

ただで始める流体解析

interFoamを用いた貯水槽排水計算

2012.3.10

1/50

interFoamを用いた貯水槽排水計算

1. 計算概要
2. モデルの設定
3. モデルの理論解
4. メッシュの設定
5. 計算条件の設定
6. 計算結果
7. まとめ
8. 質疑・応答・その他

2012.3.10

2/50

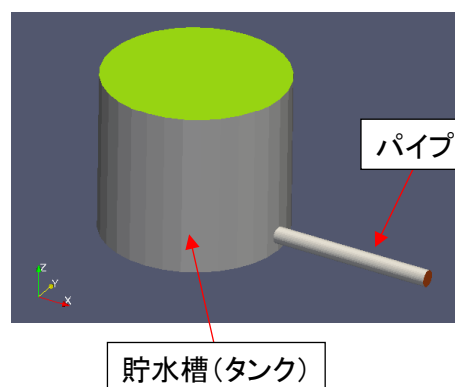
1. 計算概要

interFoamの使い方を理解するため、理論解(多分?)が存在する貯水槽排水問題を計算し、結果を比較する。

2. モデルの設定

- ・貯水槽(以下タンク)の底面部にパイプが接続されており、そこから流体が流れ出る場合のタンク液面の変化をinterFoamで計算する。
- ・タンクの形状としては円柱と立方体、パイプの形状としては円柱と直方体とする。

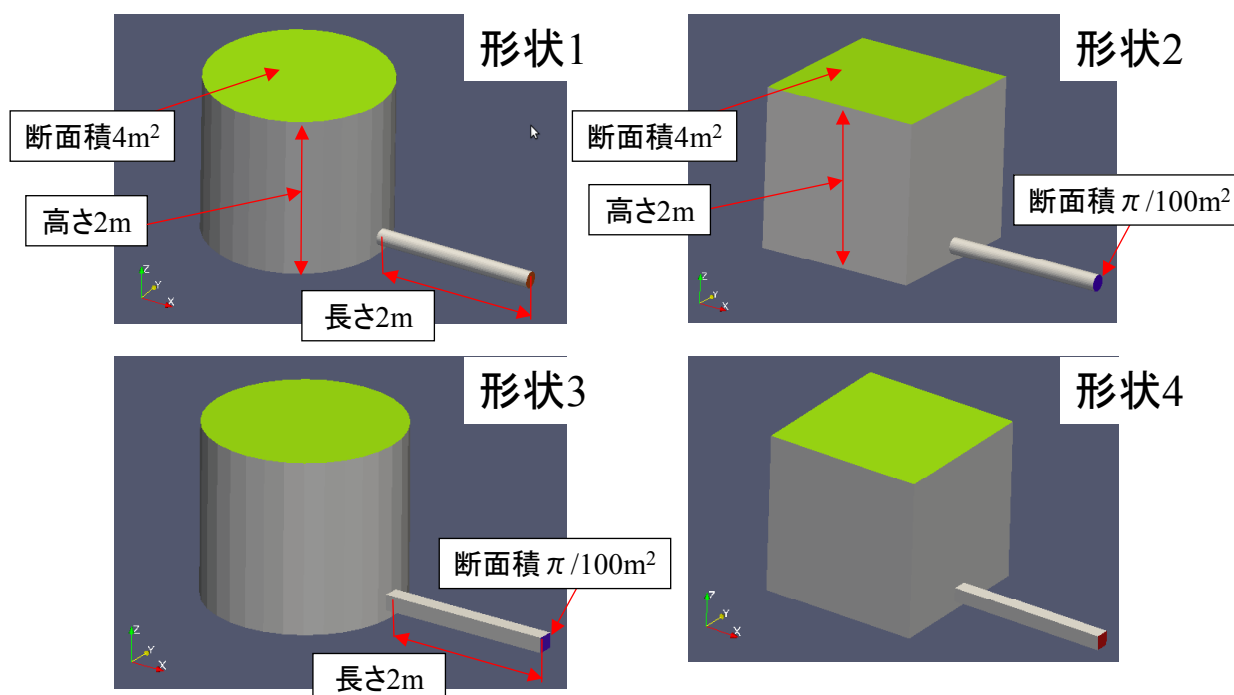
形状設定	形状1	形状2	形状3	形状4
タンク形状	円柱形	立方体	円柱形	立方体
断面(m^2)	4	4	4	4
半径(m)	1.128379		1.128379	
辺長さ(m)		2		2
高さ(m)	2	2	2	2
パイプ形状	円柱形	円柱形	直方体	直方体
断面(m^2)	$\pi/100$	$\pi/100$	$\pi/100$	$\pi/100$
半径(m)	0.1	0.1		
辺長さ(m)			0.177245	0.177245
長さ(m)	2	2	2	2



2012.3.10

3/50

2. モデルの設定



2012.3.10

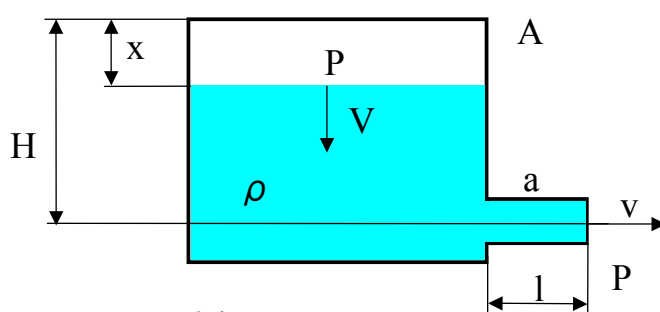
4/50

3. モデルの理論解

2012.3.10

5/50

3. モデルの理論的解 その1(損失を考慮しない場合)



P:大気圧
 ρ :液体密度
 A:貯水槽の断面積
 a:パイプの断面積
 H:貯水槽の高さ
 x:水面の位置
 V:水面の降下速度
 v:パイプからの排出速度
 l:パイプの長さ

●質量保存則

$$AV = av \quad \text{式(1)}$$

●ベルヌーイの定理

$$\frac{V^2}{2} + \frac{P}{\rho} + g(H - x) = \frac{v^2}{2} + \frac{P}{\rho} \quad \text{式(2)}$$

式(1), (2)からVを求める。

$$V = \sqrt{\frac{2g(H-x)}{\left(\frac{A}{a}\right)^2 - 1}} \quad \text{式(3)}$$

水面の降下する速度は

$$V = \frac{dx}{dt} \quad \text{式(4)}$$

2012.3.10

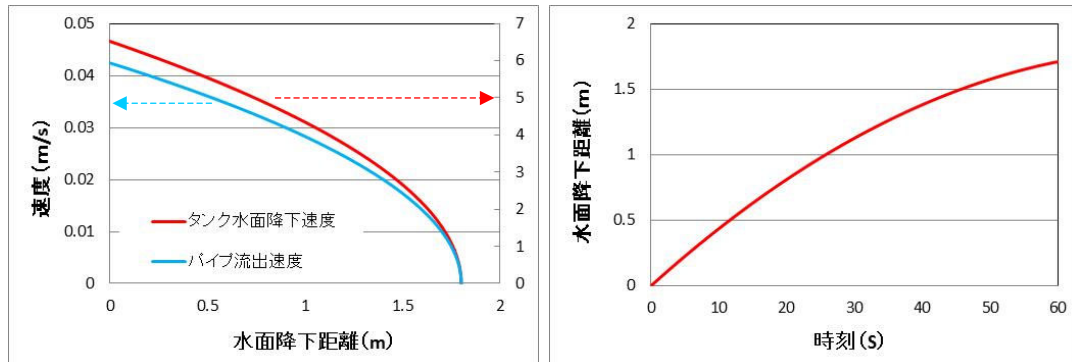
6/50

3. モデルの理論的解 その1(損失を考慮しない場合)

式(3)と式(4)からtとxの関係は、微分方程式を展開していくと

$$t = \frac{\sqrt{H} - \sqrt{H-x}}{\sqrt{\frac{g}{2\left\{\left(\frac{A}{a}\right)^2 - 1\right\}}}} \quad \text{式(5) または} \quad x = H - \left(\sqrt{H} - t \sqrt{\frac{g}{2\left\{\left(\frac{A}{a}\right)^2 - 1\right\}}} \right)^2 \quad \text{式(6)}$$

水面降下速度，パイプ流出速度の水面降下距離の関係を図に示す。

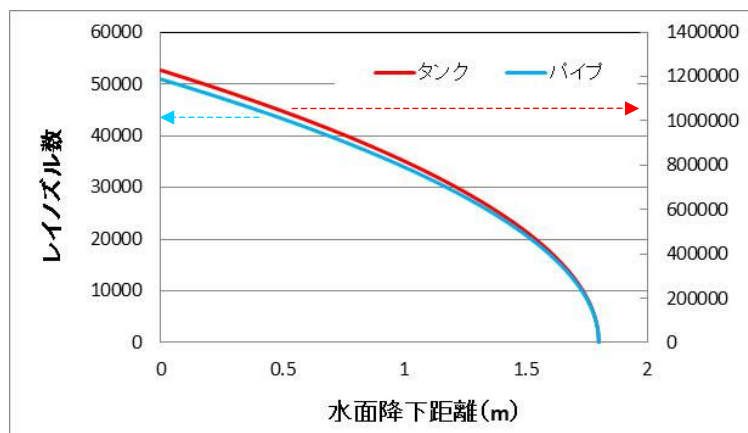


2012.3.10

7/50

3. モデルの理論的解 その1(損失を考慮しない場合)

パイプ，タンクのレイノズル数を計算すると乱流状態となっている。



タンク
直径 2.2

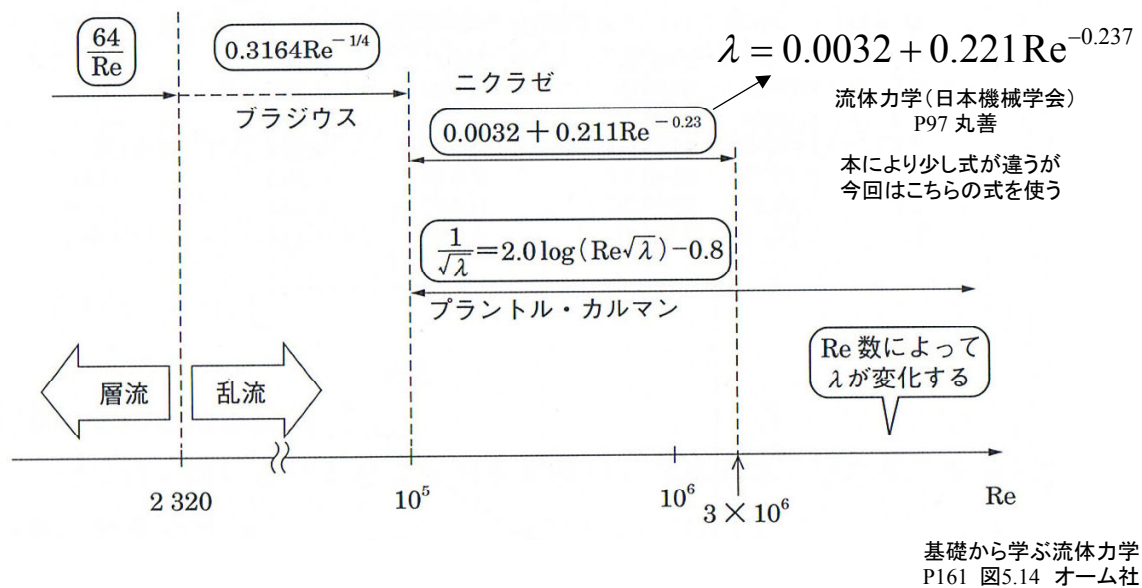
パイプ
直径 0.2

2012.3.10

8/50

3. モデルの理論的解 その1(損失を考慮しない場合)

