ただで始める流体解析

OpenFOAM 1.7.xによる
U字パイプの内部流れ解析（LES）

まず最初にお断り事項から

2011/6/16にOpenFOAM 2.0.0がリリースされましたが，本資料でのメッシュ作成，計算結果等はバージョンは1.7.xで行っています。一部メッシュ作成において2.0.xのsHMのスライドがありますが，計算には利用していません。

U字パイプの内部流れ解析

1. 解析概要
2. モデル
3. メッシュ作成(1.7.x+snapEdge)
4. LESモデル
5. パッチ，計算条件の設定
6. 計算設定
7. 計算結果
8. まとめ
9. 参照資料

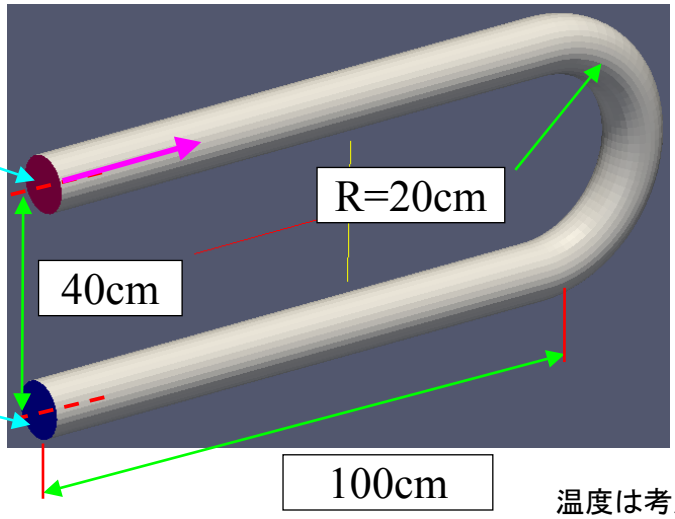
1. 解析概要

直径が等しいU字パイプの中を空気が流れる簡単な解析をLESモデルで行います。

2. モデル

inlet
直径10cm
p:勾配0
v:流入速度10m/s

outlet
直径10cm
p:大気開放
v:勾配0



温度は考慮しない

2. モデル:レイノズル数の計算

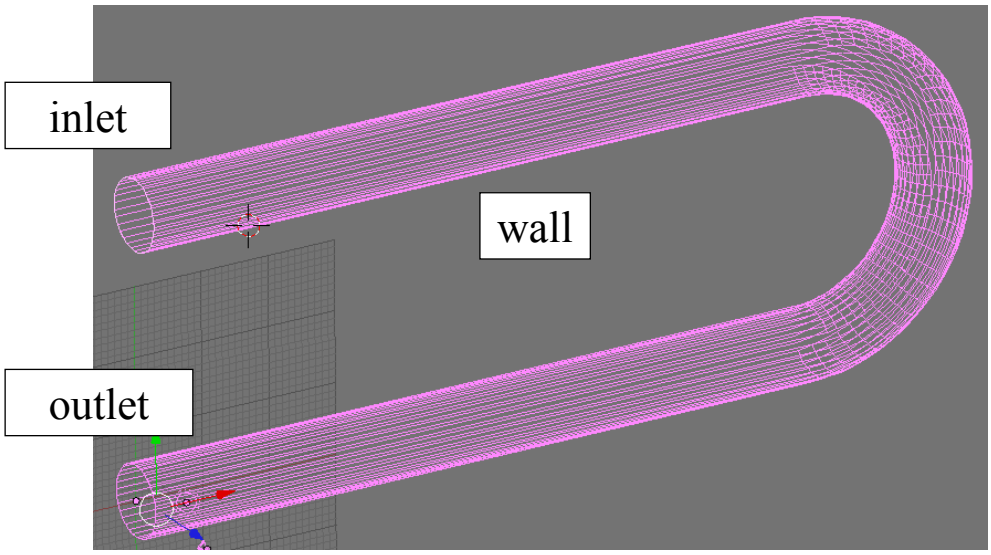
このモデルの場合は

$$\mathrm{Re} = \frac{\text{代表速度} \times \text{代表長さ}}{\text{動粘性係数}} = \frac{10 \times L}{1.54 \times 10^{-5}}$$

管内流れであれば，代表長さ(L)に管径を取る場合が一般的と言われているため，本計算ではL=0.1として，レイノズル数はRe=約65000になります。

2. モデル：Blender

モデルを単位cmで作成し，Named STLでinlet，outlet，wallの3つに分けて書き出します（第4回勉強会と同じ形状）。



2. モデル：単位変換

・作成したSTLモデル（単位cm）を計算する単位（m）に変更。scaleの引数で変換の倍率を指定。青枠が作成したモデルファイル，赤枠が単位を修正したモデルファイルとなっています。解析には修正したファイルを用います。

```
surfaceTransformPoints inlet.stl inlet_m.stl -scale "(0.01 0.01 0.01)"
surfaceTransformPoints outlet.stl outlet_m.stl -scale "(0.01 0.01 0.01)"
surfaceTransformPoints wall.stl wall_m.stl -scale "(0.01 0.01 0.01)"
```

・変換の確認：インレットの場合例

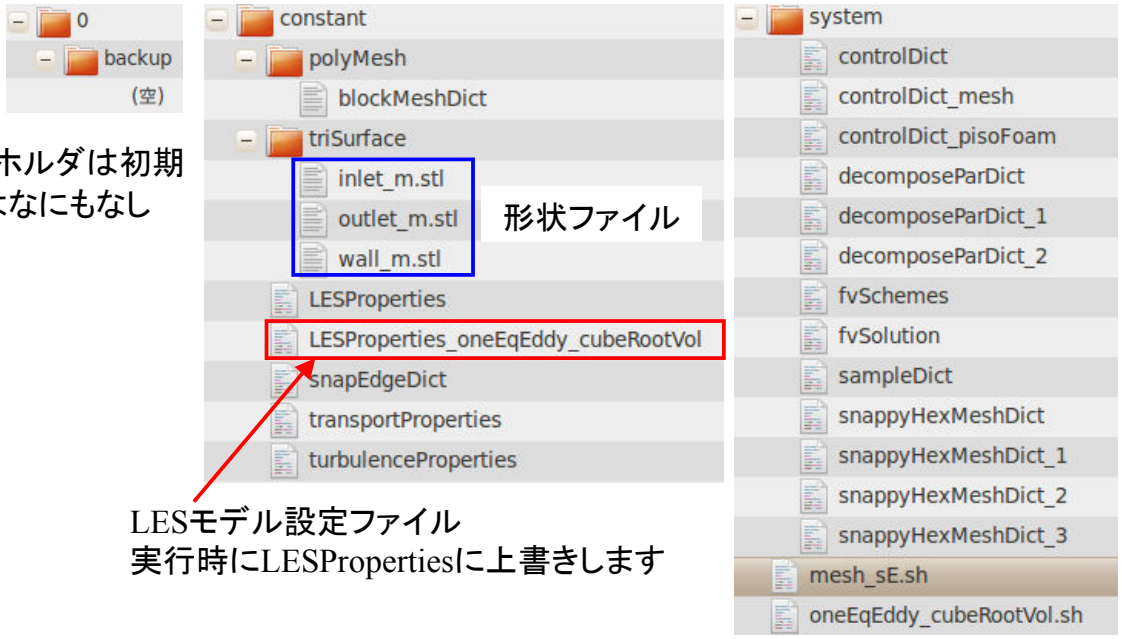
```
solid inlet
facet normal -1.0 0.0 0.0
  outer loop
    vertex 0.0 35.0958595276 0.975463747978
    vertex 0.0 35.0958595276 -0.975440263748
    vertex 0.0 34.9997940063 1.15581224236e-05
  endloop
endfacet
```

→

```
solid inlet
  facet normal -1 0 -0
    outer loop
      vertex 0 0.350959 0.00975464
      vertex 0 0.350959 -0.0097544
      vertex 0 0.349998 1.15581e-07
    endloop
  endfacet
```


3. メッシュ作成

・ファイル構成の例（初期状態）



3. メッシュ作成 snapEdgeについて

オープンCAE第4回勉強会　SM資料　18-23ページ参照

3. メッシュ作成（snapEdgeを入れる:1CUPの場合の例）

①snapEdgeに必要なファイルのリンク	ln -s constant/triSurface/wall_m.stl
	ln -s constant/triSurface/inlet_m.stl
	ln -s constant/triSurface/outlet_m.stl
②メッシュ用のcontrolDictのコピー	cp -r system/controlDict_mesh system/controlDict
③blockMeshの実施	blockMesh　→ スライド13に例
④sHM用のsnappyHexMeshのコピー	cp -r system/snappyHexMeshDict_1 system/snappyHexMeshDict
⑤sHM(castellatedMesh)の実施	snappyHexMesh　→ スライド15に例
⑥snapEdgeの実施	snapEdge
⑦sHM用のsnappyHexMeshのコピー	cp -r system/snappyHexMeshDict_2 system/snappyHexMeshDict
⑤sHM(doLayers)の実施	snappyHexMesh　→ スライド18に例
⑥ checkMeshの実施	checkMesh
⑦リナンバリングの実施	renumberMesh -latestTime
⑧必要なメッシュファイルの移動	mv constant/polyMesh constant/polyMesh_blockMesh
	mv 5/polyMesh constant
⑨テンポラリファイルの削除	rm -r 1
	rm -r 2
	rm -r 3
	rm -r 4
	rm -r 5

適当なshファイルに書いておいてbatchで実施

checkMeshでエラーがある場合には，5のホルダにsetホルダが出来ます。この下にエラーした要素の情報が入るため，rm –r 5を行うとそのデータが消えるため注意して下さい。

3．メッシュ作成（snapEdgeを入れる:4CUPの場合の例）

①snapEdgeに必要なファイルのリンク	ln -s constant/triSurface/wall_m.stl ln -s constant/triSurface/inlet_m.stl ln -s constant/triSurface/outlet_m.stl
②メッシュ用のcontrolDictのコピー	cp -r system/controlDict_mesh system/controlDict
③blockMeshの実施	blockMesh　スライド13に例
④領域分割用のdecomposeParDictのコピー	cp -r system/decomposeParDict_1 system/decomposeParDict　スライド19に例
⑤decomposeParの実施	decomposePar
⑥ファイル（stl）のコピー	cp -r constant/triSurface/ processor0/constant/ cp -r constant/triSurface/ processor1/constant/ cp -r constant/triSurface/ processor2/constant/ cp -r constant/triSurface/ processor3/constant/
⑦sHM用のsnappyHexMeshのコピー	cp -r system/snappyHexMeshDict_1 system/snappyHexMeshDict
⑧sHM(castellatedMesh)の実施	mpirun -np 4 snappyHexMesh -parallel　スライド15に例
⑨reconstructParMeshの実施	reconstructParMesh -time 2 -mergeTol 1e-6
⑩不要ファイルの削除	rm -r processor0 rm -r processor1 rm -r processor2 rm -r processor3
⑪snapEdgeの実施	snapEdge

3．メッシュ作成（snapEdgeを入れる:4CUPの場合の例 続き）

⑫領域分割用のdecomposeParDictのコピー	cp -r system/decomposeParDict_2 system/decomposeParDict　スライド20に例
⑬decomposeParの実施	decomposePar
⑭sHM用のsnappyHexMeshのコピー	cp -r system/snappyHexMeshDict_2 system/snappyHexMeshDict
⑮sHM(doLayers)の実施	mpirun -np 4 snappyHexMesh -parallel　スライド18に例
⑯reconstructParMeshの実施	reconstructParMesh -time 4 -mergeTol 1e-6
⑰checkMeshの実施	checkMesh
⑱リナンバリングの実施	renumberMesh -time 4
⑲必要なメッシュファイルの移動	mv constant/polyMesh constant/polyMesh_blockMesh mv 5/polyMesh constant
⑳テンポラリファイルの削除	rm -r 2 rm -r 3 rm -r 4 rm -r 5 rm -r processor0 rm -r processor1 rm -r processor2 rm -r processor3

checkMeshでエラーがある場合には，5のホルダにsetホルダが出来ます。この下にエラーした要素の情報が入るため，rm –r 5を行うとそのデータが消えるため注意して下さい。

適当なshファイルに書いておいてbatchで実施

3. メッシュ作成

controlDict_meshの例

```
application     simpleFoam;
startFrom       latestTime;
startTime       0;
stopAt          endTime;
endTime         800;
deltaT          1;
writeControl    timeStep;
writeInterval   200;
purgeWrite      0;
writeFormat     ascii;
writePrecision  6;
writeCompression compressed;
timeFormat      general;
timePrecision   6;
runTimeModifiable yes;
```

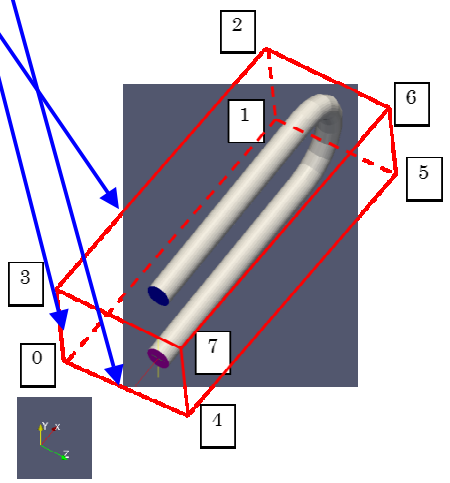
今回の解析では，pisoFoamによるLES解析を最終的には実施しますが，メッシュ作成にはsimpleFoamで使ったファイルを使い回しています。

3. メッシュ作成

blockMeshDictの例

```
FoamFile
{
    version     2.0;
    format      ascii;
    class       dictionary;
    object      blockMeshDict;
}

convertToMeters 0.01;

vertices
(
    (-10 -10 -10) // 0
    (130 -10 -10) // 1
    (130  50 -10) // 2
    (-10  50 -10) // 3
    (-10 -10  10) // 4
    (130 -10  10) // 5
    (130  50  10) // 6
    (-10  50  10) // 7
);

blocks
(
    hex (0 1 2 3 4 5 6 7) (28 12 4) simpleGrading (1 1 1)

);

edges ();
patches ();
mergePatchPairs ();
```

分割数の設定



参考：ユーザーガイド　2.1.1.1 メッシュ生成

3. メッシュ作成

snappyHexMeshDict_1の例

```
FoamFile
{
   version     2.0;
   format      ascii;
   class       dictionary;
   object      snappyHexMeshDict;
}

castellatedMesh true;
snap            true;
addLayers       false;

geometry
{
        wall_m.stl
        {
                type triSurfaceMesh;
                name wall;
        }

        inlet_m.stl
        {
                type triSurfaceMesh;
                name inlet;
        }
        outlet_m.stl
        {
                type triSurfaceMesh;
                name outlet;
        }

        box1
        {
                type searchableBox;
                min (-10 -20 -20);
                max (140  60  20);
        }
};

castellatedMeshControls
{
   maxLocalCells 1000000;
   maxGlobalCells 2000000;
   minRefinementCells 10;
   maxLoadUnbalance 0.10;
   nCellsBetweenLevels 3;
   features
   (
      //{
      //    file "someLine.eMesh";
      //    level 2;
      //}
   );
```

