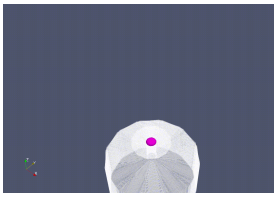
IJ吐出2D計算 (blueCFD-core-2024)



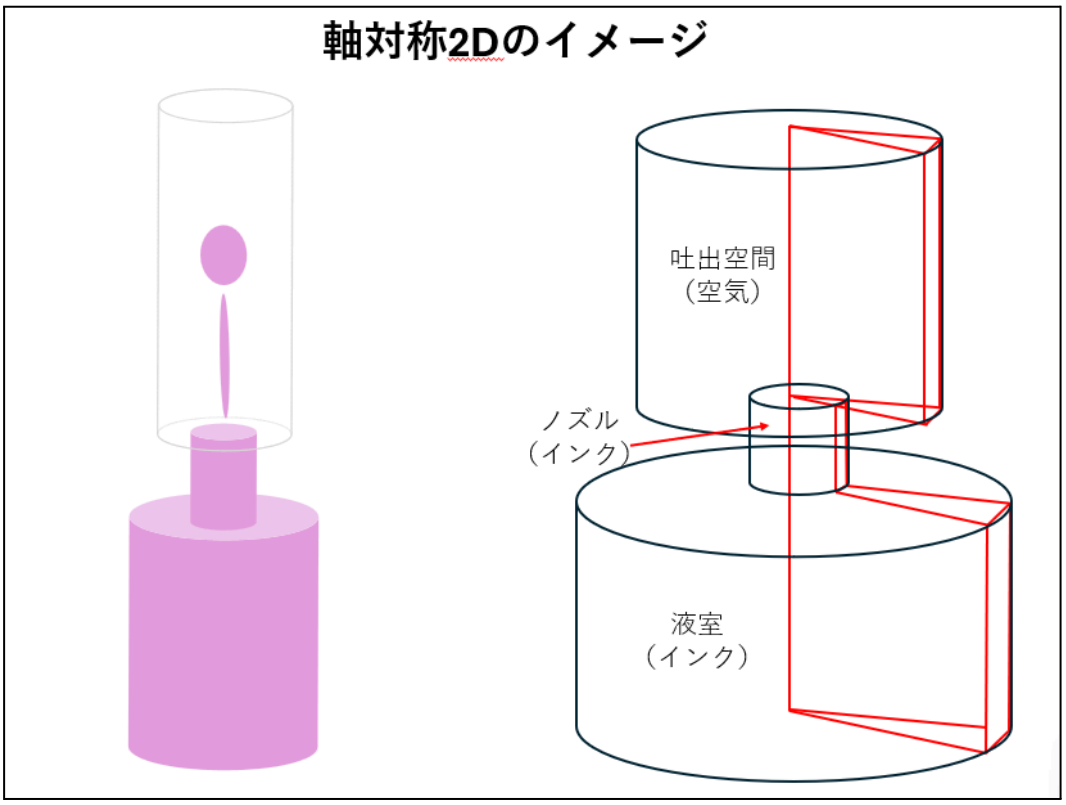
作成 2025/5/27

1　準備

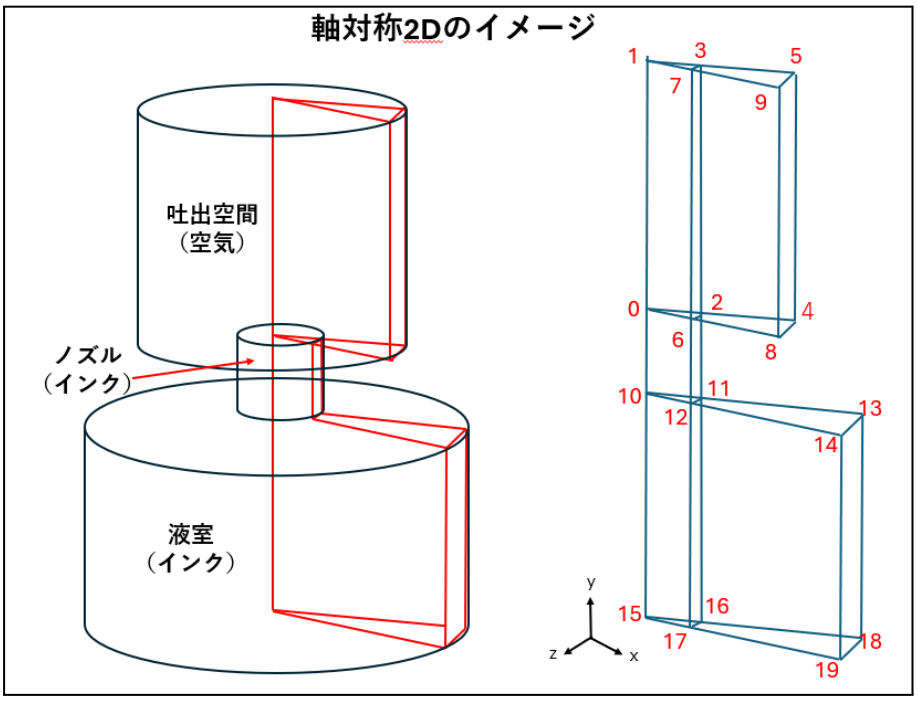
・前のチュートリアルで使ったcapillaryRise2D を　コピーしてIJeject と名前を変える.
・ターミナルを開きカレントディレクトリを IJeject にする.
・前の計算結果が残っていたら，./Allcleant ⏎（ピリオド スラッシュ Allclean) を実行して初期化しておく.
注：blueCFD-core-2024 では ./ はなくても動作するが，unix では動作しないので，ここは ./ をつけてスクリプトを呼ぶクセをつけておいたほうがよい.

