Lab 1 GEO5015: Linux and OpenFOAM

Modelling wind and dispersion in urban environments

TU Delft

Learning Objectives:

After this class, you will be able to:

- How to run openFOAM in your machines
- Install and run paraView

Activities:

This class is hands-on, so you will need to follow my steps with your own computer, so no activities will be carried out, but I will still ask you questions….

What we will learn might be very new to some, and not so new to others, if you have any doubt it is very important to solve it asap, so you don't get lost

Activities:

This class is hands-on, so you will need to follow my steps with your own computer, so no activities will be carried out, but I will still ask you questions….

What we will learn might be very new to some, and not so new to others, if you have any doubt it is very important to solve it asap, so you don't get lost

Remember you will use what I show you today, to run your own case for the assignment!!!

Outline of the class:

1. What is openFOAM?

2.openFOAM test case structure

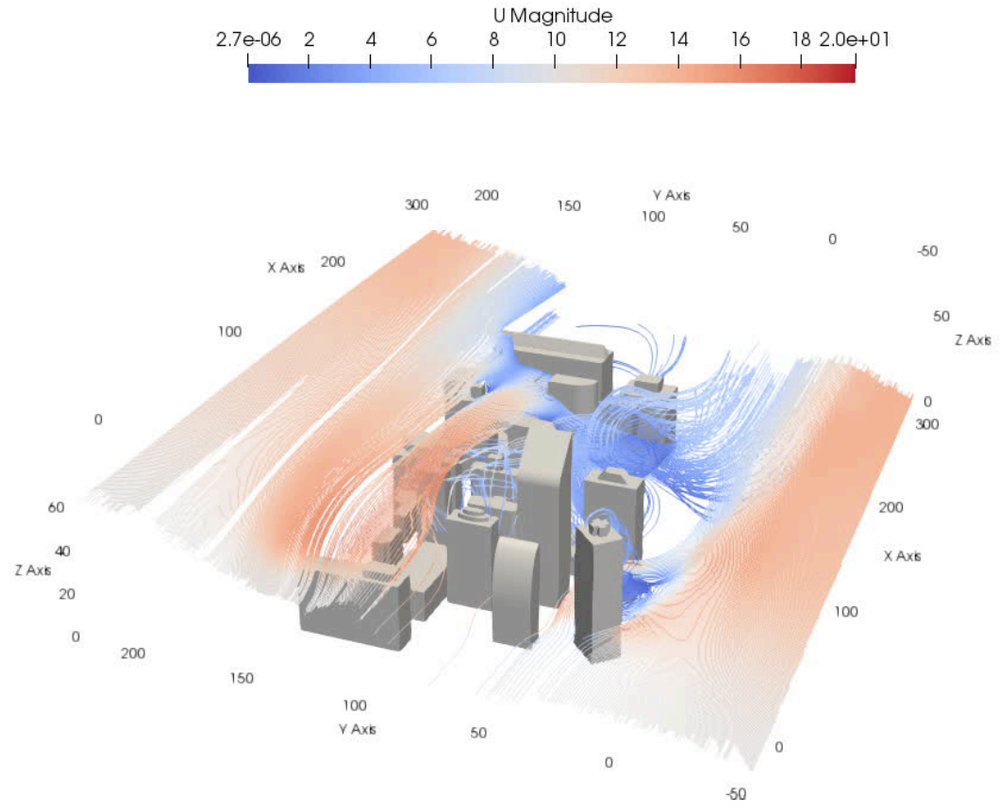
3. How to run openFOAM test case

4. Visualise the results with paraView

TU Delft

What is openFOAM?

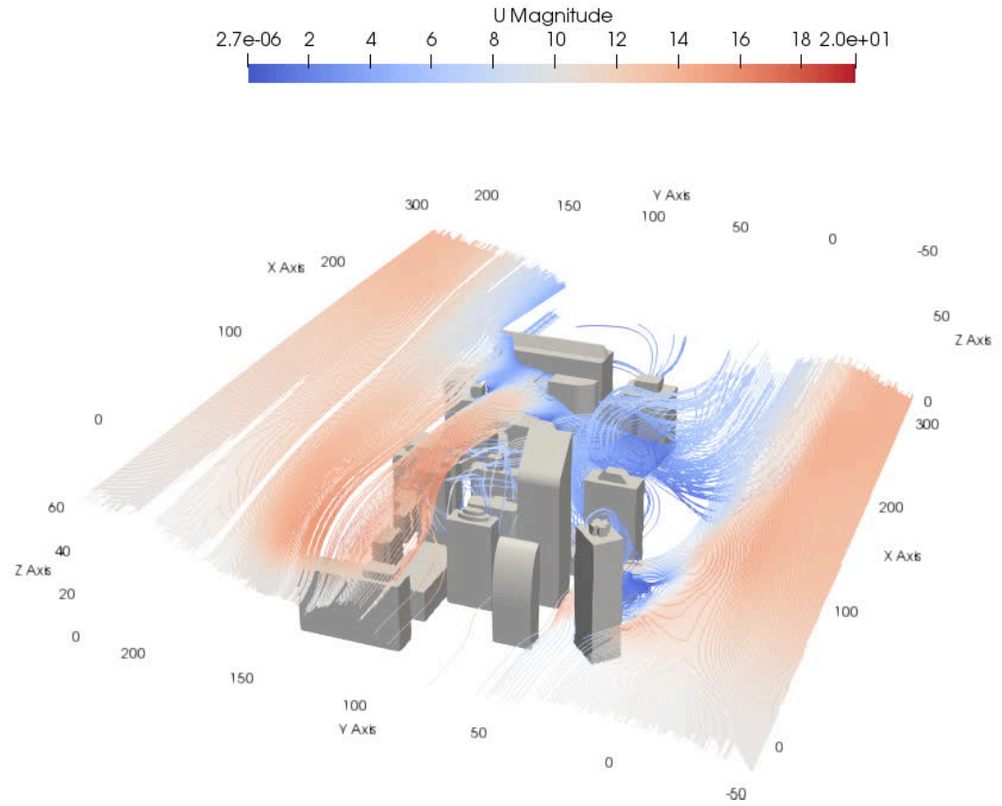
- It is a collection of about 250 applications built from a collection of 100 software libraries (modules)
- Each application performs a specific task within a CFD workflow: snappyHexMesh will generate a mesh for complex geometry for example a car, while simpleFoam will then simulate steady-state , turbulent, incompressible flow around that car.



TU Delft

What is openFOAM?

- It is a collection of about 250 applications built from a collection of 100 software libraries (modules)
- Each application performs a specific task within a CFD workflow: snappyHexMesh will generate a mesh for complex geometry for example a car, while simpleFoam will then simulate steady-state , turbulent, incompressible flow around that car.



TU Delft

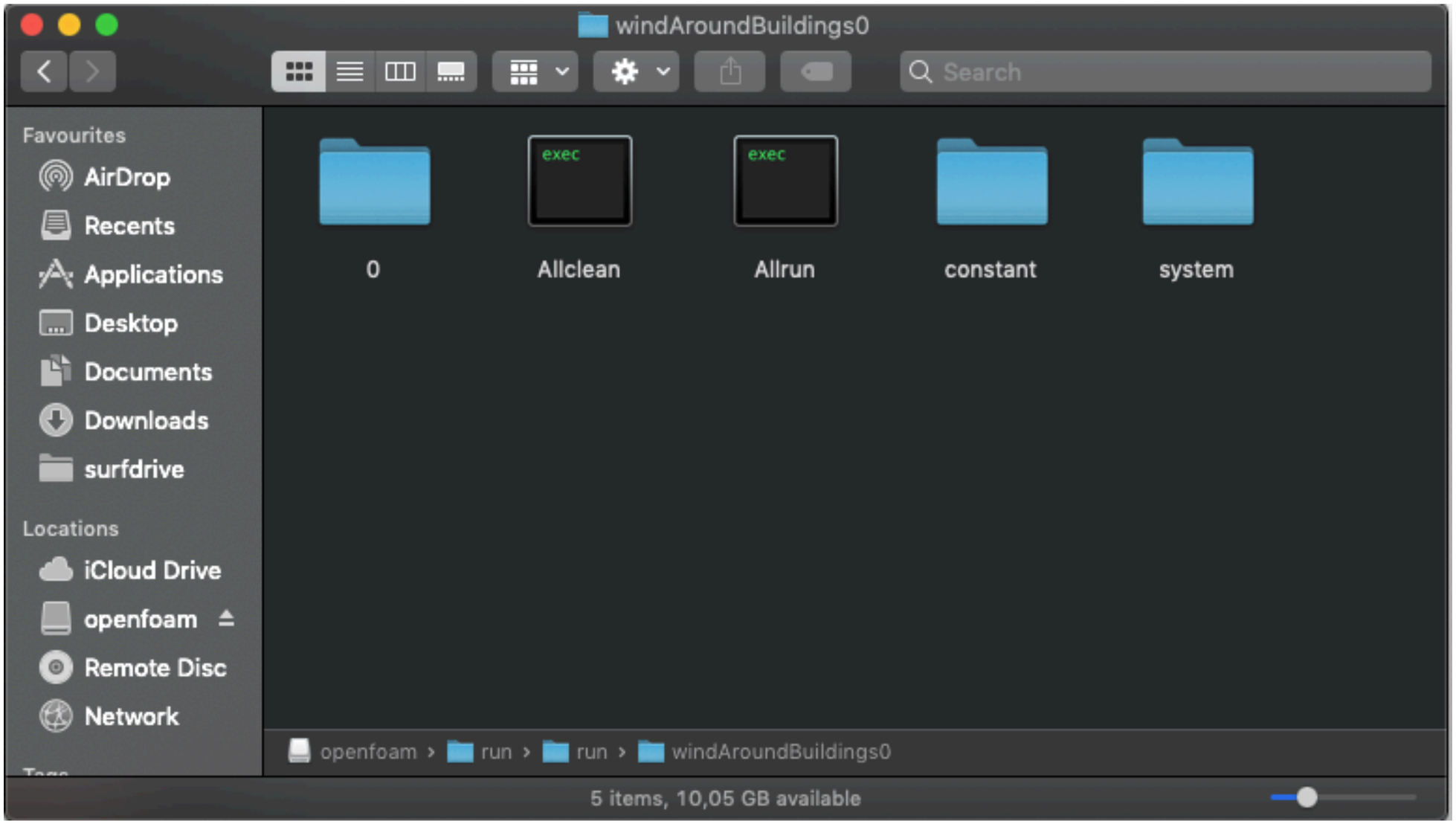
openFOAM test case structure

Folders tree code:		Symbolic path:
Applications	Solvers, utilities, test: where the equations are implemented. They have as output an app generally	$FOAM_APP
SRC	Core of the code where the physics, the boundary conditions, the mesh, and many other libraries are hosted	$FOAM_SRC
Tutorials	Multiple tutorials to run, with different physics and goals	$FOAM_TUTORIALS
Etc	OF configuration, caseDicts to start from, codeTemplates to implement new apps or libraries	$FOAM_ETC
Run	Your folder, where you will run your cases. If you don't have it, you can create it with mkdir -p $FOAM_RUN	$FOAM_RUN

openFOAM test case structure

- Folders tree in case:

```
cp -r $FOAM_TUTORIALS/incompressible/simpleFoam/windAroundBuildings $FOAM_RUN/
```

