OpenFoam

Porous medium

Ehsan Mahravan

July 25, 2018

Please send me your feedback, comments, and suggestions

mahravan.ehsan@gmail.com

Copy the following tutorial to your working folder:
$FOAM_TUTORIALS/incompressible/porousSimpleFoam/angledDuctImplicit

Simple Questions

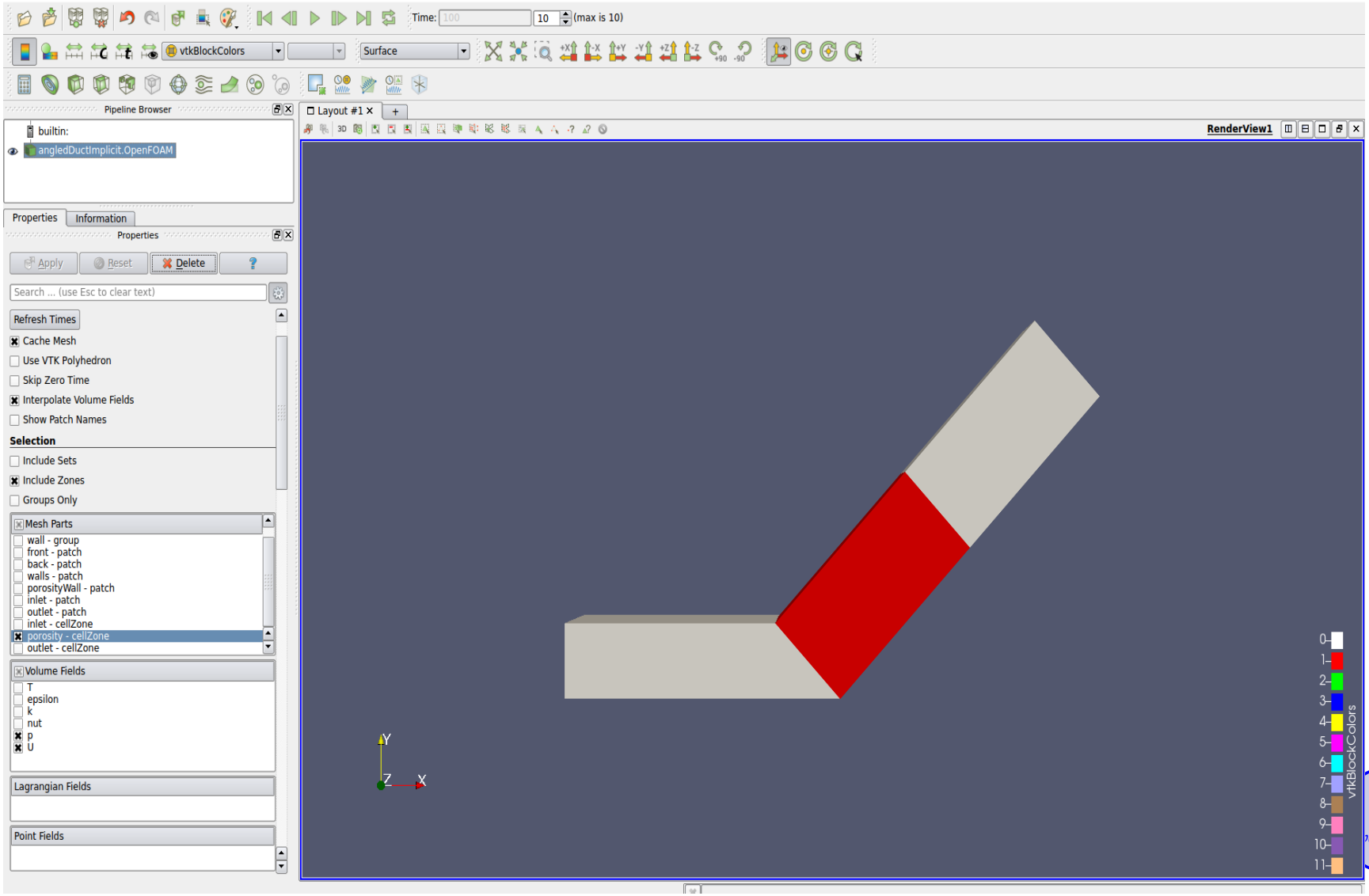
1. What are steps to run this tutorial?
2. where is porous media in geometry?
3. where porous inputs are given?

```
1  #!/bin/sh
2  cd ${0%/*} || exit 1    # Run from this directory
3
4  m4 system/blockMeshDict.m4 > system/blockMeshDict
5
6  # Source tutorial run functions
7  . $WM_PROJECT_DIR/bin/tools/RunFunctions
8
9  runApplication blockMesh
10 runApplication `getApplication`
```