2　計算対象の把握

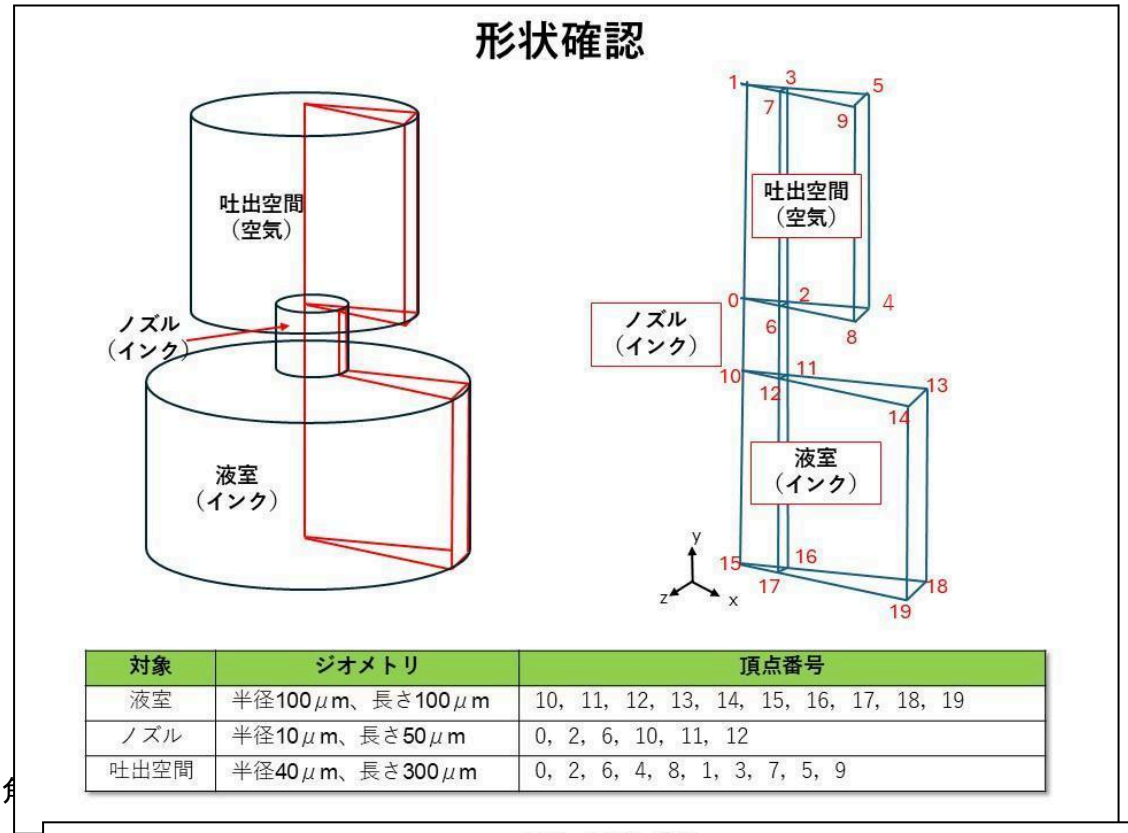
・下左図のようなインクジェットを考える．ヘッド液室形状は円柱でインク（マゼンタ，ピンク色）が重点されている．液室先端にノズルがあり，液室内の圧力波が作用することで吐出空間（空気中）に液滴を吐出する.



