# twoLiquidMixingDyMFoamを用いたタンクでの塩水混合解析（その1）

TM

# はじめに

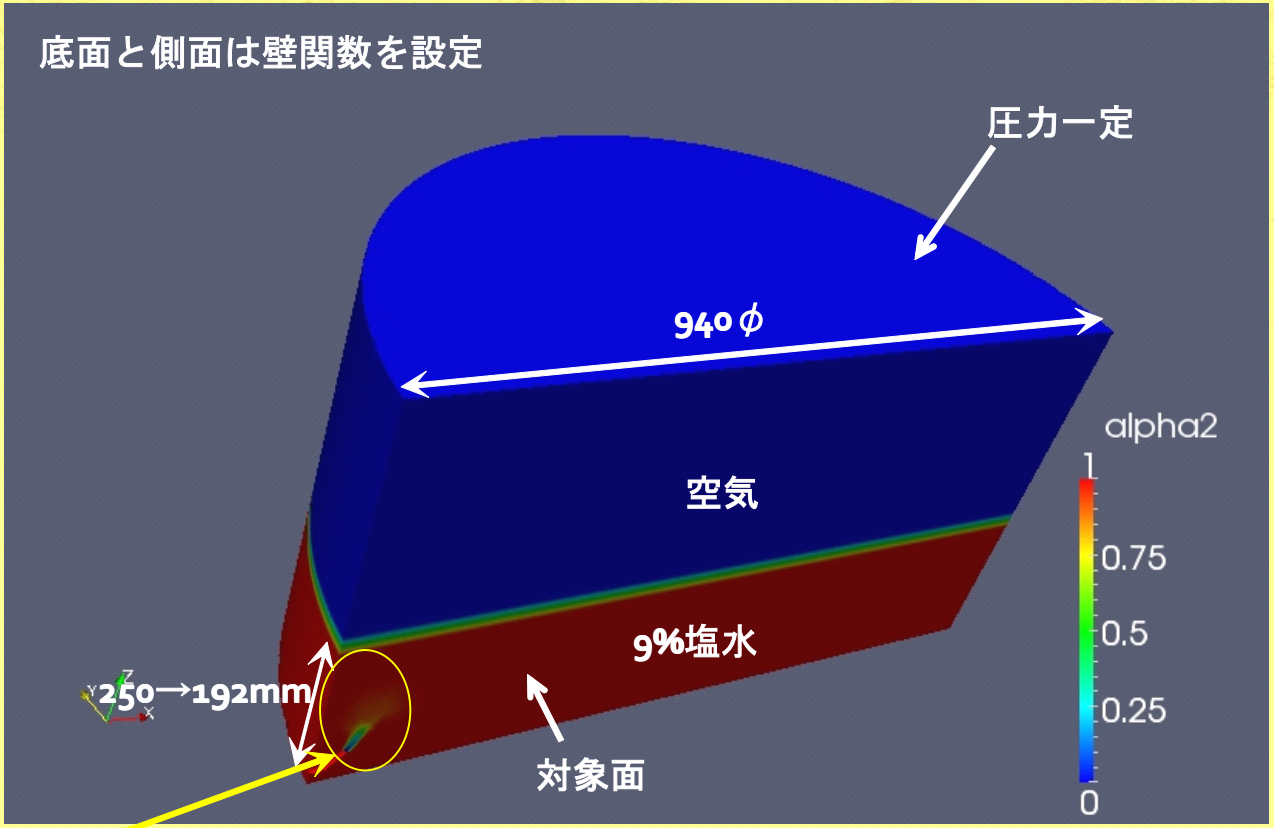
- タンク内の水と塩水の混合、空気との界面の解析を/multiphase/interMixingFoamで実施中

→計算量が膨大で計算時間が長い

→計算量を減らしたい

- /multiphase/twoLiquidMixingFoamで境界を移動できれば、空気部分の計算量を減らせる。計算も安定になるかもしれない。
- /incompressible/pimpleDyMFoam/movingConeを参考に、twoLiquidMixingFoamを改造したtwoLiquidMixingDyMFoamを紹介
- OpenFOAM　ver.2.1.0（他のverではコンパイル不能）