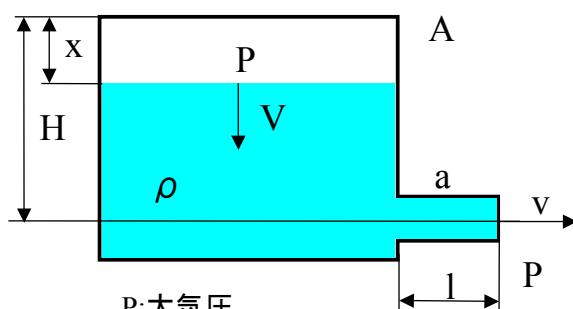
パイプでの最大Reは 1.2×10^6 程度, タンクでの最大Reは 9.6×10^4 程度である。このため損失をニクラゼ等の式で与える必要が出てくる(?)。



2012.3.10

9/50

3. モデルの理論的解 その2(パイプの流れに層流を仮定した場合)



P:大気圧
 ρ :液体密度
 A:貯水槽の断面積
 a:パイプの断面積
 d:パイプの直径
 H:貯水槽の高さ
 x:水面の位置
 V:水面の降下速度
 v:パイプからの排出速度
 l:パイプの長さ
 λ :パイプの管摩擦係数

●質量保存則

$$AV = av \quad \text{式(1)}$$

●ベルヌーイの定理,ダルシー・ワイエスバツバの摩擦損失を組み合わせる

$$\frac{V^2}{2} + \frac{P}{\rho} + g(H - x) = \frac{v^2}{2} + \frac{P}{\rho} + \lambda \frac{l}{d} \frac{v^2}{2}$$

式(7)

●パイプ内の流れを層流とすると管摩擦係数はレイノズル数の関数となる。

$$\lambda = \frac{64\nu}{vd} = \frac{64}{\text{Re}} \quad \text{式(8)}$$

式(1),(7),(8)より

2012.3.10

10/50

3. モデルの理論的解 その2(パイプの流れに層流を仮定した場合)

$$V = \frac{-\bar{B} + \sqrt{\bar{B}^2 - 4\bar{A}\bar{C}}}{2\bar{A}} \quad \text{式(9)}$$

$$\bar{A} = \left(\frac{A}{a}\right)^2 - 1 \quad \bar{B} = \frac{64\nu l A}{d^2 a} \quad \bar{C} = -2g(H - x)$$

水面の降下する速度は

$$V = \frac{dx}{dt} \quad \text{式(4)}$$

式(4),(9)から

$$\frac{dx}{dt} = \frac{-\frac{64\nu l A}{d^2 a} + \sqrt{\left(\frac{64\nu l A}{d^2 a}\right)^2 + 8g\left\{\left(\frac{A}{a}\right)^2 - 1\right\}(H - x)}}{2\left\{\left(\frac{A}{a}\right)^2 - 1\right\}} \quad \text{式(10)}$$

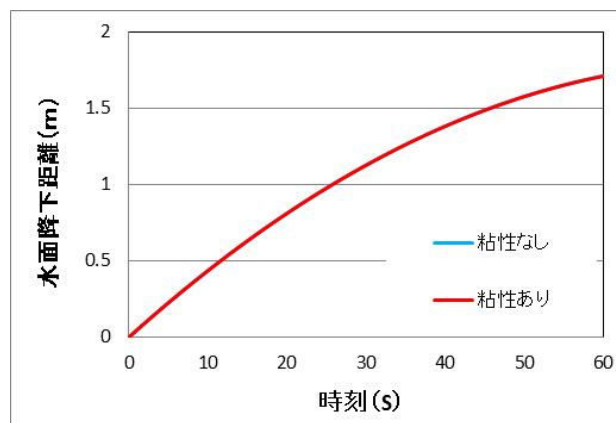
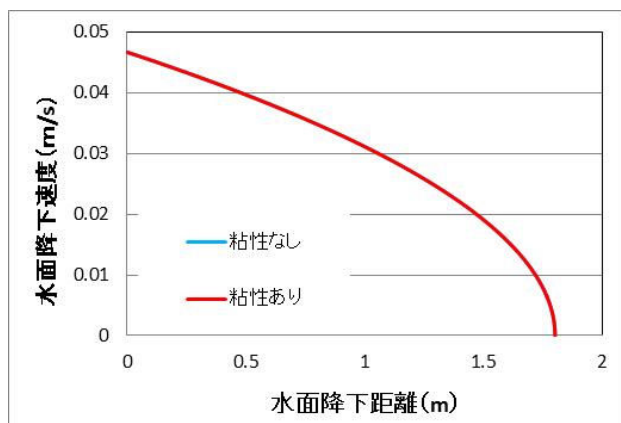
2012.3.10

11/50

3. モデルの理論的解 その2(パイプの流れに層流を仮定した場合)

式(10)の微分方程式を解けば良いが(学生時代に微積を勉強しておくべくであった), 手計算では簡単に求められないので, 4次のルンゲクッタ法を用いて解く。

層流を仮定した場合, 損失なし(粘性なし, モデルの理論解 その1)の場合と差がほとんど出ない。



2012.3.10

12/50

3. モデルの理論的解 その3(パイプの流れに乱流を仮定した場合)

●質量保存則

$$AV = av \quad \text{式(1)}$$

●ベルヌーイの定理

$$\frac{V^2}{2} + \frac{P}{\rho} + g(H - x) = \frac{v^2}{2} + \frac{P}{\rho} + \lambda \frac{l}{d} \frac{v^2}{2} \quad \text{式(7)}$$

●ニコラゼの式

$$\lambda = 0.0032 + 0.221 \operatorname{Re}^{-0.237}$$

$$\operatorname{Re} = \frac{vd}{\nu} \quad \text{式(11)}$$

式(1),(7),(11)から

$$\frac{V^2}{2} + g(H - x) = \left(\frac{A}{a}\right)^2 \frac{V^2}{2} + \left\{ 0.0032 + 0.221 \left(\frac{d}{\nu} \frac{A}{a} V\right)^{-0.237} \right\} \frac{l}{d} \frac{A^2 V^2}{2a^2} \quad \text{式(12)}$$

2012.3.10

13/50

3. モデルの理論的解 その3(パイプの流れに乱流を仮定した場合)

式(12)を時間で微分して変形すると式(13)になる。

$$\frac{dV}{dt} = \frac{g}{1 - \left(\frac{A}{a}\right)^2 - 0.0032 \frac{lA^2}{da^2} - 0.1948115 \left(\frac{dA}{\nu a}\right)^{-0.237} \frac{l}{d} \frac{A^2}{a^2} V^{-0.237}} \quad \text{式(13)}$$

これも手計算では到底解けそうにないため、4次のルンゲ
クッタ法を用いて式(4), (13)を連立させて解く。

$$\frac{dx}{dt} = V \quad \text{式(4)}$$

時刻0でのxの初期値は、 $x_{t=0} = 0$

時刻0でのVの初期値は、式(12)でx=0として求めるが、手計算は難しいので
2分法を利用して求める。

$$\frac{V^2}{2} + gH = \left(\frac{A}{a}\right)^2 \frac{V^2}{2} + \left\{ 0.0032 + 0.221 \left(\frac{d}{\nu} \frac{A}{a} V\right)^{-0.237} \right\} \frac{l}{d} \frac{A^2 V^2}{2a^2} \quad \text{式(14)}$$

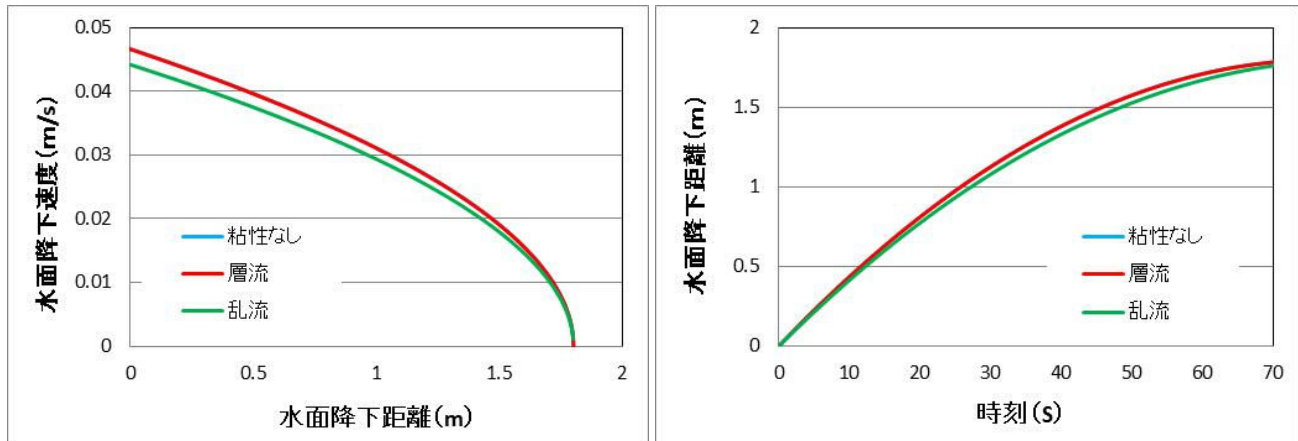
式(12)でx=0とした式

2012.3.10

14/50

3. モデルの理論的解 その3(パイプの流れに乱流を仮定した場合)

式を解いた結果から、水面降下速度、水面降下距離と時間の関係を示す。乱流の影響でタンクの水面降下速度が粘性なし、粘性あり(層流)に比較して少し遅い。このため、水面降下距離が時刻歴でみると先の2つのモデルと比較して若干少ない。



2012.3.10

15/50

4. メッシュの設定

2012.3.10

16/50

4. メッシュの設定:blockMeshDict

```

blockMeshDict
/*----- C++ -----*/
//
// \ \ \ \ \ Field | OpenFOAM: The Open Source CFD
// \ \ \ \ \ Operation | Version: 2.1.x
// \ \ \ \ \ And | Web: www.OpenFOAM.org
// \ \ \ \ \ Manipulation |
//
FoamFile
{
    version      2.0;
    format       ascii;
    class        dictionary;
    object       blockMeshDict;
}

// *****

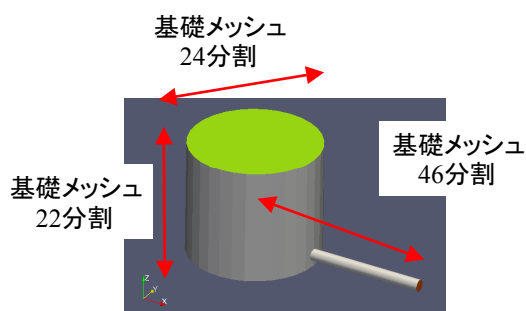
convertToMeters 1;

vertices
(
    (-1.2 -1.2 -0.1)
    ( 3.4 -1.2 -0.1)
    ( 3.4  1.2 -0.1)
    (-1.2  1.2 -0.1)
    (-1.2 -1.2  2.1)
    ( 3.4 -1.2  2.1)
    ( 3.4  1.2  2.1)
    (-1.2  1.2  2.1)
);

blocks
(
    hex (0 1 2 3 4 5 6 7) (46 24 22) simpleGrading (1 1 1)
);

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

```



幅(mm)	基礎分割数	基礎メッシュ(0)	基礎メッシュ(1)	基礎メッシュ(2)	基礎メッシュ(3)	基礎メッシュ(4)	基礎メッシュ(5)
4600	46	100	50	25	12.5	6.25	3.125
2400	24	100	50	25	12.5	6.25	3.125
2200	22	100	50	25	12.5	6.25	3.125

