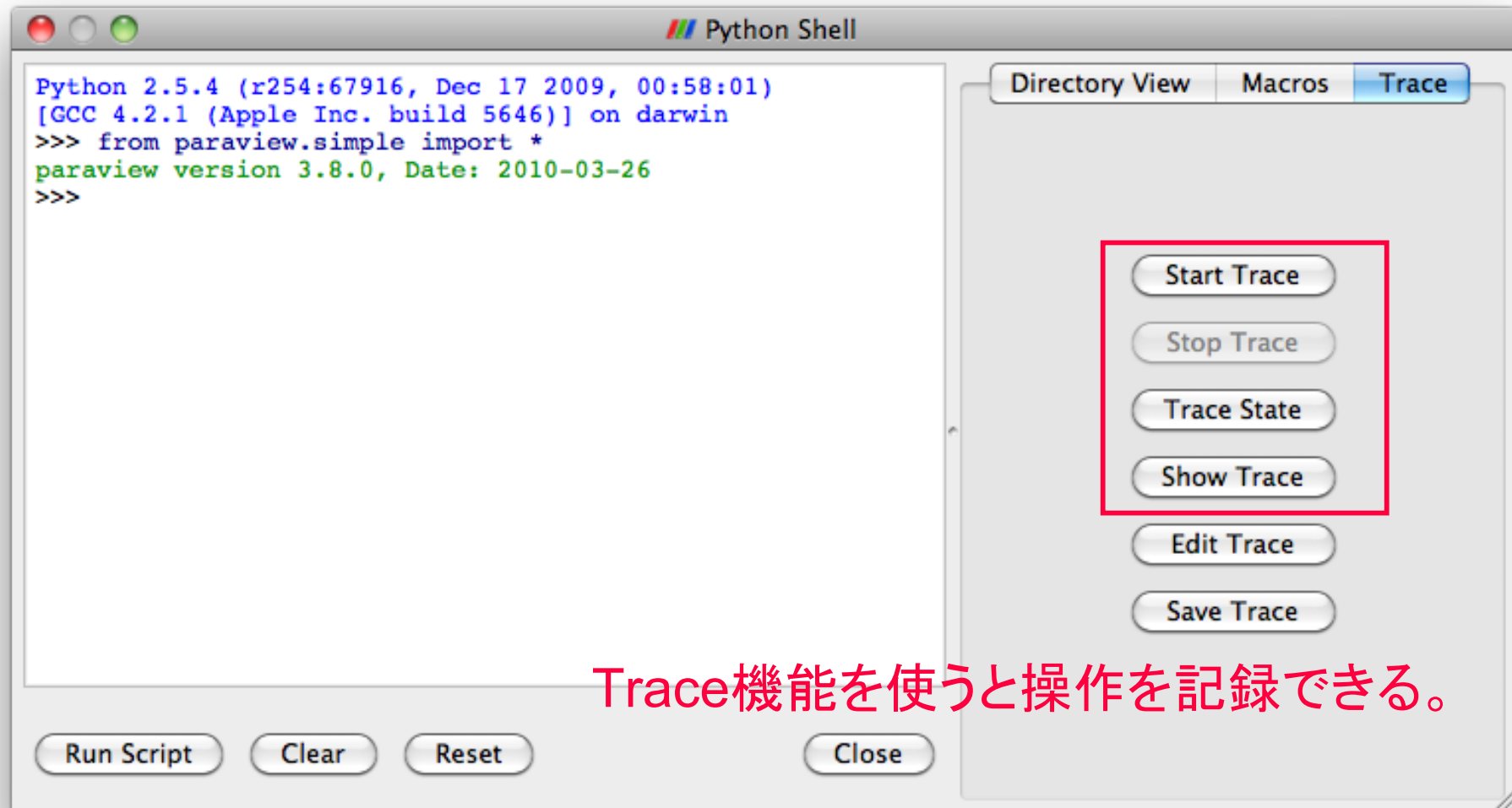
pythonスクリプトを用いたParaviewの操作

■Python Shell



pythonスクリプトを用いたParaviewの操作

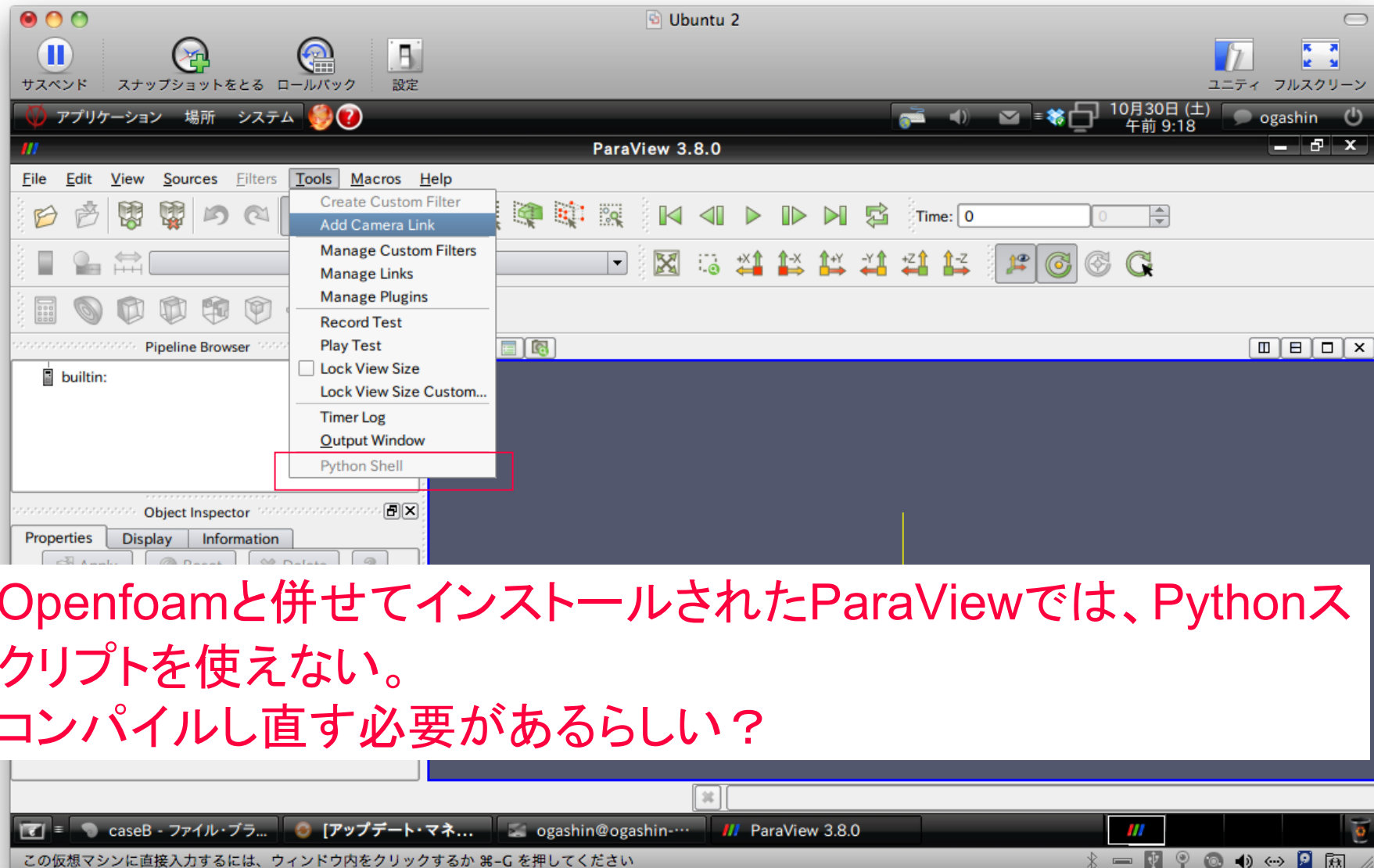
■ Trace機能



Trace機能を使うと操作を記録できる。

pythonスクリプトを用いたParaviewの操作

■ Openfoamと併せてインストールされたparaview



Openfoamと併せてインストールされたParaViewでは、Pythonスクリプトを使えない。
コンパイルし直す必要があるらしい？

pythonスクリプトを用いたParaviewの操作

■paraFoamからの結果データ出力

SaveData→ vtm形式

○ ○ ○.vtmファイルと○ ○ ○フォルダが作成される。
両方が必要。

pythonスクリプトを用いたParaviewの操作

■pythonスクリプト

```
#---- Trace output ----#
```

```
import sys  
import os
```

```
try: paraview.simple  
except: from paraview.simple import *
```

```
#対象ファイル名
```

```
test_vtm = XMLMultiBlockDataReader(  
FileName=['/Users/ogatashinya/Desktop/caseA_res.vtm'] )
```


pythonスクリプトを用いたParaviewの操作

■pythonスクリプト

```
Slice1 = Slice( SliceType="Plane" )  
Glyph4 = Glyph( GlyphType="Arrow" )  
Glyph4.Vectors = ['POINTS', 'U']  
Glyph4.GlyphType = "Arrow"
```