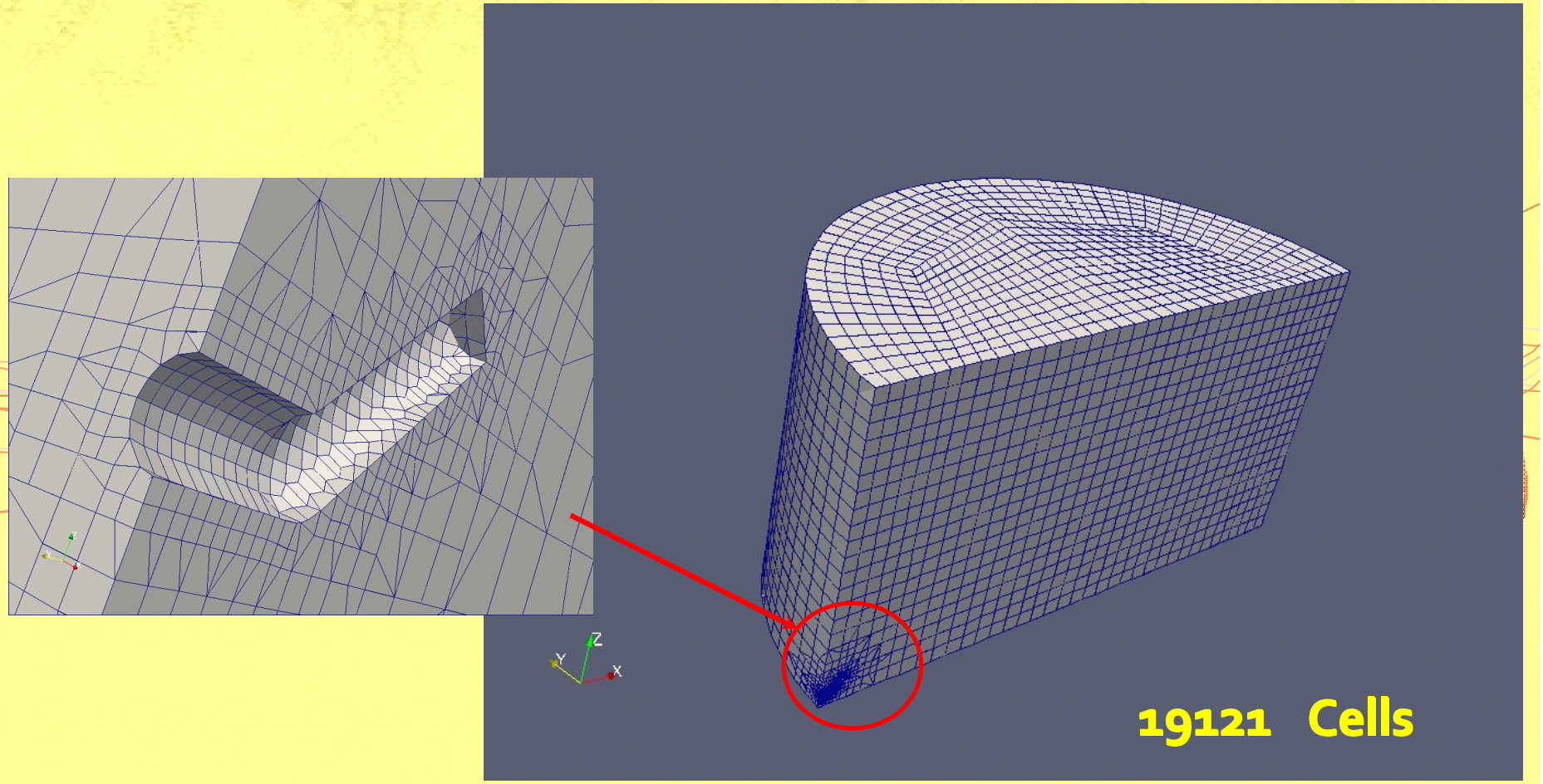
# 3次元解析



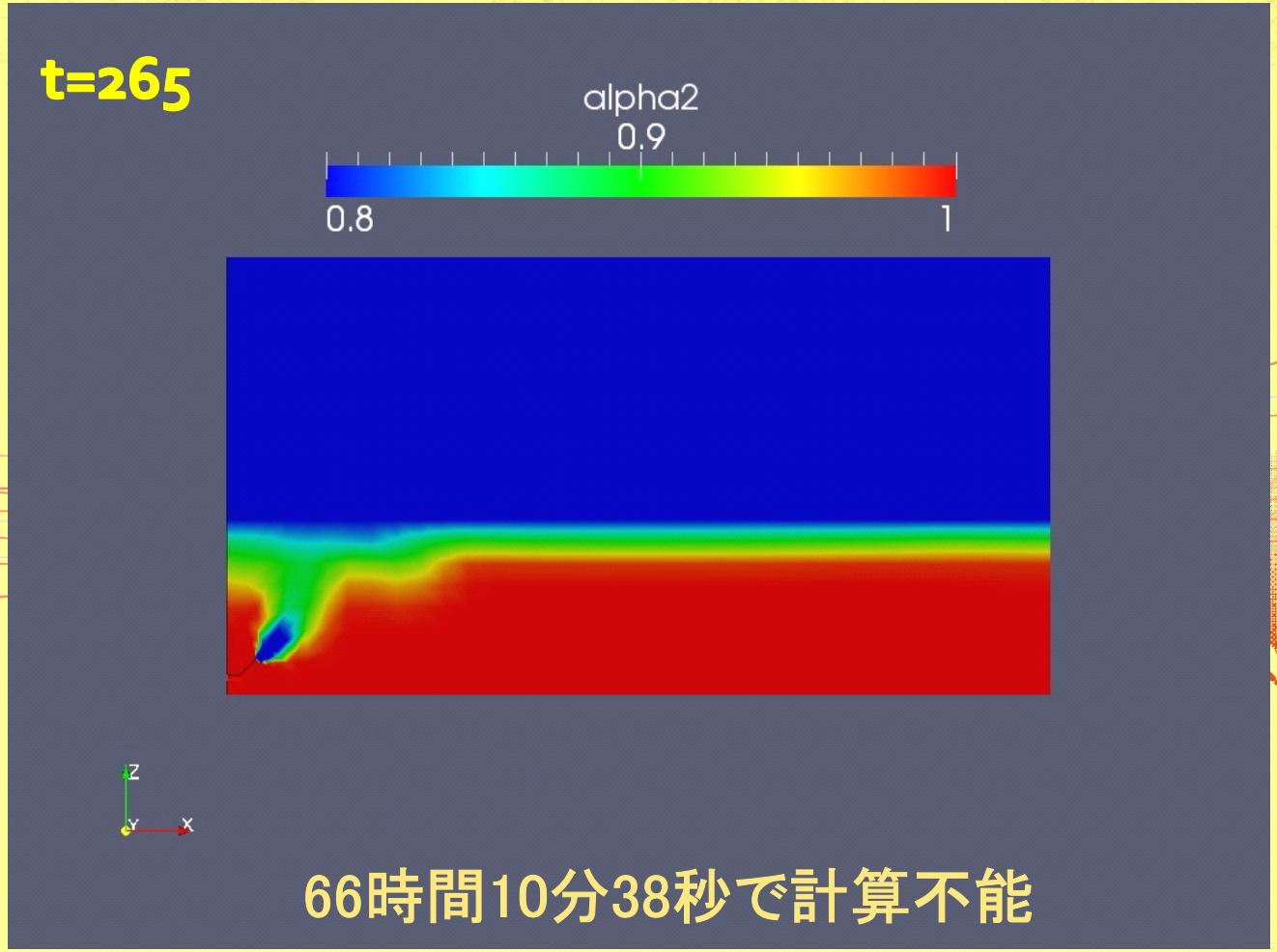

噴流なので
乱流解析が
必要

塩水タンクへの純水の注水をシミュレーション
3次元非定常層流解析
（ノズル径8φ、ノズルでの平均流速0.8m/s）

# InterMixingFoamのメッシュ



# t=265sec




66時間10分38秒で計算不能

# solvers/multiphase/twoLiquidMixingDyMFoamの作成

Make/
  files
  options

UEqn.H
alpahEqn.H
createFields.H
pEqn.H
twoLiquidMixingFoam.C

Make/
  files
  options

UEqn.H
alpahEqn.H
createFields.H
pEqn.H
twoLiquidMixingDyMFoam.C
correctPhi.H

correctPhi.H は
/solvers/incompressible/pimpleFoam/pimpleDyMFoam/からcopy
赤字ファイルは修正要

# twoLiquidMixingFoam.Cの修正（その１）→twoLiquidMixingDyMFoam.C

赤字を追加

```
#include "fvCFD.H"
#include "twoPhaseMixture.H"
#include "turbulenceModel.H"
#include "pimpleControl.H"
#include "dynamicFvMesh.H"
#include "IObasicSourceList.H"

// * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * //

int main(int argc, char *argv[])
{
    #include "setRootCase.H"
    #include "createTime.H"
    #include "createDynamicFvMesh.H"          （ #include "createMesh.H"を削除）
    #include "readGravitationalAcceleration.H"
    #include "initContinuityErrs.H"
    #include "createFields.H"
    #include "readTimeControls.H"
    #include "CourantNo.H"
    #include "setInitialDeltaT.H"

    pimpleControl pimple(mesh)
```

# twoLiquidMixingFoam.Cの修正（その２）→twoLiquidMixingDyMFoam.C

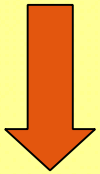
```
while (runTime.run())
 {
    #include "readTimeControls.H"
    #include "CourantNo.H"

    // Make the fluxes absolute ////////////////////
    fvc::makeAbsolute(phi, U);
    #include "setDeltaT.H"

    runTime++;

    Info<< "Time = " << runTime.timeName() << nl << endl;

    mesh.update();

    // Make the fluxes relative to the mesh motion
    fvc::makeRelative(phi, U);

    // --- Pressure-velocity PIMPLE corrector loop
```

赤字を追加

# Make/files

**twoLiquidMixingFoam.C**

**EXE = $(FOAM_APPBIN)/twoLiquidMixingFoam**



**twoLiquidMixingDyMFoam.C**

**EXE = $(FOAM_USER_APPBIN)/twoLiquidMixingDyMFoam**

# Make/options

```
EXE_INC = ¥
    -I$(LIB_SRC)/transportModels ¥
    -I$(LIB_SRC)/transportModels/incompressible/lnInclude ¥
    -I$(LIB_SRC)/transportModels/interfaceProperties/lnInclude ¥
    -I$(LIB_SRC)/turbulenceModels/incompressible/turbulenceModel ¥
    -I$(LIB_SRC)/finiteVolume/lnInclude ¥
    -I$(LIB_SRC)/dynamicFvMesh/lnInclude ¥
    -I$(LIB_SRC)/dynamicMesh/lnInclude ¥
    -I$(LIB_SRC)/meshTools/lnInclude

EXE_LIBS = ¥
    -ltwoPhaseInterfaceProperties ¥
    -lincompressibleTransportModels ¥
    -lincompressibleTurbulenceModel ¥
    -lincompressibleRASModels ¥
    -lincompressibleLESModels ¥
    -lfiniteVolume ¥
    -l dynamicFvMesh ¥
    -ltopoChangerFvMesh ¥
    -ldynamicMesh ¥
    -lmeshTools
```

