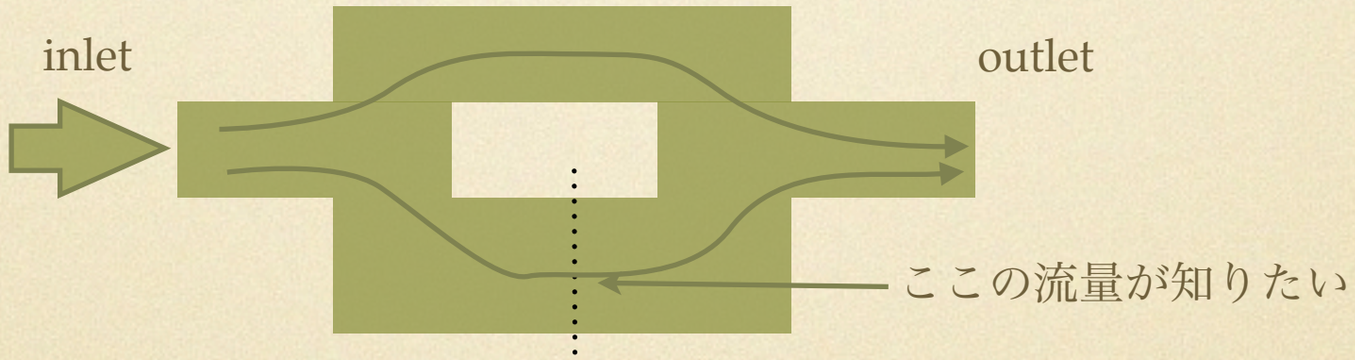
OpenFOAMの解析結果から
ParaViewで任意断面の流量を取得

y.imagawa

動機

- 出口流量ならswak4Foamを使って取得できる
- 流路が分岐する形状での流量測定方法は？



- 分岐流路にFaceSetを設定→swak4Foam(前処理)

  任意断面のFaceSetが簡単に作成できる？（難しい？）
- ParaViewで任意断面作成→流量測定(後処理)

  Sliceで任意断面は簡単に作成可能。利用機会が多い。

利用環境

ParaView標準のOpenFOAMの結果を読み込む。
paraFoamは利用していない。

OpenFOAM(ver.2.1.x)