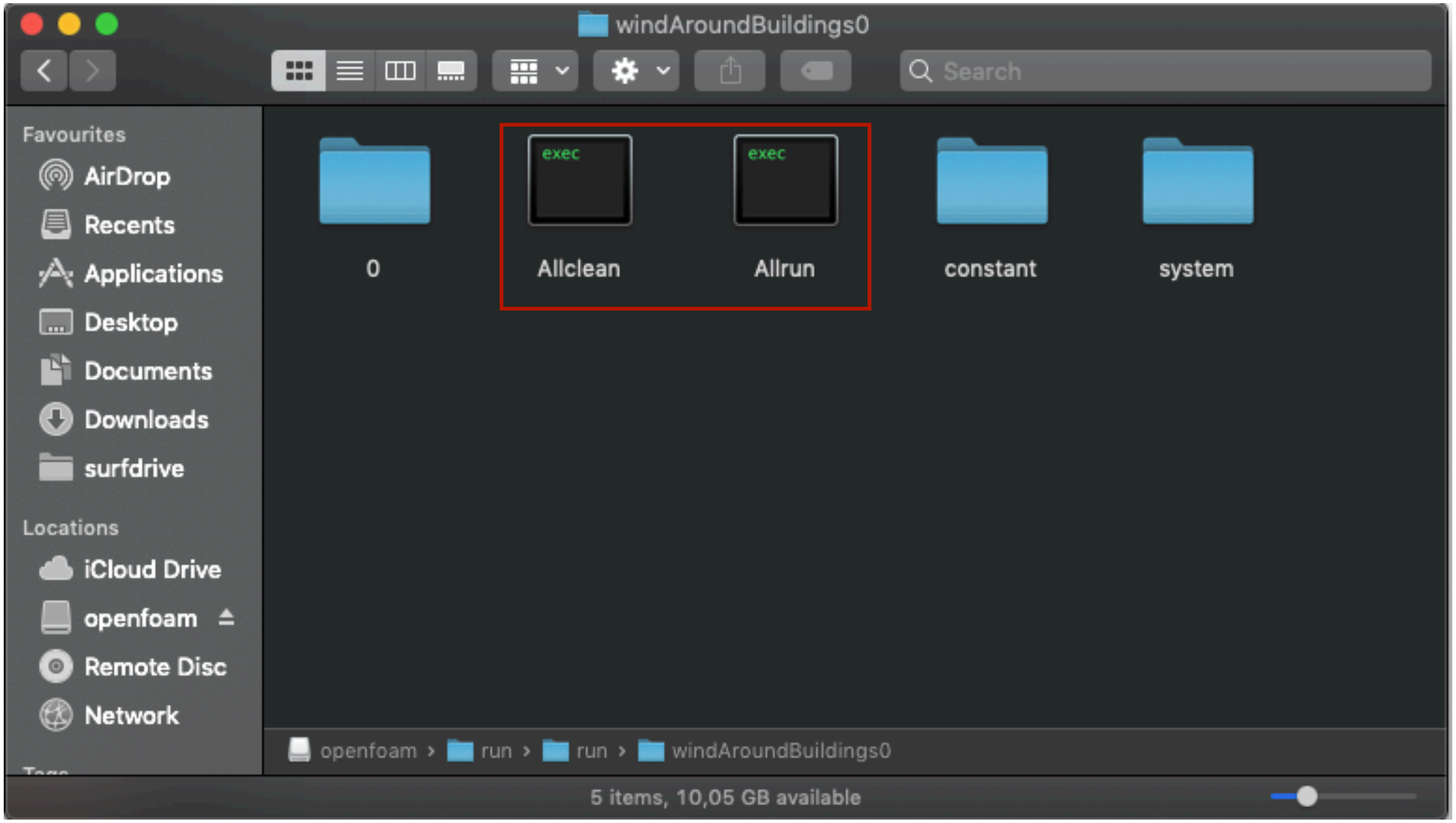


TU Delft

openFOAM test case structure

- Folders tree in case:

```
cp -r $FOAM_TUTORIALS/incompressible/simpleFoam/windAroundBuildings $FOAM_RUN/
```



TU Delft

openFOAM test case structure

- Allrun:

```
#!/bin/sh
cd ${0%/*} || exit 1    # run from this directory

# Source tutorial run functions
. $WM_PROJECT_DIR/bin/tools/RunFunctions

runApplication surfaceFeatures

runApplication blockMesh
runApplication snappyHexMesh -overwrite

runApplication $(getApplication)

#------------------------------------------------------------------------------
```

openFOAM test case structure

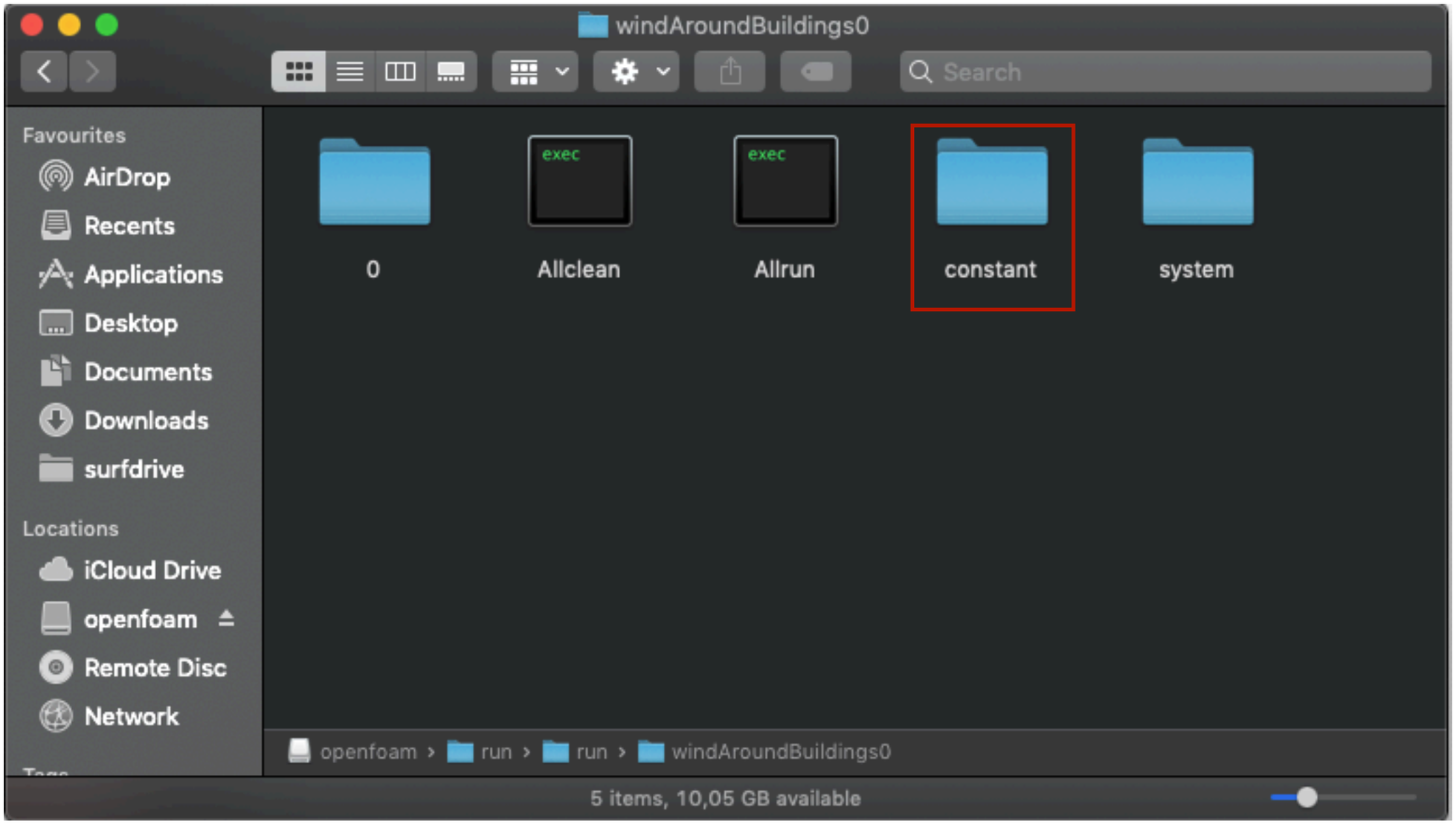
- Allclean:

```
#!/bin/sh
cd ${0%/*} || exit 1    # run from this directory

# Source tutorial clean functions
. $WM_PROJECT_DIR/bin/tools/CleanFunctions

cleanCase

rm -rf constant/extendedFeatureEdgeMesh > /dev/null 2>&1
rm -f constant/triSurface/buildings.eMesh > /dev/null 2>&1
rm -f constant/polyMesh/boundary > /dev/null 2>&1

#------------------------------------------------------------------------------
```

openFOAM test case structure

- Folders tree in case:

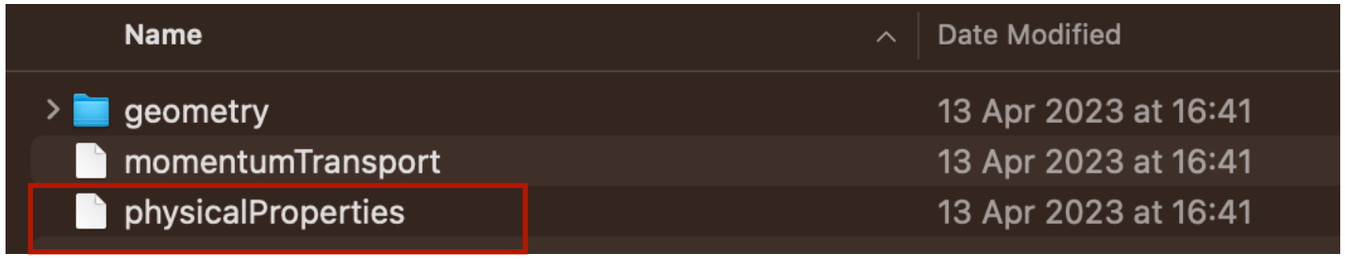
```
cp -r $FOAM_TUTORIALS/incompressible/simpleFoam/windAroundBuildings $FOAM_RUN/
```



TU Delft

openFOAM test case structure

- Constant:




```
/*--------------------------------*- C++ -*----------------------------------*\
  =========                 |
  \\      /  F ield         | OpenFOAM: The Open Source CFD Toolbox
   \\    /   O peration     | Website:  https://openfoam.org
    \\  /    A nd           | Version:  10
     \\/     M anipulation  |
\*---------------------------------------------------------------------------*/
FoamFile
{
    format      ascii;
    class       dictionary;
    object      physicalProperties;
}
// * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * //

viscosityModel  constant;

nu              [0 2 -1 0 0 0 0] 1.5e-05;

// ************************************************************************* //
~
```

openFOAM test case structure

- Constant:

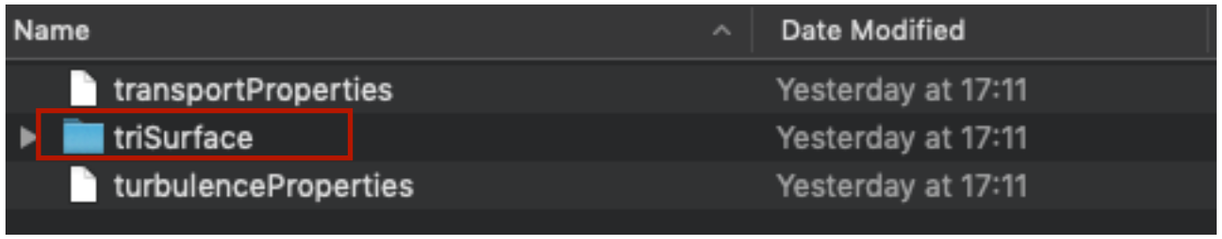

```
/*--------------------------------*- C++ -*----------------------------------*\
  =========                 |
  \\      /  F ield         | OpenFOAM: The Open Source CFD Toolbox
   \\    /   O peration     | Website:  https://openfoam.org
    \\  /    A nd           | Version:  10
     \\/     M anipulation  |
\*---------------------------------------------------------------------------*/
FoamFile
{
    format      ascii;
    class       dictionary;
    object      RASProperties;
}
// * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * //

simulationType RAS;

RAS
{
    model           kEpsilon;

    turbulence      on;

    printCoeffs     on;
}

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

TU Delft

openFOAM test case structure

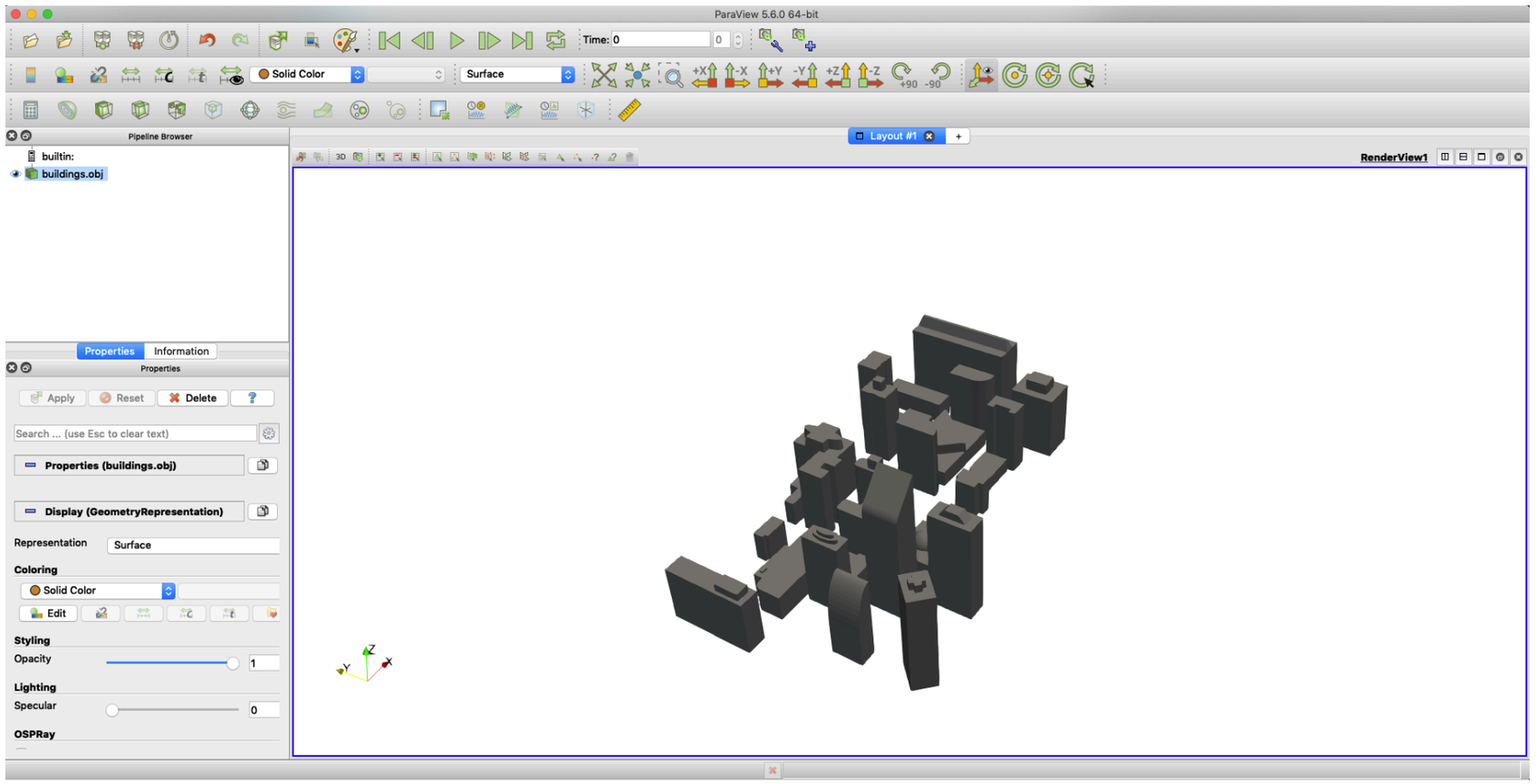
- Constant:

Geometry —>



gzip -d buildings.obj.gz

Open buildings.obj with Paraview

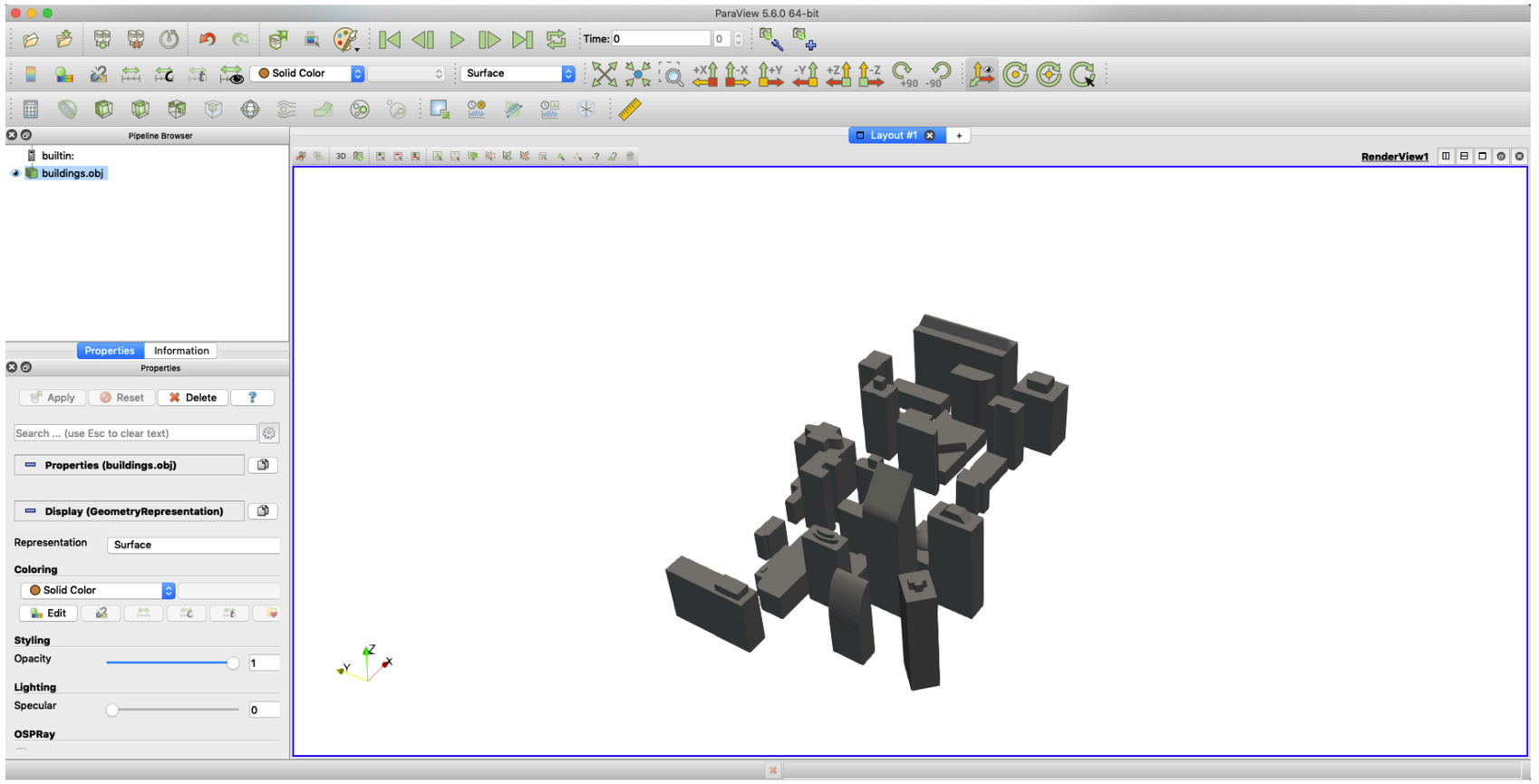


TU Delft

Let’s install paraView now!

Go to: https://www.paraview.org/download/

To visualise an openFOAM case you always need a dummy file with any name and .foam extension, like windAroundBuildings.foam

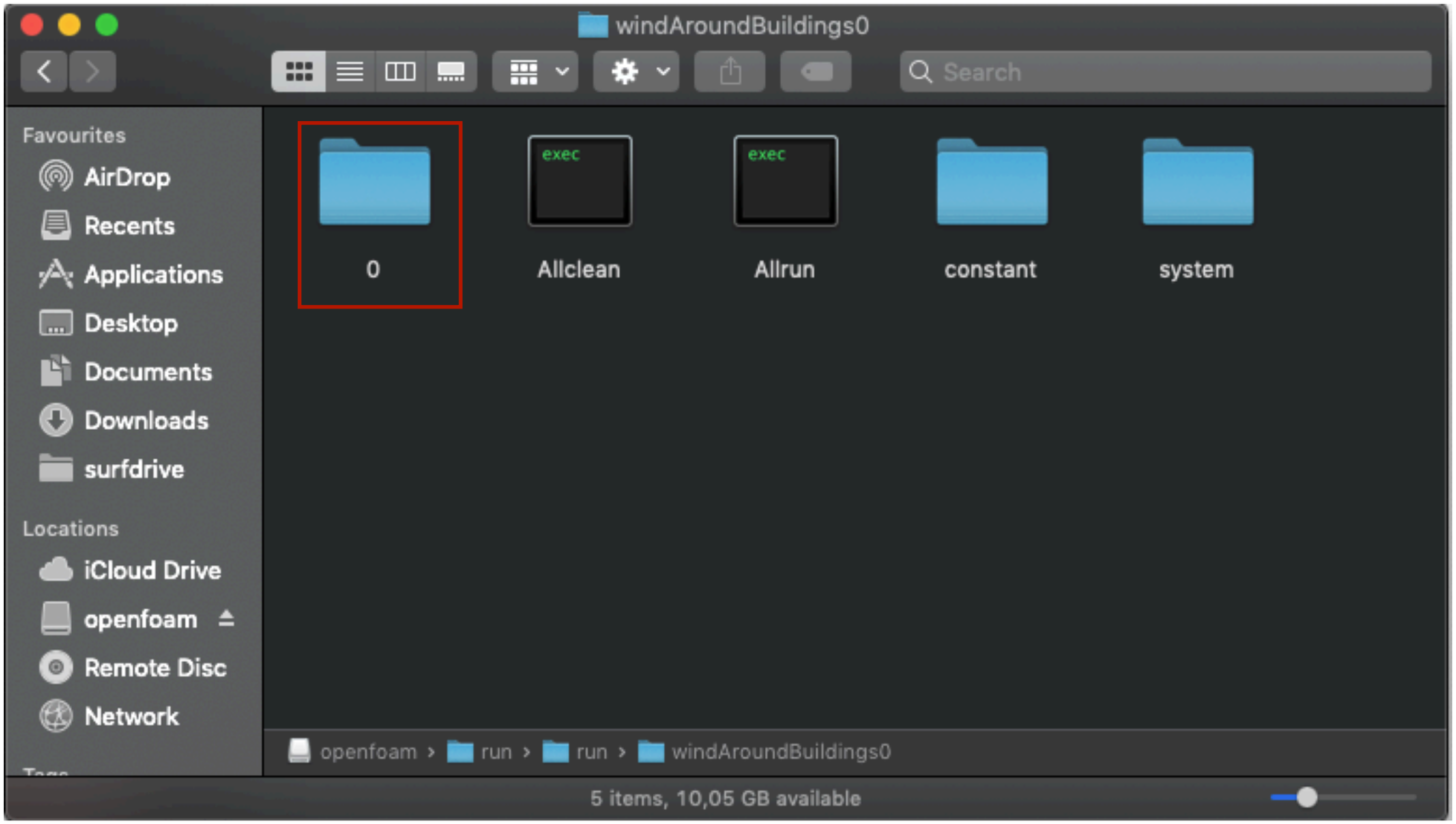


TU Delft

openFOAM test case structure

- Folders tree in case:

```
cp -r $FOAM_TUTORIALS/incompressible/simpleFoam/windAroundBuildings $FOAM_RUN/
```



TU Delft

openFOAM test case structure

- 0: contains the initial and boundary conditions for the fields to be modelled

  The basic boundary conditions are:

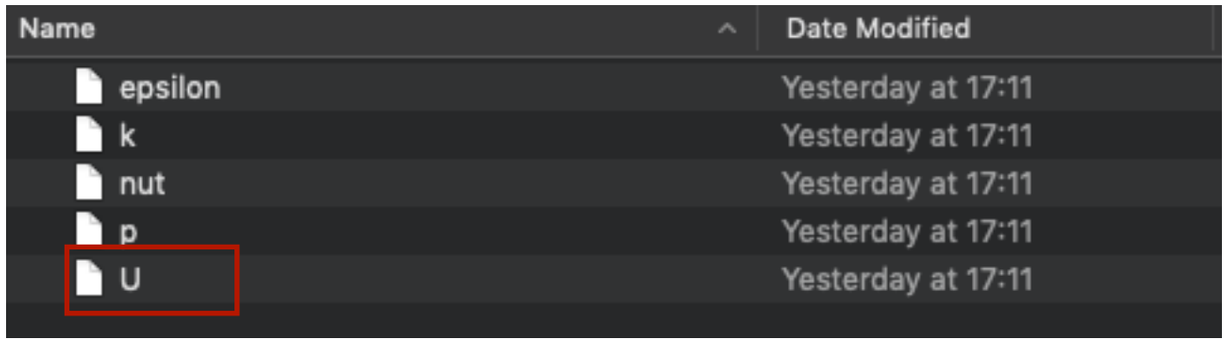
  - fixedValue: value of field is specified
  - fixedGradient: value of the normal field gradient is specified
  - zeroGradient: normal field gradient is null (0)
  - calculated: patch field value is calculated from other patches
  - mixed: mixed fixedValue/fixedGradient condition depending on a valueFraction (0(refGradient)<vf<1 (refValue))
  - directionMixed: mixed conditions with tensorial valueFraction to allow different conditions in normal and tangential direction of a vector patch field (fixedValue in tangential, zeroGradient in normal direction)

  simpleFoam -listScalarBCs -listVectorBCs

  foamInfo totalPressure

openFOAM test case structure

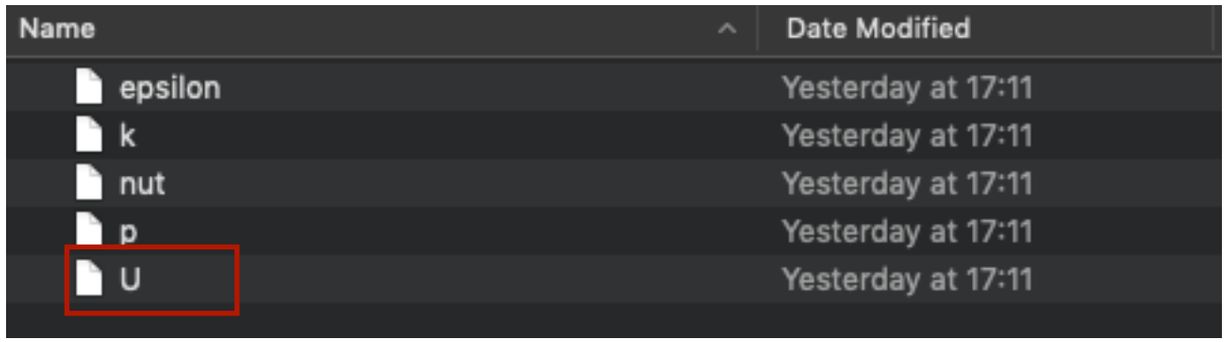
- 0: U - velocity initial and boundary values:




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

Uinlet          (10 0 0);

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

internalField   uniform (0 0 0);

boundaryField
{
    inlet
    {
        type            fixedValue;
        value           uniform $Uinlet;
    }

    outlet
    {
        type            pressureInletOutletVelocity;
        value           uniform (0 0 0);
    }

    wall
    {
        type            noSlip;
    }

    #includeEtc "caseDicts/setConstraintTypes"
}

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

openFOAM test case structure

- 0: U - velocity initial and boundary values:




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

Uinlet          (10 0 0);

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

internalField   uniform (0 0 0);

boundaryField
{
    inlet
```