#画像繰り返し出力

```
for i in range(20):
```

#切断面の指定

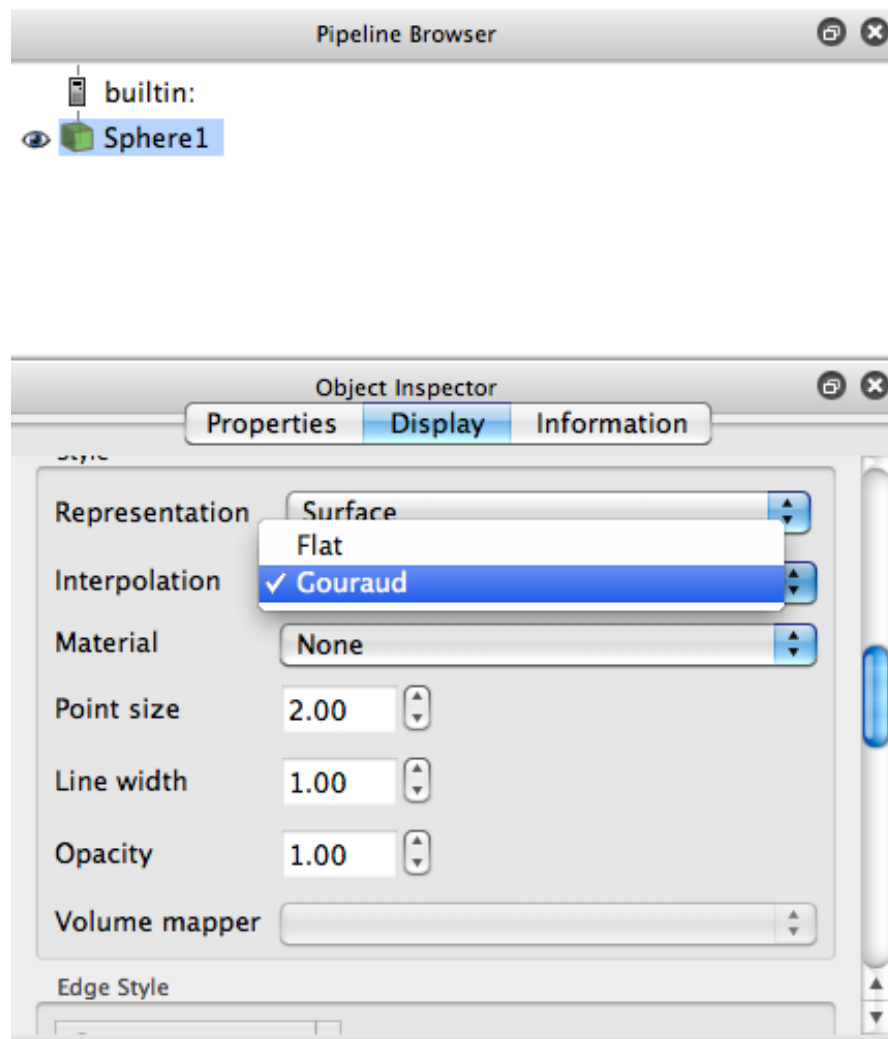
```
Slice1.SliceType.Origin = [0.25, -0.55 + i * 0.055, 0.899]
```

#画像出力

```
WriteImage('/Users/ogatashinya/Desktop/caseA_res/ %(i)d .jpg' % locals() )
```

Paraviewの補間機能

■ interpolation(補間)