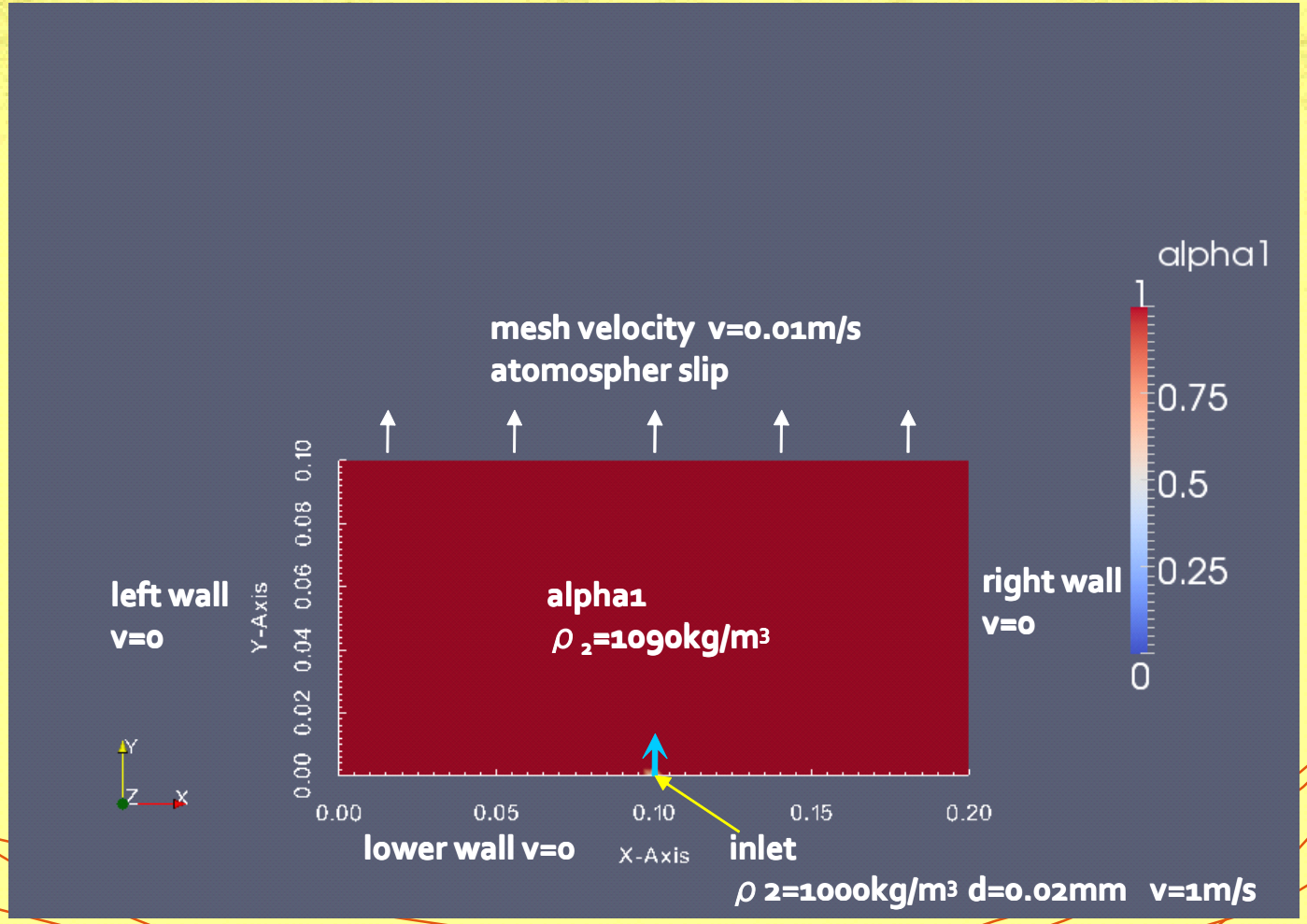
# 残りの作業

①createFields.Hの最後に次を追加

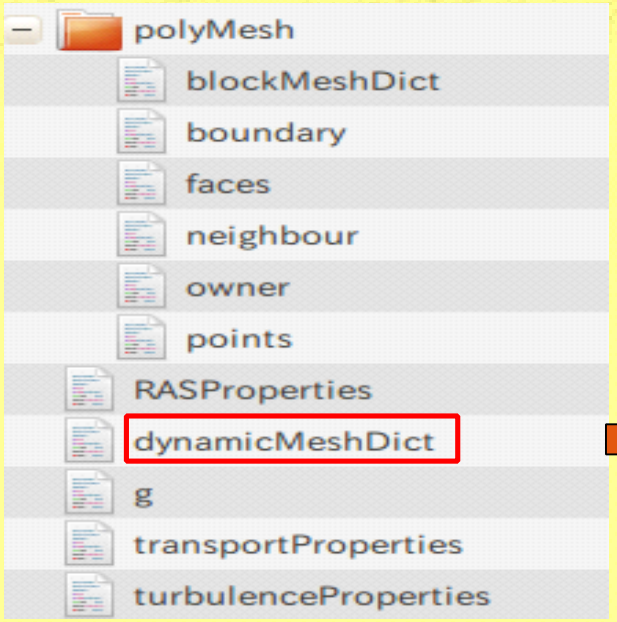
IObasicSourceList sources(mesh);

②twoLiquidMixingDyMFoamでwmake

# 2次元モデル


mesh velocity v=0.01m/s
atomospher slip
left wall
v=0
alpha1
ρ 2=1090kg/m3
right wall
v=0
lower wall v=0
inlet
ρ 2=1000kg/m3 d=0.02mm v=1m/s
Y-Axis
X-Axis
0.00
0.02
0.04
0.06
0.08
0.10
0.00
0.05
0.10
0.15
0.20
alpha1
1
0.75
0.5
0.25
0

# dynamicMeshDict

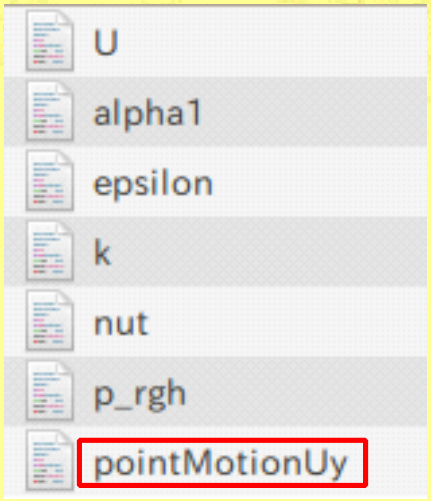



dynamicMeshDict

```
/*--------------------------------*- C++ -*----
| =========                      |
| ¥¥      /  F ield              | OpenFOAM: The Open
|  ¥¥    /   O peration          | Version:  2.1.0
|   ¥¥  /    A nd                | Web:      www.Open
|    ¥¥/     M anipulation       |
¥*---------------------------------------------
FoamFile
{
    version     2.0;
    format      ascii;
    class       dictionary;
    location    "constant";
    object      dynamicMeshDict;
}
// * * * * * * * * * * * * * * * * * * * * * * * * *

dynamicFvMesh   dynamicMotionSolverFvMesh;

motionSolverLibs ( "libfvMotionSolvers.so" );

solver          velocityComponentLaplacian;

diffusivity     directional ( 1 200 0 );


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

0以外の数字ならOK

# pointMotionUy




```
17 dimensions      [0 1 -1 0 0 0 0];
18
19 internalField   uniform 0;
20
21 boundaryField
22 {
23     inlet
24     {
25         type            fixedValue;
26         value           uniform 0;
27     }
28     leftWall
29     {
30         type            slip;
31     }
32     rightWall
33     {
34         type            slip;
35     }
36     lowerWall
37     {
38         type            fixedValue;
39         value           uniform 0;
40     }
41     atmosphere
42     {
43         type            fixedValue;
44         value           uniform 0.01;
45     }
46     defaultFaces
47     {
48         type            empty;
49     }
50 }
```

界面の速度
v=1cm/s

# fvscheme

```
25 {
26     default         Gauss linear;
27 }
28
29 divSchemes
30 {
31     default         none;
32
33     div(rho*phi,U)  Gauss linear;
34     div(phi,alpha1) Gauss vanLeer;
35     div(phi,k)      Gauss limitedLinear 1;
36     div(phi,epsilon) Gauss upwind;
37     div(((rho*nuEff)*dev(grad(U).T()))) Gauss linear;
38 }
39
40 laplacianSchemes
41 {
42     default         Gauss linear corrected;
43     laplacian(diffusivity,cellMotionU) Gauss linear uncorrected;
44 }
45
46 interpolationSchemes
47 {
48     default         linear;
49 }
```

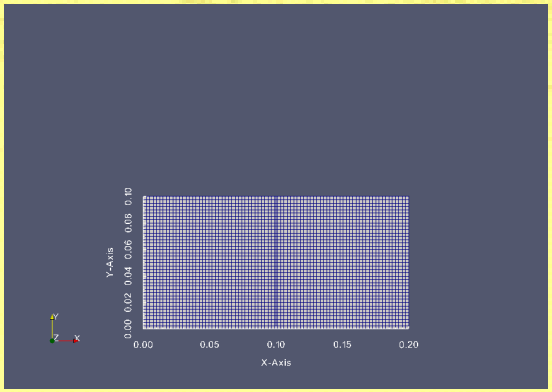
追加

# fvsolution

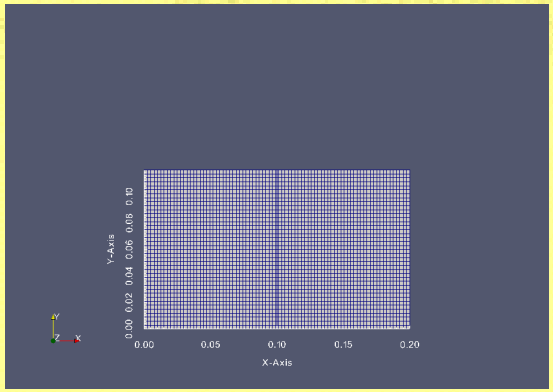
```
    "(U|k|epsilon)Final"
    {
        solver          PBiCG;
        preconditioner  DILU;
        tolerance       1e-08;
        relTol          0;
    }

    cellMotionUy
    {
        solver          PCG;
        preconditioner  DIC;
        tolerance       1e-08;
        relTol          0;
    }
}