モデルに使うstlの設定

refinementRegionsを設定するための領域

3. メッシュ作成

snappyHexMeshDict_1の例

```
refinementSurfaces
   {
      wall1
      {
         level (3 3);
      }
      inlet
      {
         level (3 3);
      }
      outlet
      {
         level (3 3);
      }
   }

   resolveFeatureAngle 30;
```

メッシュの密度設定

管壁面部のメッシュ密度を同じに切る為levelのmin，maxを同じにしている

```
refinementRegions
   {
      box1
      {
         mode inside;
         levels ((1E15 3));
      }
   }
   locationInMesh (0.05 -0.01 -0.01);
}

snapControls
{
   nSmoothPatch 3;
   tolerance 4.0;
   nSolveIter 30;
   nRelaxIter 5;
}

addLayersControls
{
   relativeSizes true;
   layers
   {
      wall1_wall
      {
         nSurfaceLayers 6;
      }
```

メッシュの切り始め点

レイヤー層の設定

3. メッシュ作成

snappyHexMeshDict_1の例

```
    expansionRatio 1.0;
    finalLayerThickness 0.3;
    minThickness 0.05;
    nGrow 1;
    featureAngle 30;
    nRelaxIter 3;
    nSmoothSurfaceNormals 1;
    nSmoothNormals 3;
    nSmoothThickness 10;
    maxFaceThicknessRatio 0.5;
    maxThicknessToMedialRatio 0.3;
    minMedianAxisAngle 130;
    nBufferCellsNoExtrude 0;
    nLayerIter 50;
}

meshQualityControls
{
    maxNonOrtho 65;
    maxBoundarySkewness 20;
    maxInternalSkewness 4;
    maxConcave 80;
    minFlatness 0.5;
    minVol 1e-13;
    minArea -1;
    minTwist 0.02;
    minDeterminant 0.001;
    minFaceWeight 0.02;
    minVolRatio 0.01;
    minTriangleTwist -1;
    nSmoothScale 4;
    errorReduction 0.75;
}
debug 0;
mergeTolerance 1E-6;

//
*****************************************************
******************** //
```

パラメータはほぼデフォルトを使用

3. メッシュ作成

snappyHexMeshDict_2の例

```
FoamFile
{
    version     2.0;
    format      ascii;
    class       dictionary;
    object      snappyHexMeshDict;
}

castellatedMesh false;
snap            false;
addLayers       true;
```

これ以降はsnappyHexMeshDict_1と同じ

3．メッシュ作成

decomposeParDict_1の例

```
numberOfSubdomains 4;

method        hierarchical;
//method        metis;
//method        parMetis;

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

hierarchicalCoeffs
{
   n            (2 2 1);
   delta         0.001;
   order         xyz;
}

manualCoeffs
{
   dataFile       "cellDecomposition";
}
```

コア数が4のため4としています。

```
metisCoeffs
{
   //n              (2 1 1);
   //cellWeightsFile     "constant/cellWeightsFile";
}
```

ユーザガイドの3.4.1に詳細がのっています。メッシュ分割にscotch，metisの方法を使うとなぜかエラーが出たので，ここではhierarchicalで分割をしています

Ver2.0では，無くなっているようです。

3．メッシュ作成

decomposeParDict_2の例

```
numberOfSubdomains 4;

method        hierarchical;
//method        metis;
//method        parMetis;

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

hierarchicalCoeffs
{
   n            (2 2 1);
   delta         0.001;
   order         xyz;
}

manualCoeffs
{
   dataFile       "cellDecomposition";
}
```

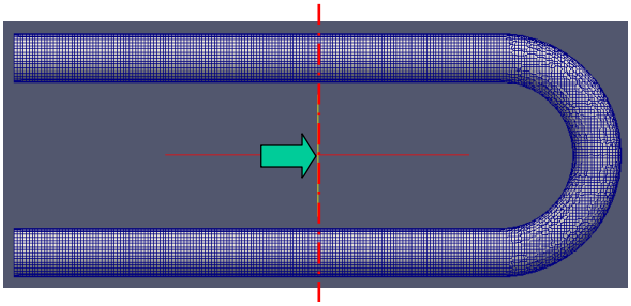
コア数が4のため4としています。

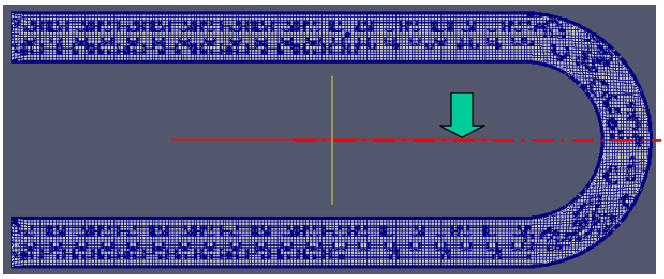
```
metisCoeffs
{
   //n              (2 1 1);
   //cellWeightsFile     "constant/cellWeightsFile";
}
```

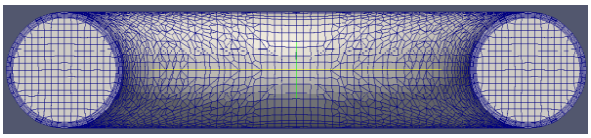
今回はhierarchicalを利用しています。scotch，metisも利用できますが，ほんの少し計算時間が速くなる程度でした。

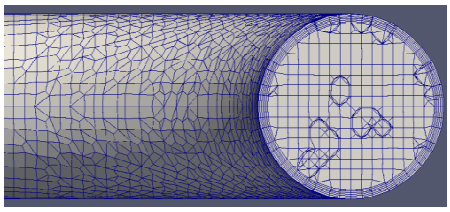
Ver2.0では，無くなっているようです。

3. メッシュ作成　1.7.x+snapEdge









このようなメッシュが出来ます。

3. メッシュ作成　OpenFOAM 2.0

OpenFOAM 2.0の新機能としてsanppyHexMeshに入ってきたFeature Edgeの機能をちょっと試してみました。

スライド23-26にメッシュを作る設定，出来上がったメッシュを示しています。設定が悪いのか，出来たメッシュはあれれ？？？でした。

回避策等はどうすれば良いか，お分かりの方は教えて下さい。

3．メッシュ作成（1CUPの場合の例）

2.0.x

①メッシュ用のcontrolDictのコピー	cp -r system/controlDict_mesh system/controlDict
②blockMeshの実施	blockMesh
③feature edgeの処理 （2.0からの新機能）	surfaceFeatureExtract -includedAngle 150 -writeObj constant/triSurface/wall_m.stl wall_m
④snappyHexMeshの実施	snappyHexMesh
⑤ checkMeshの実施	checkMesh
⑥リナンバリングの実施	renumberMesh -latestTime
⑦必要なメッシュファイルの移動	mv constant/polyMesh constant/polyMesh_blockMesh mv 4/polyMesh constant
⑨テンポラリファイルの削除	rm -r 1 rm -r 2 rm -r 3 rm -r 4

適当なshファイルに書いておいてbatchで実施

3．メッシュ作成（snappyHexMeshの例）

2.0.x

```
FoamFile
{
   version     2.0;
   format      ascii;
   class       dictionary;
   object      snappyHexMeshDict;
}

castellatedMesh true;
snap            true;
addLayers       true;

geometry
{
wall_m.stl
{
                    type triSurfaceMesh;
                    name wall;
}
inlet_m.stl
{
                    type triSurfaceMesh;
                    name inlet;
}
outlet_m.stl
{
                    type triSurfaceMesh;
                    name outlet;
}
      box1
      {
          type searchableBox;
          min (-10 -20 -20);
          max (140  60  20);
      }
};

castellatedMeshControls
{
   maxLocalCells 1000000;
   maxGlobalCells 2000000;
   minRefinementCells 0;
   nCellsBetweenLevels 1;
   features
   (
      {
         file "wall_m.eMesh";
         level 0;
      }
   );
   refinementSurfaces
   {
      wall
      {
         // Surface-wise min and max refinement level
         level (3 3);
      }

      inlet
      {
         // Surface-wise min and max refinement level
         level (3 3);
      }
outlet
      {
         // Surface-wise min and max refinement level
         level (3 3);
      }
   }
   resolveFeatureAngle 30;
   refinementRegions
   {
      box1
      {
         mode inside;
         levels ((1E15 3));
      }
   }
   locationInMesh (0.05 -0.01 -0.01);
   allowFreeStandingZoneFaces true;
}

snapControls
{
   nSmoothPatch 3;
   tolerance 1.0;
   nSolveIter 300;
   nRelaxIter 5;
   nFeatureSnapIter 10;
}
```

3. メッシュ作成（snappyHexMeshの例）

2.0.x

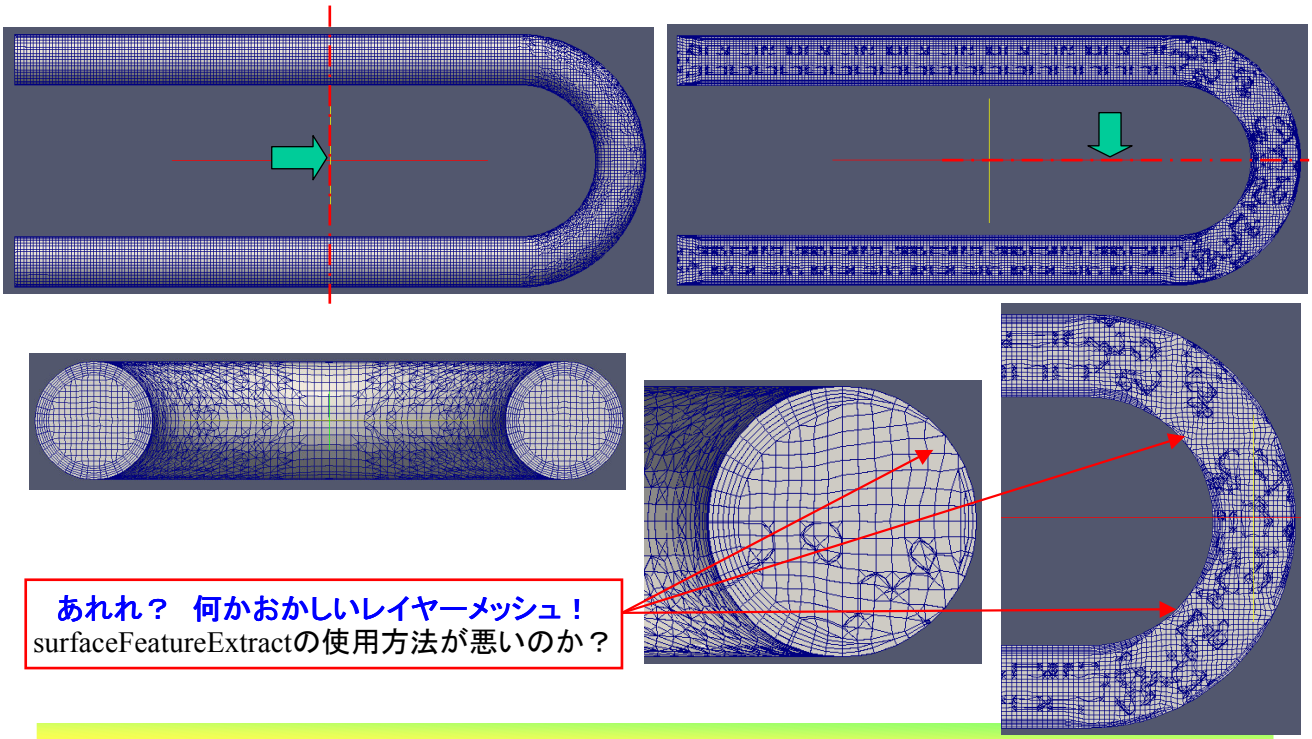
```
addLayersControls
{
   relativeSizes true;
   layers
   {
     wall_wall
      {
         nSurfaceLayers 6;
      }
   }
   expansionRatio 1.0;
   finalLayerThickness 0.3;
   minThickness 0.05;
   nGrow 0;
   featureAngle 30;
   nRelaxIter 3;
   nSmoothSurfaceNormals 1;
   nSmoothNormals 3;
   nSmoothThickness 10;
   maxFaceThicknessRatio 0.5;
   maxThicknessToMedialRatio 0.3;
   minMedianAxisAngle 90;
   nBufferCellsNoExtrude 0;
   nLayerIter 50;
   nRelaxedIter 20;
}
```

```
meshQualityControls
{
   maxNonOrtho 65;
   maxBoundarySkewness 20;
   maxInternalSkewness 4;
   maxConcave 80;
   minVol 1e-13;
   minTetQuality 1e-30;
   minArea -1;
   minTwist 0.05;
   minDeterminant 0.001;
   minFaceWeight 0.05;
   minVolRatio 0.01;
   minTriangleTwist -1;
   nSmoothScale 4;
   errorReduction 0.75;
}

debug 0;
mergeTolerance 1E-6;
```

各パラメータは
tutorials/mesh/snappyHexMesh/flange
を参照して設定しています。

3. メッシュ作成

2.0.x



あれれ？　何かおかしいレイヤーメッシュ！
surfaceFeatureExtractの使用方法が悪いのか？

4. LESモデル:どんなモデルが使えるか？

1.7.x

No	LESModel	
1	DeardorffDiffStress	DearDorff差応力モデル
2	LRRDiffStress	LRR差応力モデル
3	Smagorinsky	Smagorinskyモデル
4	SpalartAllmaras	Spalart-Allmarasモデル
5	SpalartAllmarasDDES	Spalart-Allmaras 遅延型分離渦シミュレーションモデル
6	SpalartAllmarasIDDES	Spalart-Allmaras 改良遅延型分離渦シミュレーションモデル
7	dynOneEqEddy	同時k方程式粘性モデル
8	homogeneousDynSmagorinsky	同時Smagorinskyモデル
9	kOmegaSSTSAS	k-ω-SSTスケール適応シミュレーションモデル
10	laminar	
11	locDynOneEqEddy	局部同時k方程式粘性モデル
12	mixedSmagorinsky	Smagorinskyとスケール相似の混合モデル
13	oneEqEddy	k方程式粘性モデル
14	spectEddyVisc	スペクトル渦粘性モデル

たくさんあるモデル！

表のように14のモデルが使える。ただし，1.7.1のマニュアル第3章とは少し異なっています。
その理由は，使用バージョンが1.7.xで，右のようなアナウンスが出ているため。

Renamed dynSmagorinsky to homogeneousDynSmagorinsky to avoid further confusion concerning the averaging used in this model.
Removed dynMixedSmagorinsky.
https://github.com/OpenCFD/OpenFOAM-1.7.x/commit/890df193ffbda24a2ca85686b7890d543f61531d