- 計算の実行
- ケースフォルダ※に.foamファイル（空ファイル）を作成

※解析用のデータを保存しているフォルダ
0, constant, system等のフォルダがある

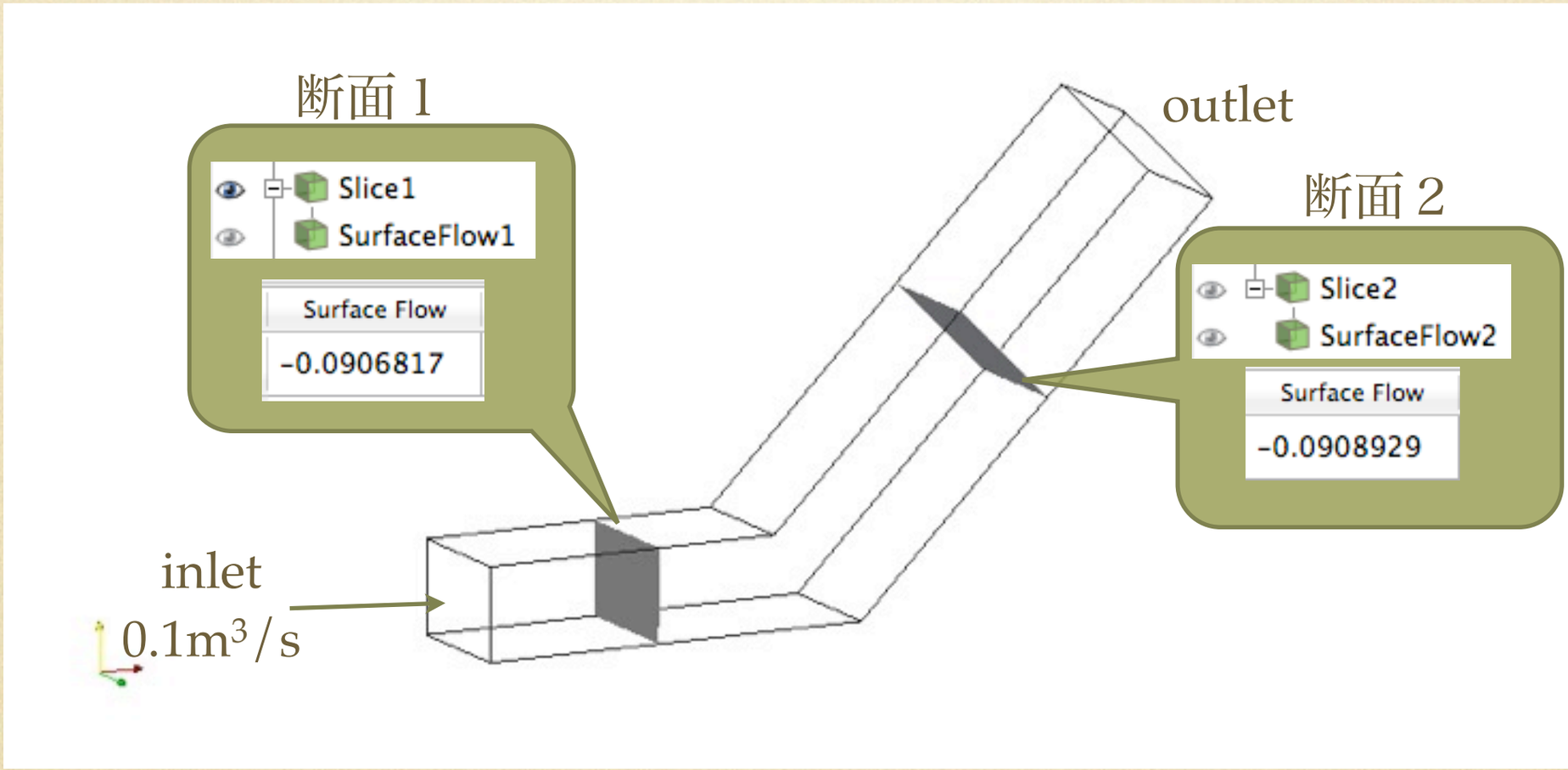
ParaView(ver. 3.14.0)

- .foamファイル（空ファイル）の読み込み

SurfaceFlowフィルタ

流量を計算してくれそうなParaViewのフィルタを試す

tutorials/incompressible/porousSimpleFoam

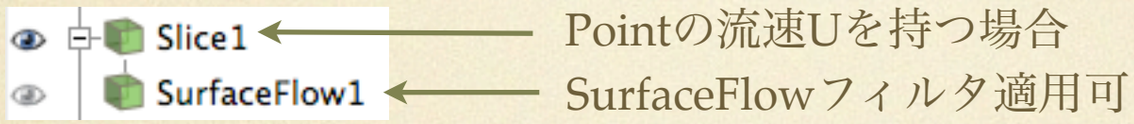


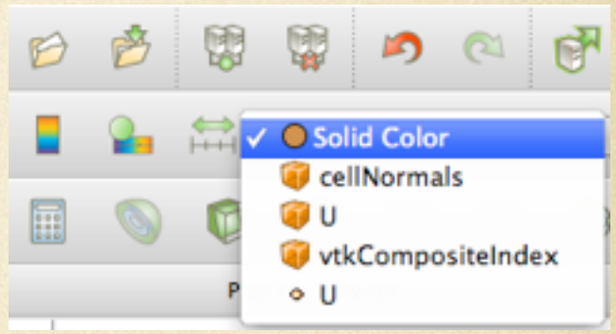
- SurfaceFlowフィルタの計算値がおかしい?
- 流量の計算値を確認するのに手間がかかる（spreadsheet view等）