PIMPLE
{
    momentumPredictor   yes;
    nOuterCorrectors    1;
    nCorrectors         2;
    nNonOrthogonalCorrectors 0;
    pRefValue           0;
    pRefPoint           (0.1 0.1 0.1);
}
```

追加

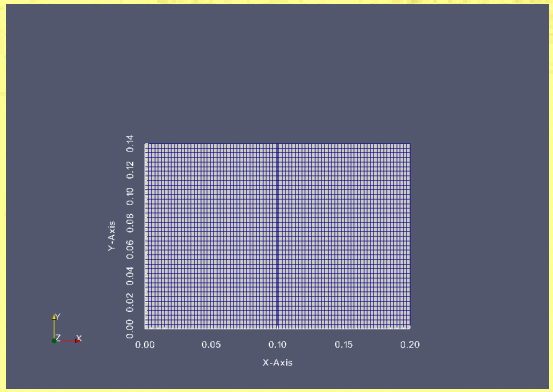
# mesh



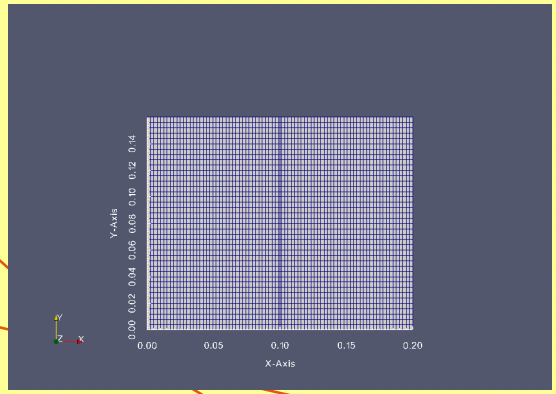

t=0.0



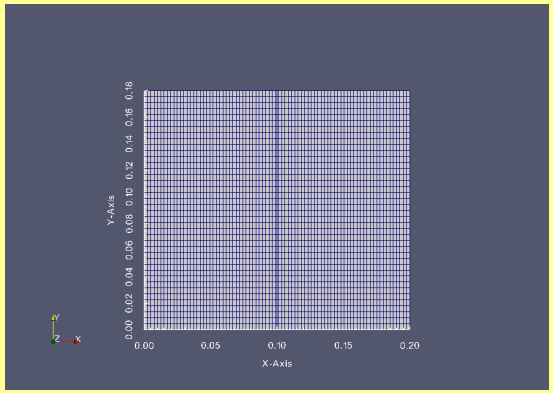

t=2.0



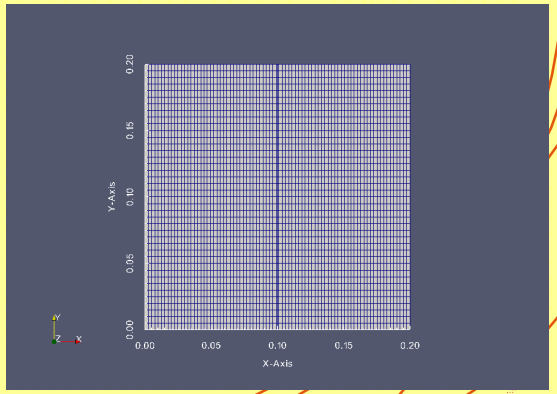

t=4.0




t=6.0

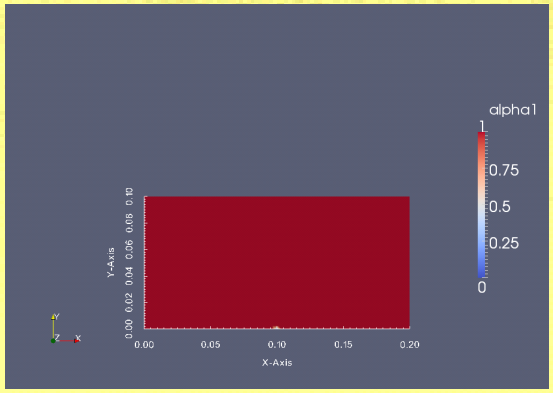