4. LESモデル:モデルの見つけ方

どんなモデルが使えるか分からない場合には，設定をわざと間違えると，選択できるモデルが示されます。
チュートリアルにあるincompressible/pisoFoam/les/pitzDailyのconstant/LESpropertiesのLESModelを間違えて計算を実行します。

```
LESModel        AASmagorinsky;

delta           cubeRootVol;

printCoeffs     on;

cubeRootVolCoeffs
{
    deltaCoeff      1;
}
```

間違ったLESモデルをわざと設定する

```
--> FOAM FATAL ERROR:
Unknown LESModel type AASmagorinsky

Valid LESModel types are :

14
(
DeardorffDiffStress
LRRDiffStress
Smagorinsky
SpalartAllmaras
SpalartAllmarasDDES
SpalartAllmarasIDDES
dynOneEqEddy
homogeneousDynSmagorinsky
kOmegaSSTSAS
laminar
locDynOneEqEddy
mixedSmagorinsky
oneEqEddy
spectEddyVisc
)
```

有効なモデルが書き出される

4. LESモデル：今回計算に利用したモデル

第3回勉強会でTM様からSmagorinsky，oneEqEddyの2モデルで十分とお教え頂いたので，まずはこの2モデルとこれらのdynモデルの4モデルを計算に利用します。（dynSmagorinskyモデルは，1.7.xでは，homogeneousDynSmagorinskyとなっています）

乱流値の初期設定

項目	値
I(みだれ強さ)	0.05
U(流入速度 m/s)	10
k	0.375

（オープンCAE第3回勉強会　SM資料　11/122ページ参照）

4. LESモデル：どんなフィルタモデルが使えるか？

LESモデルを使う場合には，フィルタサイズΔ（フィルタ幅のスケール）というものを設定する必要があります。constant/LESPropertiesにあるdeltaで設定します。

LESdelta	IDDESDelta
	Prandtl
	cubeRootVol
	maxDeltaxyz
	smooth
	vanDriest

LESdeltaも先のLESモデルと同じようにわざと間違えて有効な設定をチェックする

```
--> FOAM FATAL ERROR:
Unknown LESdelta type AcubeRootVol

Valid LESdelta types are :

6
(
IDDESDelta
Prandtl
cubeRootVol
maxDeltaxyz
smooth
vanDriest
)
```

有効な設定が書き出される

$$\upsilon_{SGS} = (C_S\Delta)^2|\bar{S}| \qquad |\bar{S}| = (\bar{S}_{ij})(\bar{S}_{ij})^{\frac{1}{2}} \qquad \bar{S}_{ij} = \frac{1}{2}\left(\frac{\partial \bar{u}_i}{\partial x_j} + \frac{\partial \bar{u}_j}{\partial x_i}\right)$$

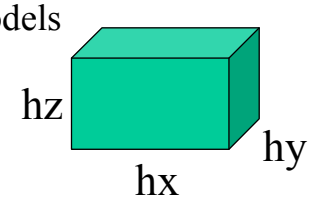
構造解析のひずみテンソルのようなもの?

4. LESモデル：今回計算に利用したフィルタ

今回は，maxDeltaxyz，cubeRootVol，vanDriestの3種類とします。
各モデルでフィルタ幅をどう決めているか？

① maxDeltaxyz
src/tubulenceModels/LES/LESdeltas/maxDeltaxyzを見ると
maxDeltaxyz takes the maximum of the three dimensions per cell
max(hx, hy, hz). Valid for structures hexahedral cells only
Δ=hx，hy，hzで一番大きいもの

② cubeRootVol
src/tubulenceModels/LES/LESdeltas/cubeRootVolを見ると
Simple cube-root of cell volume delta used in LES models
$\Delta = (hx \times hy \times hz)^{1/3}$



4. LESモデル：今回計算に利用したフィルタ

③ vanDriest（ΔはcubeRootVolを採用）
src/tubulenceModels/incompressible/LES/vanDriestDeltaを見ると
Simple cube-root of cell volume delta used in incompressible LES models
ソースをチェックすると，

```
delta_ = min
 (
    static_cast<const volScalarField&>(geometricDelta_()),
    (kappa_/Cdelta_)*((scalar(1) + SMALL) - exp(-y/ystar/Aplus_))*y
 );
```

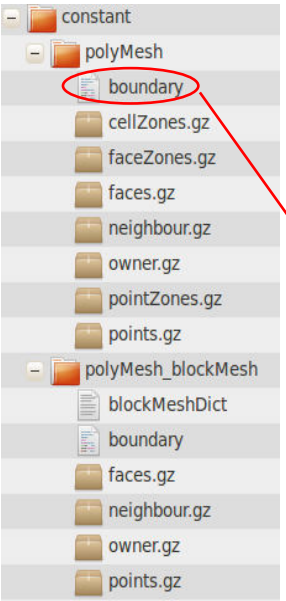
$$\Delta = \min\left[\frac{\kappa y}{C_\Delta}\left(1 - e^{\frac{-y^+}{A^+}}\right), V^{1/3}\right] \quad (13)$$

where κ is the von Karman constant, y is the distance to the closest wall, C_Δ and A^+ are a model constants ($C_\Delta = 0.158$ and $A^+ = 26$ by default in OpenFOAM®), V is the volume of the computational cell. This approach is in the spirit of Van Driest (Van Driest, 1956) damping function for μ_{SGS} (De Villiers, 2006)

壁面近傍でsgs渦粘性が0になるように減衰効果を入れています

5. パッチ，計算条件の設定

constant/polyMesh/boundaryの設定を手修正




```
4
(
    defaultFaces
    {
        type        empty;
        nFaces      0;
        startFace   576078;
    }
    wall_wall
    {
        type        wall;
        nFaces      18228;
        startFace   576078;
    }
    inlet_inlet
    {
        type        wall;
        nFaces      200;
        startFace   594306;
    }
    outlet_outlet
    {
        type        wall;
        nFaces      200;
        startFace   594506;
    }
)
```

```
4
(
    defaultFaces
    {
        type        empty;
        nFaces      0;
        startFace   576078;
    }
    wall_wall
    {
        type        wall;
        nFaces      18228;
        startFace   576078;
    }
    inlet_inlet
    {
        type        patch;
        nFaces      200;
        startFace   594306;
    }
    outlet_outlet
    {
        type        patch;
        nFaces      200;
        startFace   594506;
    }
)
```


5. パッチ，計算条件の設定

- 適当なチュートリアルからU,p,k,nuSgsなどを0のディレクトリ下にコピーします。今回は，tutorials/incompressible/pisoFoam/les/pitzDaily/0から　U, p,　k, nuSgsを持ってきています。ただ，そのままでは解析に利用できません。
- pyFoamCreateBoundaryPatches.pyでpatchを設定し，条件書き換えます。

```
pyFoamCreateBoundaryPatches.py --clear-unused 0/U
pyFoamCreateBoundaryPatches.py --clear-unused 0/p
pyFoamCreateBoundaryPatches.py --clear-unused 0/k
pyFoamCreateBoundaryPatches.py --clear-unused 0/nuSgs
```

DEXCS2010にはpyFoamは導入済，その他では事前にpyFoamの組み込みが必要となります

5. パッチ，計算条件の設定　U

```
dimensions [ 0 1 -1 0 0 0 0 ];

internalField uniform (0 0 0);

boundaryField
{
 outlet_outlet
 {
  type          inletOutlet;
  inletValue     uniform (0 0 0);
  value          uniform (0 0 0);
 }
 wall_wall
 {
  type fixedValue;
  value uniform (0 0 0);
 }
 defaultFaces
 {
  type empty;
 }
 inlet_inlet
  {
   type          turbulentInlet;
   referenceField  uniform (10 0 0);
   fluctuationScale (0.02 0.01 0.01);
   value          uniform (10 0 0);
  }
 }
```

OpenFOAM-1.7.X/src/finiteVolume/fields/fvPatchFields/derived/tubulentinlet/turbulentinletFvPatchField.H

Description
Generate a fluctuating inlet condition by adding a random component to a reference (mean) field.
Input:
referenceField
Mean field.
fluctuationScale
RMS fluctuation, provided as the fraction of the mean field.
alpha
temporal correlation factor;
the fraction of the new random component added to the previous time-step (defaults to 0.1).

5. パッチ，計算条件の設定　p

```
dimensions [ 0 2 -2 0 0 0 0 ];

internalField uniform 0;

boundaryField
{
 outlet_outlet
 {
  type fixedValue;
  value uniform 0;
 }
 wall_wall
 {
  type zeroGradient;
 }
 defaultFaces
 {
  type empty;
 }
 inlet_inlet
  {
   type zeroGradient;
  }
 }
```

5. パッチ，計算条件の設定　k

```
dimensions [ 0 2 -2 0 0 0 0 ];

internalField uniform 9.375;

boundaryField
{
 outlet_outlet
 {
  type inletOutlet;
  inletValue uniform 9.375;
  value uniform 9.375;
 }
 wall_wall
 {
  type  fixedValue;
  value uniform 0;
 }
 defaultFaces
 {
  type empty;
 }
 inlet_inlet
  {
   type turbulentIntensityKineticEnergyInlet;
   intensity 0.05;
   value uniform 1;
  }
}
```

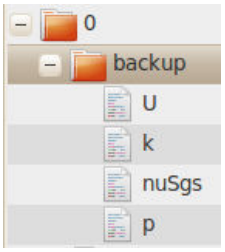
RASモデルでは壁関数を使って設定しますが，LESではどのように設定するのか良くわまりません。このため，今回は0として設定しています。アドバイスをお願いします。

5. パッチ，計算条件の設定　nuSgs

```
dimensions [ 0 2 -1 0 0 0 0 ];

internalField uniform 0;

boundaryField
{
 outlet_outlet
  {
   type calculated;
   value uniform 0;
  }
 wall_wall
  {
   type calculated;
   value uniform 0;
  }
 defaultFaces
  {
   type empty;
  }
 inlet_inlet
  {
   type calculated;
   value uniform 0;
  }
}
```

5. パッチ，計算条件の設定

条件を設定したファイルを0/backupのホルダに入れます。解析時にpotentialFoamを行うとファイルが上書きされるため，バックアップとして保存しておくと良いと思います。




```
dimensions [ 0 1 -1 0 0 0 0 ];

internalField uniform (0 0 0);

boundaryField
{
  outlet_outlet
  {
    type            inletOutlet;
    inletValue      uniform (0 0 0);
    value           uniform (0 0 0);
  }
  wall_wall
  {
    type fixedValue;
    value uniform (0 0 0);
  }
  defaultFaces
  {
```

初期設定のU

```
dimensions      [0 1 -1 0 0 0 0];

internalField   nonuniform List<vector>
194836
(
(-54.3181 1.01546 -5.50033)
(-54.2557 -1.2624 -5.49051)
(-51.02 0.0259961 1.30462)
(-55.1298 2.16323 -4.99315)
(-50.6428 0.318888 -7.63208)
(-52.2493 0.858139 -0.723717)
(-55.1601 -2.34651 -4.40758)
(-51.0309 -0.120264 1.28233)
(-50.6507 -0.15605 -7.66378)
(-52.247 -1.00596 -0.65066)
(-51.0185 0.156285 1.4435)
(-51.2366 -0.163042 0.514924)
(-50.7552 0.309466 -3.05779)
(-53.4092 0.0362489 0.543641)
(-51.4743 0.624065 -6.72558)
(-57.0482 0.753875 -3.41997)
(-52.783 1.73578 -0.685113)
(-49.0704 0.380471 -5.48405)
(-53.8612 0.0732078 0.084205)
(-51.1181 -0.25222 1.37534)
```

potentailFoam後のU

5. パッチ，計算条件の設定　fvSolution

```
solvers
{
  "(p|pFinal)"
   {
     solver          GAMG;
     tolerance       1e-7;
     relTol          0.1;
     smoother        GaussSeidel;
     nPreSweeps      0;
     nPostSweeps     2;
     cacheAgglomeration on;
     agglomerator    faceAreaPair;
     nCellsInCoarsestLevel 10;
     mergeLevels     1;
   };
 "(U|k|epsilon|omega|R|nuTilda|B)"
   {
     solver          smoothSolver;
     smoother        GaussSeidel;
     tolerance       1e-6;
     relTol          0.1;
     nSweeps         1;
   };
```

```
SIMPLE
{
   nNonOrthogonalCorrectors 0;
   convergence              1e-3;
}

PISO
{
   nCorrectors     2;
   nNonOrthogonalCorrectors 0;
}

relaxationFactors
{
   p          0.3;
   U          0.7;
   k          0.7;
   epsilon    0.7;
   omega      0.7;
}
```

5. パッチ，計算条件の設定　fvSchemes

```
ddtSchemes
{
   default        backward;
}
gradSchemes
{
   default        Gauss linear;
   grad(p)        Gauss linear;
   grad(U)        Gauss linear;
}
divSchemes
{
   default        none;
   div(phi,U)      Gauss limitedLinearV 1;
   div(phi,k)      Gauss limitedLinear 1;
   div(phi,epsilon) Gauss limitedLinear 1;
   div(phi,omega)  Gauss limitedLinear 1;
   div(phi,B)      Gauss limitedLinear 1;
   div(B)          Gauss limitedLinear 1;
   div(phi,nuTilda) Gauss limitedLinear 1;
   div((nuEff*dev(grad(U).T()))) Gauss
linear;
}
laplacianSchemes
{
   default        Gauss linear corrected;
}
interpolationSchemes
{
   default        linear;
   interpolate(U)  linear;
}
snGradSchemes
{
   default        corrected;
}
fluxRequired
{
   default        no;
   p;
}
```


5. パッチ，計算条件の設定　controlDict

```
application     pisoFoam;
startFrom       startTime;
startTime       0;
stopAt          endTime;
endTime         0.6;
deltaT          2e-04;
writeControl    timeStep;
writeInterval   300;
purgeWrite      0;
writeFormat     ascii;
writePrecision  6;
writeCompression uncompressed;
timeFormat      general;
timePrecision   6;
runTimeModifiable yes;
functions
(
   inlet_FlowRate
   {
      type            faceSource;
      functionObjectLibs ("libfieldFunctionObjects.so");
      enabled         true;
      outputControl   outputTime;
      outputInterval  0.01;
      log             true;
      valueOutput     true;
      source          patch;
      sourceName      inlet_inlet;
      operation       sum;
      fields
      (
         phi
      );
   }

   outlet_FlowRate
   {
      type            faceSource;
      functionObjectLibs ("libfieldFunctionObjects.so");
      enabled         true;
      outputControl   outputTime;
      outputInterval  0.01;
      log             true;
      valueOutput     true;
      source          patch;
      sourceName      outlet_outlet;
      operation       sum;
```