・吐出空間，ノズル，液室を軸対称2次元（2D)で計算する．右図のように楔形の領域を計算対象とする．計算領域は5つのブロックからなり，各点に番号をつける．

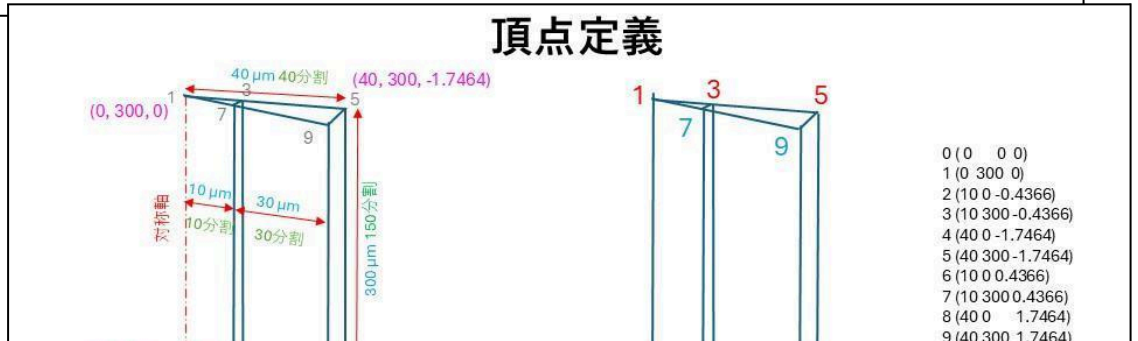


・実際のディメンジョンを想定する．

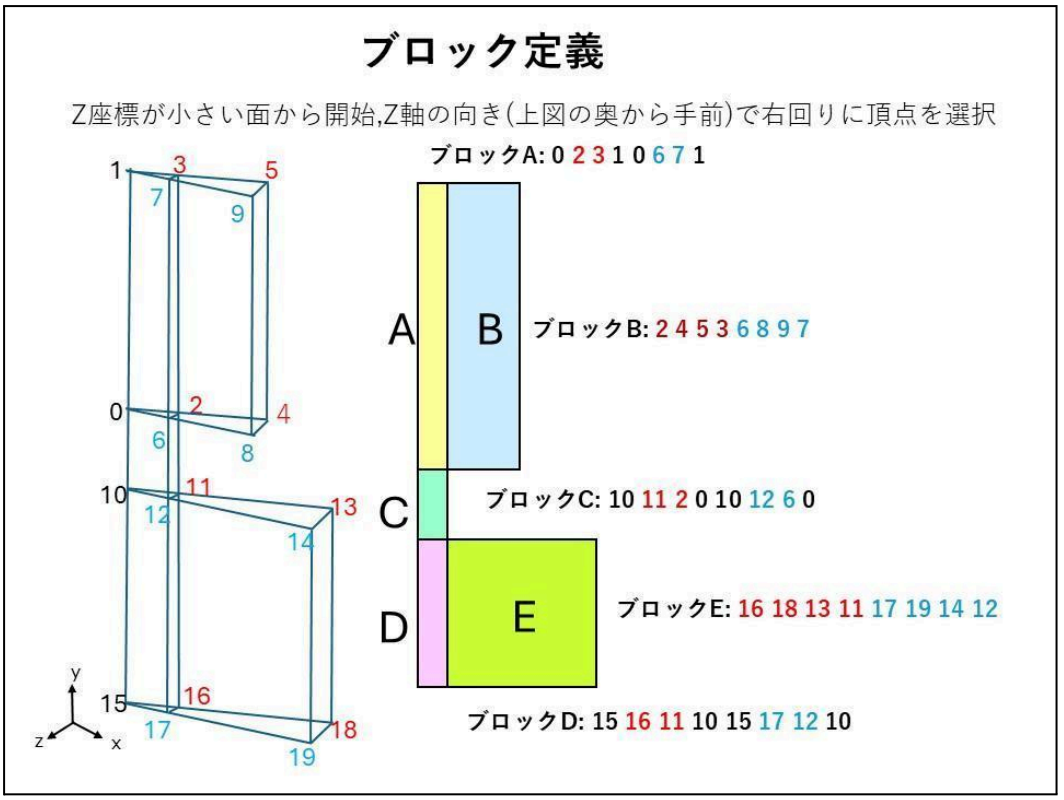


対象	ジオメトリ	頂点番号
液室	半径100μm、長さ100μm	10，11，12，13，14，15，16，17，18，19
ノズル	半径10μm、長さ50μm	0，2，6，10，11，12
吐出空間	半径40μm、長さ300μm	0，2，6，4，8，1，3，7，5，9

・角



・5つのブロックを構成する点をピックアップする.



・境界（boundary)を定義する.

面定義1

boundary( )で面を定義する

面名称ごとに、境界typeと面(構成頂点)を定義する

今回の面名称は6個

面名称

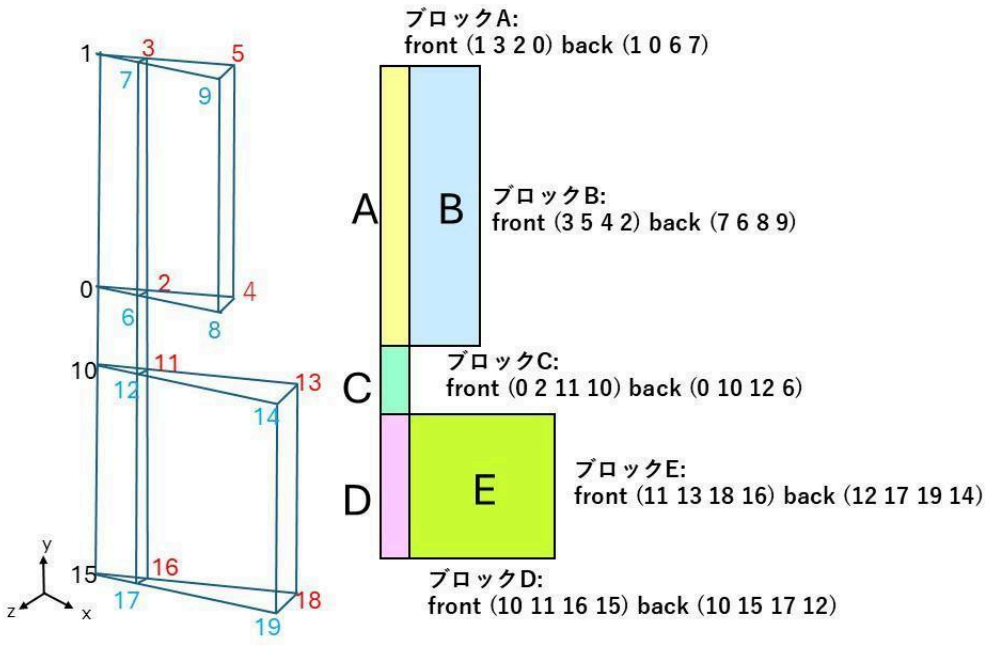
- axis（対称軸）
- inlet（流体の流入面）
- atmosphere（大気開放面および流体の流出面）
- walls（壁面）
- front、back（軸対称の回転角方向面、2次元化の非考慮面）

境界type	意味	面名称
patch	流入・流出境界	inlet, atmosphere
empty	2次元形状前後の面，対称軸	axis
wedge	円筒座標系の軸対称面（楔形状）	front, back
wall	壁面境界	walls