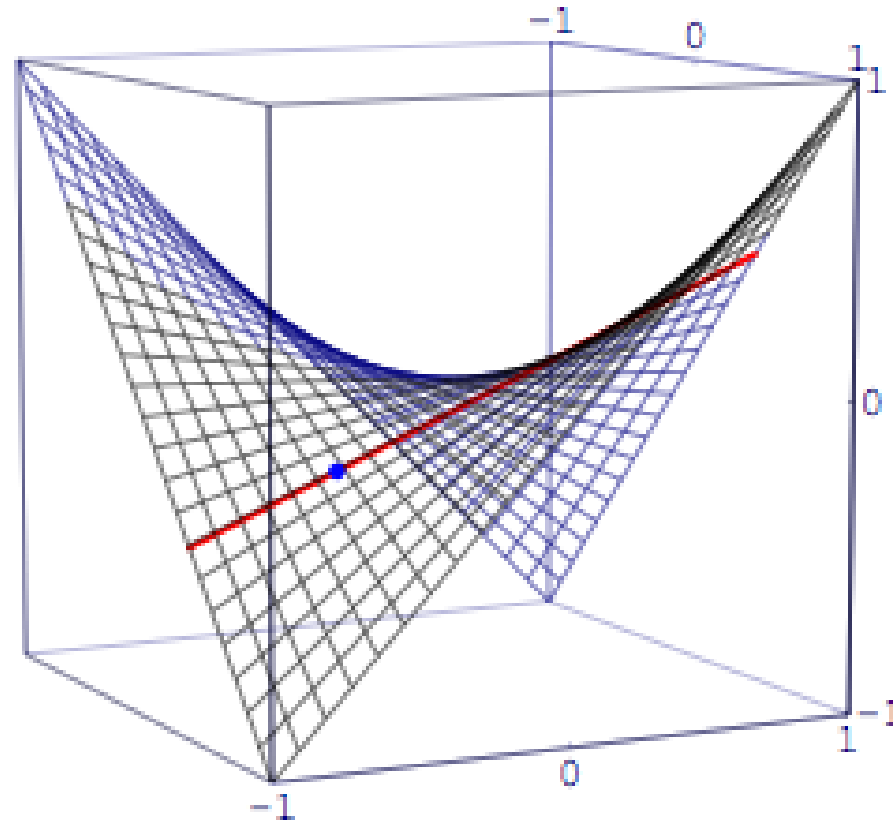


FlatとGouraudが選べる

Gouraudでは、双一次補間が
用いられる

Paraviewの補間機能

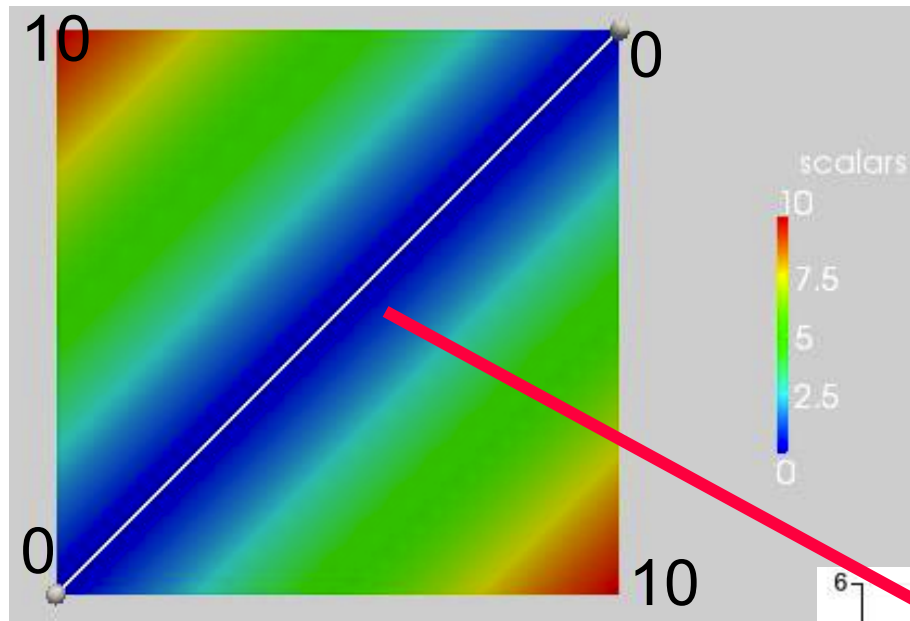
双一次曲線の例



4節点を滑らかにつなぐ

Paraviewの補間機能

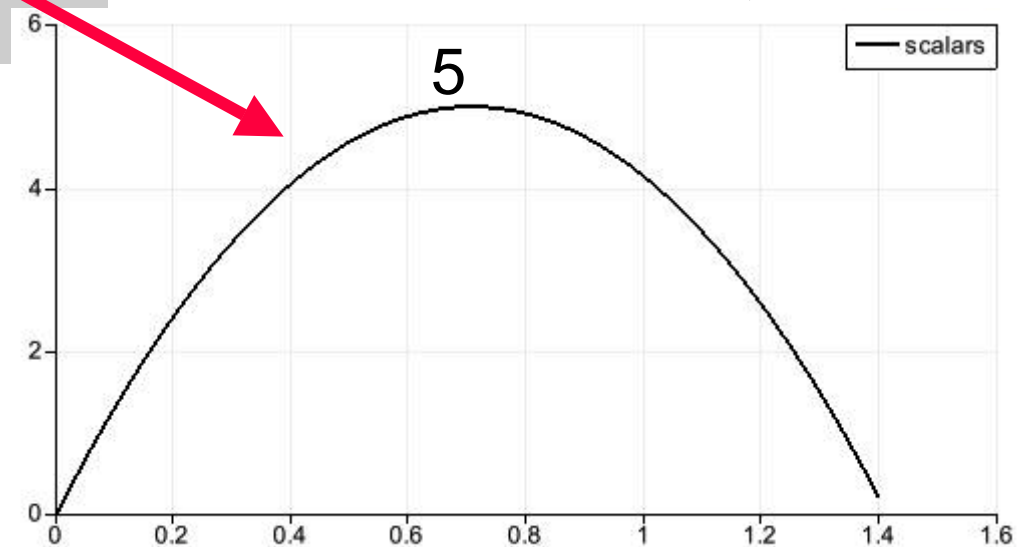