SurfaceFlowフィルタ

流量の算出方法が思っていたのと違う

・Pointでのデータがあると"SurfaceFlow"を適用できる

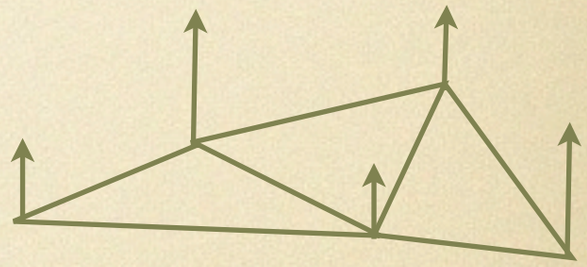




・Cellデータだけでは"SurfaceFlow"を適用できない

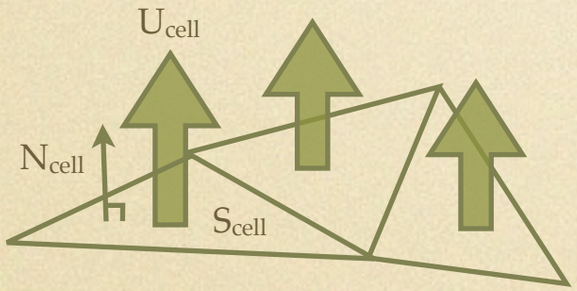
SurfaceFlowフィルタは

Pointでの流速Uを用いて流量計算している？



$SurfaceFlow=\sum U_{point}$?

・Cellのデータを利用して計算したい



$\phi=\sum S_{cell} \times U_{cell} \cdot N_{cell}$

断面の流量は

セル面積(S_{cell})×セル流速(U_{cell})・法線ベクトル(N_{cell})