2012.3.10

17/50

4. メッシュの設定:snappyHexMeshDict

```

snappyHexMeshDict
format      ascii;
class       dictionary;
object      snappyHexMeshDict;
}

// *****

// Which of the steps to run
castellatedMesh true;
snap           true;
//addLayers    true;
addLayers      false; レイヤー層はなし

// Geometry. Definition of all surfaces. All
// searchableSurface.
// Surfaces are used
// - to specify refinement for any mesh cell
// - to specify refinement for any mesh cell
// - to 'snap' the mesh boundary to the surface
geometry
{
    inlet_m.stl
    {
        type triSurfaceMesh;
        name inlet_surface;
    }

    outlet_m.stl
    {
        type triSurfaceMesh;
        name outlet_surface;
    }

    wall_m.stl
    {
        type triSurfaceMesh;
        name wall_surface;
    }
}

```

stlモデルの設定

```

snappyHexMeshDict
// Expressed as fraction of perfect b
// nProcs). 0=balance always.
maxLoadUnbalance 0.10;

// Number of buffer layers between di
// 1 means normal 2:1 refinement rest
// refinement.
nCellsBetweenLevels 1;

// Explicit feature edge refinement
// Specifies a level for any cell int
// This is a featureEdgeMesh, read fr
features
{
    file      "wall_m.eMesh";
    level     0;
}

```

feature Edgeは
壁部のみ設定

```

snappyHexMeshDict_1
// every cell intersecting a surfa
// The second level is the maximum
// intersections where the interse
// angle > resolveFeatureAngle get

refinementSurfaces
{
    inlet_surface
    {
        // Surface-wise min and ma
        level (0 0);
    }

    outlet_surface
    {
        // Surface-wise min and ma
        level (0 2);
    }

    wall_surface
    {
        // Surface-wise min and ma
        level (0 0);
    }
}

```

リファインレベルの設定

2012.3.10

18/50

4. メッシュの設定

```

mesh_para.sh ✕
echo "Start mesh"

echo "Copy of controlDict_mesh"
cp -r system/controlDict_mesh system/controlDict

echo "blockMesh"
$runApplication blockMesh > log.blockMesh

echo "surfaceFeatureExtract"
$runApplication surfaceFeatureExtract -includedAngle 150 -writeObj constant/triSurface/wall_m.stl wall_m

echo "decomposePar"
cp -r system/decomposeParDict_1 system/decomposeParDict
$runApplication decomposePar > log.decomposePar_1

echo "Copy data"
cp -r constant/triSurface/ processor0/constant/
cp -r constant/triSurface/ processor1/constant/
cp -r constant/triSurface/ processor2/constant/
cp -r constant/triSurface/ processor3/constant/

echo "snappyHexMesh 4CPU"
cp -r system/snappyHexMeshDict_1 system/snappyHexMeshDict
$runApplication mpirun -np 4 snappyHexMesh -parallel > log.snappyHexMesh

echo "reconstructParMesh"
$runApplication reconstructParMesh -time 3 -mergeTol 1e-6 > log.reconstructParMesh

echo "checkMesh"
$runApplication checkMesh > log.checkMesh

echo "renumberMesh"
$runApplication renumberMesh -time 3 > log.renumberMesh

echo "rename holder name"
mv constant/polyMesh constant/polyMesh_blockMesh

```

```

echo "set data"
mv 3/polyMesh constant
rm -r 1
rm -r 2
rm -r 3
rm -r 4
rm -r processor0
rm -r processor1
rm -r processor2
rm -r processor3

echo "End of mesh"

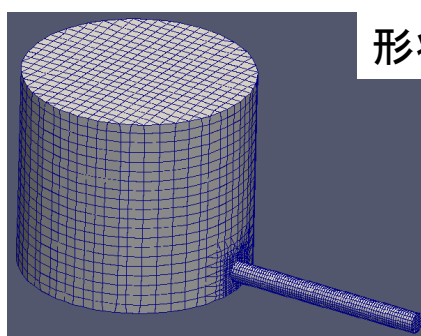
```

メッシュ作成方法は色々あると思いますが、本計算ではこうしています。
バッチで作成しています。

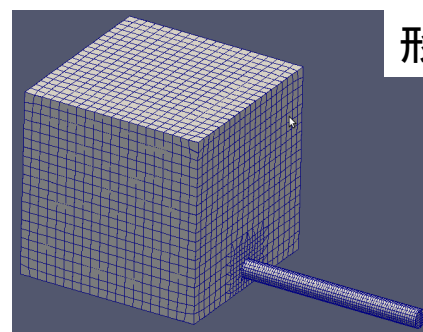
2012.3.10

19/50

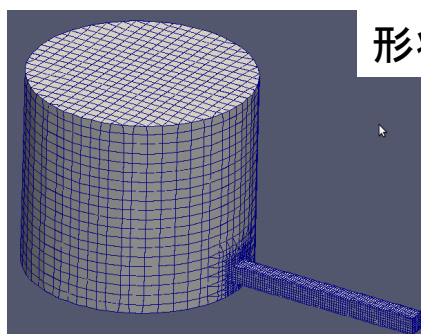
4. メッシュの設定



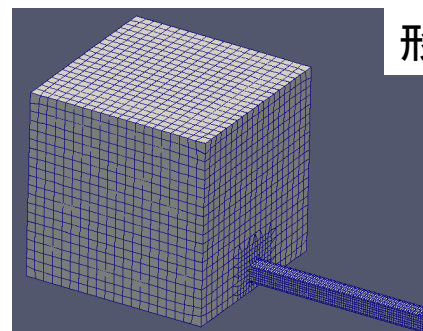
形状1



形状2



形状3



形状4

2012.3.10

20/50

5. 計算条件の設定

2012.3.10

21/50

5. 計算条件の設定:計算モデル

表に示す8モデルの計算を行う。

乱流モデル	層流			
計算モデル	CASE_3	CASE_4	CASE_5	CASE_6
形状	1	2	3	4

乱流モデル	kEpsilon			
計算モデル	CASE_3A	CASE_4A	CASE_5A	CASE_6A
形状	1	2	3	4

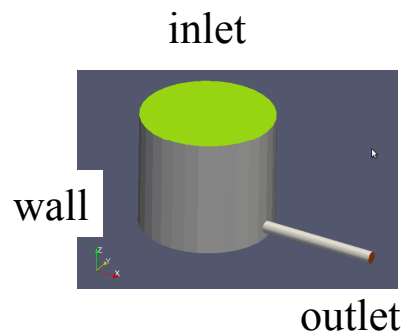
2012.3.10

22/50

5. 計算条件の設定:境界

```
U *
// -*- C++ -*-
// File generated by PyFoam - sorry for
FoamFile
{
    version 2.0;
    format ascii;
    class volVectorField;
    location "0";
    object U;
}
dimensions [ 0 1 -1 0 0 0 ];
internalField uniform (0 0 0);
boundaryField
{
    defaultFaces
    {
        type empty;
    }
    outlet_surface_Mesh
    {
        type pressureInletOutletVelocity;
        value uniform (0 0 0);
    }
    wall_surface_Mesh
    {
        type fixedValue;
        value uniform (0 0 0);
    }
    inlet_surface_Mesh
    {
        type pressureInletOutletVelocity;
        value uniform (0 0 0);
    }
}
```

```
p_rgh *
// -*- C++ -*-
// File generated by PyFoam - sorry for the
FoamFile
{
    version 2.0;
    format ascii;
    class volScalarField;
    object p_rgh;
}
dimensions [ 1 -1 -2 0 0 0 ];
internalField uniform 0;
boundaryField
{
    defaultFaces
    {
        type empty;
    }
    outlet_surface_Mesh
    {
        type totalPressure;
        p0 uniform 0;
        U U;
        phi phi;
        rho rho;
        psi none;
        gamma 1;
        value uniform 0;
    }
    wall_surface_Mesh
    {
        type buoyantPressure;
        value uniform 0;
    }
}
```



```
inlet_surface_Mesh
{
    type totalPressure;
    p0 uniform 0;
    U U;
    phi phi;
    rho rho;
    psi none;
    gamma 1;
    value uniform 0;
}
```

- DamBreakのチュートリアルを参考に設定する。
- OpenFOAMのバージョンは2.1.x

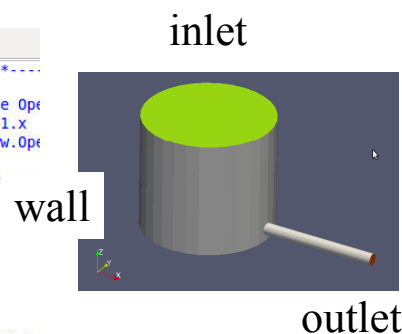
2012.3.10

23/50

5. 計算条件の設定:境界

```
alpha1 *
// -*- C++ -*-
// File generated by PyFoam - sorry for
FoamFile
{
    version 2.0;
    format ascii;
    class volScalarField;
    object alpha;
}
dimensions [ 0 0 0 0 0 0 ];
internalField uniform 0;
boundaryField
{
    defaultFaces
    {
        type empty;
    }
    outlet_surface_Mesh
    {
        type inletOutlet;
        inletValue uniform 0;
        value uniform 0;
    }
    wall_surface_Mesh
    {
        type zeroGradient;
    }
    inlet_surface_Mesh
    {
        type inletOutlet;
        inletValue uniform 0;
        value uniform 0;
    }
}
```

```
setFieldsDict *
// -*- C++ -*-
FoamFile
{
    version 2.0;
    format ascii;
    class dictionary;
    location "system";
    object setFieldsDict;
}
// *****
defaultFieldValues
{
    volScalarFieldValue alpha1 0
};
regions
{
    boxToCell
    {
        box (-1.2 -1.2 -0.8) (1.2 1.2 2.2);
        fieldValues
        {
            volScalarFieldValue alpha1 1
        };
    }
};
```



モデルの全領域を
水領域にする

2012.3.10

24/50

5. 計算条件の設定: 乱流モデル(層流), 物性値, 重力

turbulenceProperties

```

FoamFile
{
  version      2.0;
  format       ascii;
  class        dictionary;
  location     "constant";
  object       turbulenceProperties;
}

simulationType laminar;
        
```

乱流モデルは層流

transportProperties

```

FoamFile
{
  version      2.0;
  format       ascii;
  class        dictionary;
  location     "constant";
  object       transportProperties;
}

phase1
{
  transportModel Newtonian;
  nu [ 0.2 -1 0 0 0 0 ] 1e-06;
  rho [ 1 -3 0 0 0 0 ] 1000;
}

phase2
{
  transportModel Newtonian;
  nu [ 0.2 -1 0 0 0 0 ] 1.48e-05;
  rho [ 1 -3 0 0 0 0 ] 1.205;
}

sigma      sigma [ 1 0 -2 0 0 0 ] 0.07;
        
```

動粘性係数
密度

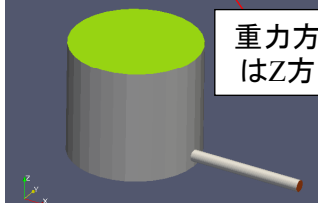
g

```

FoamFile
{
  version      2.0;
  format       ascii;
  class        uniformDimensionedVectorField;
  location     "constant";
  object       g;
}

dimensions   [0 1 -2 0 0 0];
value       ( 0 0 -9.81 );
        
```

重力方向はZ方向



2012.3.10

25/50

5. 計算条件の設定: 乱流モデル(乱流), k , ϵ

turbulenceProperties

```

FoamFile
{
  version      2.0;
  format       ascii;
  class        dictionary;
  location     "constant";
  object       turbulenceProperties;
}

simulationType RASModel;
        
```