計算時間は10m/sの流れと管長さ約2.6mから0.6secとしています。

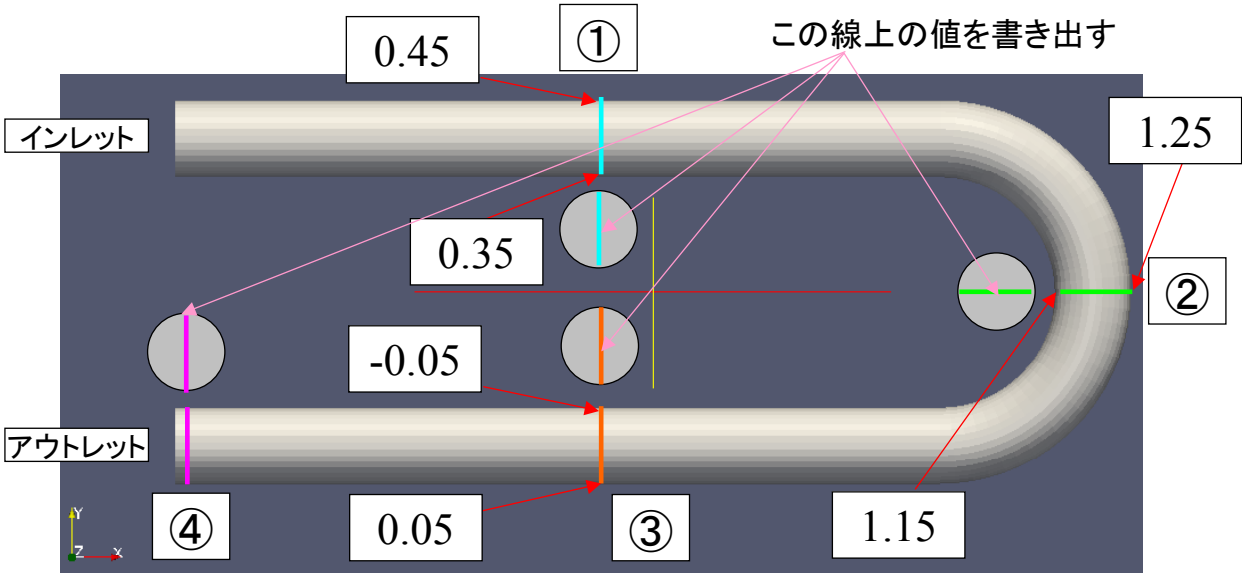
各モデルで最大クーラン数が0.8以下になっている事を確認しています。

5. パッチ，計算条件の設定　controlDict

```
    fields
    (
        phi
    );
}

fieldAverage
{
    type            fieldAverage;
    functionObjectLibs ("libfieldFunctionObjects.so");
    enabled         true;
    outputControl   outputTime;
    outputInterval  0.01;
    log             true;
    valueOutput     true;

    fields
    (
        U
        {
            mean        on;
            prime2Mean  on;
            base        time;
        }
        p
        {
            mean        on;
            prime2Mean  on;
            base        time;
        }
    );
}
);
```


5. パッチ，計算条件の設定　sampleDict

特定位置での速度,圧力，乱流エネルギーの書き出し設定
パイプ断面の中心線上で上記の値を書き出します。



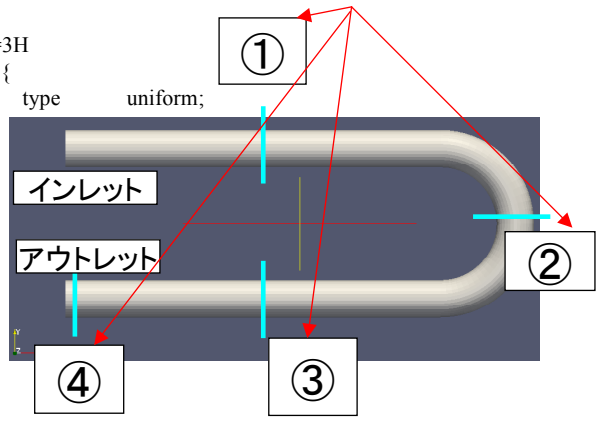
5. パッチ，計算条件の設定　sampleDict

```
FoamFile
{
   version      2.0;
   format       ascii;
   class        dictionary;
   location     system;
   object       sampleDict;
}

setFormat raw;
surfaceFormat foamFile;
interpolationScheme cellPointFace;

fields
(
   U p k
);
```

```
sets
(
   x=0H
   {
      type        uniform;
      axis        y;
      start       (0.5 0.35 0);
      end         (0.5 0.45 0);
      nPoints     50;
   }
   x=1H
   {
      type        uniform;
      axis        x;
      start       (1.15 0.2 0);
      end         (1.25 0.2 0);
      nPoints     50;
   }
   x=2H
   {
      type        uniform;
      axis        y;
      start       (0.5 -0.05 0);
      end         (0.5  0.05 0);
      nPoints     50;
   }
   x=3H
   {
      type        uniform;
      axis        y;
      start       (0.01 -0.05 0);
      end         (0.01  0.05 0);
      nPoints     50;
   }
);

surfaces
(
);
```

①　②　③　④

この4ヵ所で書き出す



参考：ユーザーガイド　6.5 データのサンプリング
$FOAM_UTILITIES/postProcessing/sampling/sampleにひな形あり

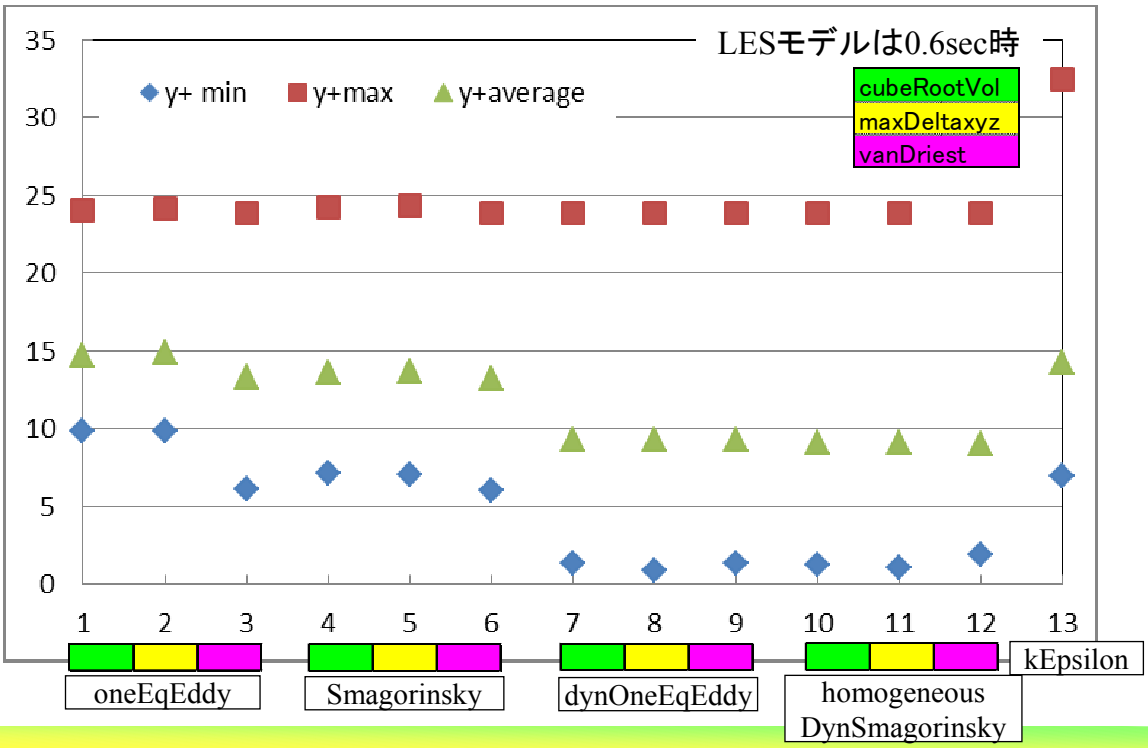
6. 計算設定

次のような設定で計算を実施します。RASモデルは比較用で，標準的なkEpsilonモデルです。LESモデルは0.6秒後まで計算しています。

No	LESモデル	デルタ	レイヤー数	minThickness	fvSchemes	fvSolution
1	oneEqEddy	cubeRootVol	6	0.05	TVD	GAMG smoothSolver
2		maxDeltaxyz				
3		vanDriest				
4	Smagorinsky	cubeRootVol				
5		maxDeltaxyz				
6		vanDriest				
7	dynOneEqEddy	cubeRootVol				
8		maxDeltaxyz				
9		vanDriest				
10	homogeneousDynSmagorinsky	cubeRootVol				
11		maxDeltaxyz				
12		vanDriest				

No	RASモデル		レイヤー数	minThickness	fvSchemes	fvSolution
13	kEpsilon		6	0.05	TVD	GAMA,smoot

7. 計算結果　各モデルのy+



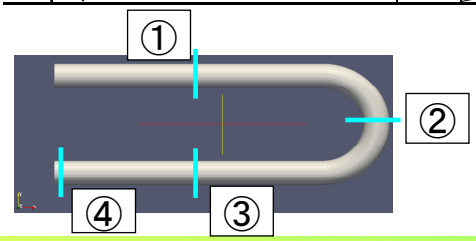
7. 計算結果　各モデルのy+

No	LESモデル	デルタ	y+ min	y+max	y+average
1	oneEqEddy	cubeRootVol	10.1	24.3	14.7
2		maxDeltaxyz	10.1	24.3	14.8
3		vanDriest	6.6	24.0	13.1
4	Smagorinsky	cubeRootVol	7.9	24.4	13.5
5		maxDeltaxyz	7.8	24.6	13.5
6		vanDriest	6.6	24.0	13.0
7	dynOneEqEddy	cubeRootVol	2.1	23.9	9.2
8		maxDeltaxyz	1.9	23.9	9.2
9		vanDriest	2.1	23.9	9.2
10	homogeneousDynSmagorinsky	cubeRootVol	1.6	23.9	9.1
11		maxDeltaxyz	1.9	23.9	9.1
12		vanDriest	1.6	23.9	9.1

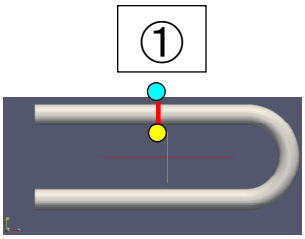
No	RASモデル		y+ min	y+max	y+average
13	kEpsilon		6.9	33.3	14.2

7. 計算結果　各モデルでの書き出しデータ一覧

No	モデル		デルタ	静圧			軸方向速度				乱流エネルギー			
				①	②	③	①	②	③	④	①	②	③	④
1	oneEqEddy	LES	cubeRootVol	○	○	○	○	○	○	○	○	○	○	○
2			maxDeltaxyz	○	○	○	○	○	○	○	○	○	○	○
3			vanDriest	○	○	○	○	○	○	○	○	○	○	○
4	Smagorinsky		cubeRootVol	○	○	○	○	○	○	○				
5			maxDeltaxyz	○	○	○	○	○	○	○				
6			vanDriest	○	○	○	○	○	○	○				
7	dynOneEqEddy		cubeRootVol	○	○	○	○	○	○	○	○	○	○	○
8			maxDeltaxyz	○	○	○	○	○	○	○	○	○	○	○
9			vanDriest	○	○	○	○	○	○	○	○	○	○	○
10	homogeneousDynSmagorinsky		cubeRootVol	○	○	○	○	○	○	○	○	○	○	○
11			maxDeltaxyz	○	○	○	○	○	○	○	○	○	○	○
12			vanDriest	○	○	○	○	○	○	○	○	○	○	○
13	kEpsilon	RAS		○	○	○	○	○	○	○	○	○	○	○



7. 計算結果　oneEqEddyモデル：①部での静圧比較

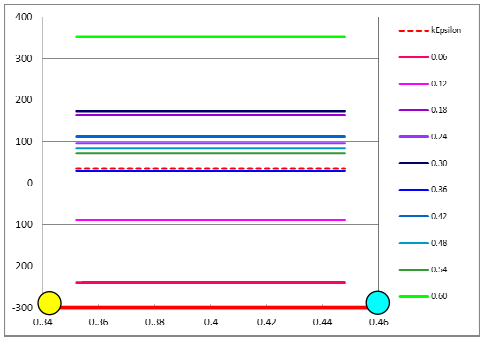


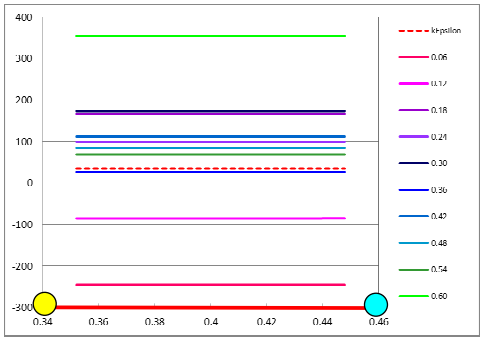
円管断面の中央部を黄丸から青丸方向に静圧を表示

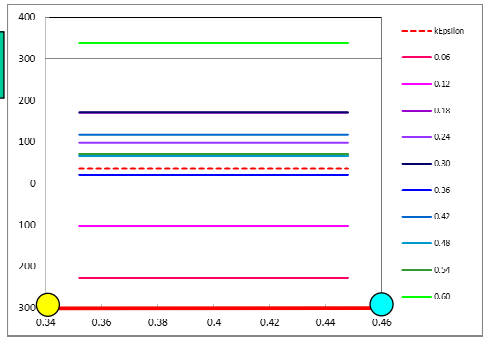
cubeRootVol　maxDeltaxyz

vanDriest

図配置



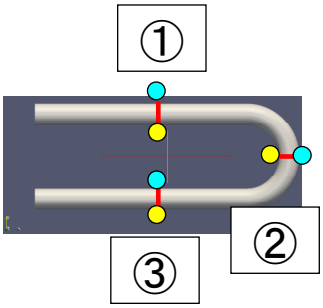




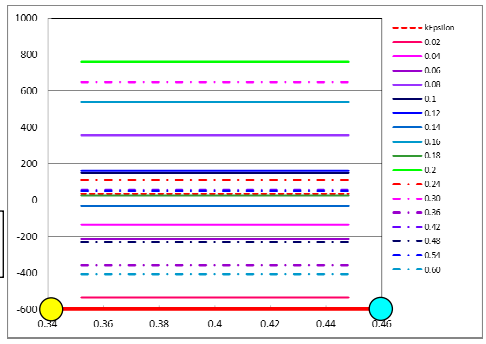
・静圧は収束しているか。cubeRootVolモデルで少し細かく計算（スライド51）。