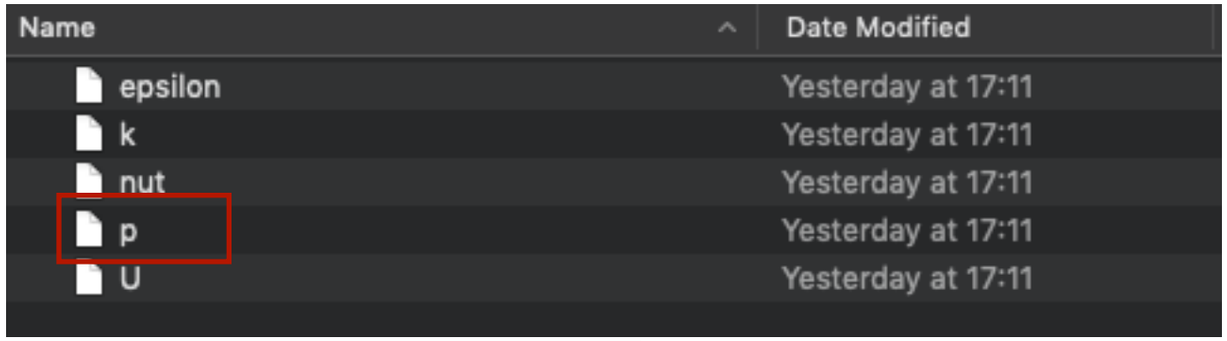
No.	Property	SI unit	USCS unit
1	Mass	kilogram (kg)	pound-mass (lbm)
2	Length	metre (m)	foot (ft)
3	Time	second (s)	second (s)
4	Temperature	Kelvin (K)	degree Rankine (°R)
5	Quantity	mole (mol)	mole (mol)
6	Current	ampere (A)	ampere (A)
7	Luminous intensity	candela (cd)	candela (cd)

```
    #includeEtc "caseDicts/setConstraintTypes"
}

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

TU Delft

openFOAM test case structure

- 0: p - pressure initial and boundary values:

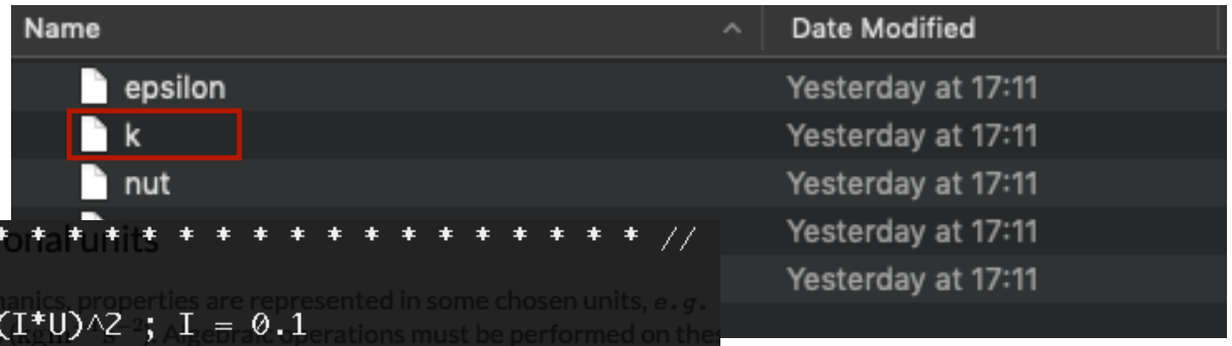



```
// * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * //

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

internalField   uniform 0;

boundaryField
{
    inlet
    {
        type            zeroGradient;
    }
    outlet
    {
        type            totalPressure;
        p0              uniform 0;
        gamma           1.4;
        value           uniform 0;
    }

    wall
    {
        type            zeroGradient;
    }

    #includeEtc "caseDicts/setConstraintTypes"
}

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

TU Delft

openFOAM test case structure

- 0: k - tke initial and boundary values:

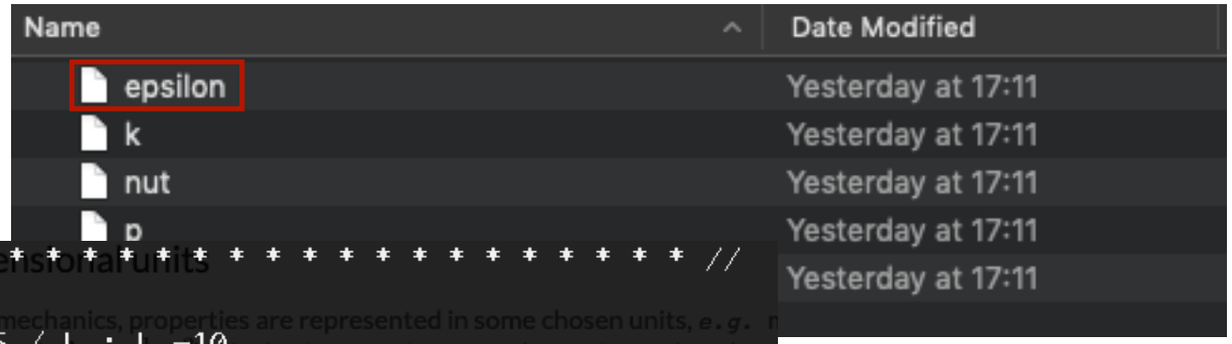



```
// * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * //

kInlet          1.5;   // approx k = 1.5*(I*U)^2 ; I = 0.1

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

internalField   uniform $kInlet;

boundaryField
{
    inlet
    {
        type            fixedValue;
        value           uniform $kInlet;
    }

    outlet
    {
        type            inletOutlet;
        inletValue      uniform $kInlet;
        value           uniform $kInlet;
    }

    wall
    {
        type            kqRWallFunction;
        value           uniform $kInlet;
    }

    #includeEtc "caseDicts/setConstraintTypes"
}

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

TU Delft

openFOAM test case structure

- 0: **ε** - dissipation initial and boundary values:

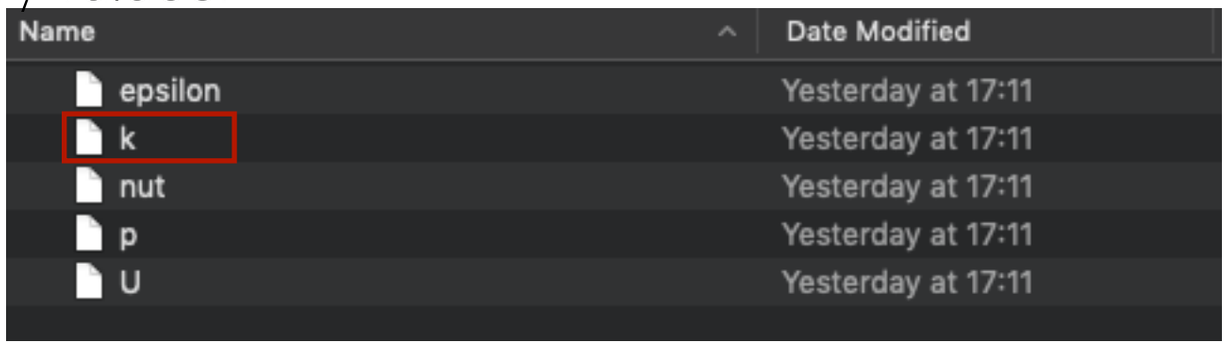



```
// * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * //

epsilonInlet  0.03; // Cmu^0.75 * k^1.5 / L ; L =10

dimensions      [0 2 -3 0 0 0 0];

internalField   uniform $epsilonInlet;

boundaryField
{
    inlet
    {
        type            fixedValue;
        value           uniform $epsilonInlet;
    }

    outlet
    {
        type            inletOutlet;
        inletValue      uniform $epsilonInlet;
        value           uniform $epsilonInlet;
    }

    wall
    {
        type            epsilonWallFunction;
        value           uniform $epsilonInlet;
    }

    #includeEtc "caseDicts/setConstraintTypes"
}

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

TU Delft

openFOAM test case structure

- 0: $\mathbf{v}_t$ - turbulent viscosity initial and boundary values:



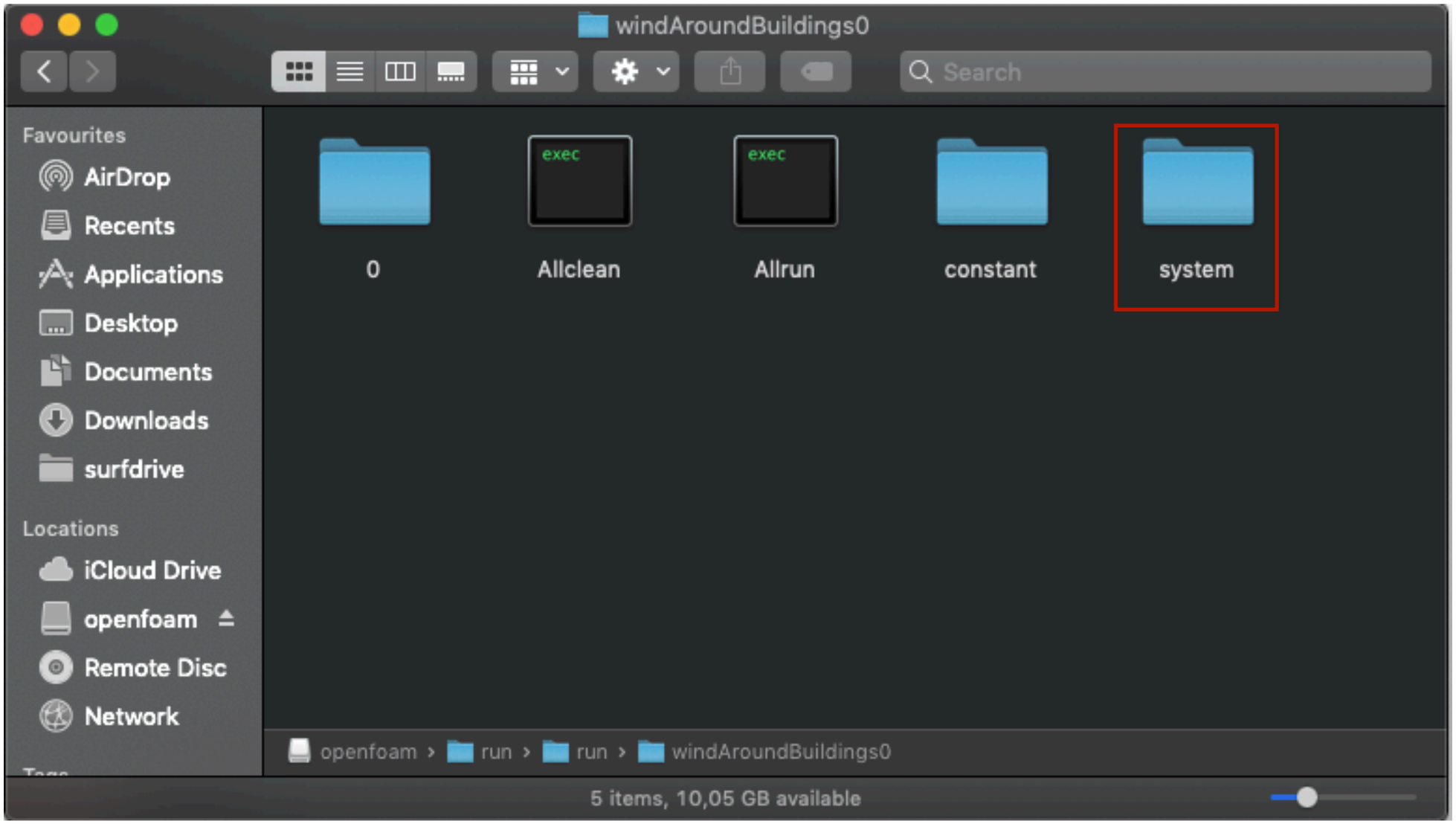

```
// * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * //

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

internalField   uniform 0;

boundaryField
{
    inlet
    {
        type            calculated;
        value           uniform 0;
    }

    outlet
    {
        type            calculated;
        value           uniform 0;
    }

    wall
    {
        type            nutkWallFunction;
        value           uniform 0;
    }

    #includeEtc "caseDicts/setConstraintTypes"
}

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

TU Delft

openFOAM test case structure

- Folders tree in case:

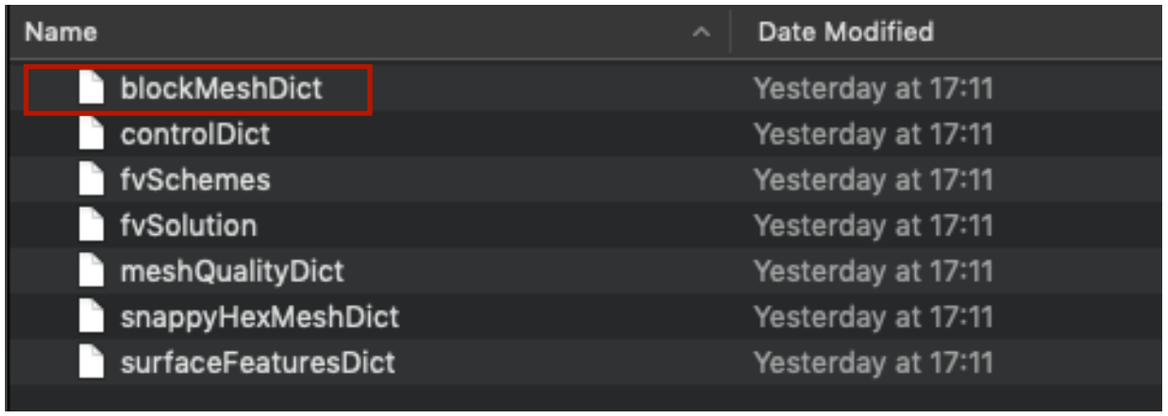
```
cp -r $FOAM_TUTORIALS/incompressible/simpleFoam/windAroundBuildings $FOAM_RUN/
```



TU Delft

openFOAM test case structure

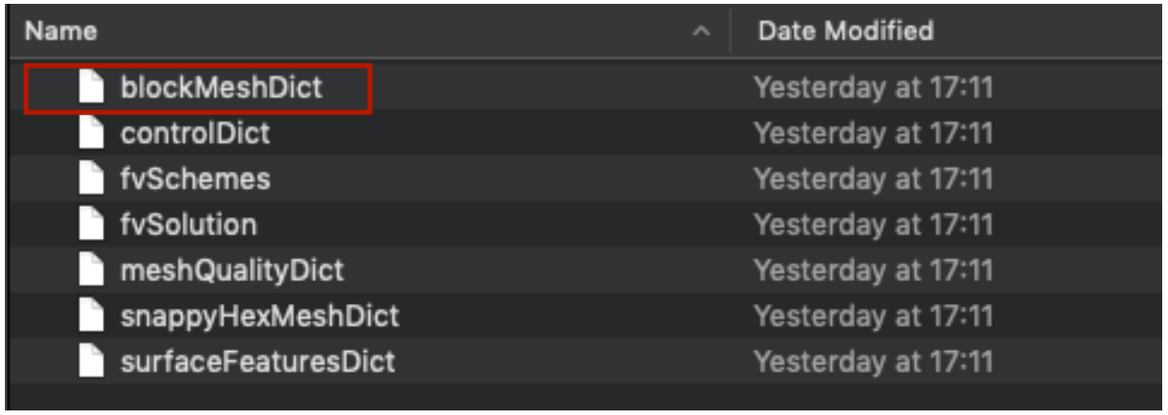
- System - BlockMeshDict:




```
boundary
(
    inlet
    {
        type patch;
        faces
        (
            (0 3 7 4)
        );
    }

    outlet
    {
        type patch;
        faces
        (
            (1 5 6 2)
        );
    }

    ground
    {
        type wall;
        faces
        (
            (0 1 2 3)
        );
    }

    frontAndBack
    {
        type symmetry;
        faces
        (
            (0 4 5 1)
            (3 2 6 7)
            (4 7 6 5)
        );
    }

);
```

openFOAM test case structure

- System - BlockMeshDict:



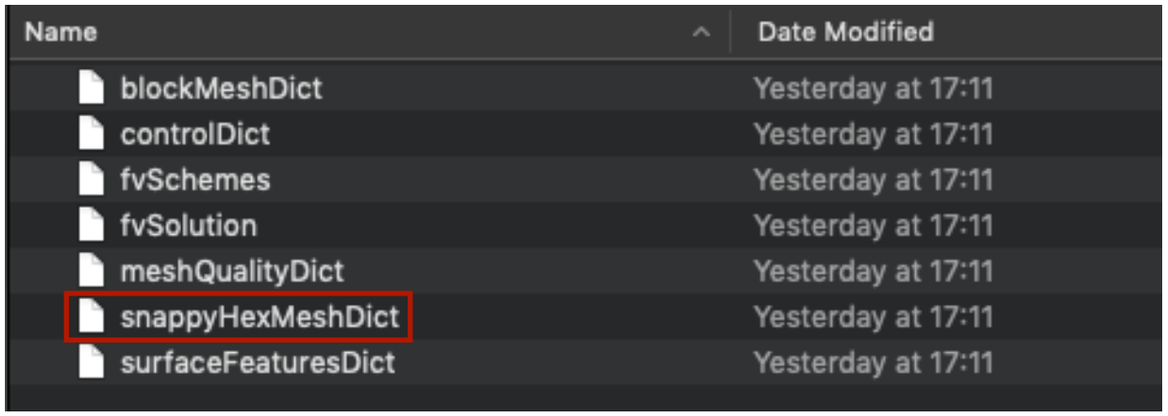

```
backgroundMesh
{
    xMin   -20; // L = 350
    xMax   330;
    yMin   -50; // L = 280
    yMax   230;
    zMin     0;
    zMax   140;
    xCells  25;
    yCells  20;
    zCells  10;
}

convertToMeters 1;

vertices
(
    ($:backgroundMesh.xMin $:backgroundMesh.yMin $:backgroundMesh.zMin)
    ($:backgroundMesh.xMax $:backgroundMesh.yMin $:backgroundMesh.zMin)
    ($:backgroundMesh.xMax $:backgroundMesh.yMax $:backgroundMesh.zMin)
    ($:backgroundMesh.xMin $:backgroundMesh.yMax $:backgroundMesh.zMin)

    ($:backgroundMesh.xMin $:backgroundMesh.yMin $:backgroundMesh.zMax)
    ($:backgroundMesh.xMax $:backgroundMesh.yMin $:backgroundMesh.zMax)
    ($:backgroundMesh.xMax $:backgroundMesh.yMax $:backgroundMesh.zMax)
    ($:backgroundMesh.xMin $:backgroundMesh.yMax $:backgroundMesh.zMax)
);
```

```
blocks
(
    hex (0 1 2 3 4 5 6 7)
    (
        $:backgroundMesh.xCells
        $:backgroundMesh.yCells
        $:backgroundMesh.zCells
    )
    simpleGrading (1 1 1)
);

edges
(
);
```

TU Delft

openFOAM test case structure

- System - SnappyHexMeshDict:



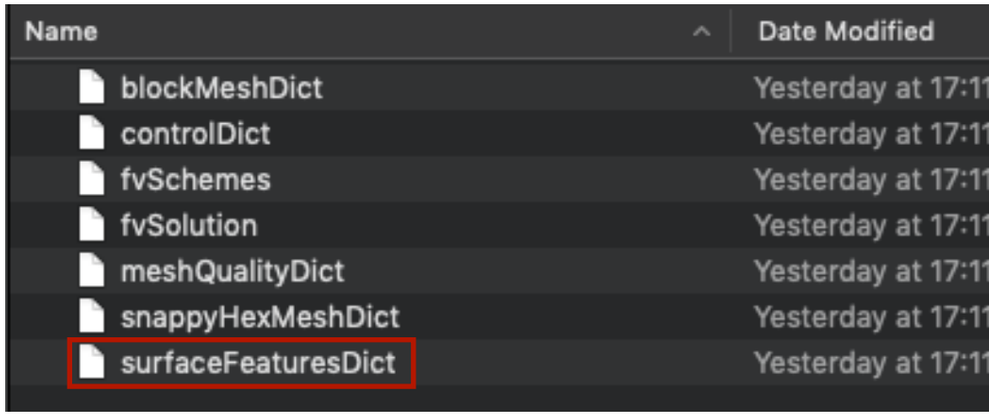

```
/*--------------------------------*- C++ -*----------------------------------*\
  =========                 |
  \\      /  F ield         | OpenFOAM: The Open Source CFD Toolbox
   \\    /   O peration     | Website:  https://openfoam.org
    \\  /    A nd           | Version:  7
     \\/     M anipulation  |
\*---------------------------------------------------------------------------*/
FoamFile
{
    version     2.0;
    format      ascii;
    class       dictionary;
    object      snappyHexMeshDict;
}
// * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * //

#includeEtc "caseDicts/mesh/generation/snappyHexMeshDict.cfg"

castellatedMesh on;
snap            on;
addLayers       off;

geometry
{
    buildings
    {
        type triSurfaceMesh;
        file "buildings.obj";
    }

    refinementBox
    {
        type searchableBox;
        min  (  0   0   0);
        max  (250 180  90);
    }
```

```
{
    features
    (
      { file  "buildings.eMesh"; level 1; }
    );

    refinementSurfaces
    {
        buildings
        {
            level (3 3);
            patchInfo { type wall; }
        }
    }

    refinementRegions
    {
        refinementBox
        {
            mode inside;
            levels ((1E15 2));
        }
    }

    locationInMesh (1 1 1);
}
```

TU Delft

openFOAM test case structure

- System - surfaceFeaturesDict:



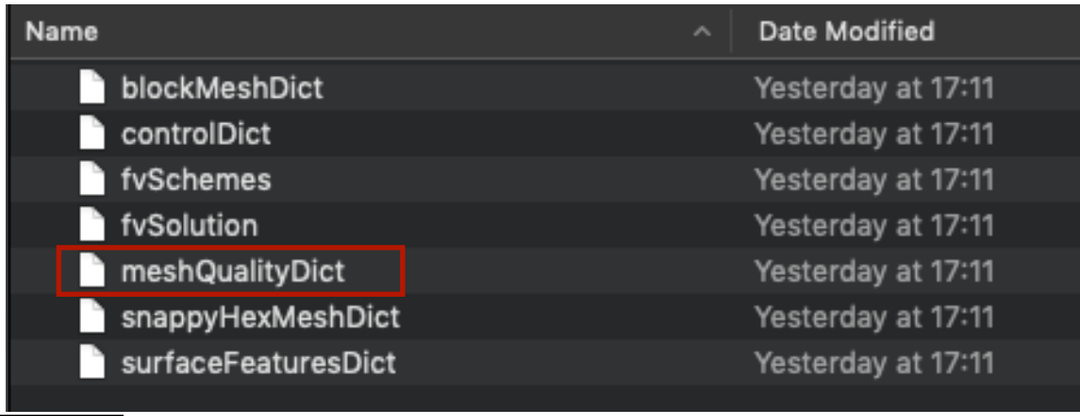

```
/*--------------------------------*- C++ -*----------------------------------*\
  =========                 |
  \\      /  F ield         | OpenFOAM: The Open Source CFD Toolbox
   \\    /   O peration     | Website:  https://openfoam.org
    \\  /    A nd           | Version:  7
     \\/     M anipulation  |
\*---------------------------------------------------------------------------*/
FoamFile
{
    version     2.0;
    format      ascii;
    class       dictionary;
    object      surfaceFeaturesDict;
}
// * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * //

surfaces ("buildings.obj");

#includeEtc "caseDicts/surface/surfaceFeaturesDict.cfg"