乱流モデルはkEpsilon

k

```

FoamFile
{
  version      2.0;
  format       ascii;
  class        volScalarField;
  location     "0";
  object       k;
}

dimensions   [0 2 -2 0 0 0];
internalField uniform 0.01;

boundaryField
{
  defaultFaces
  {
    type empty;
  }

  outlet_surface_Mesh
  {
    type        inletOutlet;
    inletValue   uniform 0.2;
    value       uniform 0.2;
  }

  wall_surface_Mesh
  {
    type        kqRWallFunction;
    value       uniform 0.01;
  }

  inlet_surface_Mesh
  {
    type        inletOutlet;
    inletValue   uniform 0.01;
    value       uniform 0.01;
  }
}
        
```

epsilon

```

FoamFile
{
  version      2.0;
  format       ascii;
  class        volScalarField;
  location     "0";
  object       epsilon;
}

dimensions   [0 2 -3 0 0 0];
internalField uniform 0.01;

boundaryField
{
  defaultFaces
  {
    type empty;
  }

  outlet_surface_Mesh
  {
    type        inletOutlet;
    inletValue   uniform 0.6;
    value       uniform 0.6;
  }

  wall_surface_Mesh
  {
    type        epsilonWallFunction;
    value       uniform 0.01;
  }

  inlet_surface_Mesh
  {
    type        inletOutlet;
    inletValue   uniform 0.01;
    value       uniform 0.01;
  }
}
        
```

2012.3.10

5. 計算条件の設定: 乱流モデル(乱流), k , ε

kEpsilonモデルのパラメータは式(15)を使って計算する。

$$k = \frac{3}{2}(UI)^2$$

$$\varepsilon = C_{\mu}^{3/4} \frac{k^{3/2}}{l_h} \quad \text{式(15)}$$

$$l_h = 0.07d_h$$

部位	タンク	パイプ
U(初期速度)	0.04793	6.102647
I(乱流強さ,%)	0.05	0.05
k(乱流エネルギー)	8.61482E-06	0.139658627
dh(水力直径)	2.2	0.2
lh(0.07dh)	0.154	0.014
C_{μ}	0.09	0.09
ε (エネルギー消散率)	2.69793E-08	0.612569679

タンクの k , ε は値が小さいため, 少し大きめの値を設定する。

2012.3.10

27/50

5. 計算条件の設定: パイプからの流出量モニタ

```

controlDict
{
    functions
    (
        ①
        {
            type            faceSource;
            functionObjectLibs ("libfieldFunctionObjects.so");
            enabled          true;
            // outputControl outputTime;
            outputControl    timeStep;
            outputInterval    5;
            log              true;
            valueOutput      true;
            source            patch;
            sourceName        outlet_surface_Plane.004;
            operation         sum;
            fields
            (
                phi
            );
        }
        massFlowSimple ②
        {
            // Just for checking whether the other massflow is correct
            type patchMassFlow;
            patches
            (
                outlet_surface_Plane.004
            );
            verbose true;
        }
    );
}

massFlowSwak ③
{
    type patchExpression;
    verbose true;
    accumulations
    (
        sum
    );
    patches
    (
        outlet_surface_Plane.004
    );
    expression "phi";
};

libs
(
    "libOpenFOAM.so"
    "libgroovyBC.so"
    "libgroovyStandardBCs.so"
    "libsimpleSwakFunctionObjects.so"
    "libswakFunctionObjects.so"
);

```

3つの方法でモニタする

2012.3.10

28/50

5. 計算条件の設定: 計算用のバッチ

```
Tank_flow.sh ✖
echo "Start of calculation of Tank flow model"

echo "Copy data"
cp -r 0/backup/alpha1 0/
cp -r 0/backup/U 0/
cp -r 0/backup/p_rgh 0/
cp -r system/controlDict_interFoam system/controlDict

echo "setFields"
$runApplication setFields

echo "decomposePar"
cp -r system/decomposeParDict_1 system/decomposeParDict
$runApplication decomposePar > log.decomposePar

echo "interFoam 4CPU"
pyFoamPlotRunner.py mpirun -np 4 interFoam -parallel > log.interFoam

echo "reconstructPar"
$runApplication reconstructPar > log.reconstructPar

echo "set data"
rm -r processor0
rm -r processor1
rm -r processor2
rm -r processor3

pyFoamPlotWatcher.py --solver-not-running-anymore --hardcopy log.interFoam

echo "End of calculation"
```