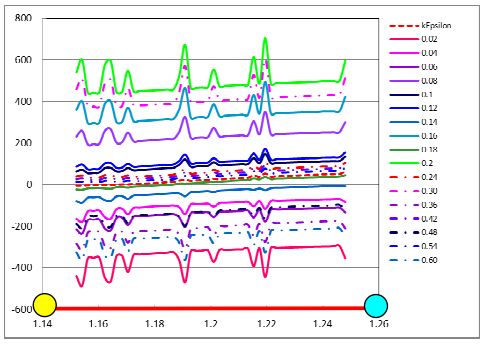
・静圧の傾向はどのフィルタサイズでも同じ

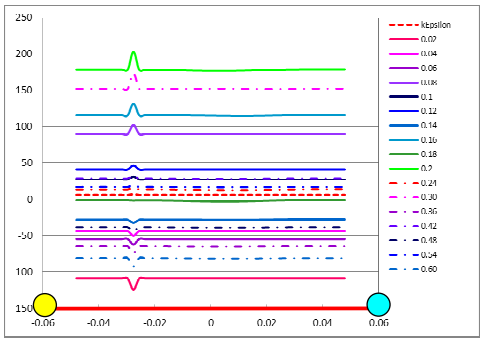
7. 計算結果　oneEqEddy cubeRootVol 静圧変動のチェック



円管断面の中央部を黄丸から青丸方向に静圧を表示

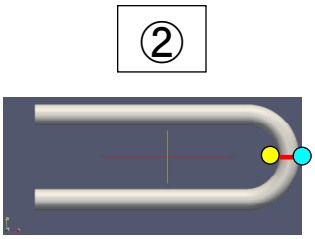






・LESモデルでは静圧はある範囲を変動している。

7. 計算結果　oneEqEddyモデル：②部での静圧比較

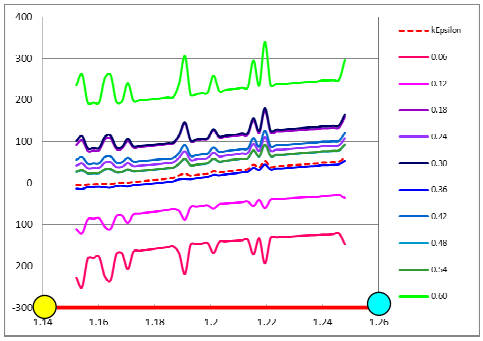


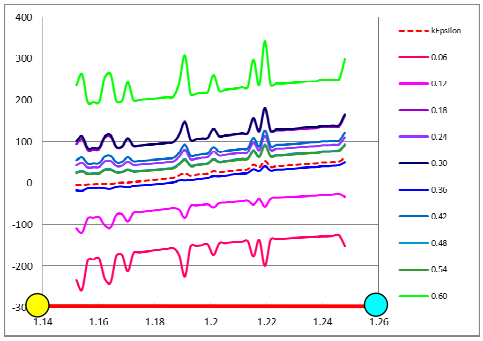
円管断面の中央部を黄丸から青丸方向に静圧を表示

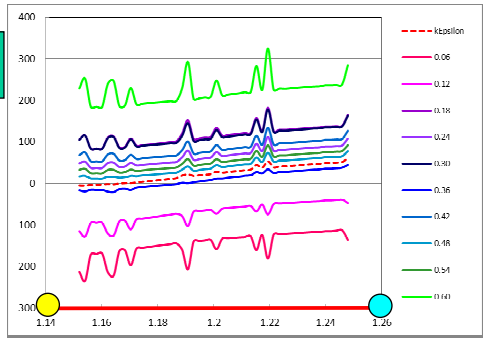
cubeRootVol　maxDeltaxyz

vanDriest

図配置





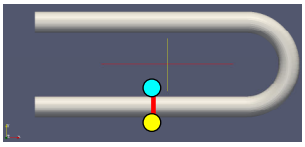


・多数スパイク状の山が出る。これはなぜか？

・静圧の傾向はどのフィルタサイズでも同じ。

7. 計算結果　oneEqEddyモデル：③部での静圧比較

③

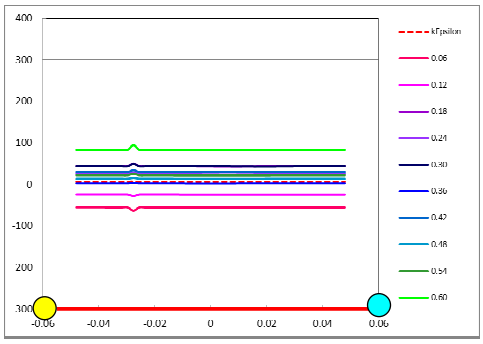


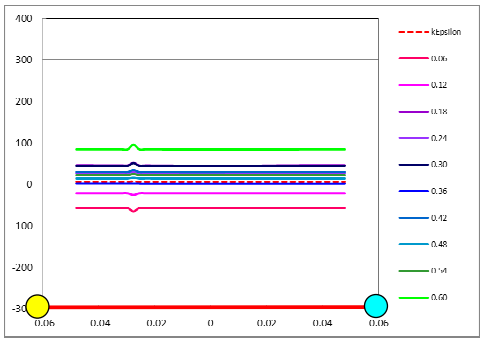
円管断面の中央部を黄丸から青丸方向に静圧を表示

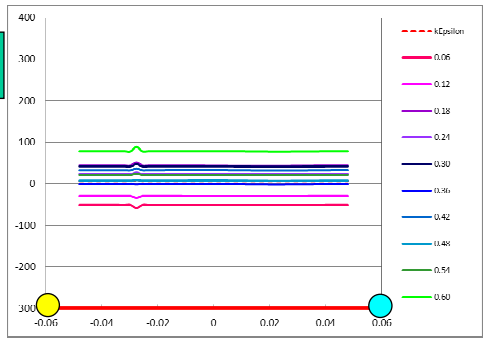
cubeRootVol　maxDeltaxyz

vanDriest

図配置



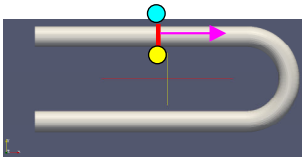




・静圧の傾向はどのフィルタサイズでも同じ。

7. 計算結果　oneEqEddyモデル：①部での軸方向速度比較

①

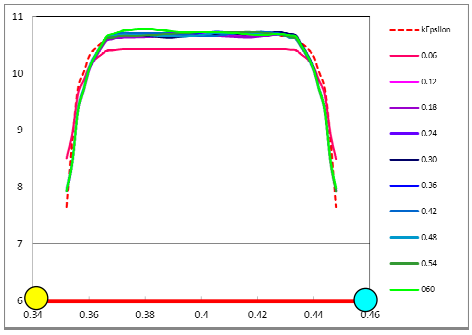


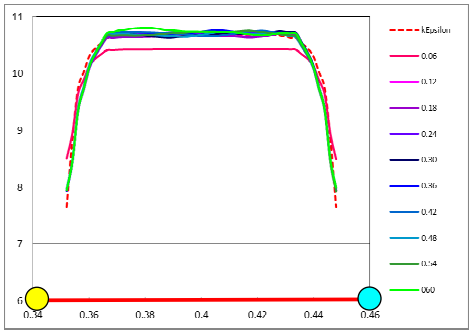
円管断面の中央部を黄丸から青丸方向に軸方向速度を表示

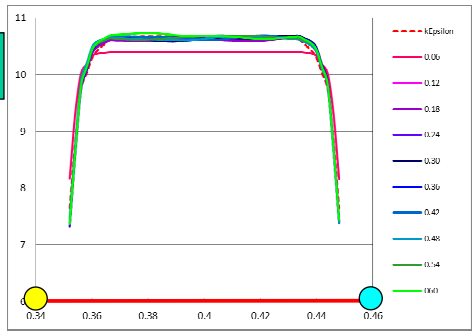
cubeRootVol　maxDeltaxyz

vanDriest

図配置







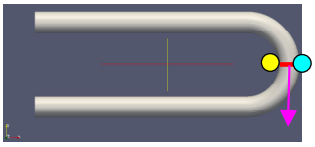
・0.12秒後で速度の形状はどのモデルも安定してきている。

・vanDriestのフィルタモデルは，他の2つのモデルと比較してより壁の近くまで流れが発達している。

・どのフィルタkEpsilonモデルのシェイプと似ている。

7. 計算結果　oneEqEddyモデル：②部での軸方向速度比較

②

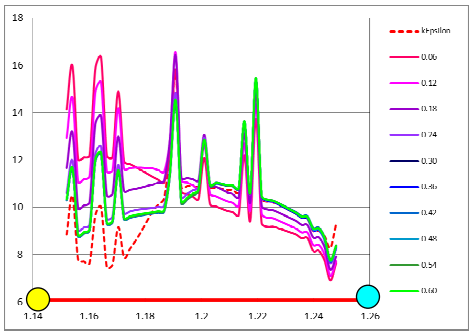


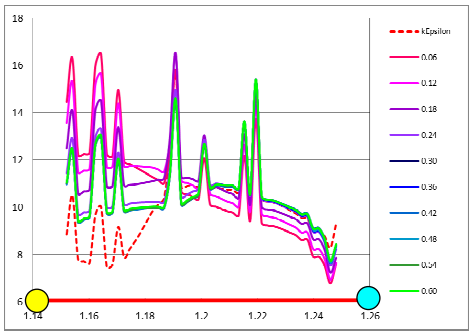
円管断面の中央部を
黄丸から青丸方向に
軸方向速度を表示

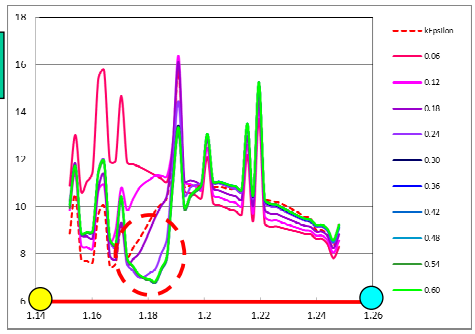
cubeRootVol　maxDeltaxyz

vanDriest

図配置





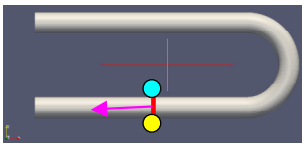


・多数スパイク状の山が出る。
　これはなぜか？

・vanDriestと他のフィルタでは速
　度シェイプが異なる。

7. 計算結果　oneEqEddyモデル：③部での軸方向速度比較

③

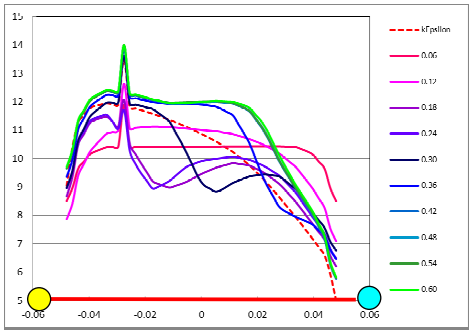


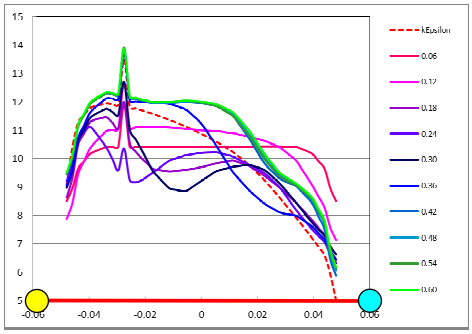
円管断面の中央部を
黄丸から青丸方向に
軸方向速度を表示

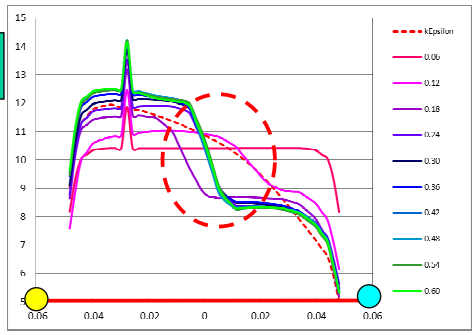
cubeRootVol　maxDeltaxyz

vanDriest

図配置







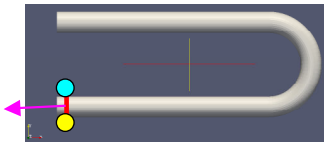
・1本スパイク状の山が出る。
　これはなぜか？

・各モデルともkEpsiloモデルのシェ
　イプとかい離がある。

・vanDriestと他のフィルタでは速
　度シェイプが異なる。

7. 計算結果　oneEqEddyモデル：④部での軸方向速度比較

④

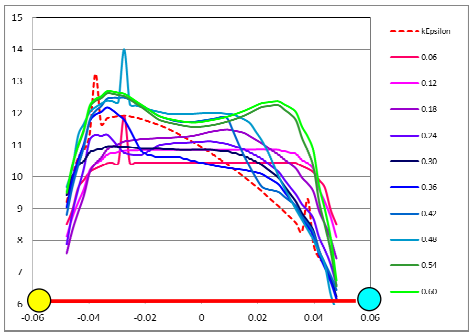


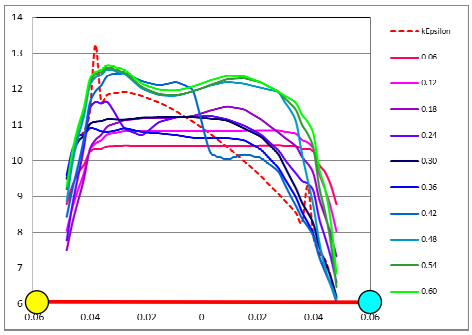
円管断面のの中央部を黄丸から青丸方向に軸方向速度を表示

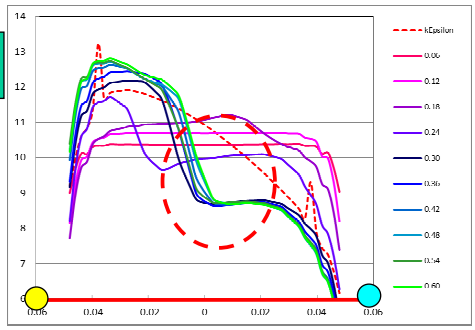
cubeRootVol　maxDeltaxyz

vanDriest

図配置





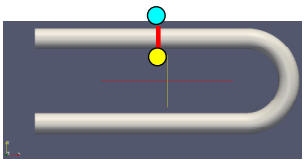


・各モデルともkEpsiloモデルのシェイプとかい離がある。

・vanDriestと他のフィルタでは速度シェイプが異なる。

7. 計算結果　oneEqEddyモデル：①部での乱流エネルギー比較

①

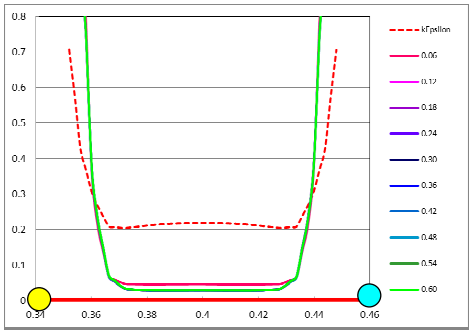