の和で求まるはず

Sliceで断面を作成した後に、断面の流量を計算するフィルタを

Programmable Filterで作ってみる

作成するフィルタ

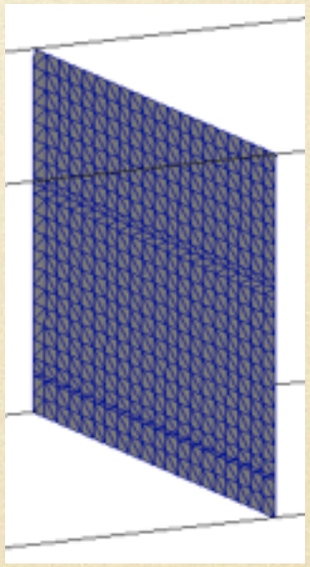
データフロー

解析データの読み込み

流量を計算したい断面をSliceで作成

Sliceで作成した断面の流量を計算

Sliceフィルタ

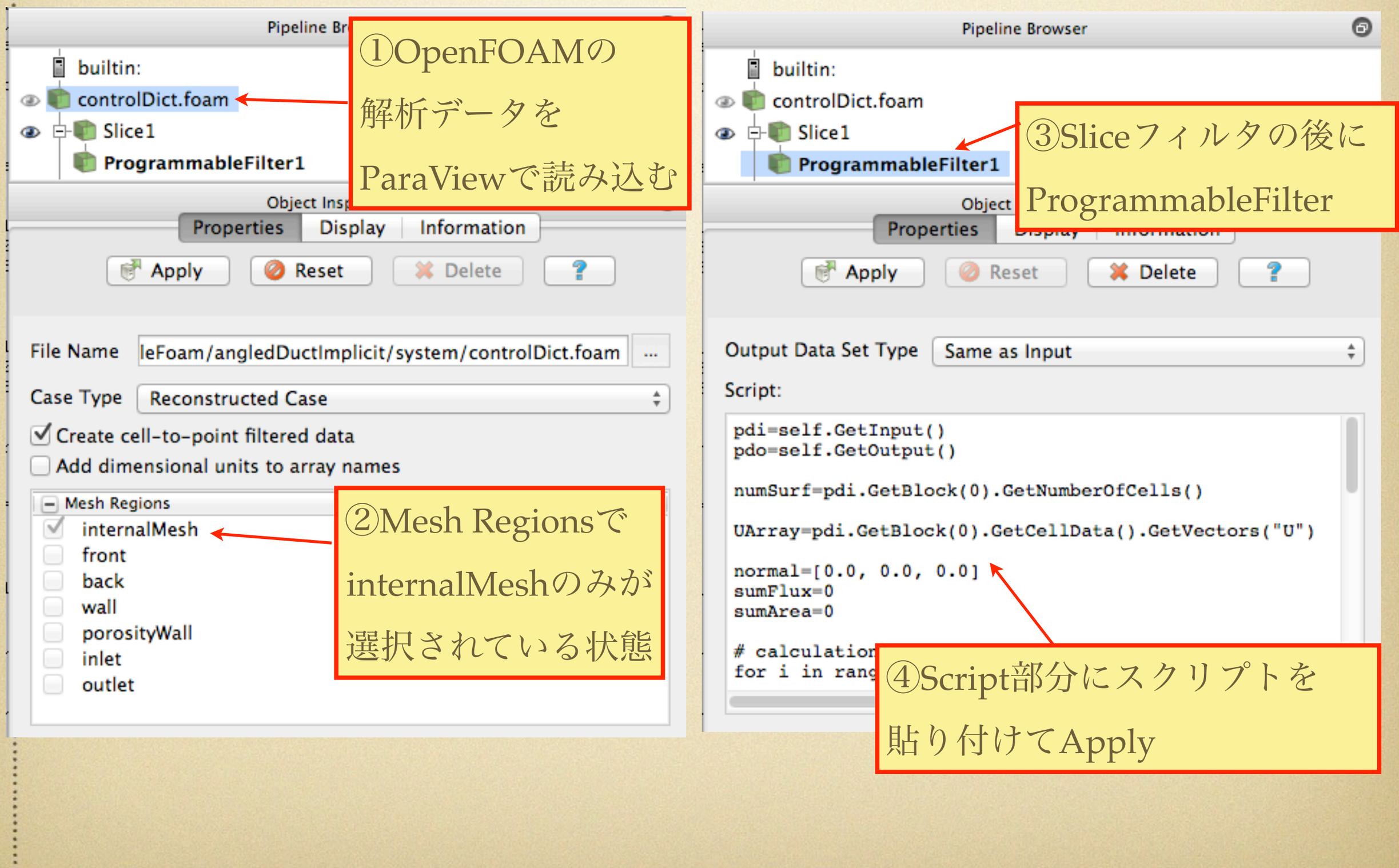


- Sliceフィルタを実行すると、断面は全て三角形に分割される
- １つ１つの三角形が持つ流速と面積を取得して計算する

Programmable Filter

- vtk-pythonを使ってフィルタを作成する機能
- 詳しくは、2011年12月のオープンCAEシンポジウムの資料を参照

今回のProgrammable Filterの使い方



フィルタスクリプト例

```
pdi=self.GetInput()
pdo=self.GetOutput()

numSurf=pdi.GetBlock(0).GetNumberOfCells()

UArray=pdi.GetBlock(0).GetCellData().GetVectors("U")

normal=[0.0, 0.0, 0.0]
sumFlux=0
sumArea=0

#断面の全てのセル(三角形)について流量を計算し、足しあわせる
for i in range(numSurf):

    #  1 つのセル(三角形)を取り出す
    cell = pdi.GetBlock(0).GetCell(i)

    # セル(三角形)を形成する 3 点の座標を取得
    p1=pdi.GetBlock(0).GetPoint(cell.GetPointId(0))
    p2=pdi.GetBlock(0).GetPoint(cell.GetPointId(1))
    p3=pdi.GetBlock(0).GetPoint(cell.GetPointId(2))
```