■ interpolation(補間)



3点間の線形補間

レンダリングと内部計算は
計算方法が異なるらしい。

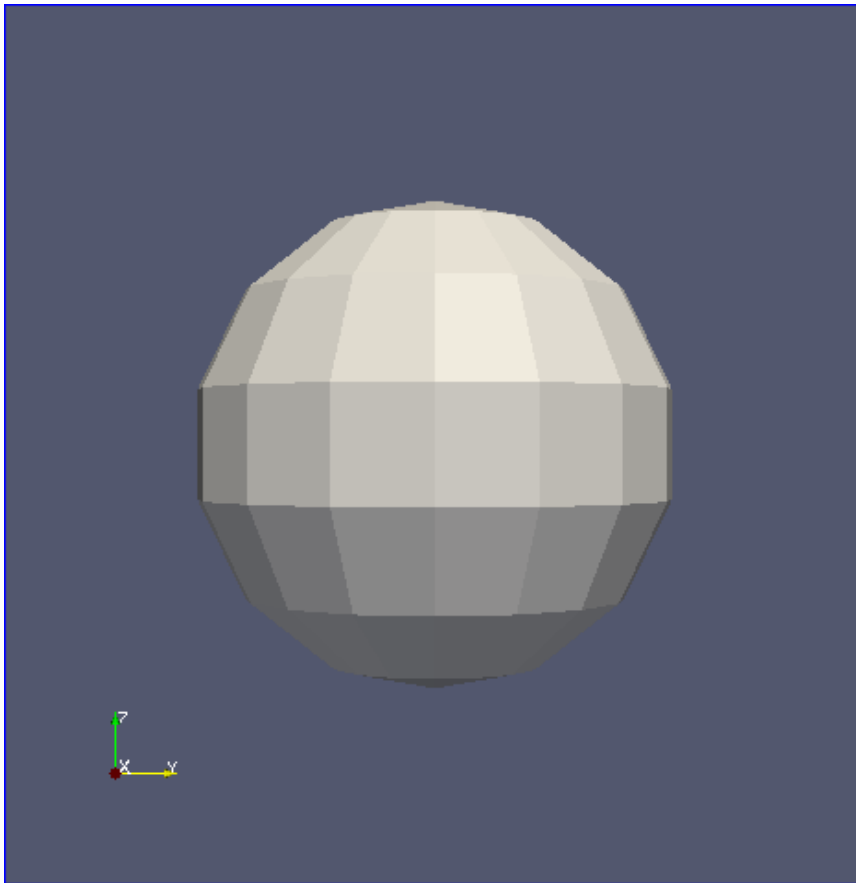
数値でみると4点間の双一次補間
がなされているように見える。



Paraviewの補間機能

■ interpolation(補間)

Flat



Gouroud