t=8.0

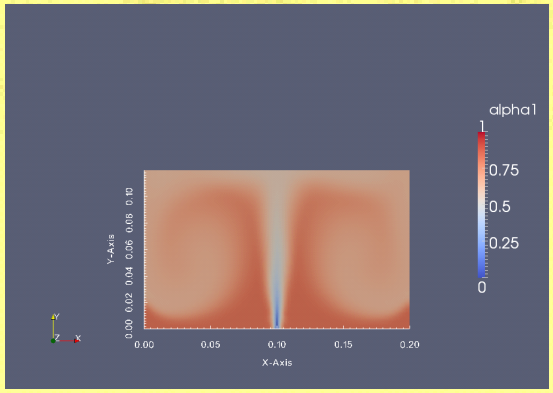



t=10.0

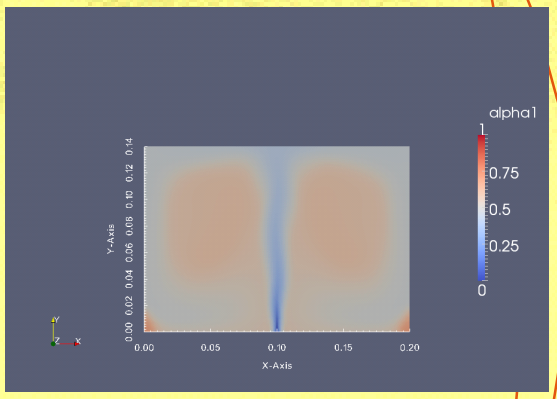
# alpha1



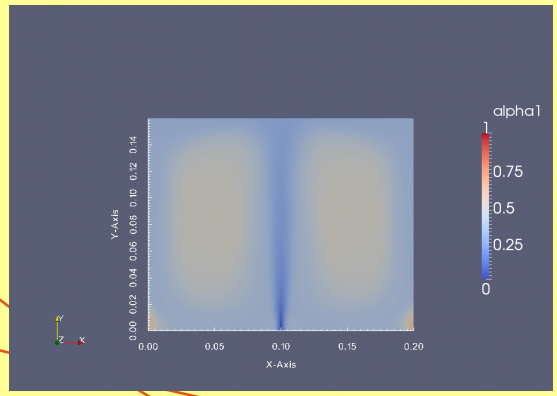

t=0.0



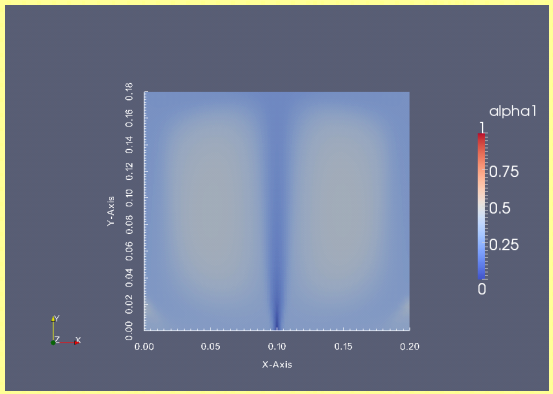

t=2.0



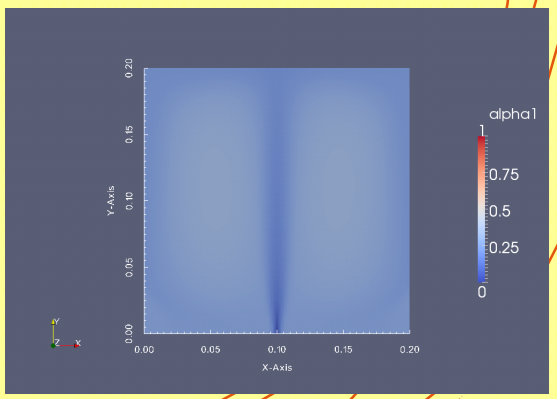

t=4.0




t=6.0

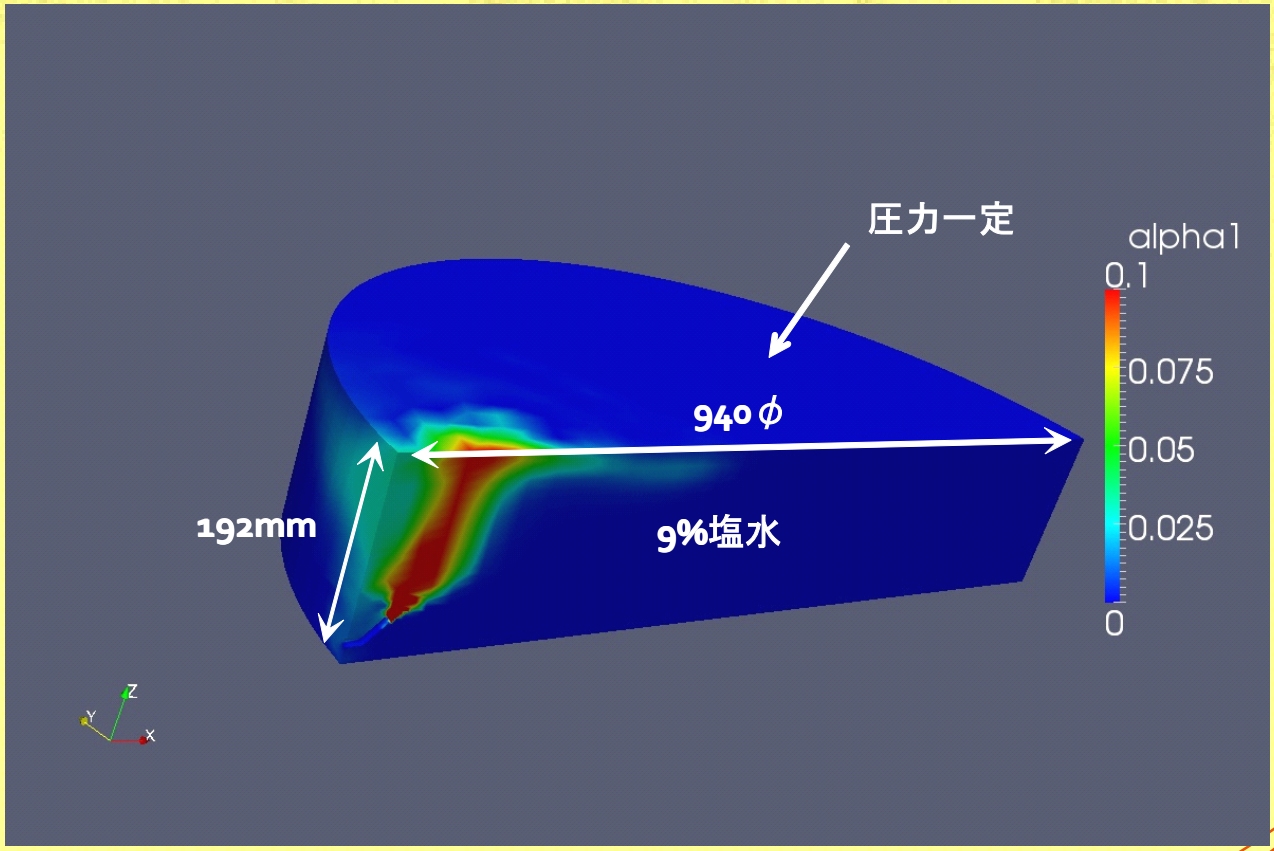



t=8.0



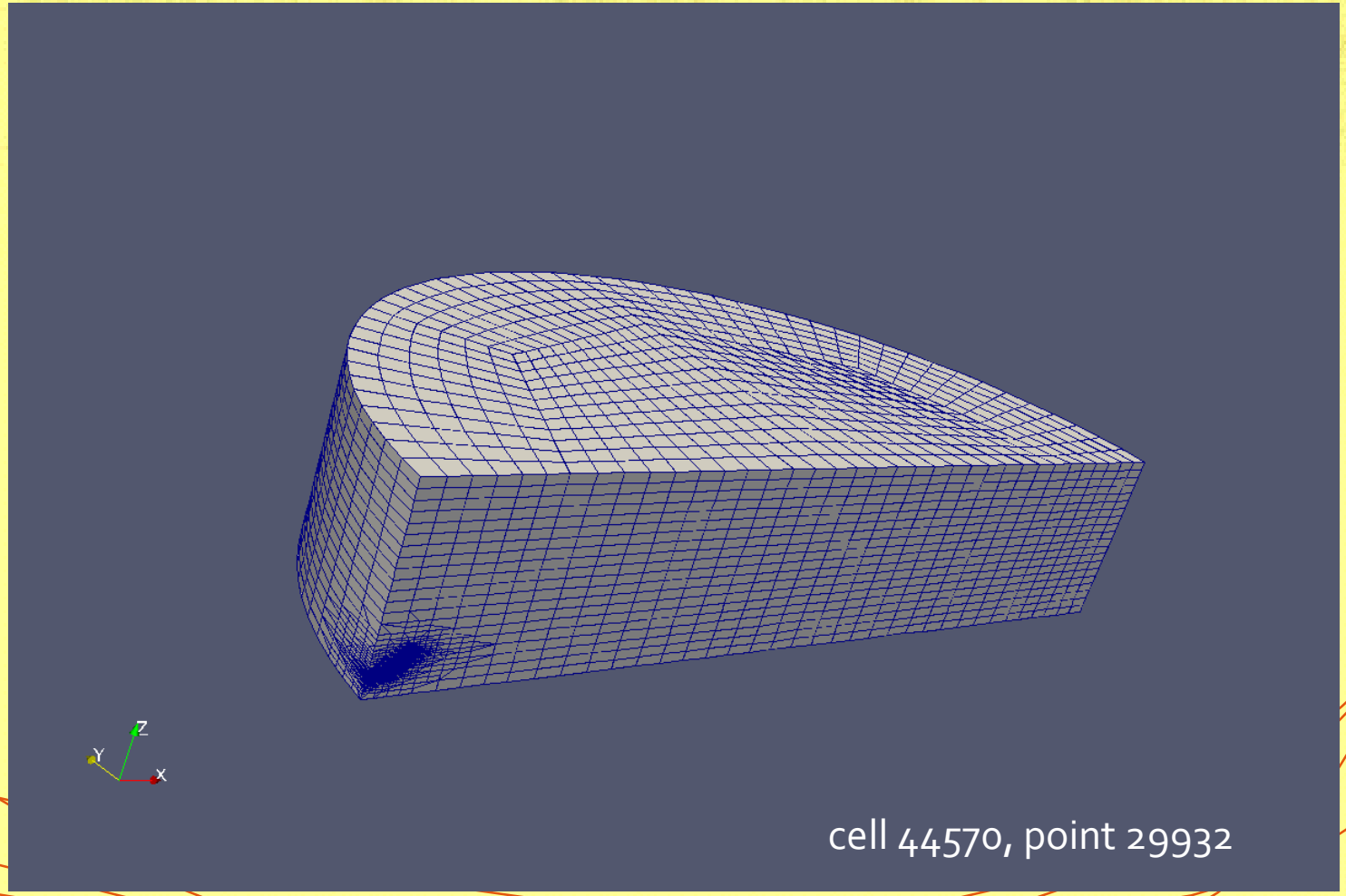

t=10.0

# 3次元解析

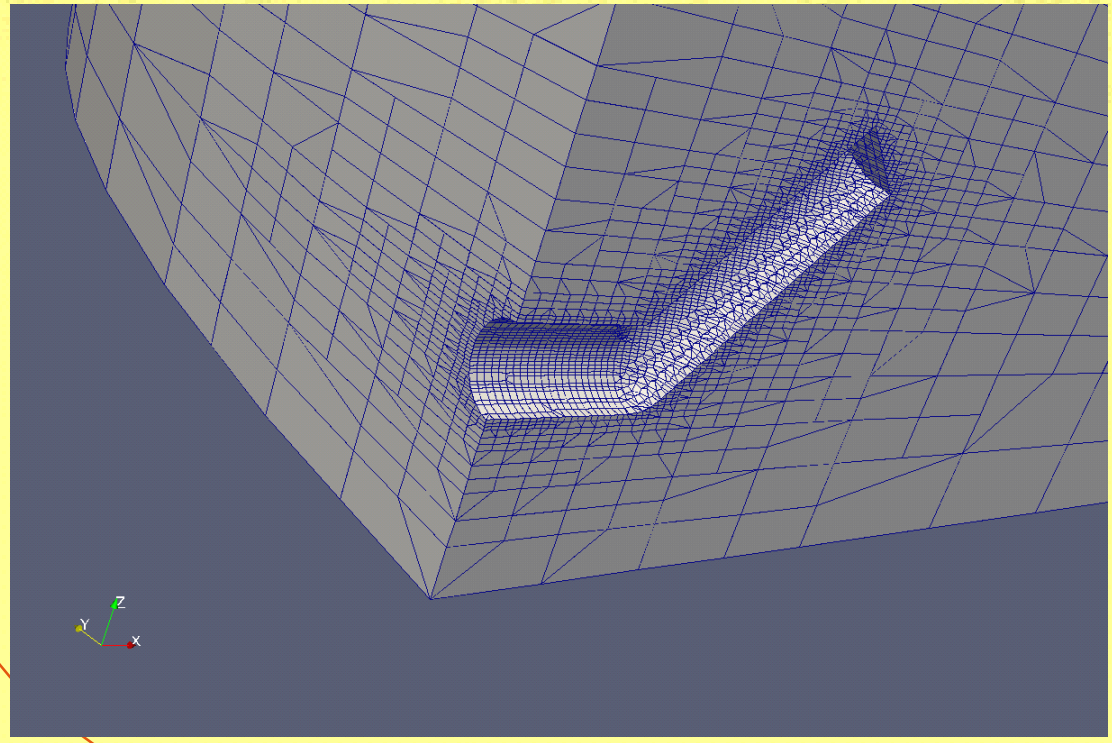



塩水タンクへの純水の注水をシミュレーション
3次元非定常層流解析