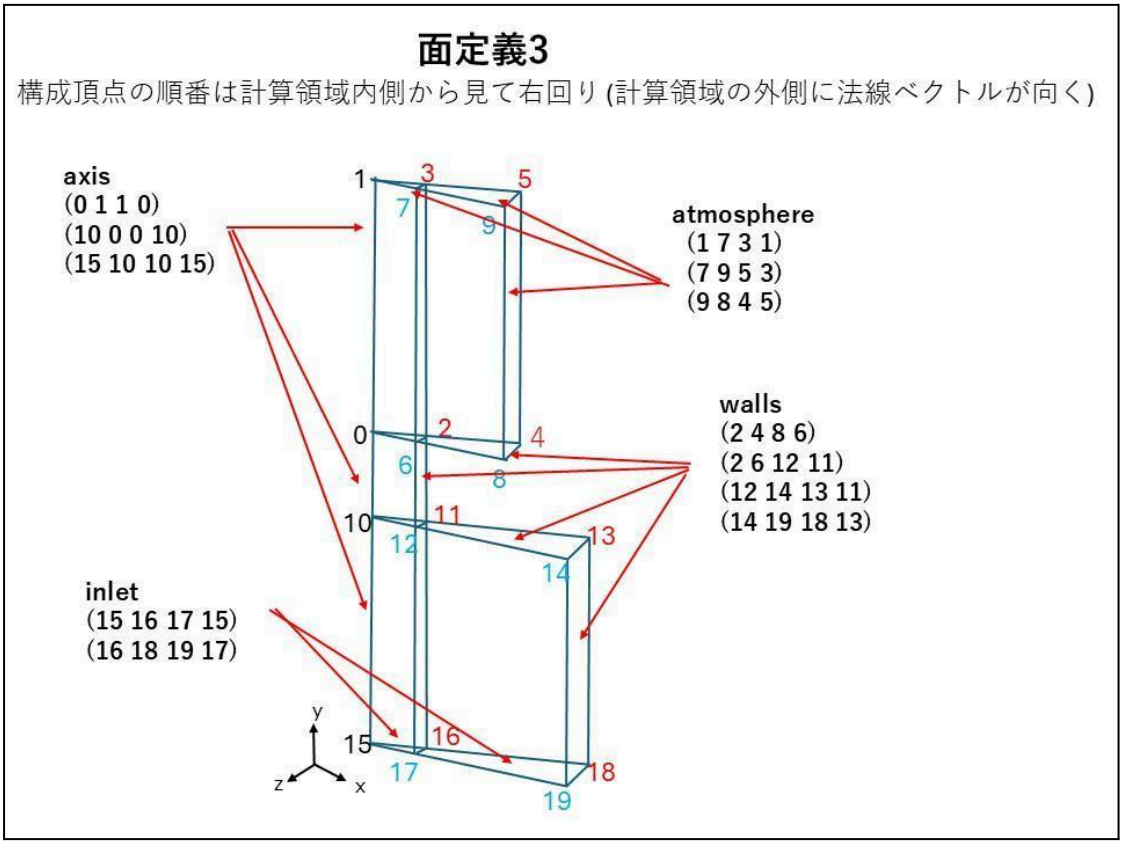
・front と backを構成する面を定義する．front はz座標が負，backは正とする．

面定義2

構成頂点の順番は計算領域内側から見て右回り (計算領域の外側に法線ベクトルが向く)



・axis, atmosphere, walls, inlet を定義する.



3　blockMeshDictの修正

・system\blockMeshDict を開く
・数値の単位をμm (convertToMeters 1e-6;) にする
・頂点（vertices）の座標を設定.
・5つのブロックの頂点6つ，分割数を設定.
・境界（boundary）を定義.
・最終的に，blockMeshDictは以下のようになる.
　自分で考えて入力するのが面倒だったらコピー＆ペーストしましょう.

```
convertToMeters 1e-6;
vertices
(
  ( 0   0  0)        //0
  ( 0  300  0)       //1
  (10  0 -0.4366)    //2
  (10 300 -0.4366)   //3
  (40 0 -1.7464)     //4
  (40 300 -1.7464)
  (10 0 0.4366)
  (10 300 0.4366)
  (40 0 1.7464)
  (40 300 1.7464)
  (0   -50 0)        //10
  (10 -50 -0.4366)
  (10 -50   0.4366)
  (100 -50 -4.366)
  (100 -50  4.366)
  (  0  -150 0)      //15
  (  10  -150 -0.4366)
  (  10  -150  0.4366)
  ( 100  -150 -4.366)
  ( 100  -150  4.366)  //19
);

blocks
(
    hex (0 2 3 1 0 6 7 1) (10 150 1) simpleGrading (1 1 1)
    hex (2 4 5 3 6 8 9 7) (30 150 1) simpleGrading (1 1 1)
    hex (10 11 2 0 10 12 6 0) (10 25 1) simpleGrading (1 1 1)
    hex (15 16 11 10 15 17 12 10) (10 50 1) simpleGrading (1 1 1)
    hex (16 18 13 11 17 19 14 12) (90 50 1) simpleGrading (1 1 1)
);

boundary
(
    inlet
    {
        type patch;
        faces
        (
          (15 16 17 15)
          (16 18 19 17)
```

```
    );
}
atmosphere
{
    type patch;
    faces
    (
      (1 7 3 1)
      (7 9 5 3)
      (9 8 4 5)
    );
}
walls
{
    type wall;
    faces
    (
      (2 4 8 6)
      (2 6 12 11)
      (12 14 13 11)
      (14 19 18 13)
    );
}
front
{
    type wedge;
    faces
    (
      (1 3 2 0)
      (3 5 4 2)
      (0 2 11 10)
      (10 11 16 15)
      (11 13 18 16)
    );
}
back
{
    type wedge;
    faces
    (
      (1 0 6 7)
      (7 6 8 9)
      (0 10 12 6)
      (10 15 17 12)
      (12 17 19 14)
  );
}
axis
{
    type empty;
    faces
    (
     (0 1 1 0)
     (10 0 0 10)
```

```
        (15 10 10 15)
      );
    }
);
```

４メッシュ作成

・blockMeshを実行すると，膨大なワーニングがでる．数分かかる？我慢する・・・のはつらい・・・controlDict で以下にすると以降このワーニングは消える．消えなければ何かがおかしい．

```
writePrecision  10;
```