2012.3.10

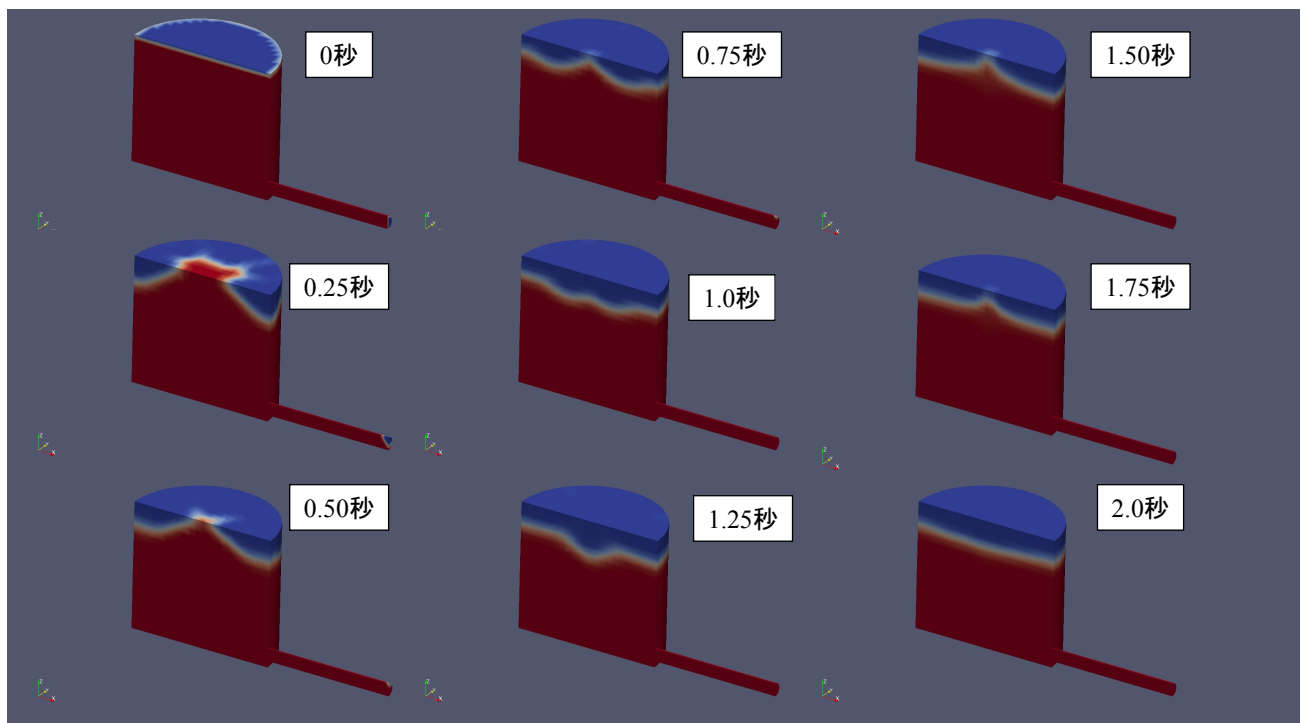
29/50

6. 計算結果

2012.3.10

30/50

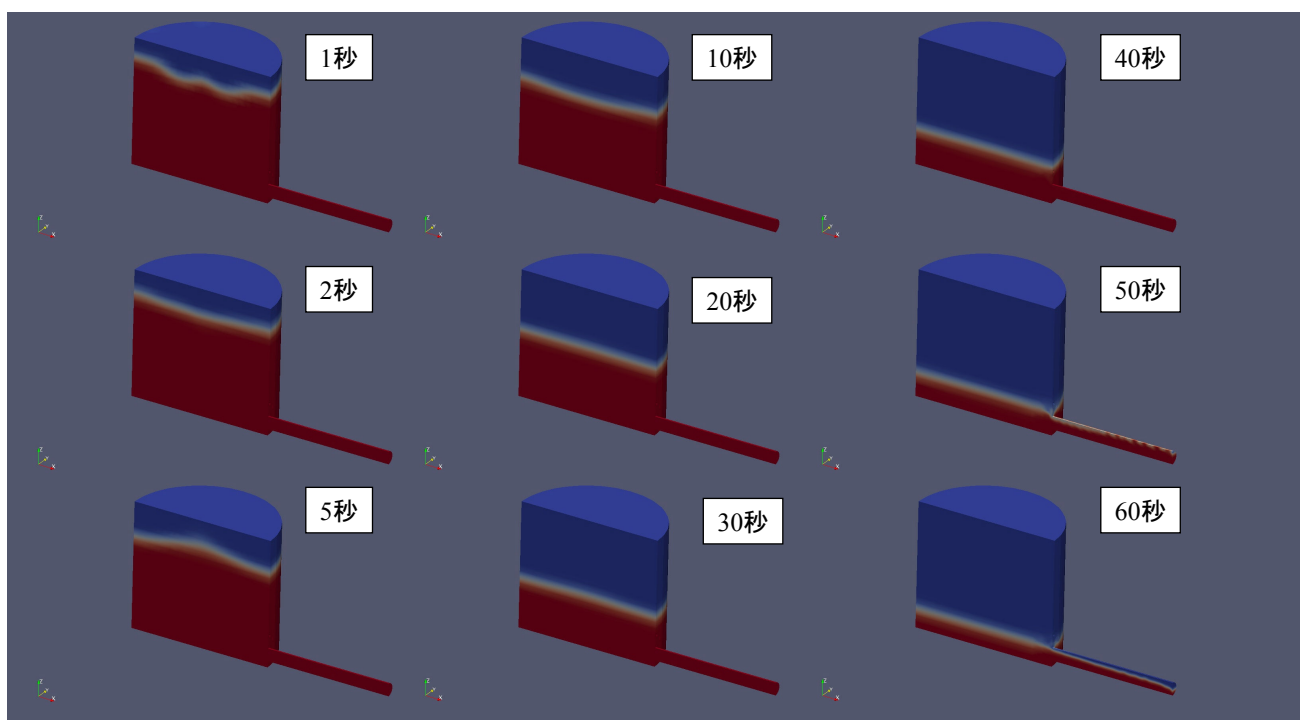
6. 計算結果 計算モデル 層流 CASE_3 初期



2012.3.10

31/50

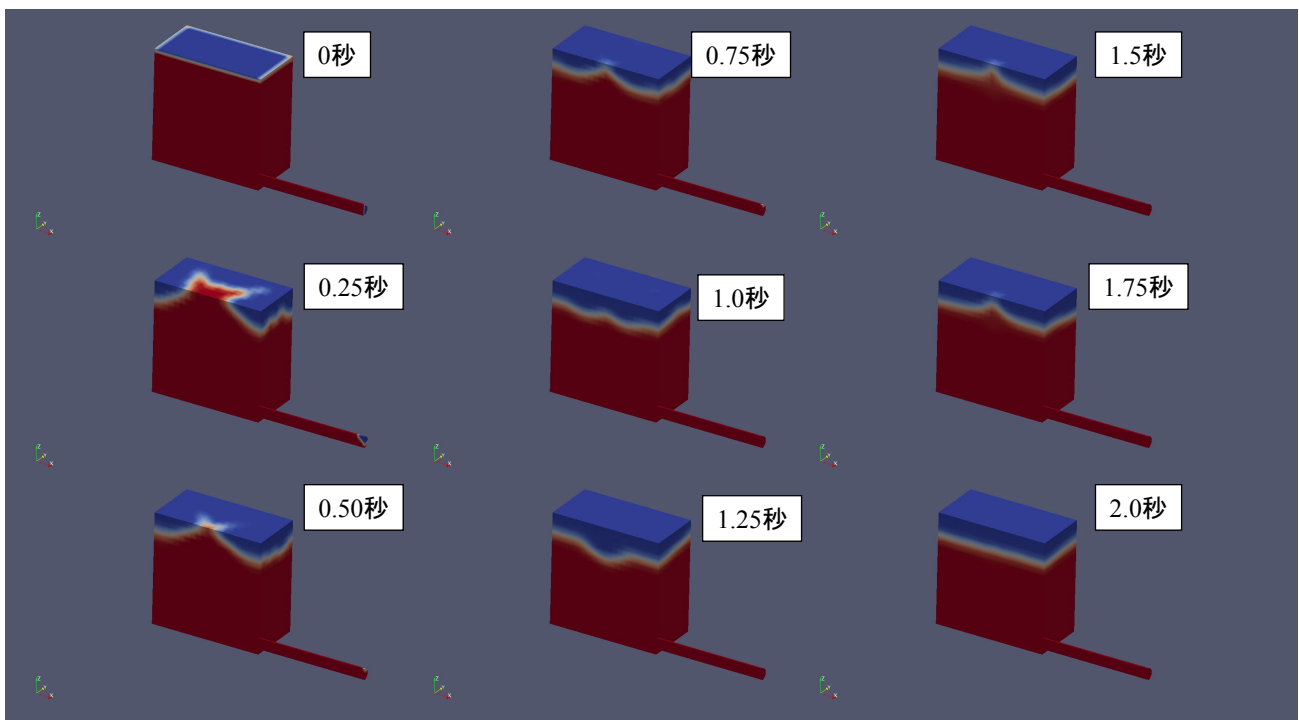
6. 計算結果 計算モデル 層流 CASE_3



2012.3.10

32/50

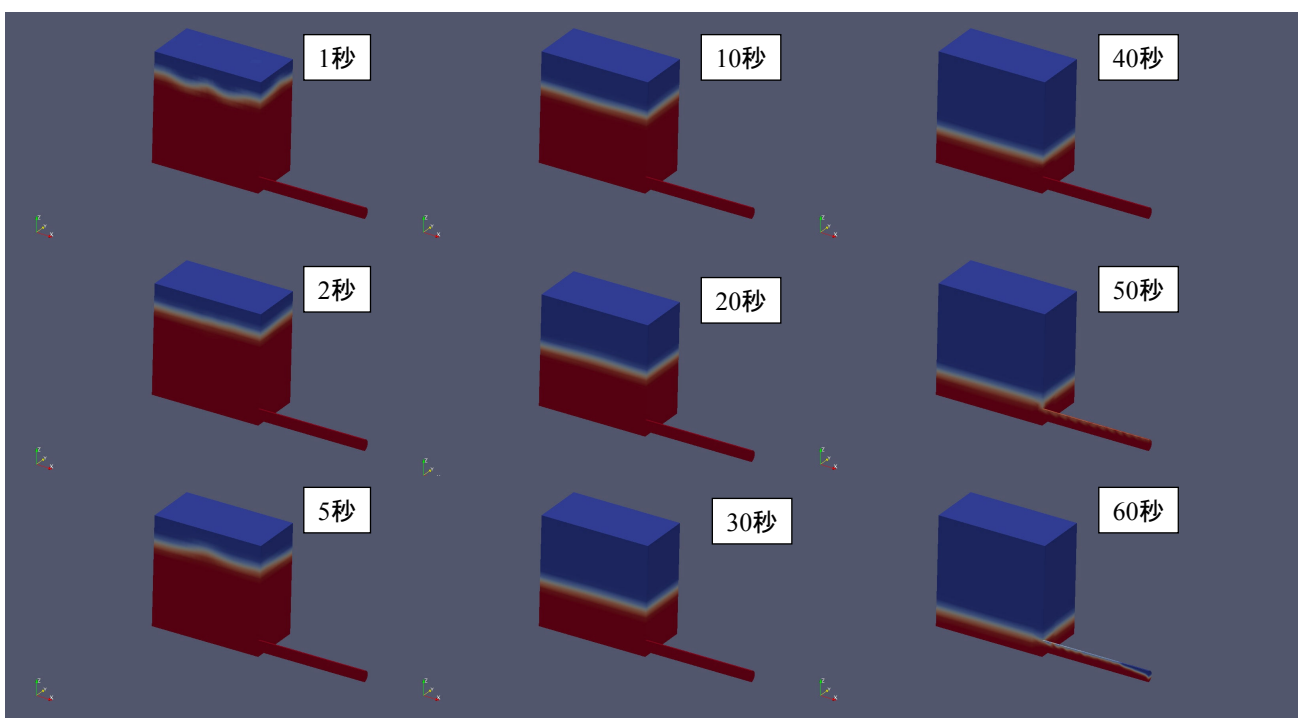
6. 計算結果 計算モデル 層流 CASE_4 初期



2012.3.10

33/50

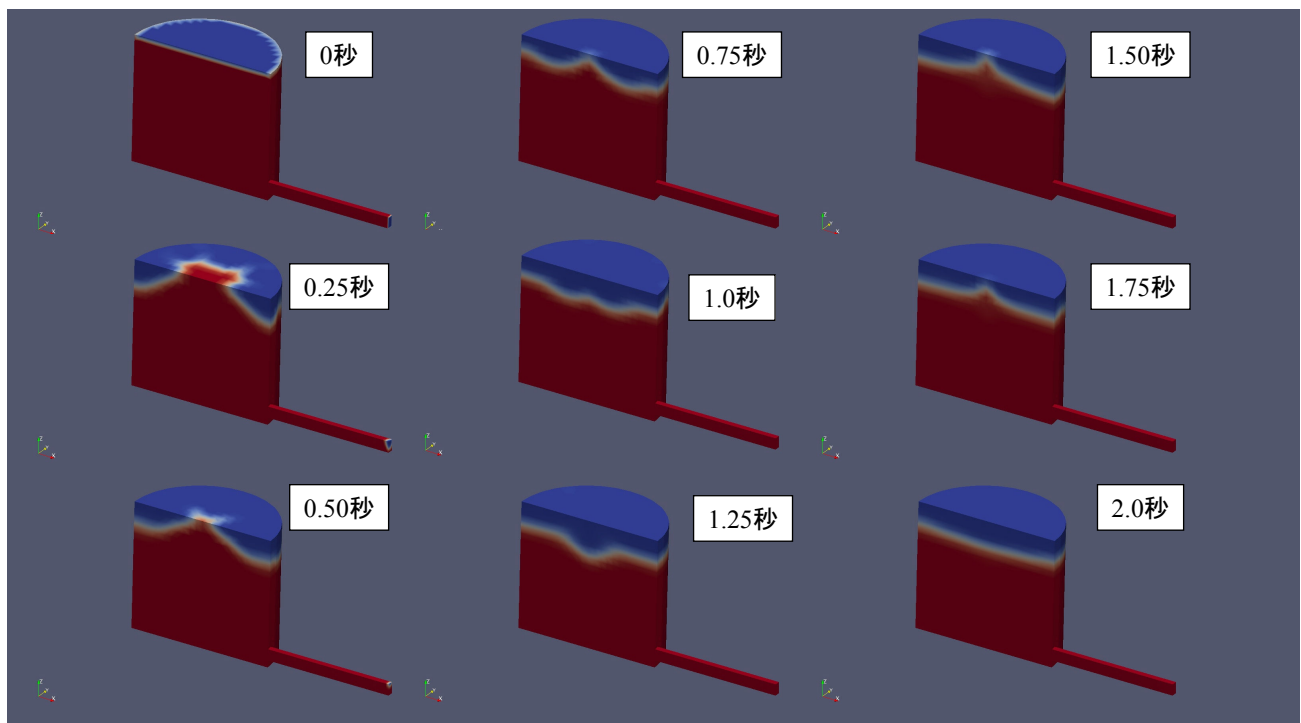
6. 計算結果 計算モデル 層流 CASE_4



2012.3.10

34/50

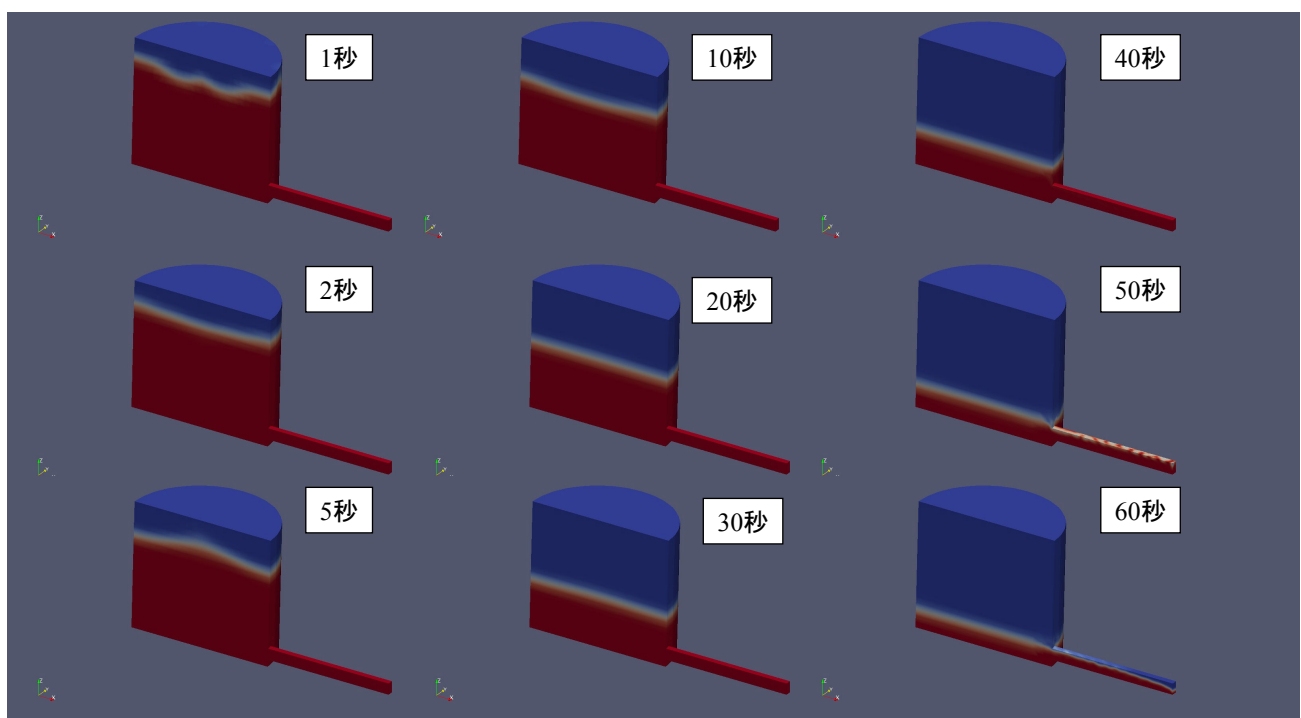
6. 計算結果 計算モデル 層流 CASE_5 初期



2012.3.10

35/50