（ノズル径8φ、ノズルでの平均流速0.8m/s）

# Mesh



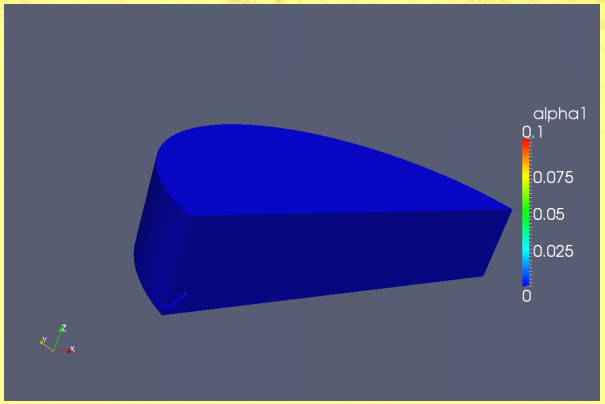
# mesh　nozzle周辺



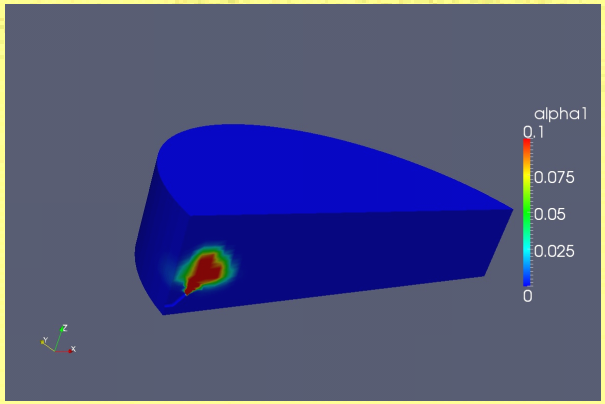
```
snappyHexMesh   features
  (
    {
      file "pipeWall.eMesh";
      level 2
    }
  );
```

level 0だとすぐ発散
ノズル角でk、ε が増大

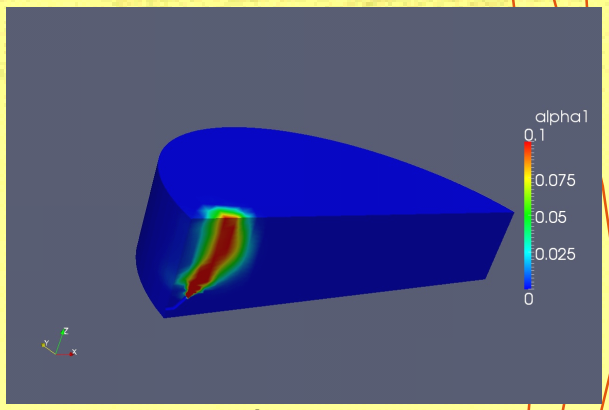
# alpha1



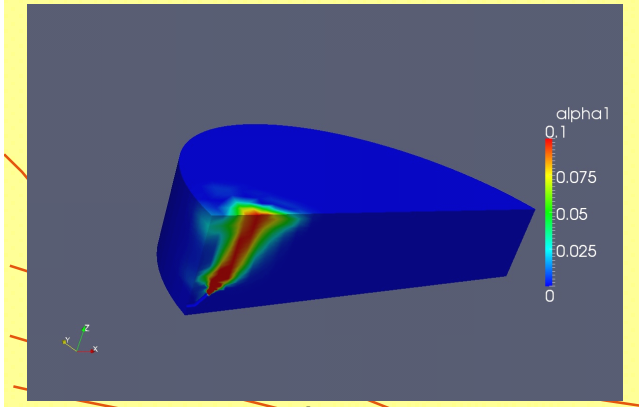

t=0.0



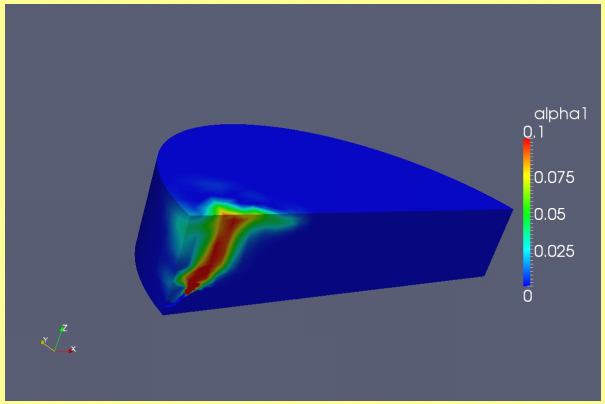

t=1.0



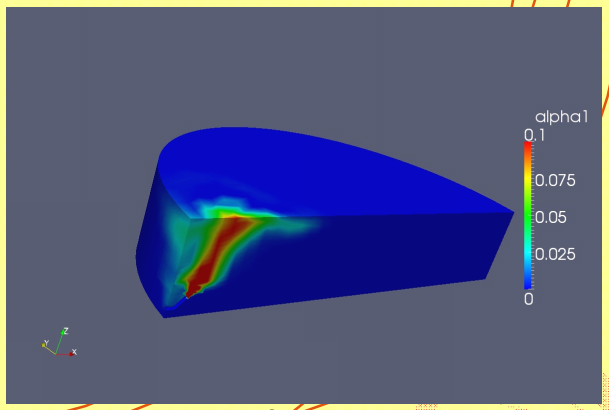