(ワーニング抜粋）

```
At local face at (9.95556e-05 -6.1e-05 4.34417e-06) the normal (-0.0490584 2.11233e-16
0.998796) differs from the average normal (0.00194868 -6.72105e-05 0.998315) by
0.00260195
Either correct the patch or split it into planar parts
--> FOAM Warning :
    From function virtual void
Foam::wedgePolyPatch::calcGeometry(Foam::PstreamBuffers&)
    in file meshes/polyMesh/polyPatches/constraint/wedge/wedgePolyPatch.C at line 70
    Wedge patch 'back' is not planar.
At local face at (9.95556e-05 -5.9e-05 4.34417e-06) the normal (-0.0490584 3.31973e-16
0.998796) differs from the average normal (0.00194868 -6.72105e-05 0.998315) by
0.00260195
Either correct the patch or split it into planar parts
--> FOAM Warning :
    From function virtual void
Foam::wedgePolyPatch::calcGeometry(Foam::PstreamBuffers&)
    in file meshes/polyMesh/polyPatches/constraint/wedge/wedgePolyPatch.C at line 70
    Wedge patch 'back' is not planar.
・・・・・
```

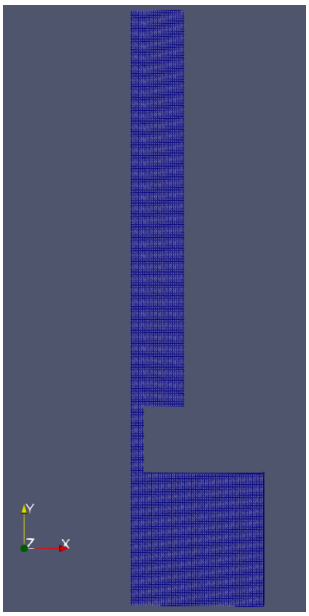
・以下で終わればOK

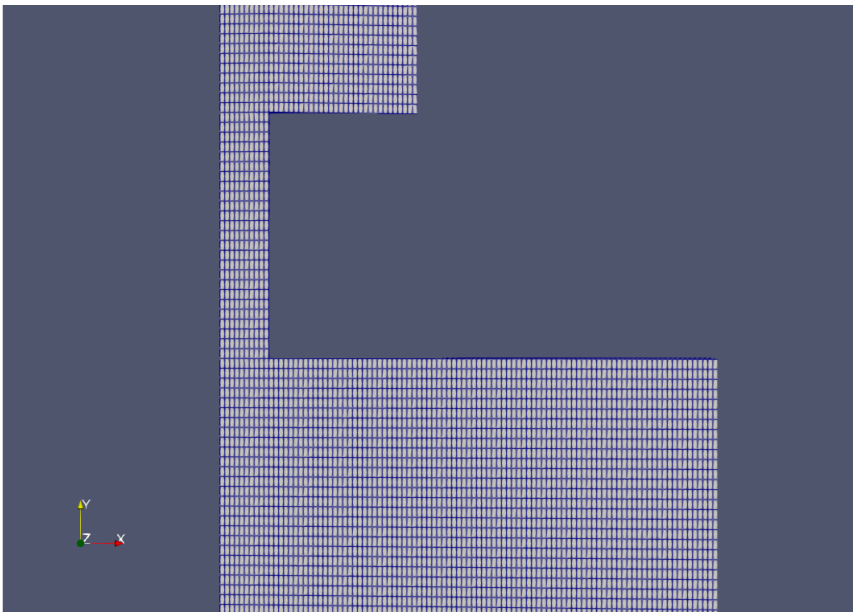
```
Writing polyMesh
----------------
Mesh Information
----------------
  boundingBox: (0 -0.00015 -4.366e-06) (0.0001 0.0003 4.366e-06)
  nPoints: 22835
  nCells: 11250
  nFaces: 44980
  nInternalFaces: 22145
----------------
Patches
----------------
  patch 0 (start: 22145 size: 100) name: inlet
  patch 1 (start: 22245 size: 40) name: atmosphere
  patch 2 (start: 22285 size: 195) name: walls
  patch 3 (start: 22480 size: 11250) name: front
```

patch 4 (start: 33730 size: 11250) name: back
patch 5 (start: 44980 size: 0) name: axis

End

・paraFoam でメッシュ確認

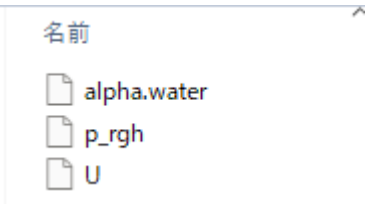




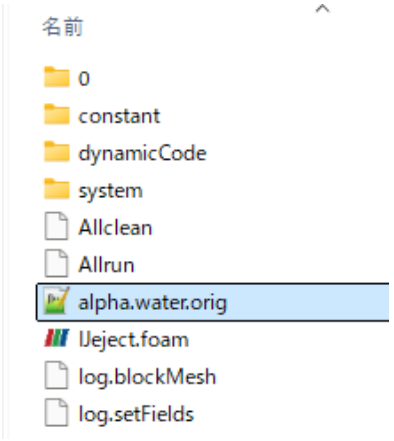
５　境界条件設定

・alpha.water.orig が 0フォルダの外にでていることを確認 (確認ポイントは2つ)

確認１：0フォルダには aloha.water.orig がないことを確認



確認２：IJejectの直下に aloha.water.origあること．こうなっていないとparaviewで0フォルダのVOF値がうまく表示されない．



・alpha.water.orig を開き，front, back, axisの境界があることを確認する．

注意：開くのはalpha.waterではない．

capillaryRise2Dのチュートリアルからもってきていればあるはず．
なければ以下をコピー＆ペーストする．

```
front
{
    type            wedge;
}
back
{
    type            wedge;
}
axis
{
    type            empty;
}
```

・接触角を45度にする. もうなっているかも？

```
  walls
   {
      type          contactAngle;
      theta0        45;
      limit         gradient;
      value         uniform 0;
   }
```