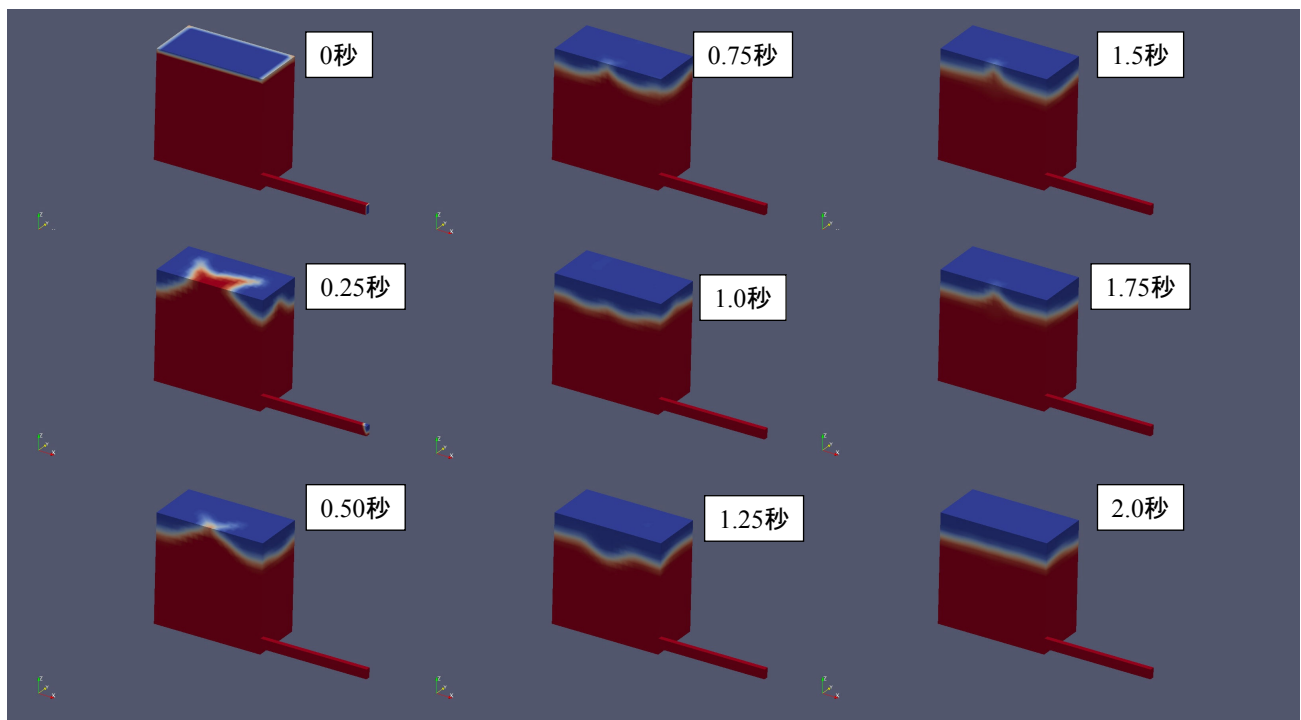
6. 計算結果 計算モデル 層流 CASE_5



2012.3.10

36/50

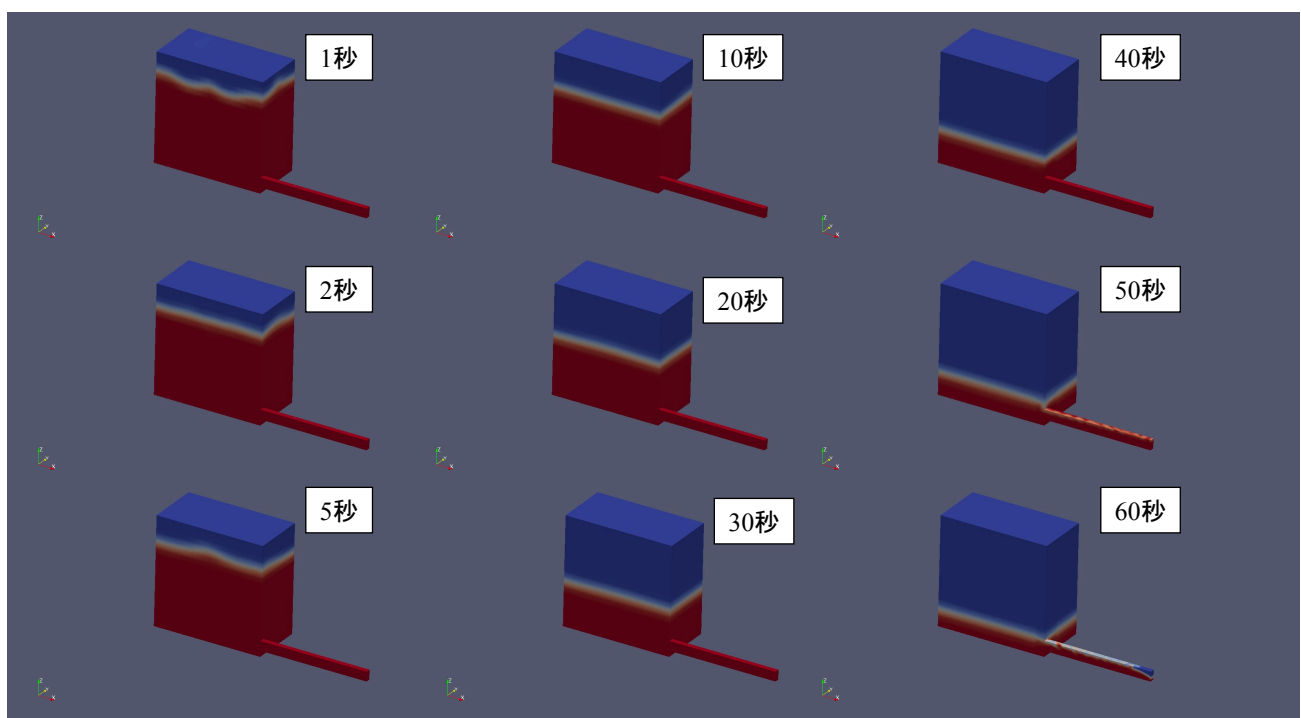
6. 計算結果 計算モデル 層流 CASE_6 初期



2012.3.10

37/50

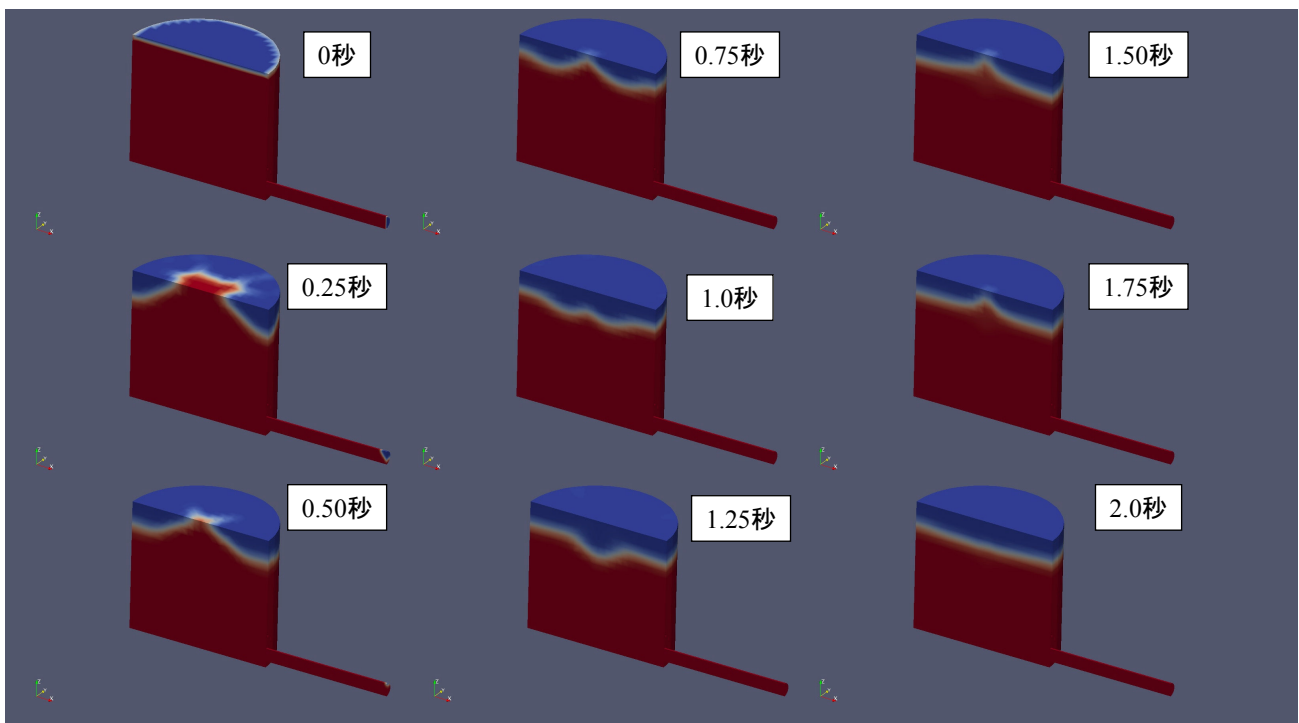
6. 計算結果 計算モデル 層流 CASE_6



2012.3.10

38/50

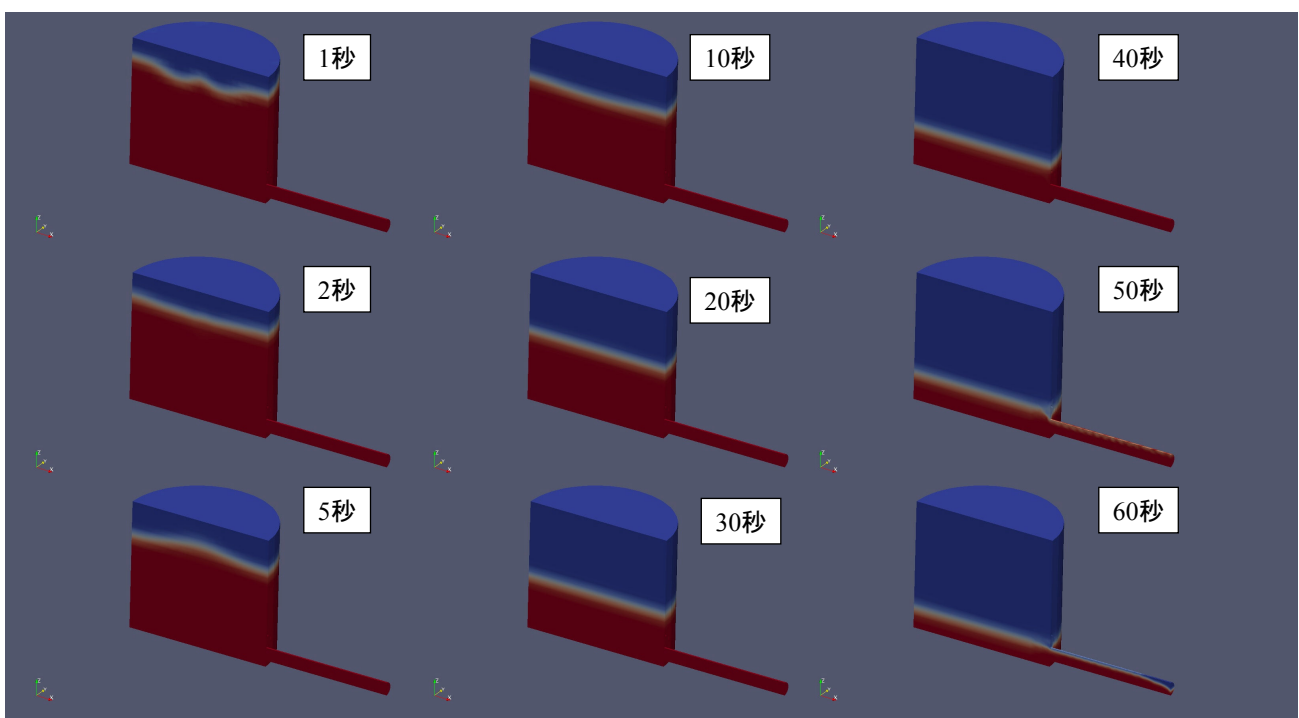
6. 計算結果 計算モデル 乱流 CASE_3A 初期



2012.3.10

39/50

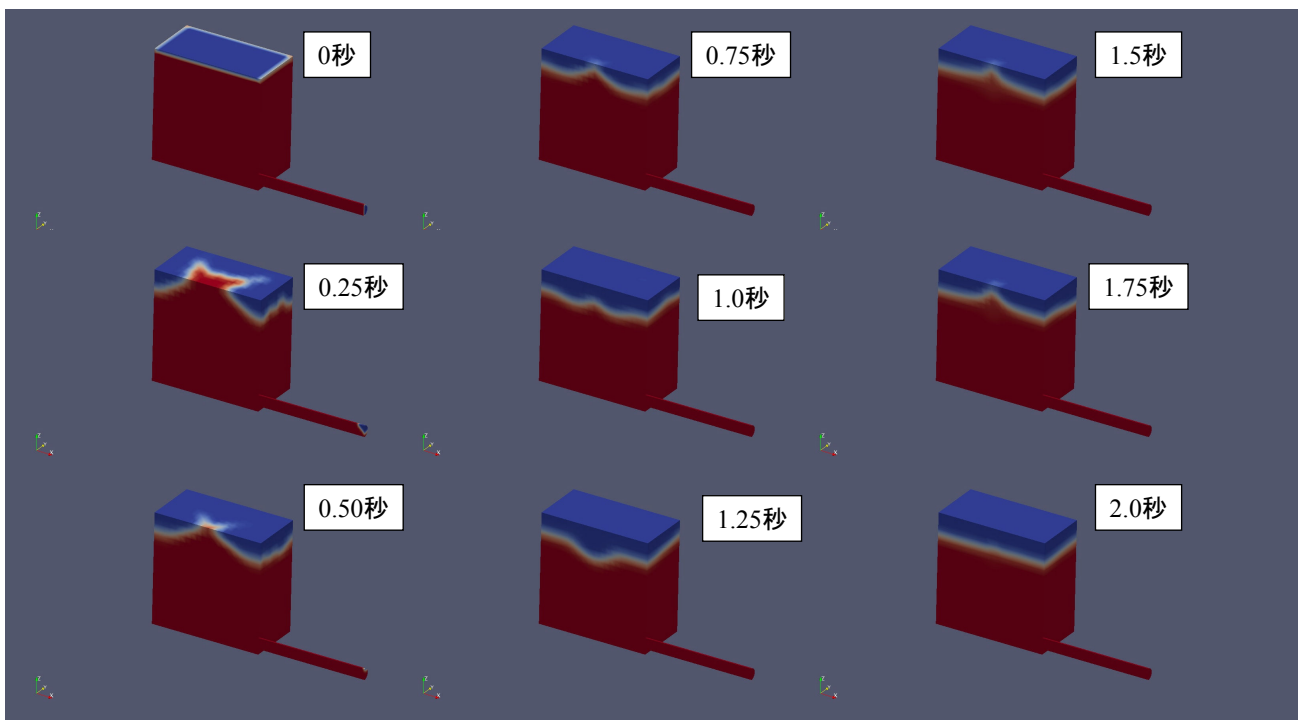
6. 計算結果 計算モデル 乱流 CASE_3A



2012.3.10

40/50

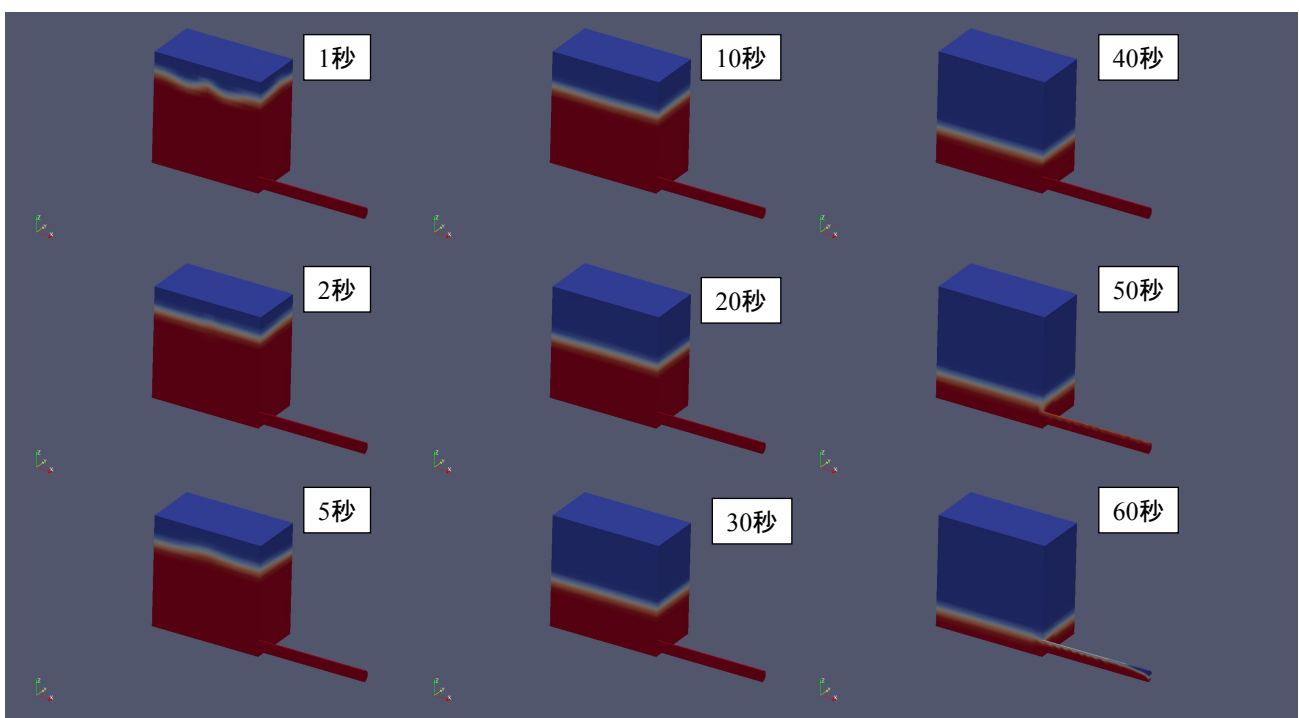
6. 計算結果 計算モデル 乱流 CASE_4A 初期



2012.3.10

41/50