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

```
/*--------------------------------*- C++ -*----------------------------------*\
| =========                 |                                                 |
| \\      /  F ield         | OpenFOAM: The Open Source CFD Toolbox           |
|  \\    /   O peration     | Website:  https://openfoam.org                  |
|   \\  /    A nd           | Version:  7                                     |
|    \\/     M anipulation  |                                                 |
\*---------------------------------------------------------------------------*/

includedAngle     150;

subsetFeatures
{
    nonManifoldEdges yes;
    openEdges        yes;
}

trimFeatures
{
    minElem          0;
    minLen           0;
}

writeObj             yes;

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

TU Delft

openFOAM test case structure

- System - meshQualityDict:

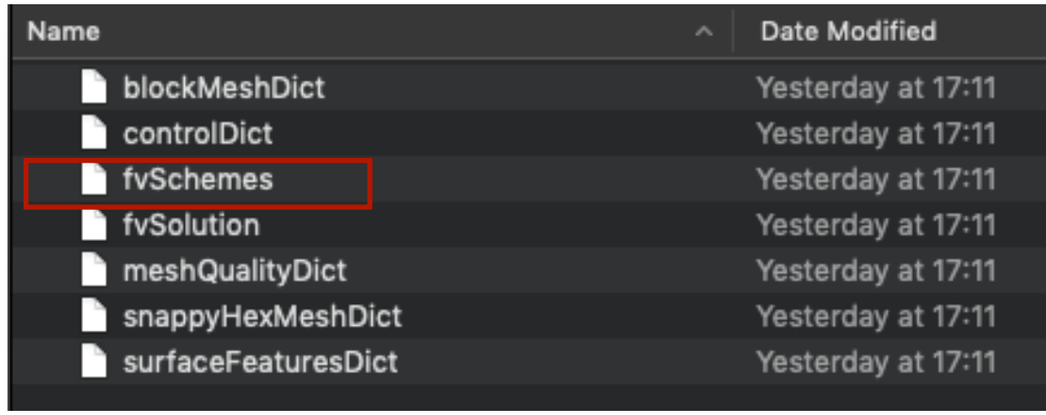



```
/*--------------------------------*- C++ -*----------------------------------*\
  =========                 |
  \\      /  F ield         | OpenFOAM: The Open Source CFD Toolbox
   \\    /   O peration     | Website:  https://openfoam.org
    \\  /    A nd           | Version:  7
     \\/     M anipulation  |
\*---------------------------------------------------------------------------*/
FoamFile
{
    version     2.0;
    format      ascii;
    class       dictionary;
    object      meshQualityDict;
}
// * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * *

#includeEtc "caseDicts/mesh/generation/meshQualityDict.cfg"

//- minFaceWeight (0 -> 0.5)
//minFaceWeight 0.02;

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

```
/*--------------------------------*- C++ -*----------------------------------*\
  =========                 |
  \\      /  F ield         | OpenFOAM: The Open Source CFD Toolbox
   \\    /   O peration     | Website:  https://openfoam.org
    \\  /    A nd           | Version:  7
     \\/     M anipulation  |
\*---------------------------------------------------------------------------*/
maxNonOrtho       65;

maxBoundarySkewness 20;

maxInternalSkewness 4;

maxConcave        80;

// Minimum cell pyramid volume; case dependent
minVol            1e-13;

minTetQuality     -1e30;

// if >0 : preserve single cells with all points on the surface if the
// resulting volume after snapping (by approximation) is larger than
// minVolCollapseRatio times old volume (i.e. not collapsed to flat cell).
//  If <0 : delete always.
//minVolCollapseRatio 0.5;

minArea           -1;

minTwist          0.02;

minDeterminant    0.001;

minFaceWeight     0.05;

minVolRatio       0.01;
```

openFOAM test case structure

- System - fvSchemes:



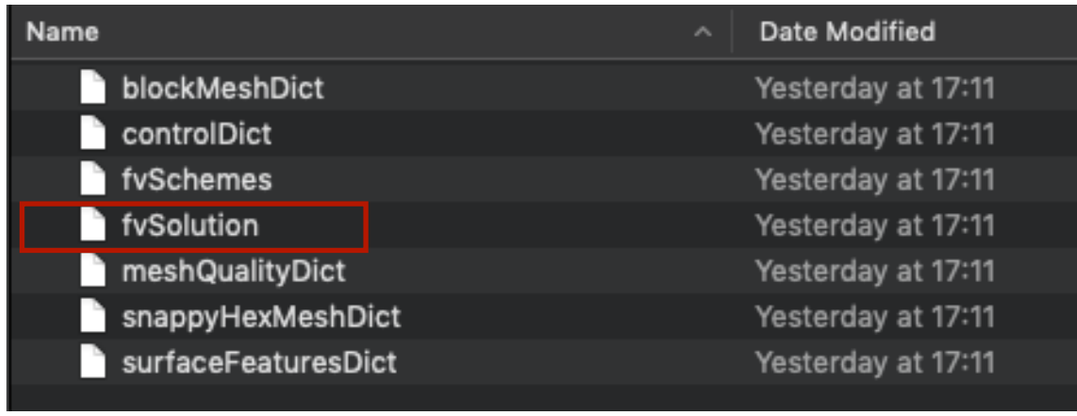

```
/*--------------------------------*- C++ -*----------------------------------*\
  =========                 |
  \\      /  F ield         | OpenFOAM: The Open Source CFD Toolbox
   \\    /   O peration     | Website:  https://openfoam.org
    \\  /    A nd           | Version:  7
     \\/     M anipulation  |
\*---------------------------------------------------------------------------*/
FoamFile
{
    version     2.0;
    format      ascii;
    class       dictionary;
    object      fvSchemes;
}
// * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * //

ddtSchemes
{
    default         steadyState;
}

gradSchemes
{
    default         Gauss linear;

    limited         cellLimited Gauss linear 1;
    grad(U)         $limited;
    grad(k)         $limited;
    grad(epsilon)     $limited;
}

divSchemes
{
    default         none;

    div(phi,U)      bounded Gauss linearUpwind limited;
```

```
laplacianSchemes
{
    default         Gauss linear corrected;
}

interpolationSchemes
{
    default         linear;
}

snGradSchemes
{
    default         corrected;
}

wallDist
{
    method meshWave;
}

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

TU Delft

openFOAM test case structure

- System - fvSolution:



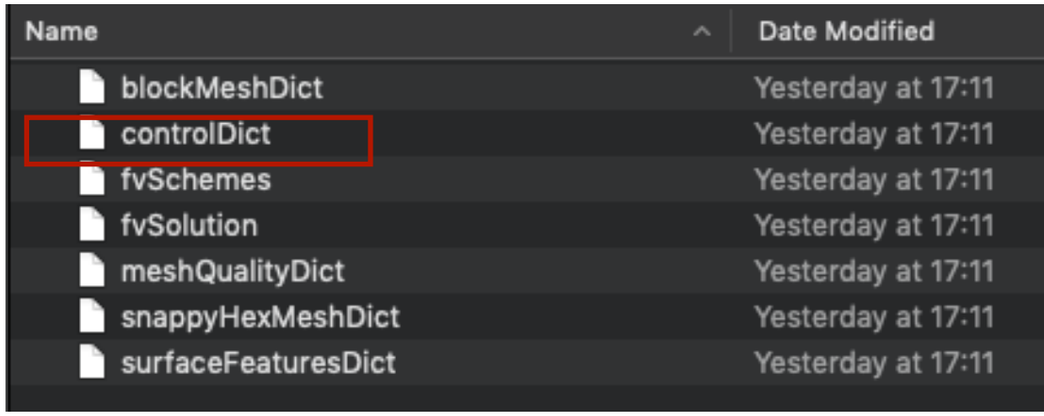

```
/*--------------------------------*- C++ -*----------------------------------*\
  =========                 |
  \\      /  F ield         | OpenFOAM: The Open Source CFD Toolbox
   \\    /   O peration     | Website:  https://openfoam.org
    \\  /    A nd           | Version:  7
     \\/     M anipulation  |
\*---------------------------------------------------------------------------*/
FoamFile
{
    version     2.0;
    format      ascii;
    class       dictionary;
    object      fvSolution;
}
// * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * //

solvers
{
    p
    {
        solver          GAMG;
        smoother        GaussSeidel;
        tolerance       1e-6;
        relTol          0.1;
    }

    "(U|k|omega|epsilon)"
    {
        solver          smoothSolver;
        smoother        symGaussSeidel;
        tolerance       1e-6;
        relTol          0.1;
    }
}
```

```
SIMPLE
{
    residualControl
    {
        p               1e-4;
        U               1e-4;
        "(k|omega|epsilon)" 1e-4;
    }
    nNonOrthogonalCorrectors 0;
    pRefCell        0;
    pRefValue       0;

}

potentialFlow
{
    nNonOrthogonalCorrectors 10;
}

relaxationFactors
{
    fields
    {
        p               0.3;
    }
    equations
    {
        U               0.7;
        "(k|omega|epsilon).*" 0.7;
    }
}

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

TU Delft

openFOAM test case structure

- System - controlDict:



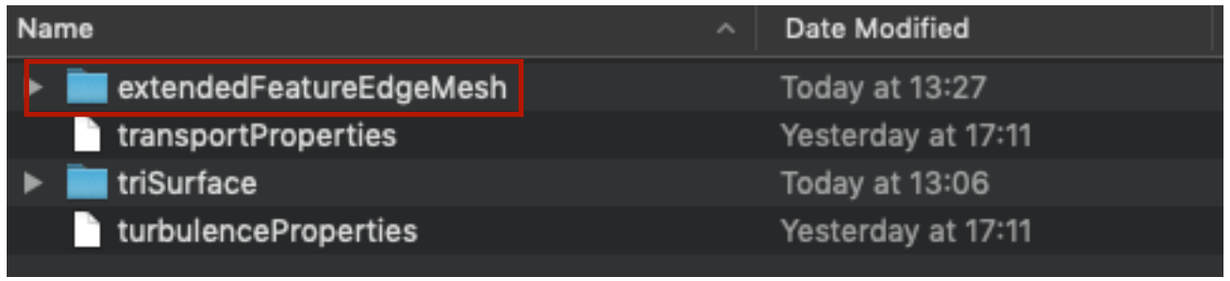

```
/*--------------------------------*- C++ -*----------------------------------*\
  =========                 |
  \\      /  F ield         | OpenFOAM: The Open Source CFD Toolbox
   \\    /   O peration     | Website:  https://openfoam.org
    \\  /    A nd           | Version:  7
     \\/     M anipulation  |
\*---------------------------------------------------------------------------*/
FoamFile
{
    version     2.0;
    format      ascii;
    class       dictionary;
    object      controlDict;
}
// * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * //

application     simpleFoam;

startFrom       latestTime;

startTime       0;

stopAt          endTime;

endTime         400;

deltaT          1;

writeControl    timeStep;

writeInterval   50;

purgeWrite      0;

writeFormat     ascii;
```

```
writePrecision  6;

writeCompression off;

timeFormat      general;

timePrecision   6;

runTimeModifiable true;

// adjustTimeStep  yes;

// maxCo           1;

functions
{
}
```

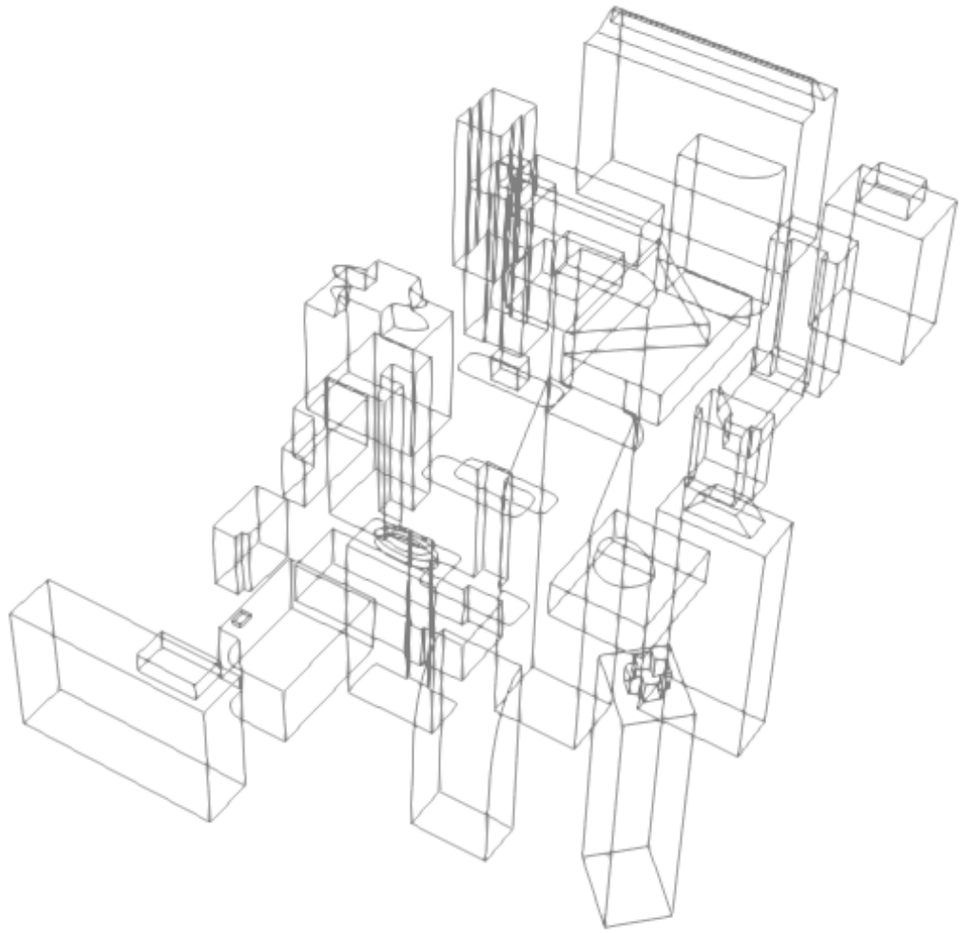
TU Delft

How to run openFOAM test case:

1. surfaceFeatures
2. blockMesh
3. snappyHexMesh
4. simpleFoam



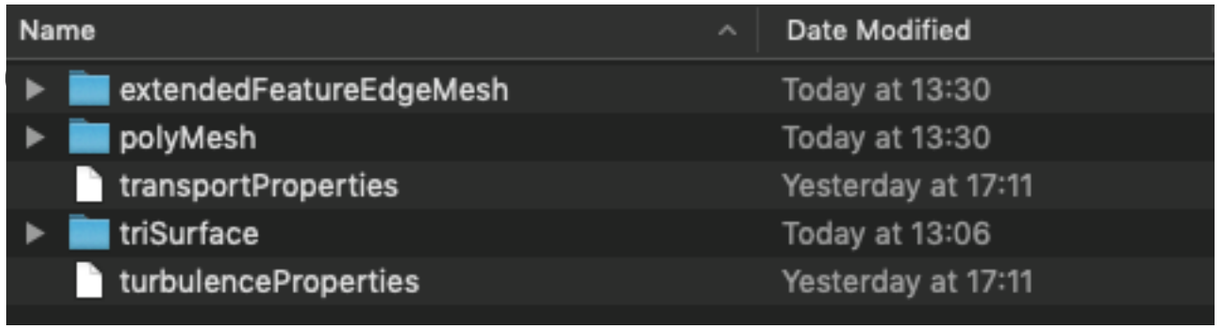
With this command we can extract the edges, which can be used within snappyHexMesh, to help the snapping process and improve the quality of the mesh generation

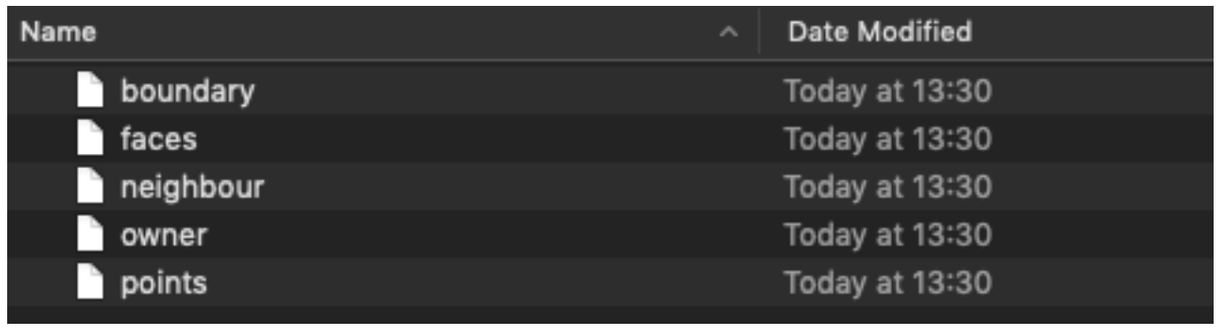


TU Delft

How to run openFOAM test

1. surfaceFeatures
2. **blockMesh**
3. snappyHexMesh
4. simpleFoam



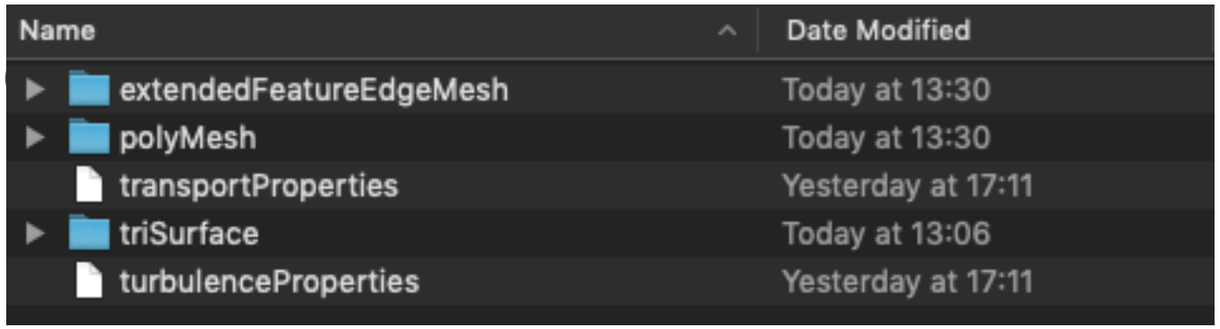


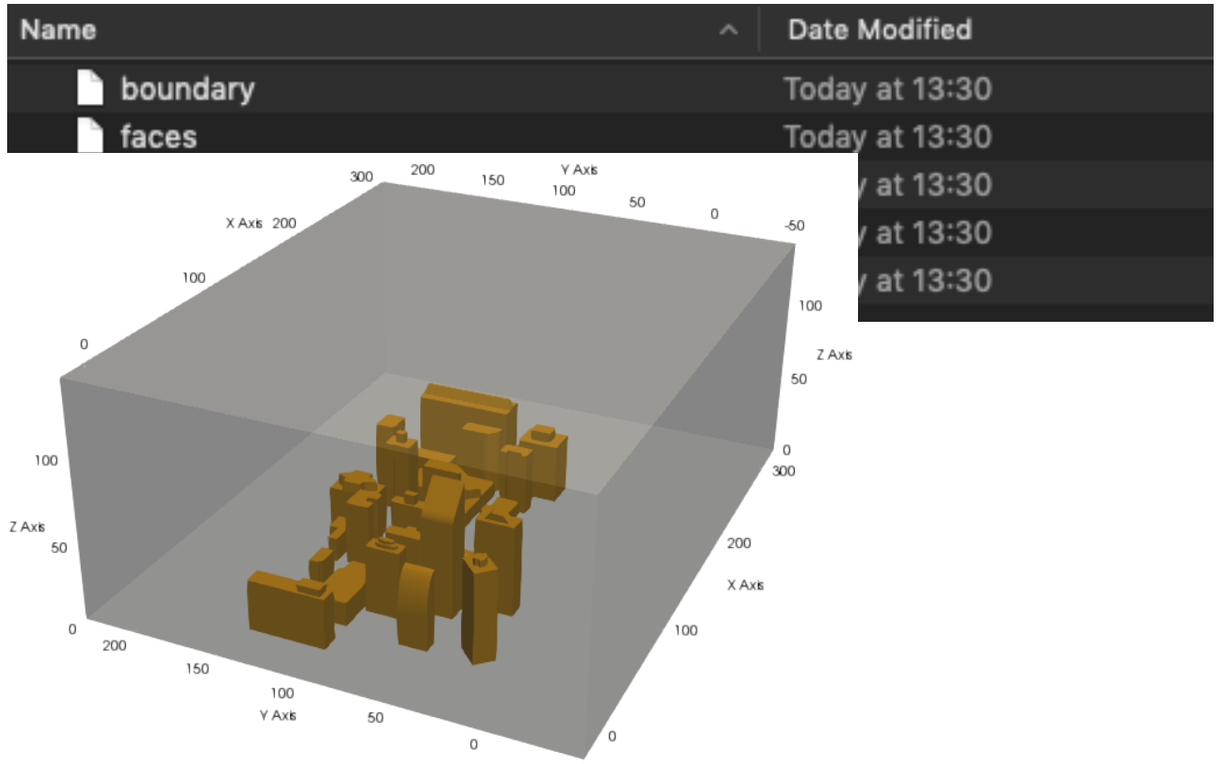

```
// * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * //

4
(
    inlet
    {
        type            patch;
        nFaces          200;
        startFace       14050;
    }
    outlet
    {
        type            patch;
        nFaces          200;
        startFace       14250;
    }
    ground
    {
        type            wall;
        inGroups        List<word> 1(wall);
        nFaces          500;
        startFace       14450;
    }
    frontAndBack
    {
        type            symmetry;
        inGroups        List<word> 1(symmetry);
        nFaces          1000;
        startFace       14950;
    }
)

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

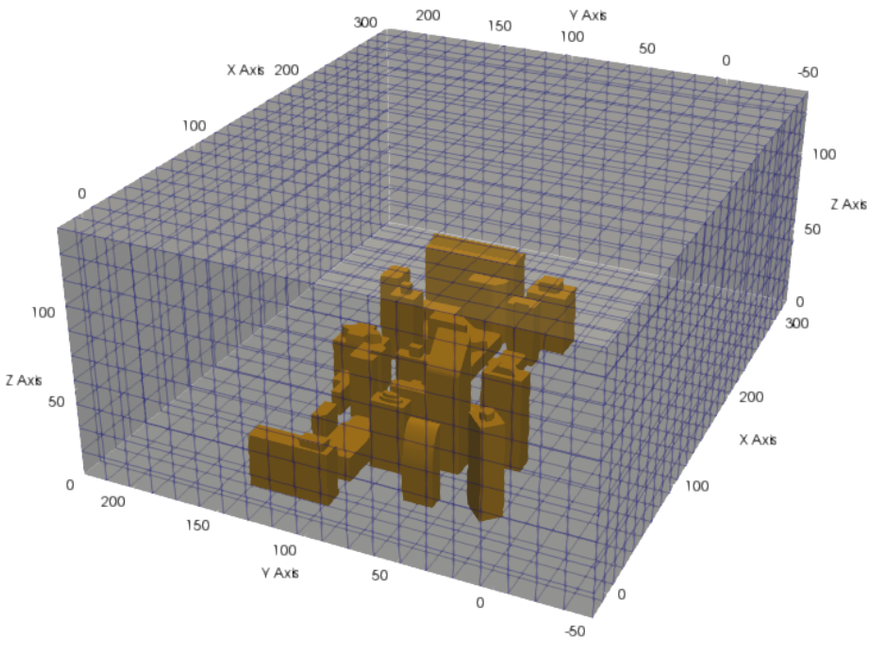
TU Delft

How to run openFOAM test

1. surfaceFeatures
2. **blockMesh**
3. snappyHexMesh
4. simpleFoam





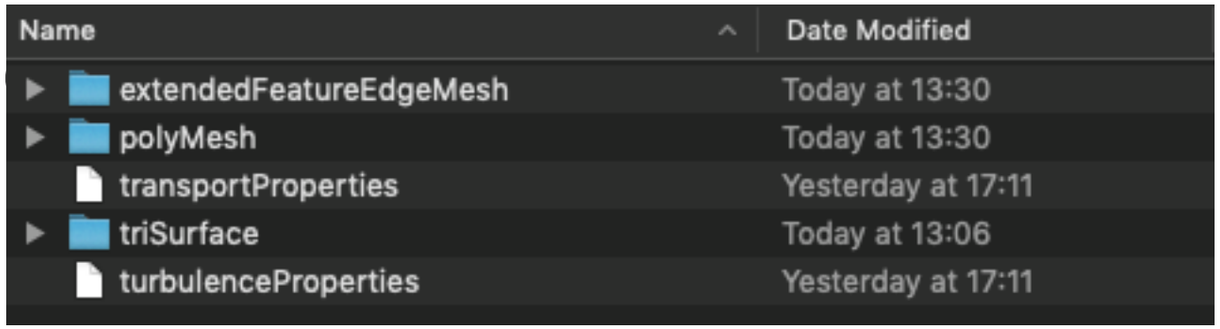


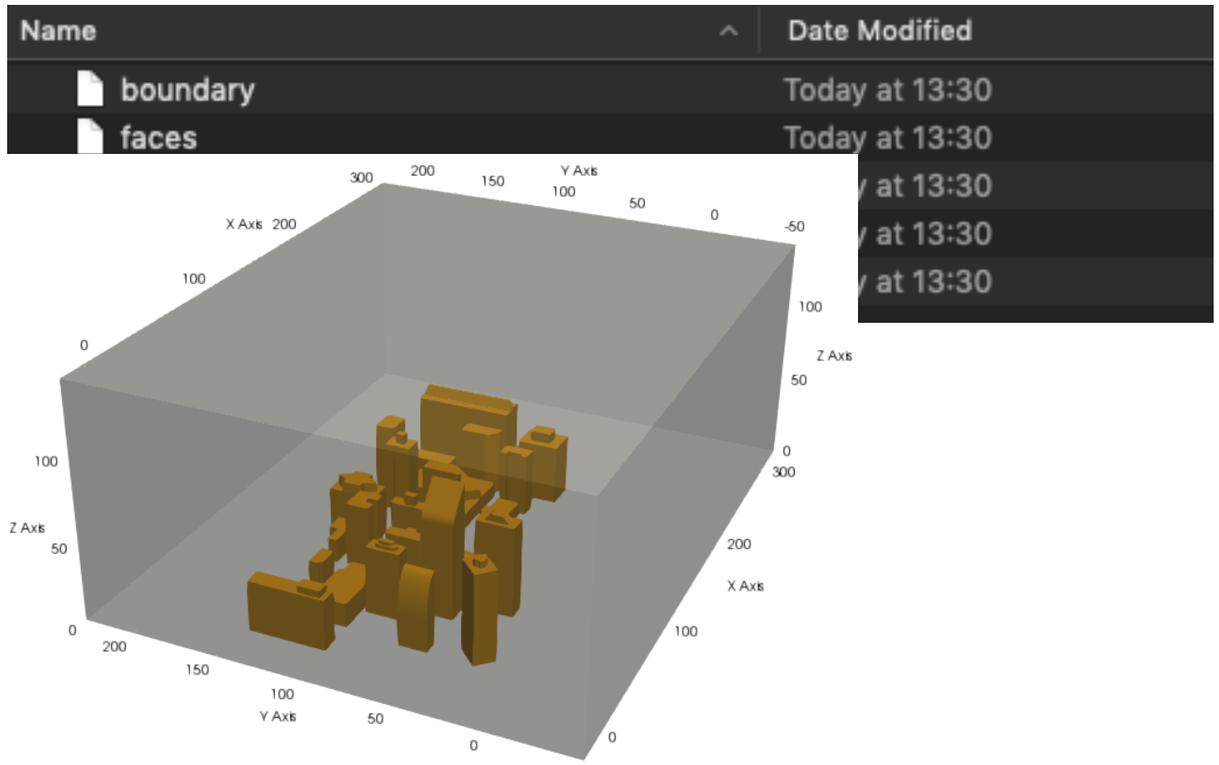