・0\p_rghファイルを開き, front, back, axisの境界があることを確認する.
capillaryRise2Dのチュートリアルからもってきていればあるはず.
なければalpha.water.orig からコピー＆ペーストする.

```
  front
   {
      type          wedge;
   }
   back
   {
      type          wedge;
   }
         axis
   {
      type          empty;
   }
```

・inlet 境界を対称境界(zeroGradient)にする.

```
    inlet
    {
//        type          fixedValue;
//        value         uniform 0;
        type          zeroGradient;
    }
```

・0\Uのファイルを開き, front, back, axisの境界があることを確認する.
capillaryRise2Dのチュートリアルからもってきていればあるはず.
なければalpha.water.orig からコピー＆ペーストする.

```
  front
   {
      type          wedge;
   }
   back
   {
      type          wedge;
   }
         axis
   {
      type          empty;
```

```
    }
```

・inlet 境界を以下に修正

```
    inlet
    {
//        type            pressureInletOutletVelocity;
      type fixedValue;
       value       uniform (0 0.1 0);
    }
```

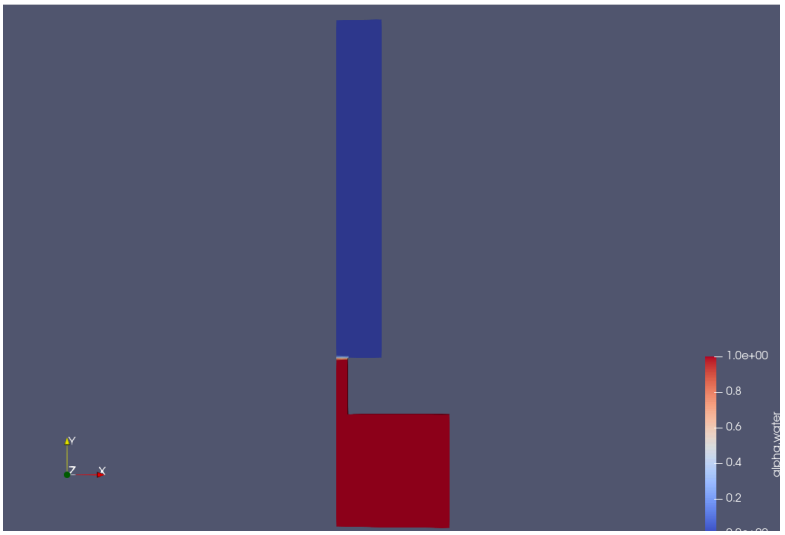
7－4　流体設定

・system\setFieldsDict　を開き以下に修正

```
regions
(
    boxToCell
    {
        box (0 -1 -1) (1 0 1);
        fieldValues
        (
            volScalarFieldValue alpha.water 1
        );
    }
);
```

6　初期化

・　./Allclean を実行
・paraFoamでメッシュ，VOF値を確認
　Skip Zero Time のチェックを外さないとVOF値は見えない.



7　物性値設定

・constant\phaseProperties　を開き以下に修正する. 単位は[N/m]=[kg m/s2 /m]＝[kg/s2].

```
phases          (water air);
sigma           0.05;
```

・physicalProperties.waterを開き以下に修正する. 値は動粘度 [m^2/s] なので注意.
　インク粘性係数は5 mPa･sだが密度1000で割って動粘度 5e-6 となる.

```
viscosityModel  constant;
nu              5e-06;
rho             1000;
```

8　計算条件設定

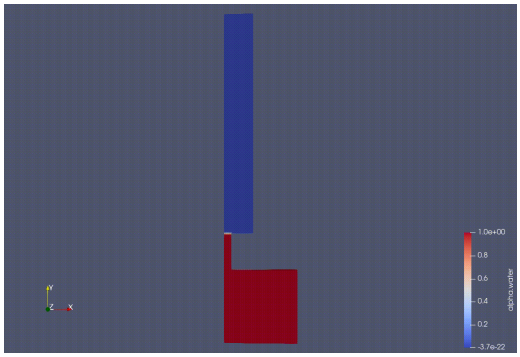
・controlDict を開き，以下に修正

```
application     foamRun;
solver          incompressibleVoF;
//startFrom       startTime;
startFrom       latestTime;
startTime       0;
stopAt          endTime;
endTime         50e-6;
deltaT          1e-6;
writeControl    adjustableRunTime;
writeInterval   2e-6;
purgeWrite      0;
writeFormat     ascii;
```

9　計算実行

・foamRunを実行する

計算が終わったらparaFoamで確認.
以下のようにノズルから液柱が吐出していればOK
一定速度のインクが流入しているので，液滴にならず液柱として吐出している.
液滴が形成できるように境界条件を変更しよう.