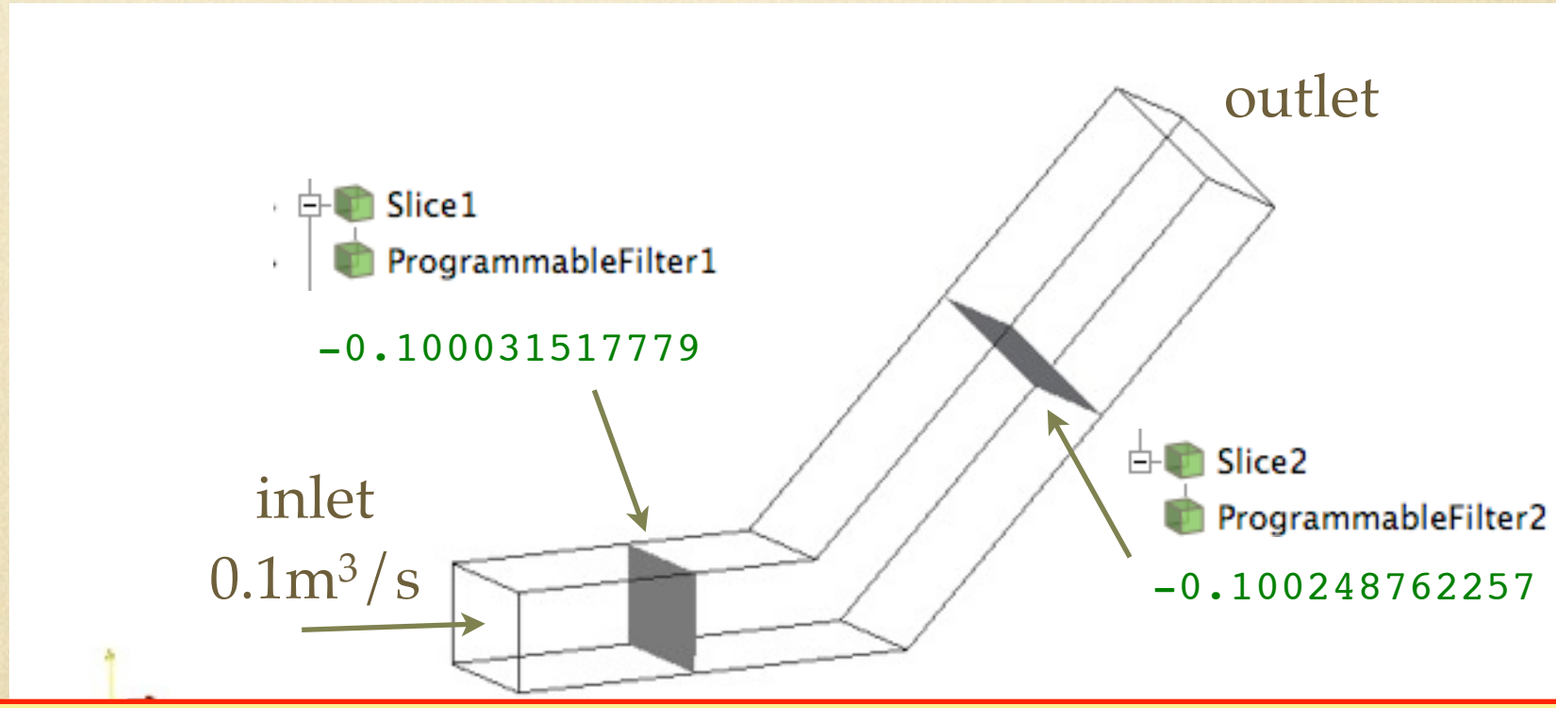
```
    # セル(三角形)のx,y,z方向の流速を取得
    U1 = UArray.GetComponent(i,0)
    U2 = UArray.GetComponent(i,1)
    U3 = UArray.GetComponent(i,2)

    # 三角形の面積を計算
    area = vtk.vtkTriangle.TriangleArea(p1, p2, p3)

    #断面の法線ベクトルを設定
    if (i==0):
        vtk.vtkTriangle.ComputeNormal(p1, p2, p3, normal)

    #流量を計算
    flux = area*(U1*normal[0]+U2*normal[1]+U3*normal[2])

    #セル流量を足し合わせる
    sumFlux += flux

    #セルの面積を足し合わせる
    sumArea += area

#断面の面積と流量の和を出力
print "Area = ", sumArea
print "Flux = ", sumFlux
```

pdi.GetBlock(0) : 前ページの「②Mesh Regions」で選択したMesh Regionのデータの0番目のことで、今回はinternalMeshのことと仮定
wallやinlet, outlet等を選択しておけば、pdi.GetBlock(1)とかするとデータを利用できる。（でも使い道がわからない）

法線ベクトル：各セル毎で法線ベクトルの向きの正負が逆になることがあるので、特定のセル(今回は0番目のセル）の法線方向に合わせる

出力結果

Output Messageウィンドウに
結果が出力される



断面流量の値は、ほぼ0.1

（値がマイナスになっているのは、Slice方向が逆のため）

まとめ

- ParaView付属のSurfaceFlowフィルタの動作が良くわからないので、Programmable Filterで、任意断面の流量を計算するフィルタを作成
- 今回のスクリプトを拡張すれば、任意断面の圧力も合わせて計算し、出力できる（と思う）
- 任意断面のデータ計算方法で、もっと簡単な方法があったら是非教えてください