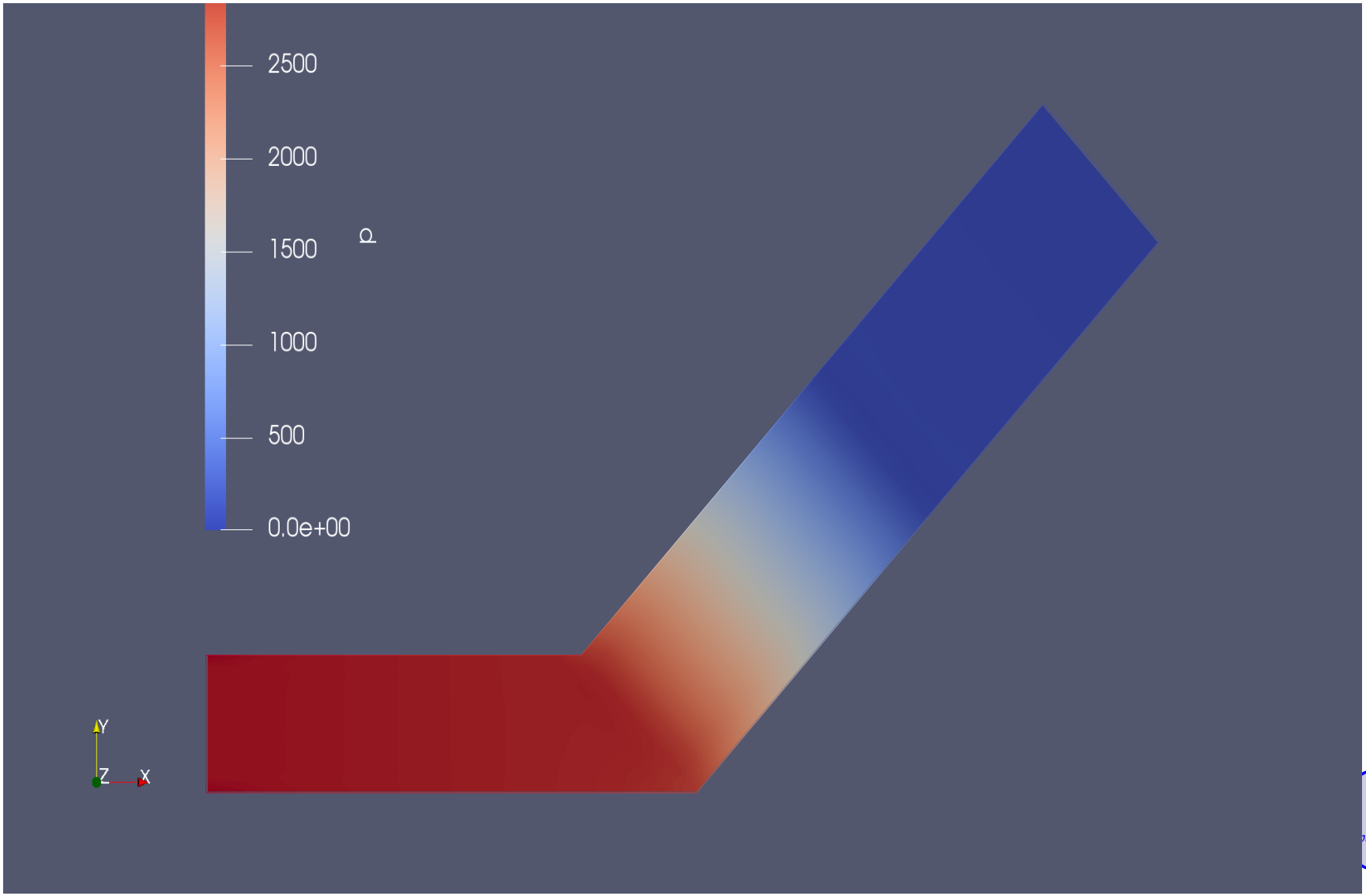
1. line 4: Creating `blockMeshDict` file using *m4*,
2. line 9: Creating mesh using blockMesh,
3. line 10: Solving the case using *porousSimpleFoam*.

1. Execute `Allrun` script by: ./ Allrun
2. Open paraview using paraFoam
3. In properties panel toggle include zone,
4. In mesh parts select prosity – cellZone,
5. In display panel select vtkBlockColors,

- The red block in the picture is porous region.



- Porous medium causes a considarable pressure drop.