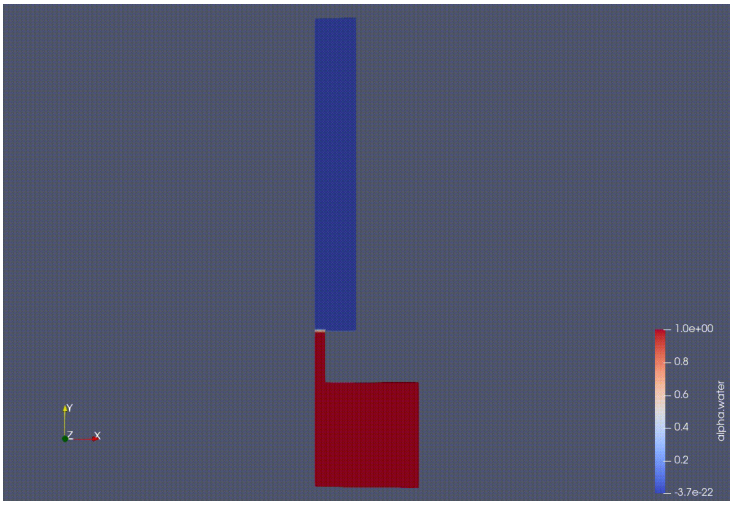


10　切断波形を設定

・0\Uを開きinletを以下のように修正

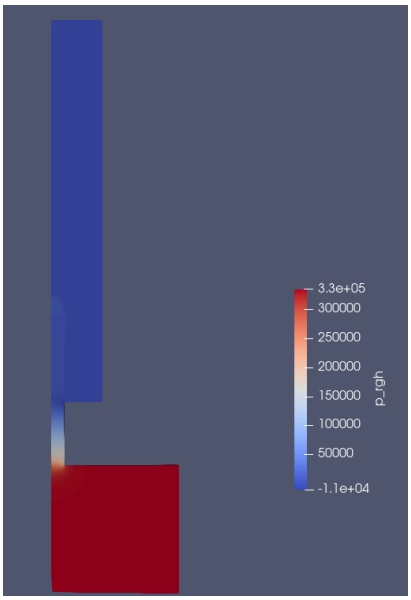
```
  inlet
   {
//       type            pressureInletOutletVelocity;
//     type fixedValue;
//          value       uniform (0 0.1 0);
       type            uniformFixedValue;
       uniformValue    table
       (
          (0 (0 0.1 0))
          (10e-6 (0 0.1 0))
          (11e-6 (0 -0.1 0))
          (20e-6 (0 -0.025 0))
          (30e-6 (0 0 0))
          (50e-6 (0 0 0))
       );
       value           uniform (0 0 0);
   }
```

・./Allclean ⏎, foamRun ⏎　を実行
・paraFoam で結果確認
今度は液滴が切断される.



11　圧力境界条件で液滴切断

・どのくらいの圧力で吐出しているかを確認する.
　圧力(p_rgh)色分け表示で，液室内の圧力を見積もると，3e5 [Pa]　以上の圧力がかかっていることがわかる.



・Uの境界条件変更
pressureInletOutletVelocity　に戻す

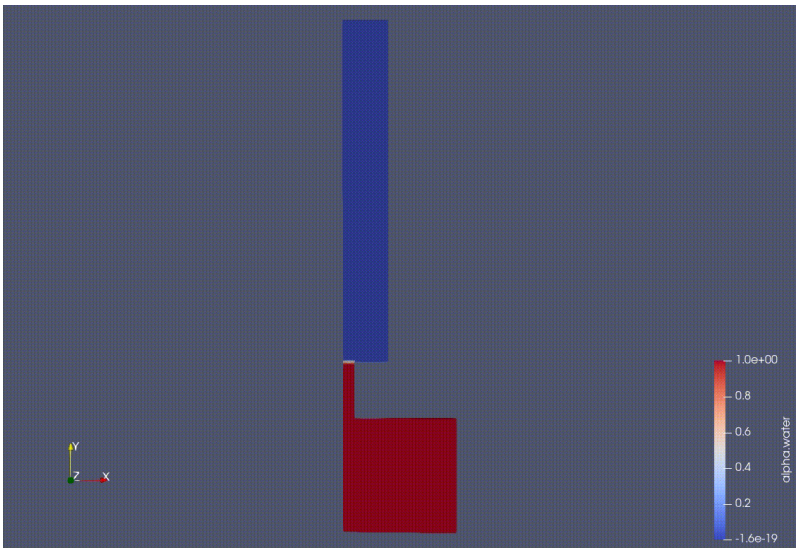
```
    inlet
    {
        type            pressureInletOutletVelocity;
        value           uniform (0 0 0);
    }
```

・p_rghの境界条件を変更

```
    inlet
    {
//      type            fixedValue;
//              value           uniform 0;
//      type            zeroGradient;
        type            uniformFixedValue;
        uniformValue    table
        (
            (0      4e5)
            (10e-6   4e5)
            (11e-6  -8e4)
            (30e-6   0e4)
            (50e-6   0e5)
        );
```

```
    value           uniform 0;
}
```

・./Allclean ⏎
foamRun ⏎ で計算実行
paraFoam で結果確認



１２　パラメータスタディー

・ノズルやヘッドの形状を色々変えて吐出への影響をみてみよう.

★裏技１　座標自動計算

●blockMeshの座標を自動的に計算する方法　(OF12 blueCFD-core-2024)
形状を変えるたびに座標を計算するのが大変なので，点情報(vertices)を自動的に算出するように変更する.

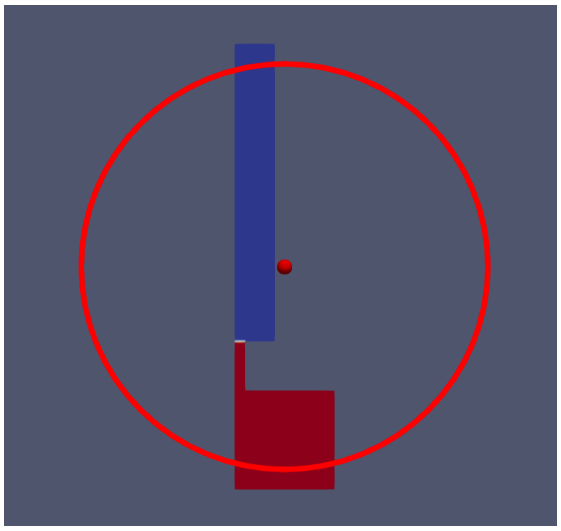
```
convertToMeters 1.0e-6;
x_air    40.0;  // air field radius
y_air    300.0; // air field length
x_nozzle 10.0;  // nozzle radius
y_nozzle 50.0;  // nozzle length
x_cavity 100.0; // cavity radius
y_cavity 150.0; // cavity length

//----- Don't change below
theta 2.5;
z_air_p    #calc " $x_air   *sin(degToRad($theta))";
z_air_n    #neg $z_air_p;
z_nozzle_p #calc " $x_nozzle*sin(degToRad($theta))";
z_nozzle_n #neg $z_nozzle_p;
z_cavity_p #calc " $x_cavity*sin(degToRad($theta))";
z_cavity_n #neg $z_cavity_p;
y_nozzle_n #neg $y_nozzle;
y_cavity_n #neg $y_cavity;

//--- reference
//rBend #calc "sqrt(scalar($rBall*$rBall - $rPipe*$rPipe))";
//zr          #calc "$radius*cos(degToRad(0.5*$angle))";
//halfWidth    #calc "$radius*sin(degToRad(0.5*$angle))";
//negHalfWidth #neg $halfWidth;
//--- reference end

vertices
(
  ( 0           0       0)     //0
  ( 0       $y_air      0)     //1
  ($x_nozzle     0  $z_nozzle_n)  //2
  ($x_nozzle $y_air  $z_nozzle_n)  //3
  ($x_air        0  $z_air_n)     //4
  ($x_air    $y_air  $z_air_n)     //5
  ($x_nozzle     0  $z_nozzle_p)   //6
  ($x_nozzle $y_air  $z_nozzle_p)   //7
  ($x_air        0  $z_air_p)     //8
```