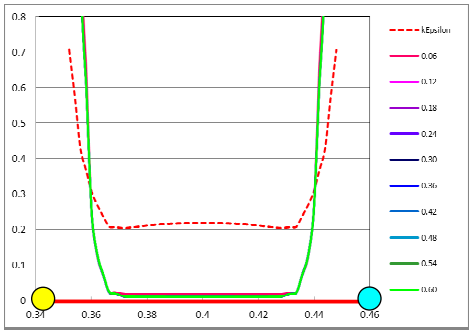
円管断面の中央部を黄丸から青丸方向に乱流エネルギーを表示

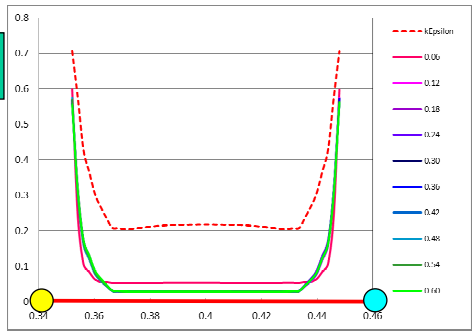
cubeRootVol　maxDeltaxyz

vanDriest

図配置







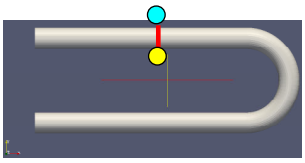
・vanDriestのフィルタと他ではシェイプが異なる。

・各モデルともkEpsiloモデルのシェイプとかい離がある。

・壁近傍においてcubeRootVol, maxDeltaxyzのフィルタは，大きなkとなっている。（スライド56も参照）。

7. 計算結果　oneEqEddyモデル：①部での乱流エネルギー比較

①

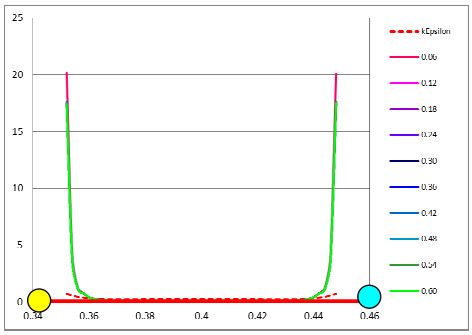


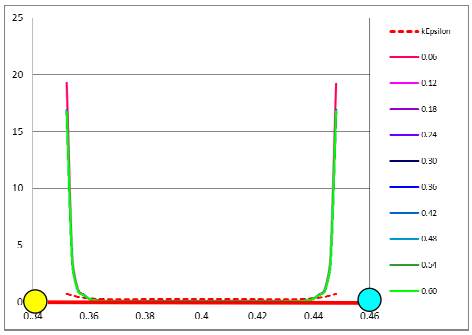
円管断面の中央部を黄丸から青丸方向に乱流エネルギーを表示

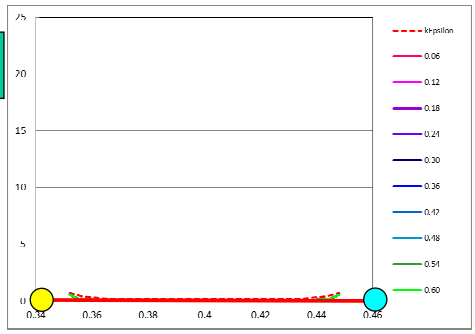
cubeRootVol　maxDeltaxyz

vanDriest

図配置



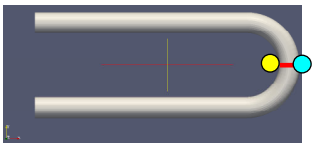




・このスライドはスライド58の全体表示。

7. 計算結果　oneEqEddyモデル：②部での乱流エネルギー比較

②

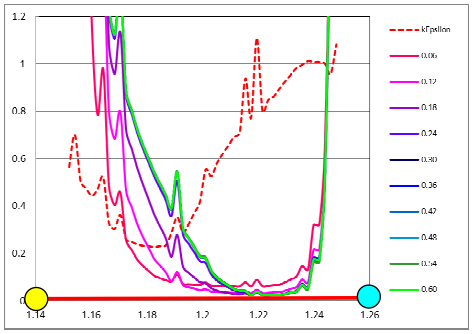


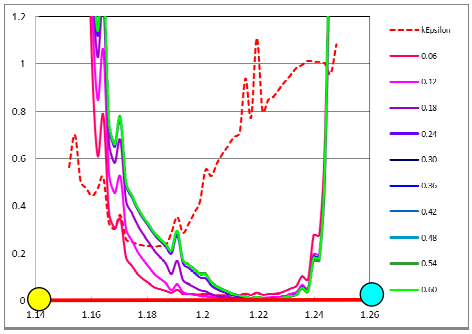
円管断面の中央部を黄丸から青丸方向に乱流エネルギーを表示

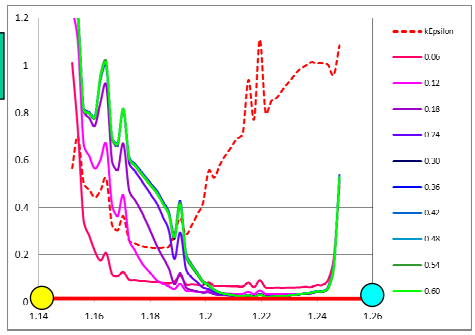
cubeRootVol　maxDeltaxyz

vanDriest

図配置



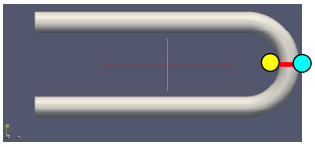




・各モデルともkEpsiloモデルのシェイプとかい離がある。

7. 計算結果　oneEqEddyモデル：②部での乱流エネルギー比較

②

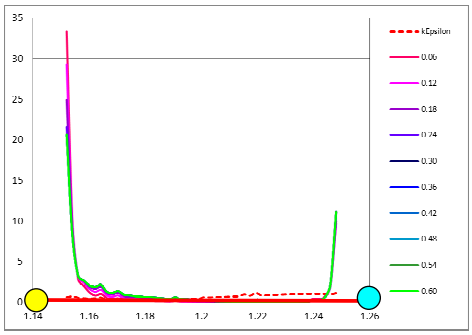


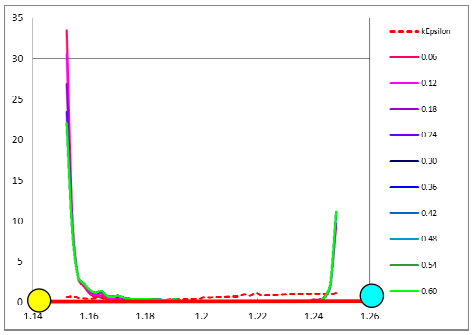
円管断面の中央部を黄丸から青丸方向に乱流エネルギーを表示

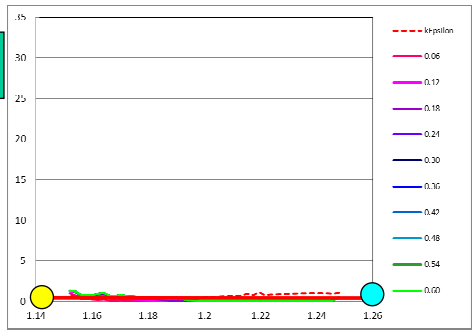
cubeRootVol　maxDeltaxyz

vanDriest

図配置





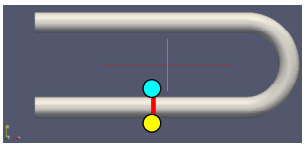


・このスライドはスライド60の全体表示。

・vanDriestのフィルタと他ではシェイプが異なる。

7. 計算結果　oneEqEddyモデル：③部での乱流エネルギー比較

③

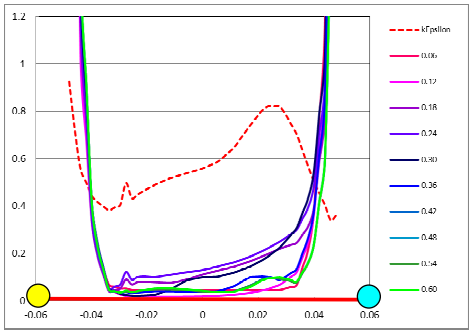


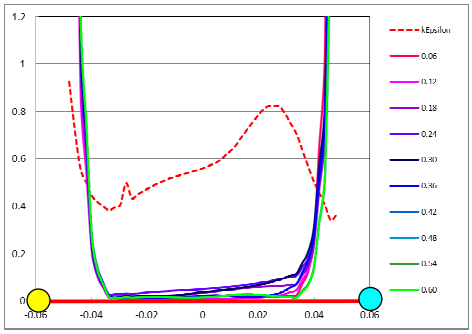
円管断面の中央部を黄丸から青丸方向に乱流エネルギーを表示

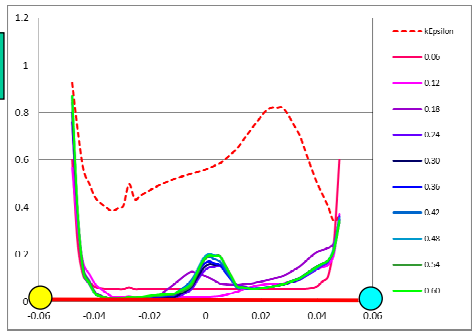
cubeRootVol　maxDeltaxyz

vanDriest

図配置



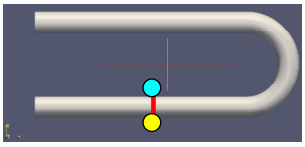




・各モデルともkEpsiloモデルのシェイプとかい離がある。

7. 計算結果　oneEqEddyモデル：③部での乱流エネルギー比較

③

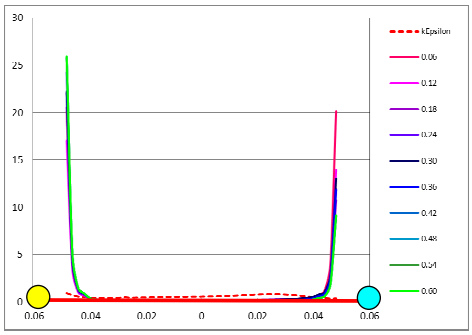


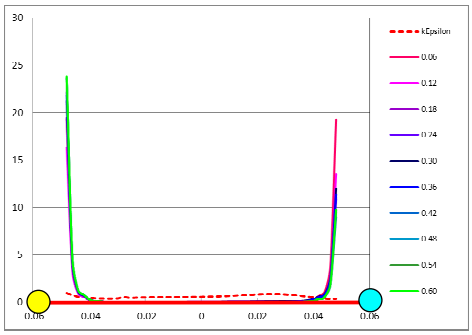
円管断面の中央部を黄丸から青丸方向に乱流エネルギーを表示

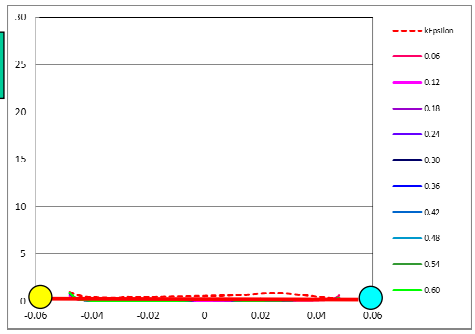
cubeRootVol　maxDeltaxyz

vanDriest

図配置





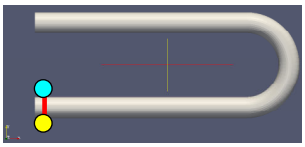


・このスライドはスライド62の全体表示

・vanDriestのフィルタと他ではシェイプが異なる。

7. 計算結果　oneEqEddyモデル：④部での乱流エネルギー比較

④

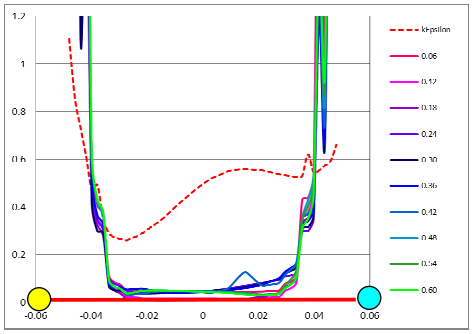


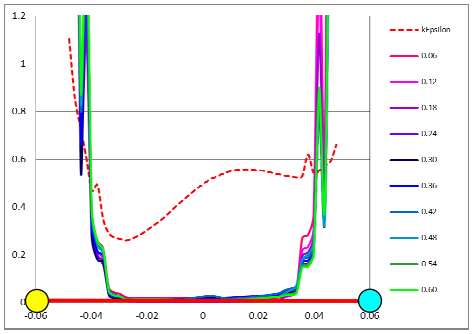
円管断面の中央部を黄丸から青丸方向に乱流エネルギーを表示

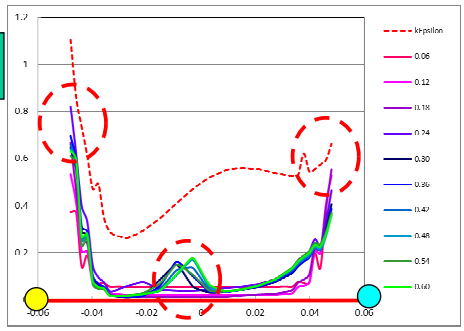
cubeRootVol　maxDeltaxyz

vanDriest

図配置





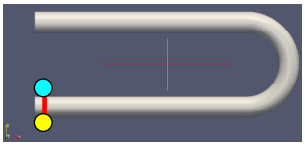


・各モデルともkEpsiloモデルのシェイプとかい離がある。

・vanDriestのフィルタと他ではシェイプが異なる。

7. 計算結果　oneEqEddyモデル：④部での乱流エネルギー比較

④

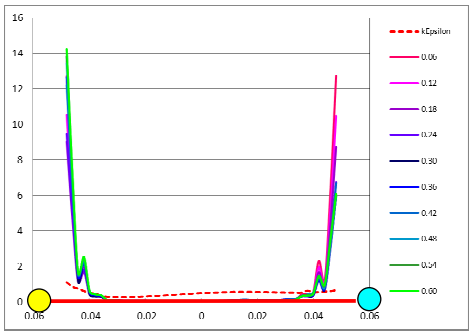


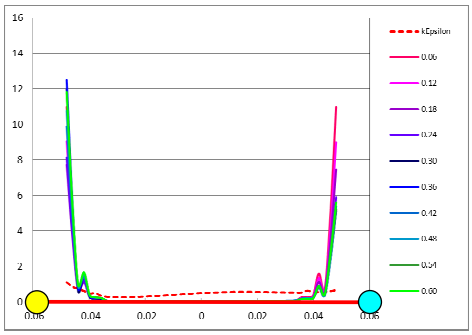
円管断面の中央部を黄丸から青丸方向に乱流エネルギーを表示