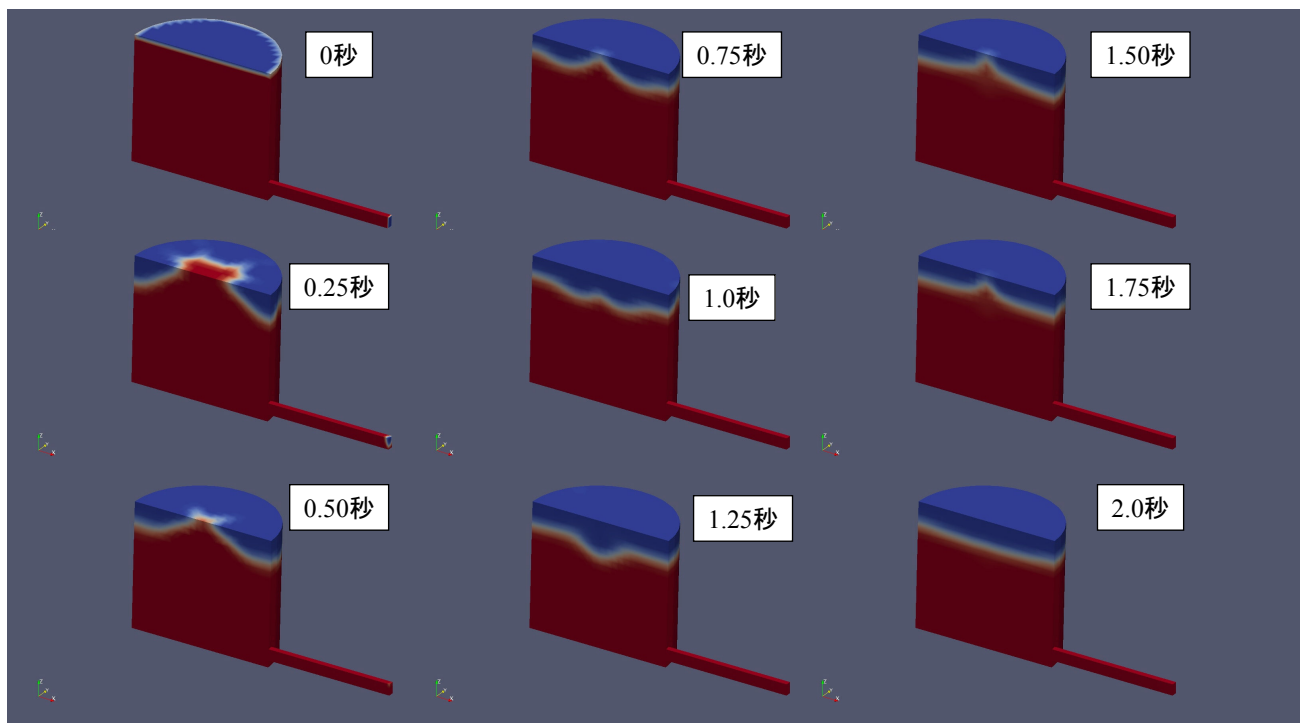
6. 計算結果 計算モデル 乱流 CASE_4A



2012.3.10

42/50

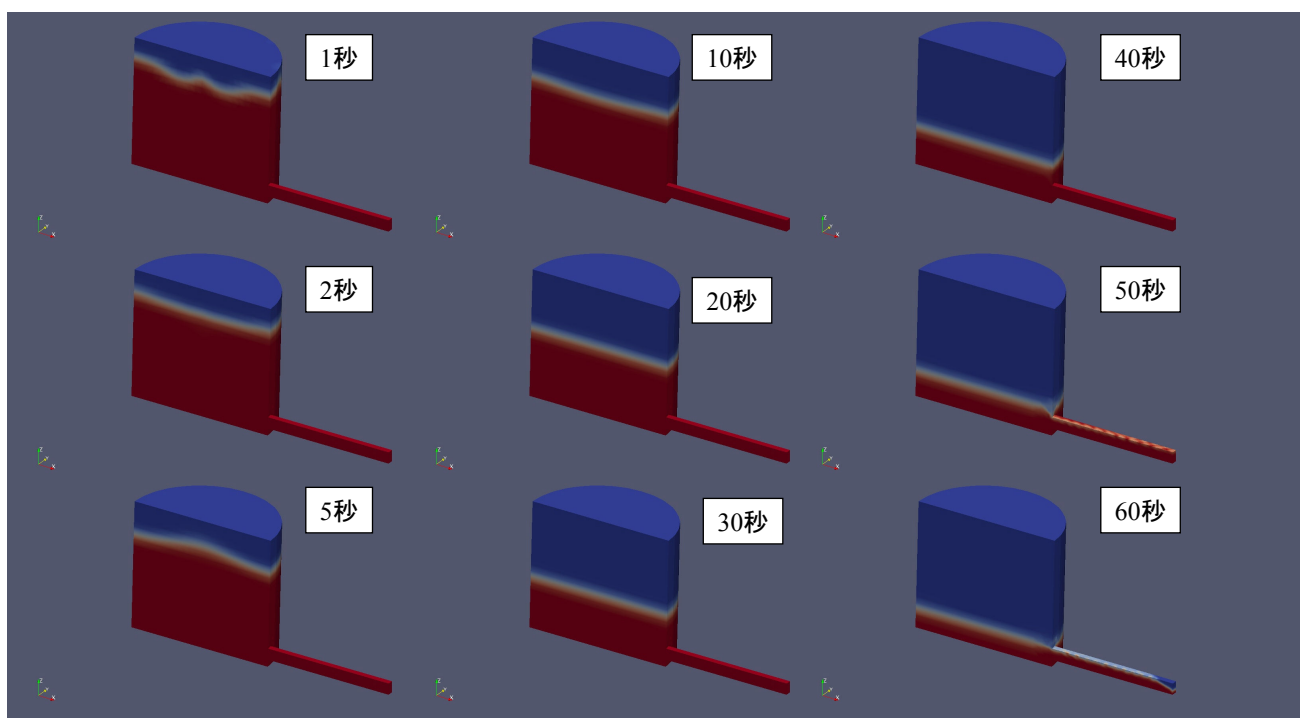
6. 計算結果 計算モデル 乱流 CASE_5A 初期



2012.3.10

43/50

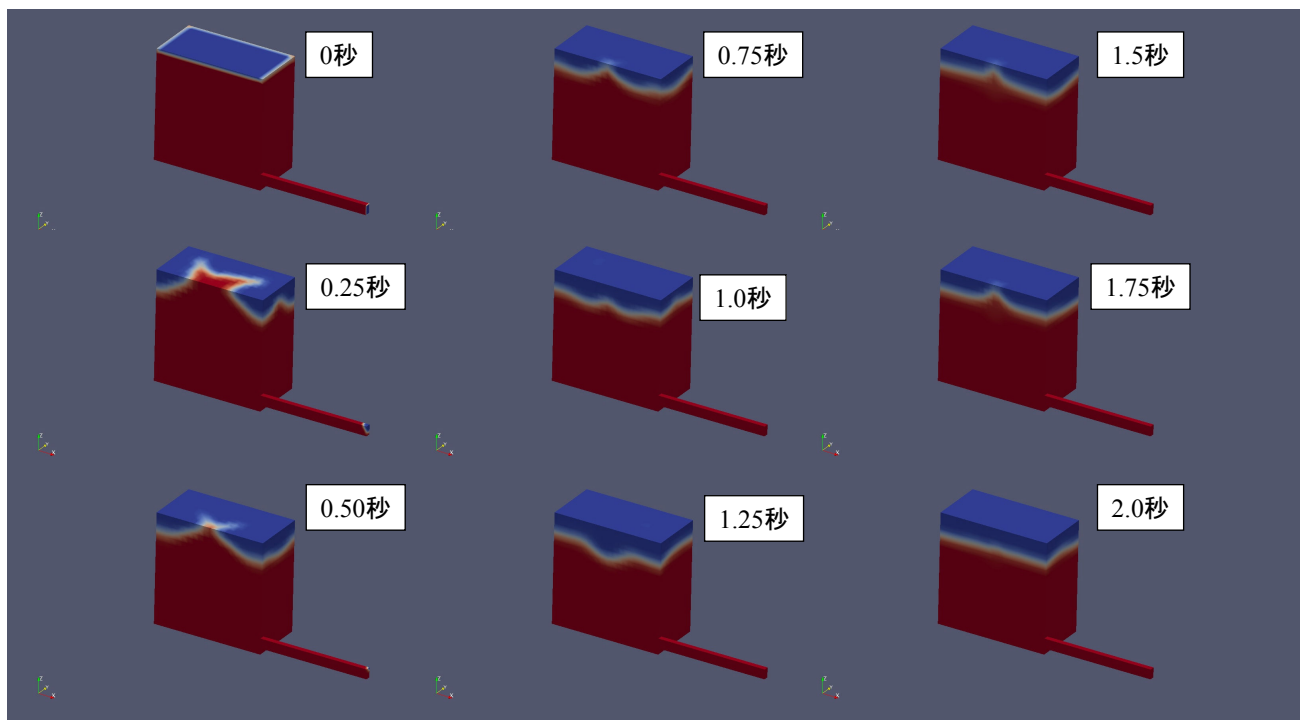
6. 計算結果 計算モデル 乱流 CASE_5A



2012.3.10

44/50

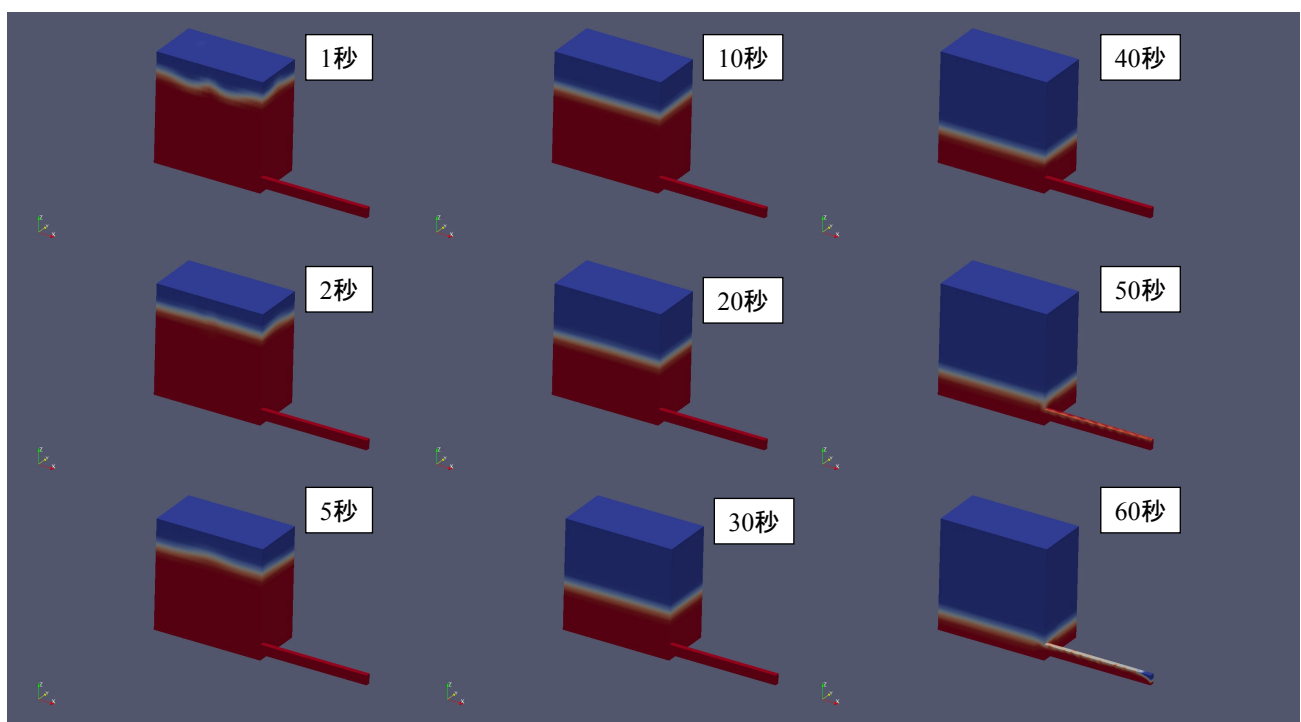
6. 計算結果 計算モデル 乱流 CASE_6A 初期



2012.3.10

45/50

6. 計算結果 計算モデル 乱流 CASE_6A

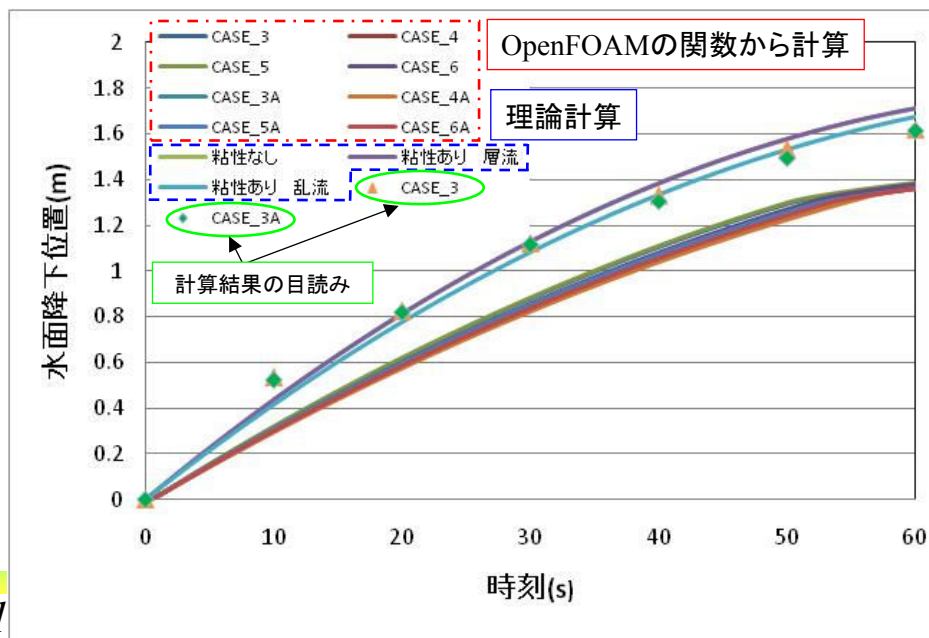


2012.3.10

46/50

6. 計算結果

水面降下位置と時間の関係を示す。OpenFOAMの関数から計算した値と計算結果から読み取った値とには差がある。計算結果から読み取った値は理論計算に近い結果になっている。(なぜだか不明)