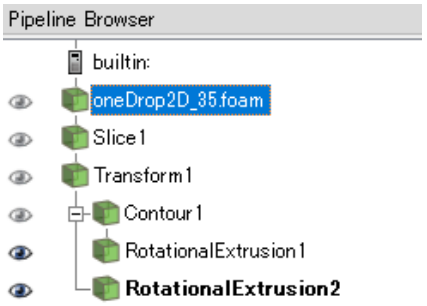
```
  ($x_air    $y_air  $z_air_p)      //9
  (0         $y_nozzle_n  0)           //10
  ($x_nozzle $y_nozzle_n  $z_nozzle_n)   //11
  ($x_nozzle $y_nozzle_n  $z_nozzle_p)   //12
  ($x_cavity $y_nozzle_n  $z_cavity_n)   //13
  ($x_cavity $y_nozzle_n  $z_cavity_p)   //14
  (  0  $y_cavity_n 0)                   //15
  ($x_nozzle  $y_cavity_n $z_nozzle_n)   //16
  ($x_nozzle  $y_cavity_n $z_nozzle_p)   //17
  ($x_cavity  $y_cavity_n $z_cavity_n)   //18
  ($x_cavity  $y_cavity_n $z_cavity_p)   //19
);
```

★裏技2　3次元表示

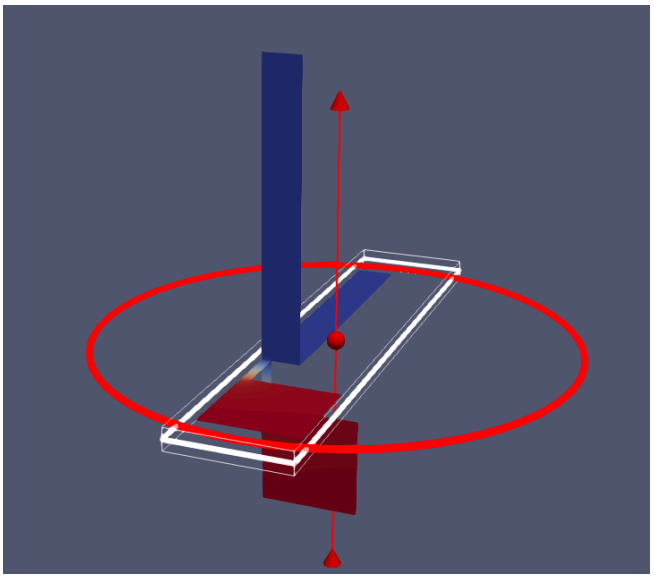
せっかくなので3Dで迫力ある動画をつくってみましょう.
参考
https://groups.google.com/forum/#!topic/openfoam/nYJ00DhzLG0

・まずSliceでZ軸を通る断面を作成

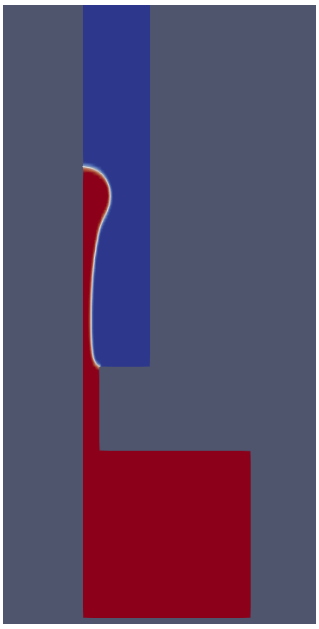




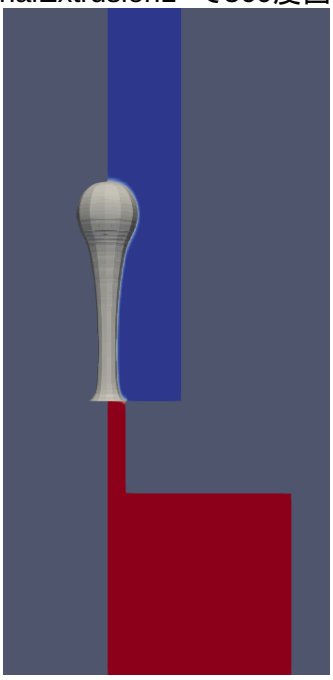
・TransformでX軸回り90度回転させる.



・Contourでalpha=0.5の境界線を作成する.



・RotationalExtrusion1　で360度回転させると立体的になる.



・もう一回 foam を読み込んで，壁だけ表示させて・・・
TransformでX軸回り90度回転
RotationalExtrusion1　で360度回転
壁を半透明にしてインクをマゼンタ色にすると
こんな絵もできる．動画にすると素敵💛．