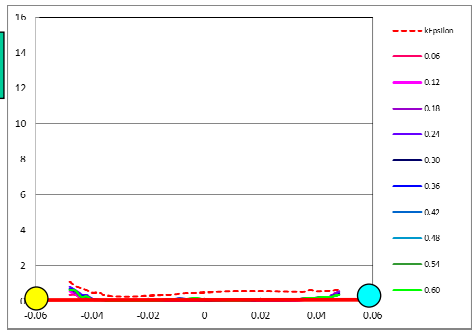
cubeRootVol　maxDeltaxyz

vanDriest

図配置





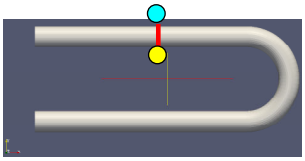


・このスライドはスライド64の全体表示

・vanDriestのフィルタと他ではシェイプが異なる。

7. 計算結果　Smagorinskyモデル：①部での静圧比較

①

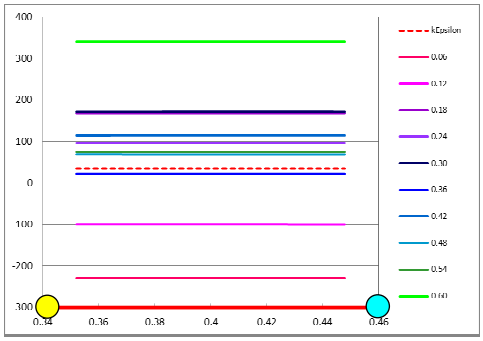


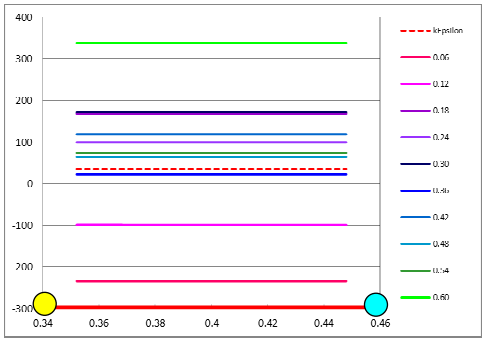
円管断面の中央部を黄丸から青丸方向に静圧を表示

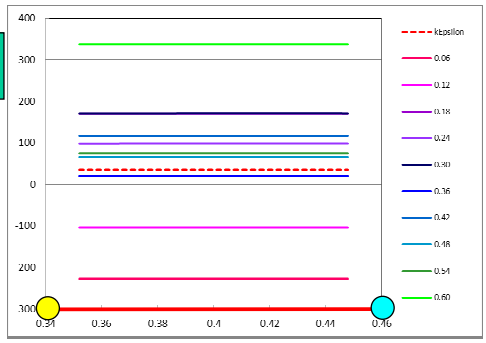
cubeRootVol　maxDeltaxyz

vanDriest

図配置



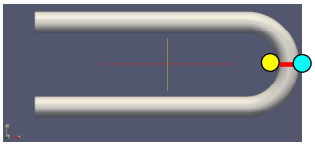




・静圧の傾向はどのフィルタサイズでも同じ。

7. 計算結果　Smagorinskyモデル:②部での静圧比較

②

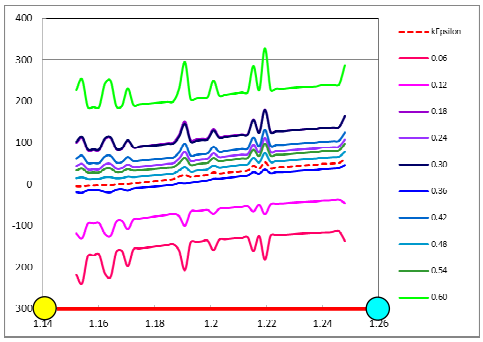


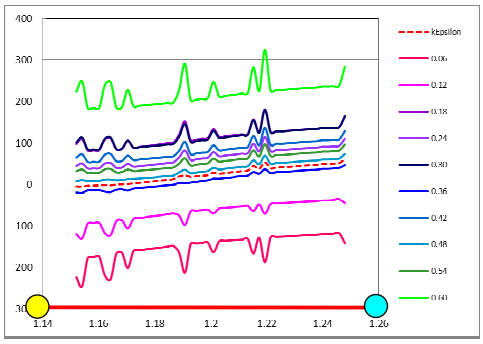
円管断面の中央部を黄丸から青丸方向に静圧を表示

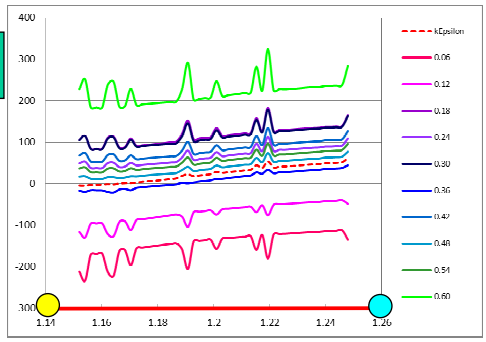
cubeRootVol　maxDeltaxyz

vanDriest

図配置





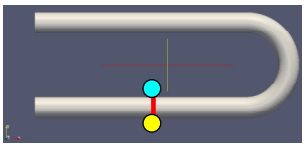


・多数スパイク状の山が出る。これはなぜか？

・静圧の傾向はどのフィルタサイズでも同じ。

7. 計算結果　Smagorinskyモデル:③部での静圧比較

③

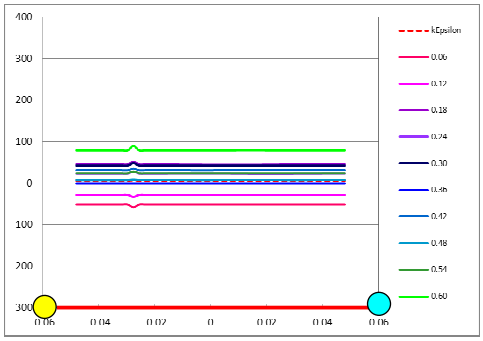


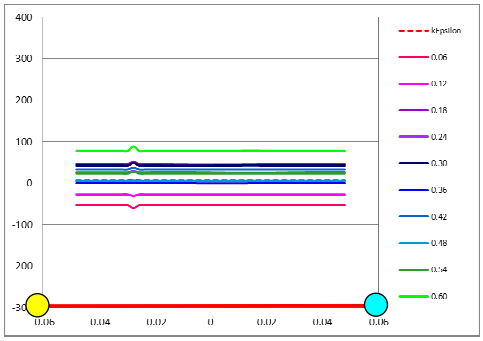
円管断面の中央部を黄丸から青丸方向に静圧を表示

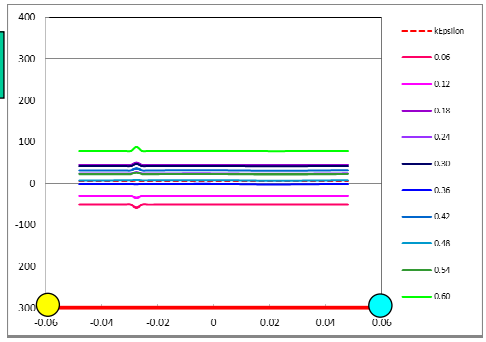
cubeRootVol　maxDeltaxyz

vanDriest

図配置



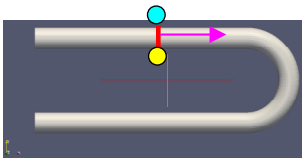




・静圧の傾向はどのフィルタサイズでも同じ。

7. 計算結果　Smagorinskyモデル:①部での軸方向速度比較

①

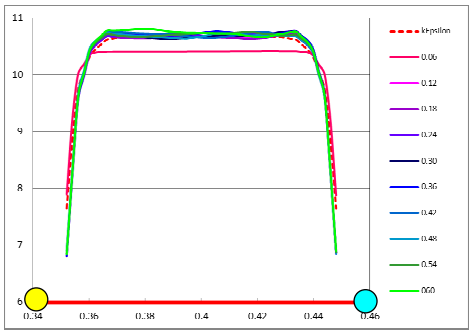


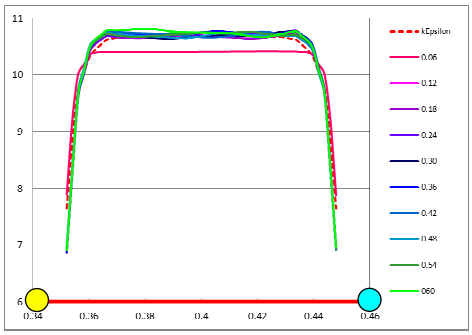
円管断面の中央部を
黄丸から青丸方向に
軸方向速度を表示

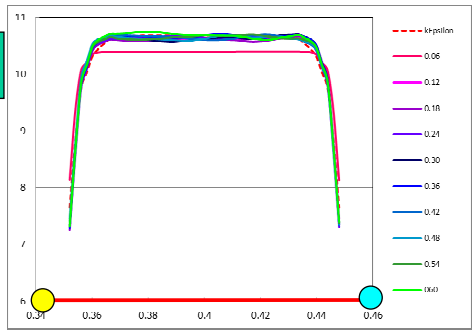
cubeRootVol　maxDeltaxyz

vanDriest

図配置







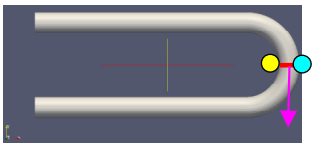
・0.12秒後あたりで速度の形状は安定してきている。

・vanDriestのフィルタモデルは，他の2つのモデルと比較してより壁の近くまで流れが発達している。

・どのフィルタkEpsilonモデルのシェイプと似ている。

7. 計算結果　Smagorinskyモデル:②部での軸方向速度比較

②

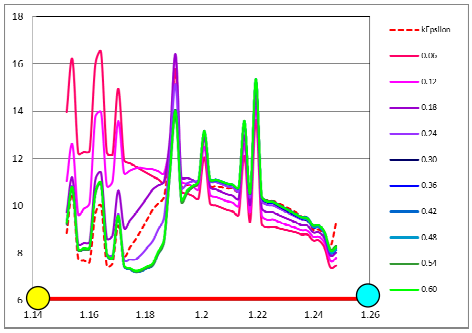


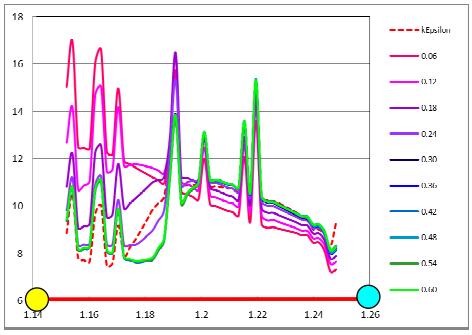
円管断面の中央部を
黄丸から青丸方向に
軸方向速度を表示

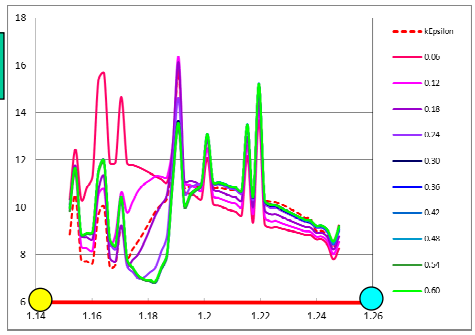
cubeRootVol　maxDeltaxyz

vanDriest

図配置



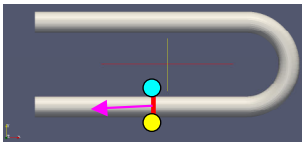




・多数スパイク状の山が出る。
これはなぜか？

7. 計算結果　Smagorinskyモデル:③部での軸方向速度比較

③

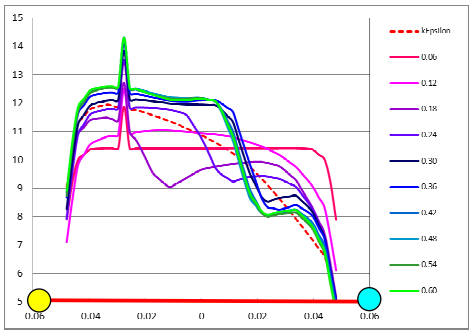


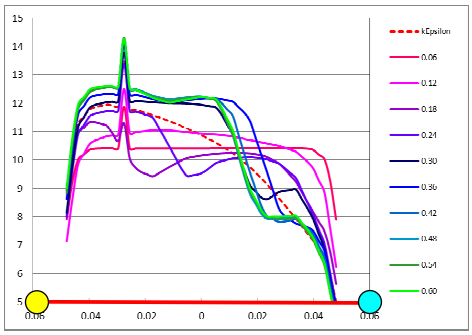
円管断面の中央部を黄丸から青丸方向に軸方向速度を表示

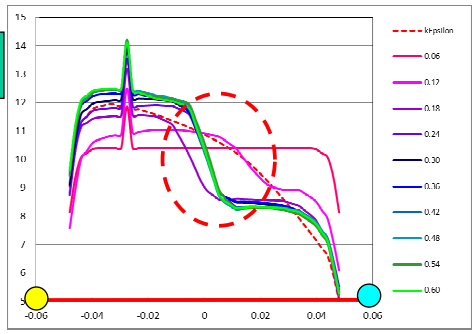
cubeRootVol　maxDeltaxyz

vanDriest

図配置





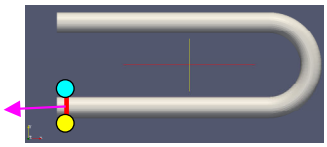


・1本スパイク状の山が出る。これはなぜか？

・vanDriestと他のフィルタでは速度シェイプが異なる。

7. 計算結果　Smagorinskyモデル:④部での軸方向速度比較

④

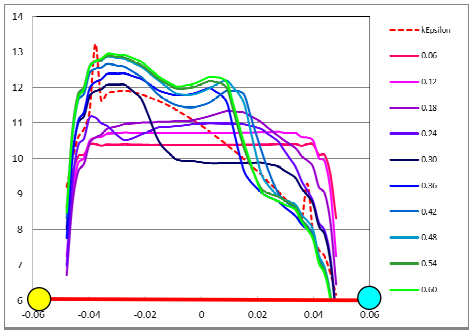


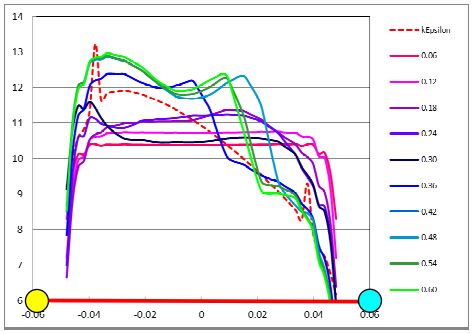
円管断面の中央部を黄丸から青丸方向に乱流エネルギーを表示

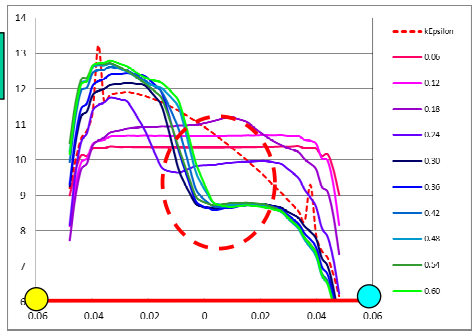
cubeRootVol　maxDeltaxyz

vanDriest

図配置