t=2.0




t=3.0

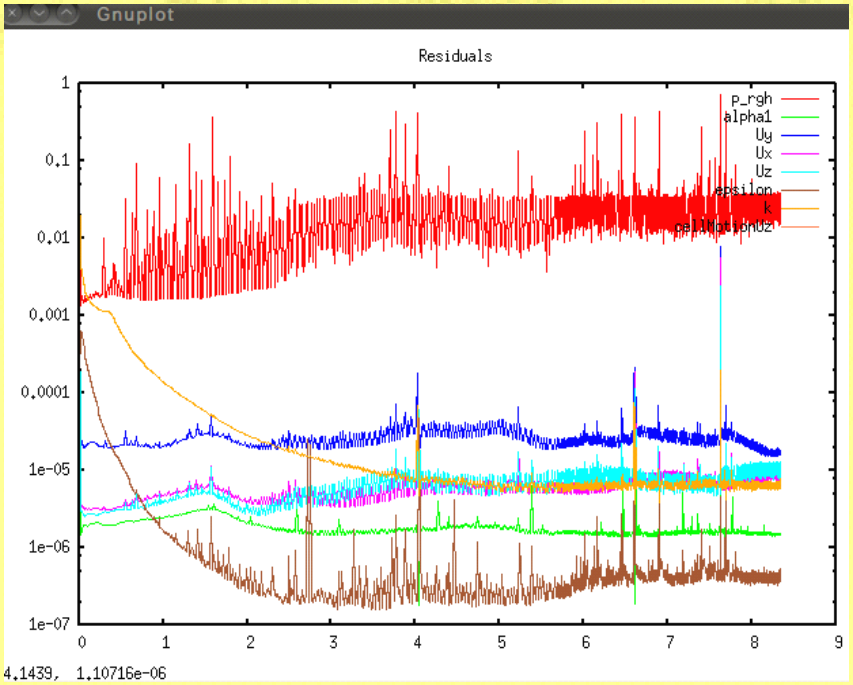


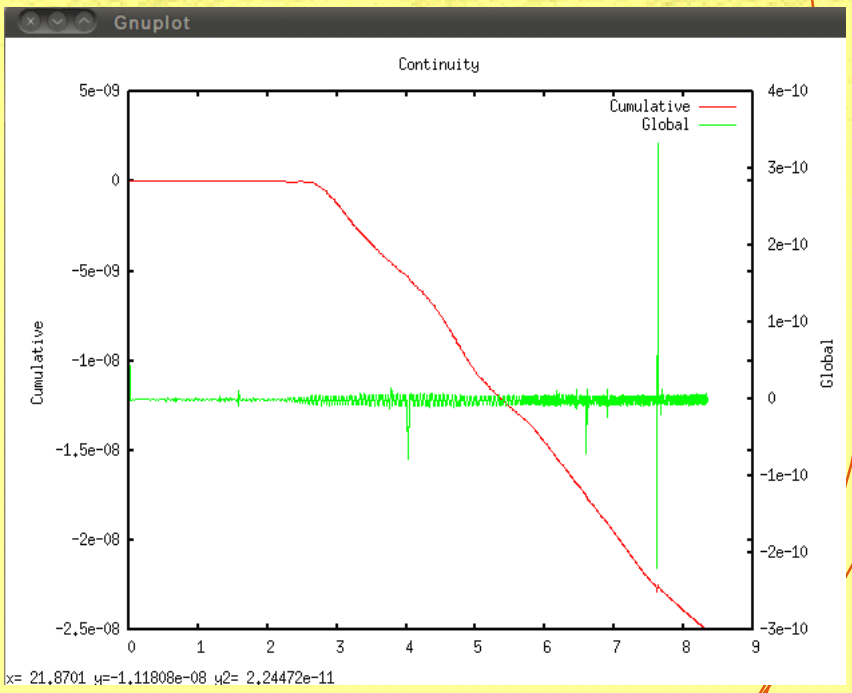

t=4.0




t=5.0

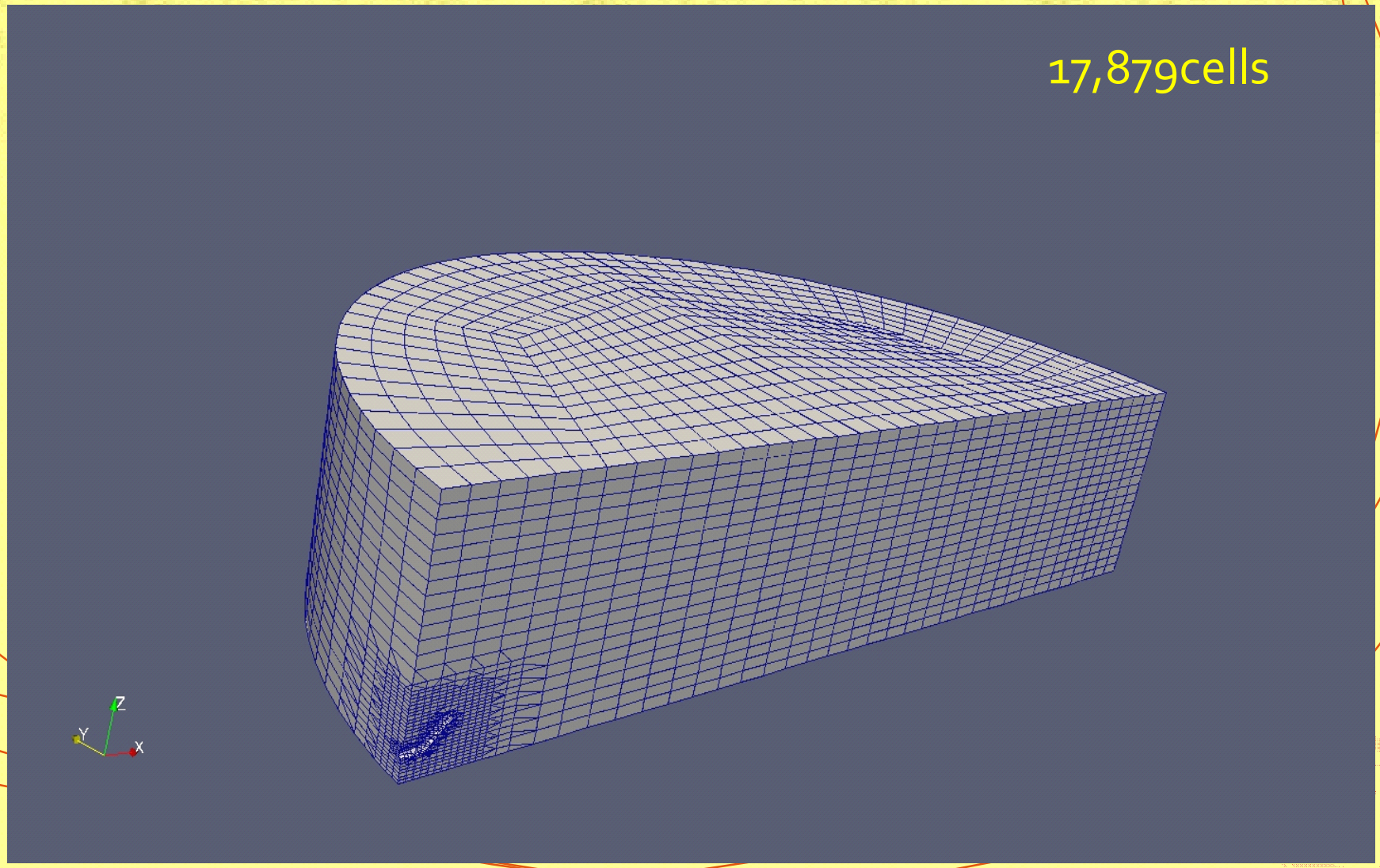
# 残差履歴



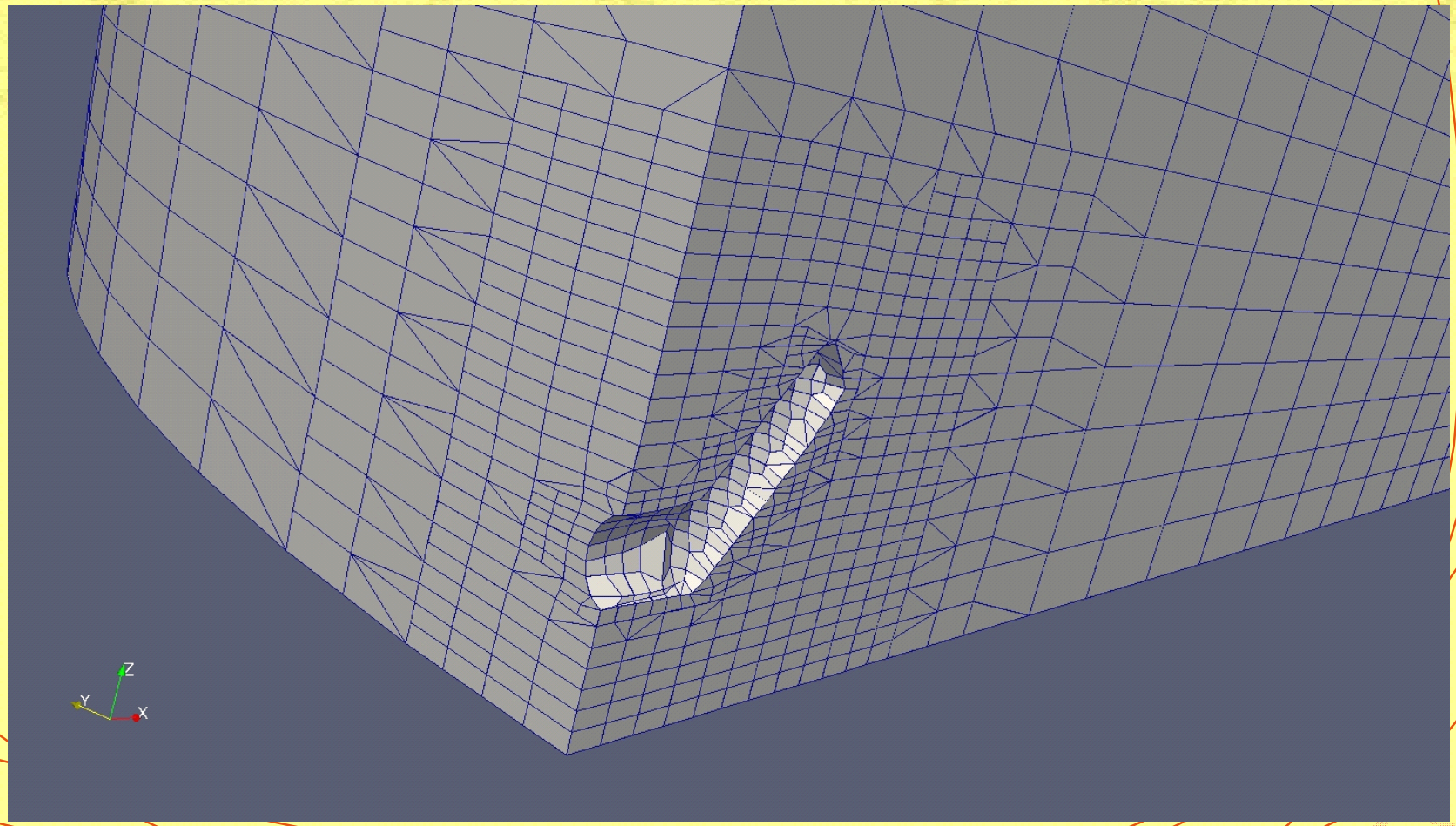




**Max co（クーラン数）1.0ではt=8.3secで発散**

# 粗いメシュ

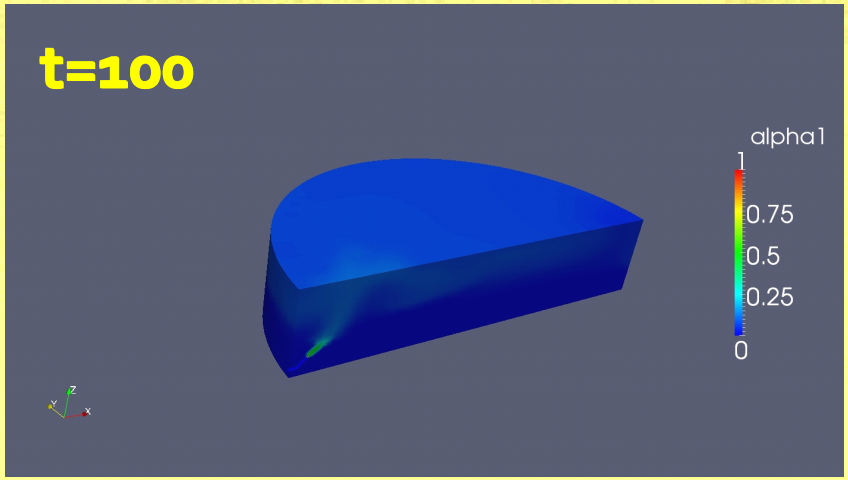


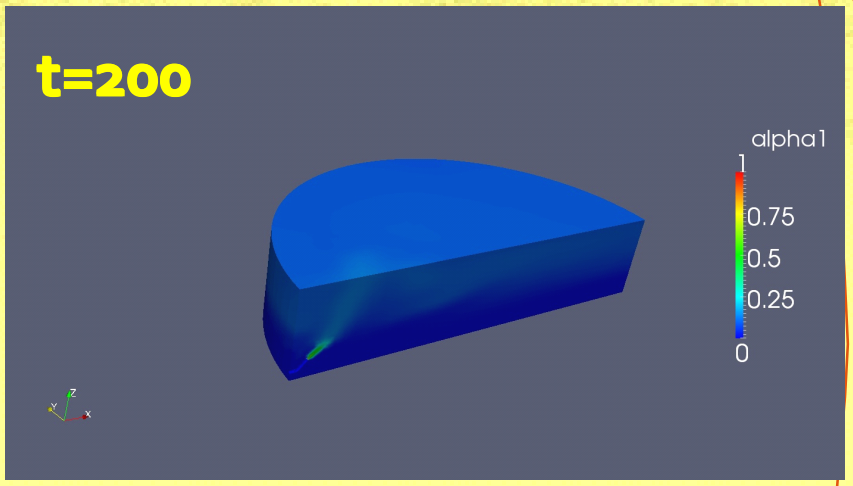
# 粗いメッシュ

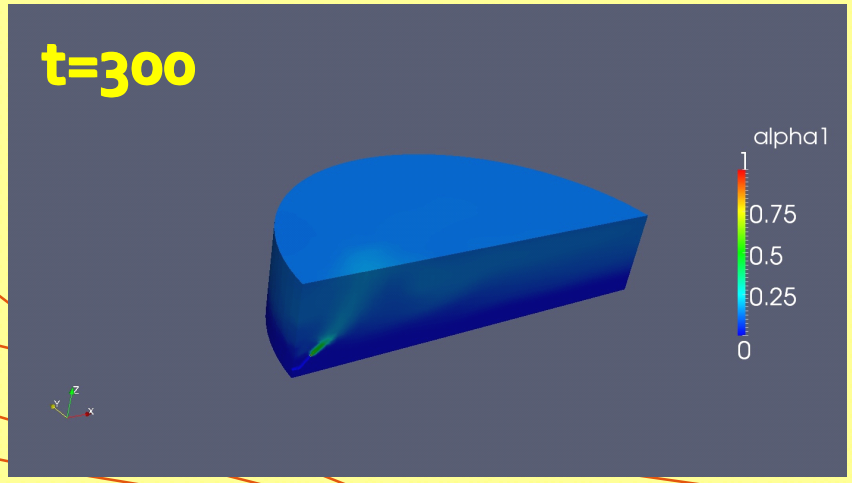


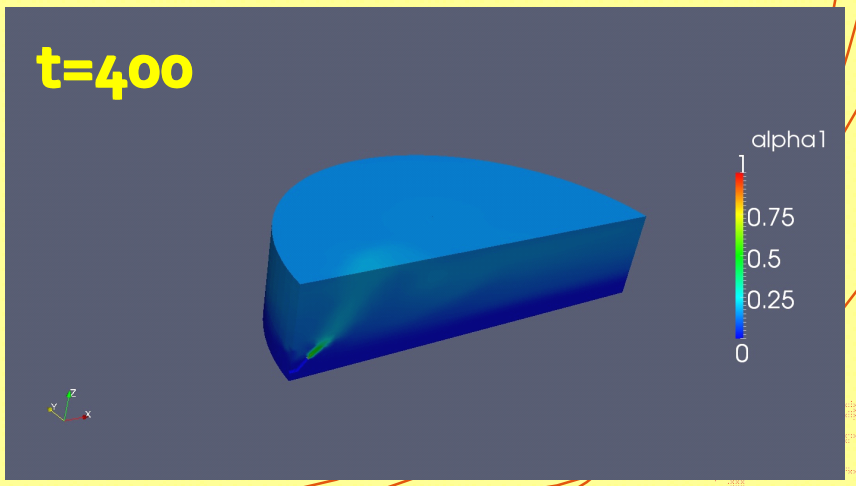

ノズル出口の精度は見た目でも不十分