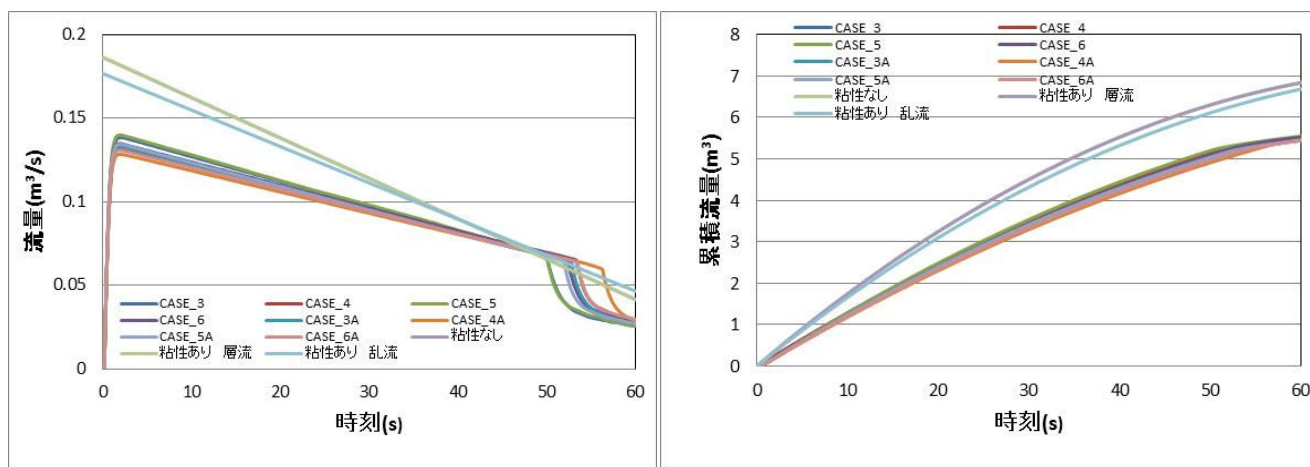


2012.3.1

7/50

6. 計算結果

流量, 累積流量についても, 先と同じ傾向となっている。



2012.3.10

48/50

7. まとめ

- ・貯水槽排水問題について理論値と計算値を比較した。
- ・排水初期の時間帯では、水面が3次元的に大きく変動している。
- ・OpenFOAMの関数から計算した値と、図から読み取った値に相違がみられる。(原因不明)

2012.3.10

49/50

8. 質疑・応答・その他

- ・排水初期の時間帯で水面が3次元的に大きく変動している理由は何にか？
- ・境界条件の設定は合っているか？
- ・関数で書き出した流量とアニメーションで書いた図から算出した流量が大きく異なる理由はどこにあるのか？
- ・計算で利用した一部のモデルは勉強会のホームページにアップ
CASE3, CASE3A OpenFOAMのバージョンは2.1, swak4Foam必要, 並列数は4。
mesh_para.shでメッシュ作成
Tank_flow.shで計算

2012.3.10

50/50