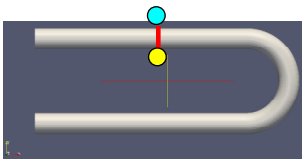
・vanDriestと他のフィルタでは速度シェイプが異なる。

7. 計算結果 Smagorinskyモデル:乱流エネルギー比較

このモデルでは乱流エネルギーがsampleDictでは抽出できません。この理由はなんでしょうか？。教えて下さい！。

7. 計算結果　dynOneEqEddyモデル:①部での静圧比較

①

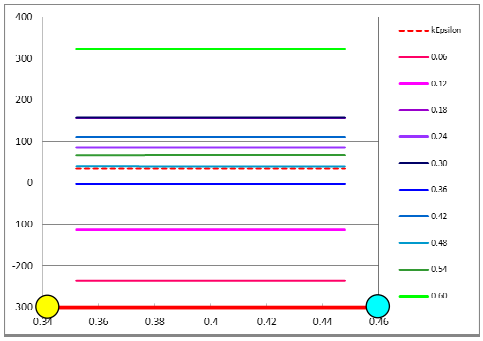


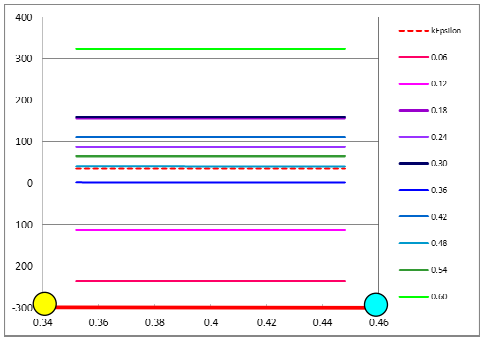
円管断面の中央部を黄丸から青丸方向に静圧を表示

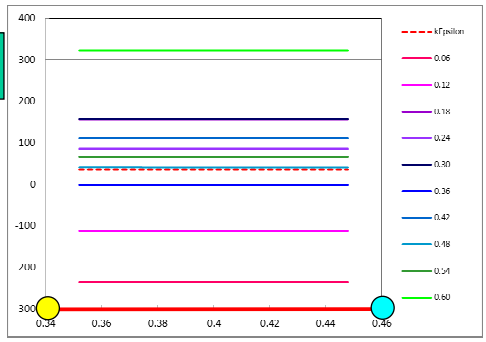
cubeRootVol　maxDeltaxyz

vanDriest

図配置



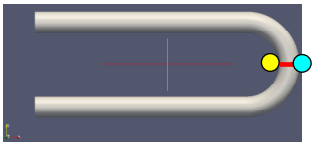




・静圧の傾向はどのフィルタサイズでも同じ。

7. 計算結果　dynOneEqEddyモデル:②部での静圧比較

②

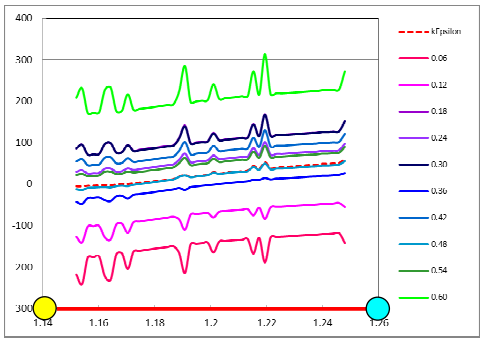


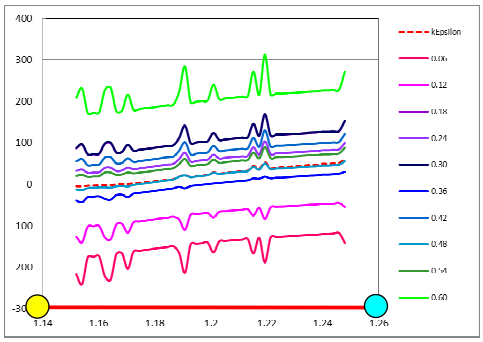
円管断面の中央部を黄丸から青丸方向に静圧を表示

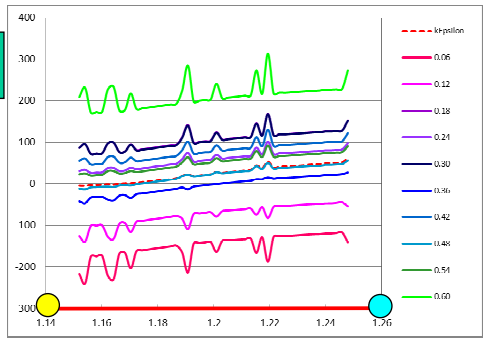
cubeRootVol　maxDeltaxyz

vanDriest

図配置





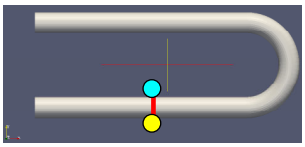


・多数スパイク状の山が出る。これはなぜか？

・静圧の傾向はどのフィルタサイズでも同じ。

7. 計算結果　dynOneEqEddyモデル:③部での静圧比較

③

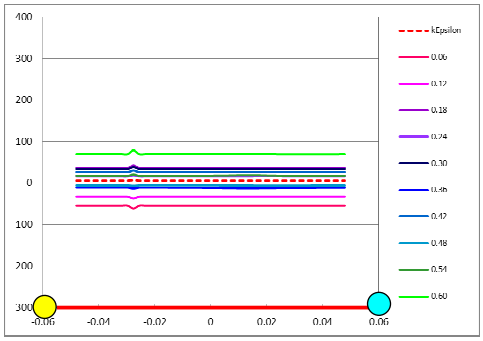


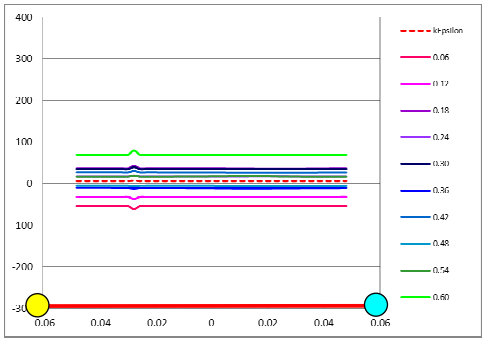
円管断面の中央部を黄丸から青丸方向に静圧を表示

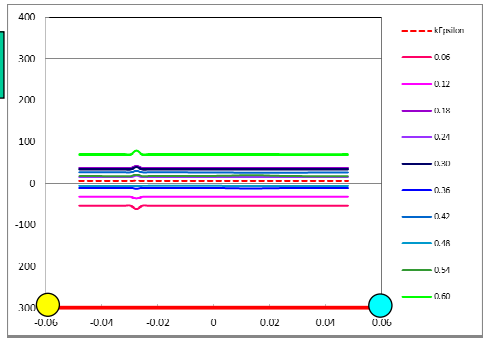
cubeRootVol　maxDeltaxyz

vanDriest

図配置



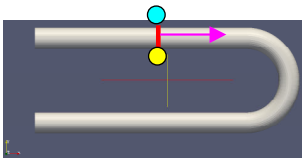




・静圧の傾向はどのフィルタサイズでも同じ。

7. 計算結果　dynOneEqEddyモデル:①部での軸方向速度比較

①

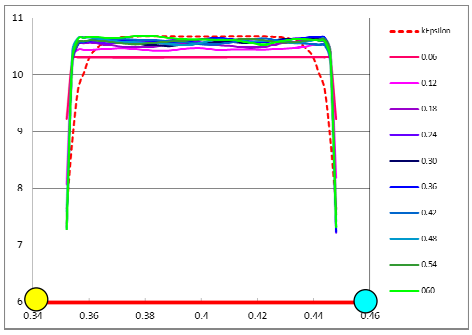


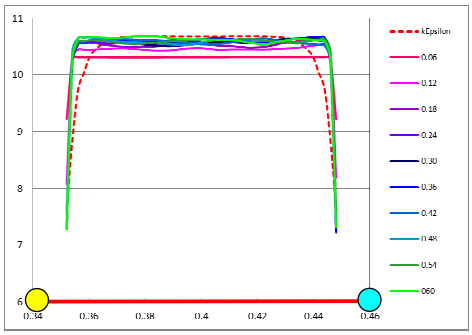
円管断面の中央部を
黄丸から青丸方向に
軸方向速度を表示

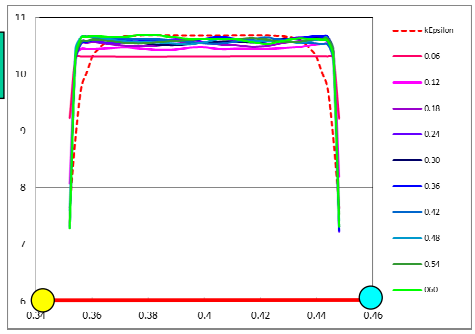
cubeRootVol　maxDeltaxyz

vanDriest

図配置







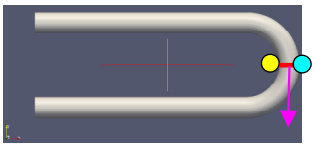
・OneEqEddyモデルと比較すると，より壁近傍まで速度が発達している。

・どのフィルタモデルもOneEqEddyモデルの場合と異なり，速度シェイプは良くあっている。

・各モデルともkEpsiloモデルのシェイプとかい離がある。

7. 計算結果　dynOneEqEddyモデル:②部での軸方向速度比較

②

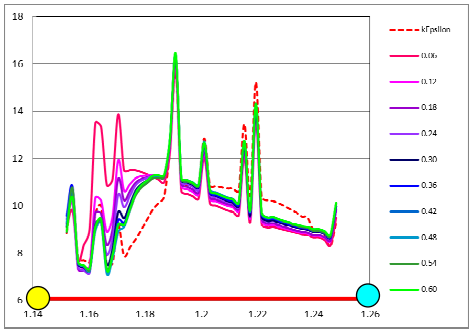


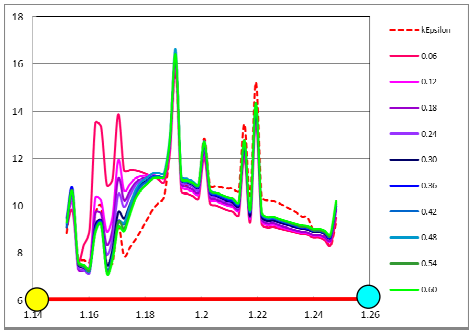
円管断面の中央部を
黄丸から青丸方向に
軸方向速度を表示

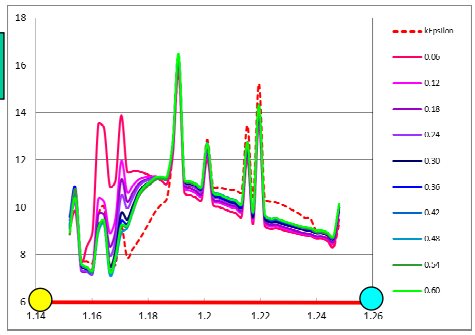
cubeRootVol　maxDeltaxyz

vanDriest

図配置





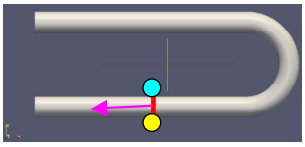


・多数スパイク状の山が出る。これはなぜか？

・各フィルタサイズともシェイプは良く似ている。

7. 計算結果　dynOneEqEddyモデル：③部での軸方向速度比較

③

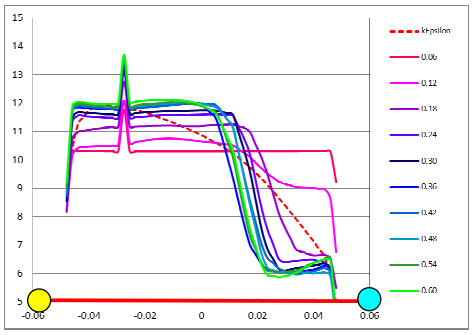


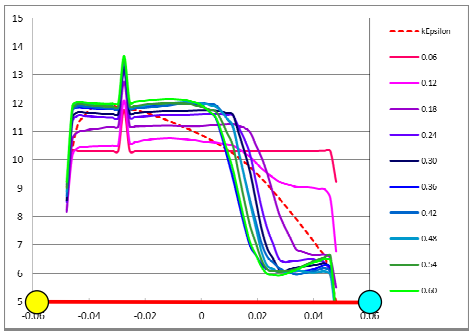
円管断面の中央部を黄丸から青丸方向に軸方向速度を表示

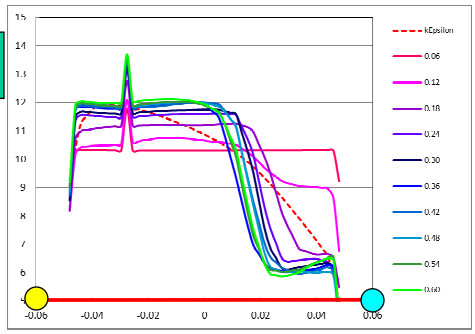
cubeRootVol　maxDeltaxyz

vanDriest

図配置





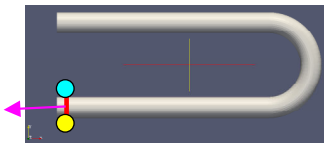


・1本スパイク状の山が出る。これはなぜか？

・各フィルタサイズともシェイプは良く似ている。

7. 計算結果　dynOneEqEddyモデル：④部での軸方向速度比較

④

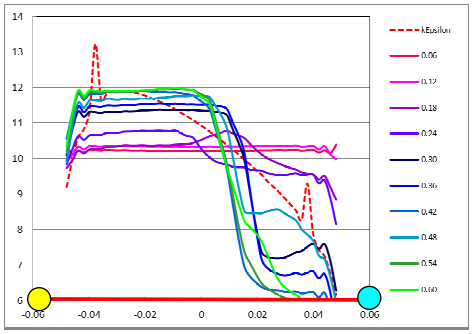


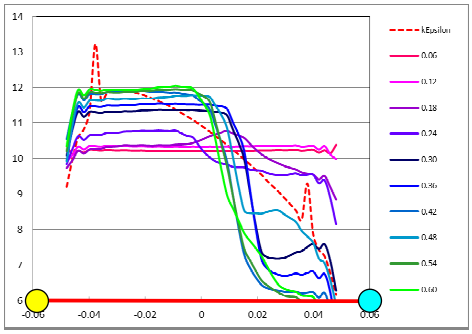
円管断面の中央部を黄丸から青丸方向に軸方向速度を表示

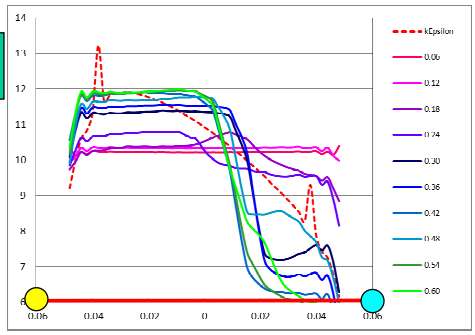
cubeRootVol　maxDeltaxyz

vanDriest

図配置







・各フィルタサイズともシェイプは良く似ている。