- `topoSet` is not used, so how the porous zone is specified?
- This zone must be specified in blockMeshDict. Where blocks are defined, one can also gives them a name and after generating the mesh it will be a seperate zone.

```
1  blocks
2  (
3      // inlet block
4      hex (0 4 5 1 2 6 7 3)
5      inlet ( 15 20 20 ) simpleGrading (1 1 1)
6
7      // porosity block
8      hex (4 8 9 5 6 10 11 7)
9      porosity ( 20 20 20 ) simpleGrading (1 1 1)
10
11     // outlet block
12     hex (8 12 13 9 10 14 15 11)
13     outlet ( 20 20 20 )  simpleGrading (1 1 1)
14 );
```

- Porosity properties are given in constant / porosityProperties
- This file can be provided only for solvers which has *poros* word in their name.
- Other solvers can also handle porosity. For these solvers one can provide porosity information in *fvOptions*.

```
porosity1
{
    type            DarcyForchheimer;

    cellZone        porosity;

    d   (5e7 -1000 -1000);
    f   (0 0 0);

    coordinateSystem
    {
        type    cartesian;
        origin  (0 0 0);
        coordinateRotation
        {
            type    axesRotation;
            e1      (0.70710678 0.70710678 0);
            e2      (0 0 1);
        }
    }
}
```

- type: Name of the model to encounter porous effect.
- cellZone: is the name of the zone for which porosity model will be applied.
- active: One can choose whether porosity model be applied or not by changing *active* entry to yes or no.
- modeleNameCoeff: Requirements of selected model.

For encountering porosity effect a source term (S) must be added to the momentum equation. This can be done using three well known porosity model as follows

1. DarcyForchheimer: Viscous and inertial resistances vectors (d and f) must be specified for diagonal of the source term added to Navier-Stocks equations. Coordinate of vectors correspond to local coordinate system.

$$S = -(\mu d + \frac{\rho|U|}{2} f)U$$

2. fixedCoeff: Two vector constants (alpha and beta are needed and for compressible flow reference density is also required.

$$S = -\rho_{ref}(\alpha + \beta|U|)U$$

3. powerLaw: Two constants (C_0 and C_1) are needed, the first in Coefficient and the second is power used in the source term.

$$S = -\rho C_0 |U|^{(C_1 - 1)} U$$

A local coordinate system which represent direction of prosity must be specified. The only available type is *cartesian* which needs an origin and orientation.
Orientation here is along angled part of the duct. *coordinateRotation* type are as follows:

1. axesRotation: The 3 rotations are defined in the axis convention around X, around Y and around Z).
2. STARCDRotation: The 3 rotations are defined in the STAR-CD convention (around `Z`, around `X'` and around `Y''`).
3. EulerCoordinateRotation: The 3 rotations are defined in the STAR-CD convention (around `Z`, around `X'` and around `Y''`) for STARCDRotation
4. cylindrical

```
coordinateRotation
{
    type STARCDRotation;// EulerCoordinateRotation
    degrees false;// true
    rotation (0 0 3.141592654);//(0 0 180)
}
coordinateRotation
{
    type         EulerRotation
    degrees      false;
    rotation     (0 0 3.141592654);
}
cylindrical
{
     type         localAxes;
     e3           (0 0 1);
}
```

```
porosity1
{
    type            fixedCoeff;
    active          yes;
    cellZone        porosity;

    fixedCoeffCoeffs
    {
    alpha        (500 -1000 -1000);
    beta         (0 0 0);

        coordinateSystem
        {
            type    cartesian;
            origin  (0 0 0);
            coordinateRotation
            {
                type    axesRotation;
                e1      (0.70710678 0.70710678 0);
                e2      (0 0 1);
            }
        }
    }
}
```

$FOAM_TUTORIALS/incompressible/pisoFoam/laminar/porousBlockage/

```
porosity1
{
    type            explicitPorositySource;

    explicitPorositySourceCoeffs
    {
        selectionMode   cellZone;
        cellZone        porousBlockage;

        type            DarcyForchheimer;

        // D 100;  // Very little blockage
        // D 200;  // Some blockage but steady flow
        // D 500;  // Slight waviness in the far wake
        D 1000; // Fully shedding behavior

        d   ($D $D $D);
        f   (0 0 0);

        coordinateSystem
        {
            type    cartesian;
            origin  (0 0 0);
            coordinateRotation
            {
                type    axesRotation;
                e1  (1 0 0);
                e2  (0 1 0);
            }
        }
    }
}
```

References

- OpenFOAM header files.
- Description of porousSimpleFoam and adding the Brinkmann model to the porous models, Reza Gooya, A course at Chalmers University of Technology.