# alpha1



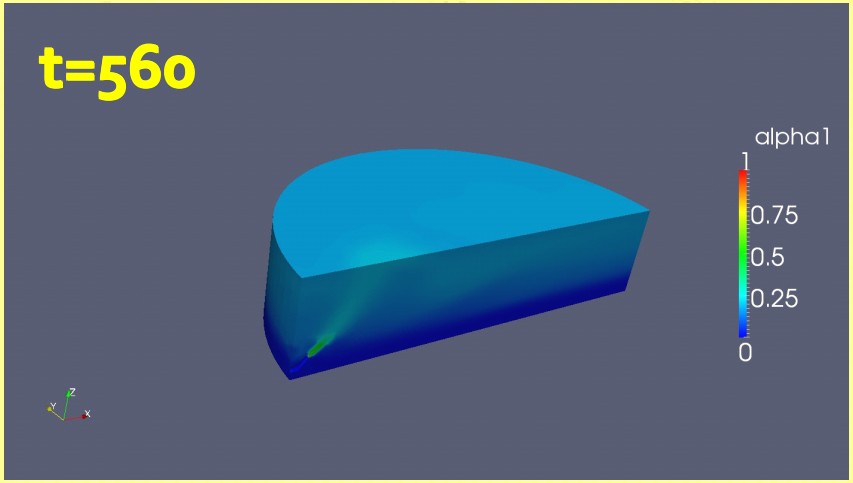



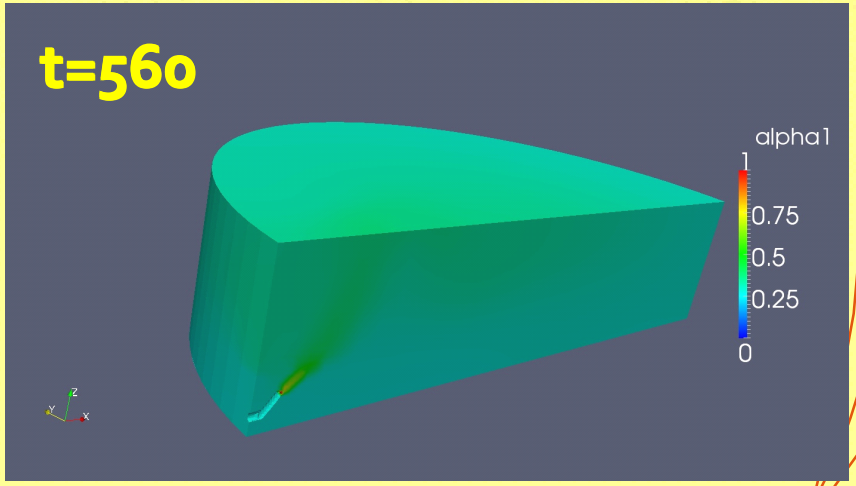




計算はしやすい。t=560secまで計算

# 乱流拡散の影響（αt）




αt=0：乱流拡散無
計算時間=22時間53分




αt=1：乱流拡散無
計算時間=9時間13分

αt→で大で拡散混合が促進。
但し、実験値はこれらの中間→精度が悪い

# おわりに

- **twoLiquidMixingFoam**改造して界面が移動する**twoLiquidMixingDyMFoam**を作成。タンク内の塩水混合解析を検討
- メッシュが数万程度では精度が不足

  →早い計算機が必要なのか検討中

# 参考文献

移動メッシュについて
http://www.geocities.jp/penguinitis2002/study/OpenFOAM/tankentai/06-dynmesh.html