```
// * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * //

4
(
    inlet
    {
        type            patch;
        nFaces          200;
        startFace       14050;
    }
    outlet
    {
        type            patch;
        nFaces          200;
        startFace       14250;
    }
    ground
    {
        type            wall;
        inGroups        List<word> 1(wall);
        nFaces          500;
        startFace       14450;
    }
    frontAndBack
    {
        type            symmetry;
        inGroups        List<word> 1(symmetry);
        nFaces          1000;
        startFace       14950;
    }
)

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

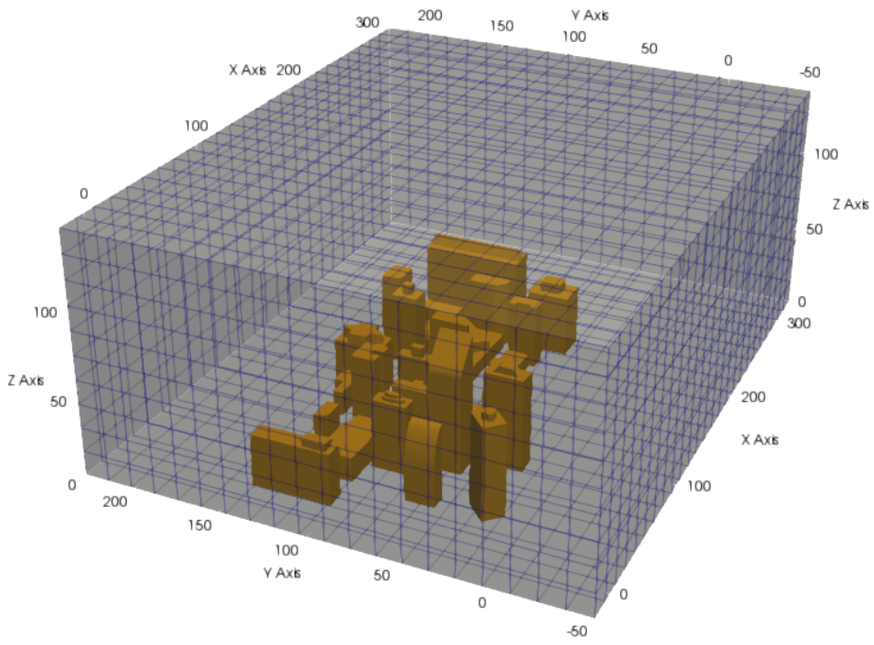
TU Delft

How to run openFOAM test

1. surfaceFeatures
2. **blockMesh**
3. snappyHexMesh
4. simpleFoam



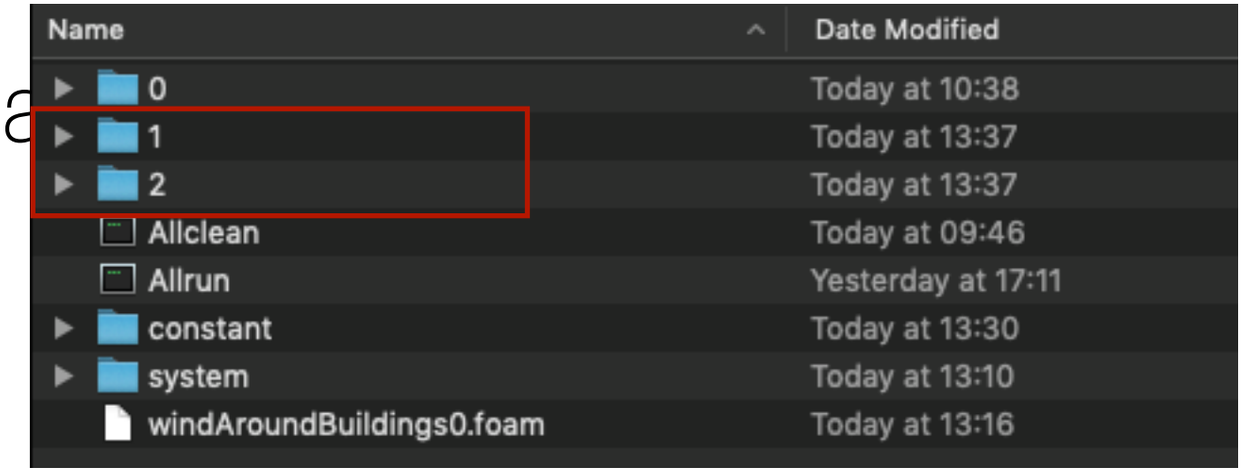





```
// * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * //

4
(
    inlet
    {
        type            patch;
        nFaces          200;
        startFace       14050;
    }
    outlet
    {
        type            patch;
        nFaces          200;
        startFace       14250;
    }
    ground
    {
        type            wall;
        inGroups        List<word> 1(wall);
        nFaces          500;
        startFace       14450;
    }
    frontAndBack
    {
        type            symmetry;
        inGroups        List<word> 1(symmetry);
        nFaces          1000;
        startFace       14950;
    }
)

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

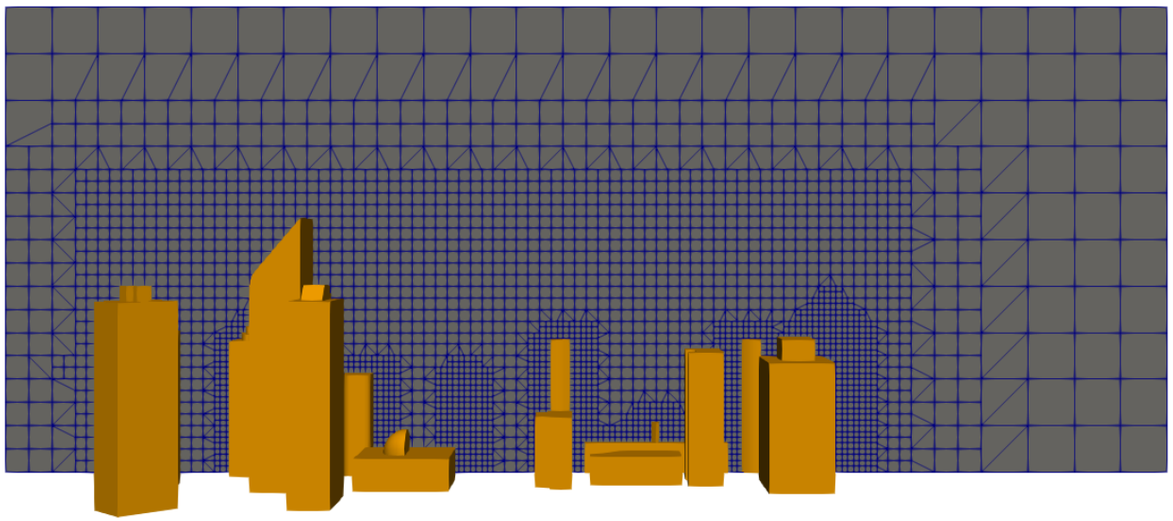
```
touch windAroundBuildings0.foam
```

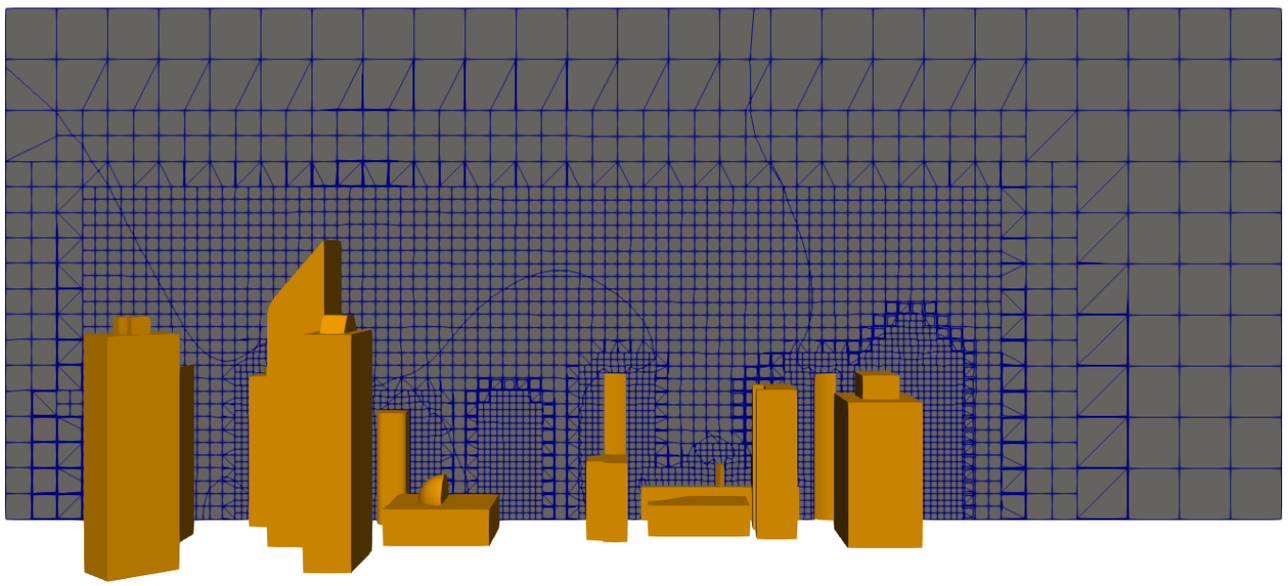
How to run openFOAM test ca



1. surfaceFeatures
2. blockMesh
3. **snappyHexMesh**
4. simpleFoam

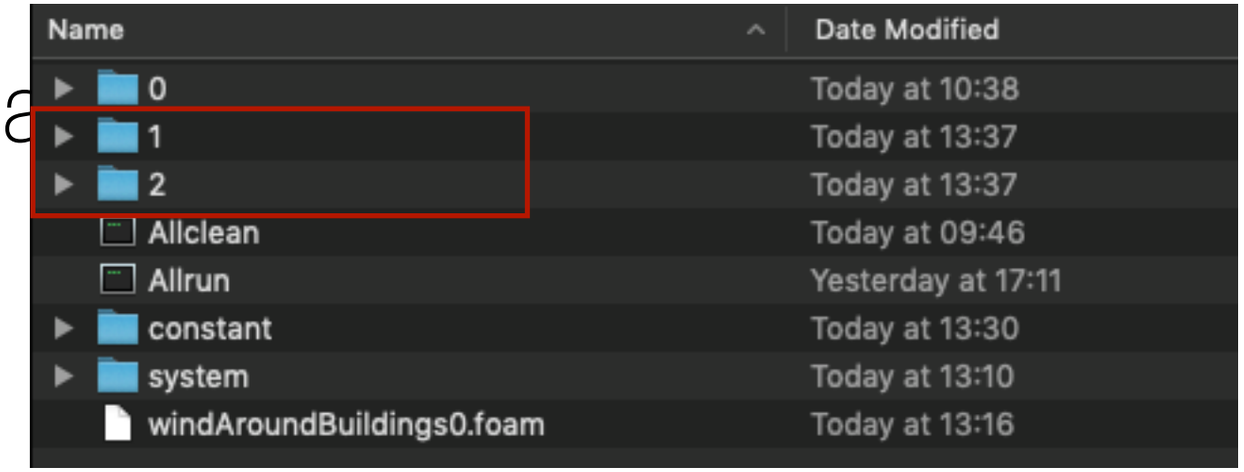
```
#includeEtc "caseDicts/mesh/generation/snappyHexMeshDict.cfg"

castellatedMesh on;
snap            on;
addLayers       off;
```





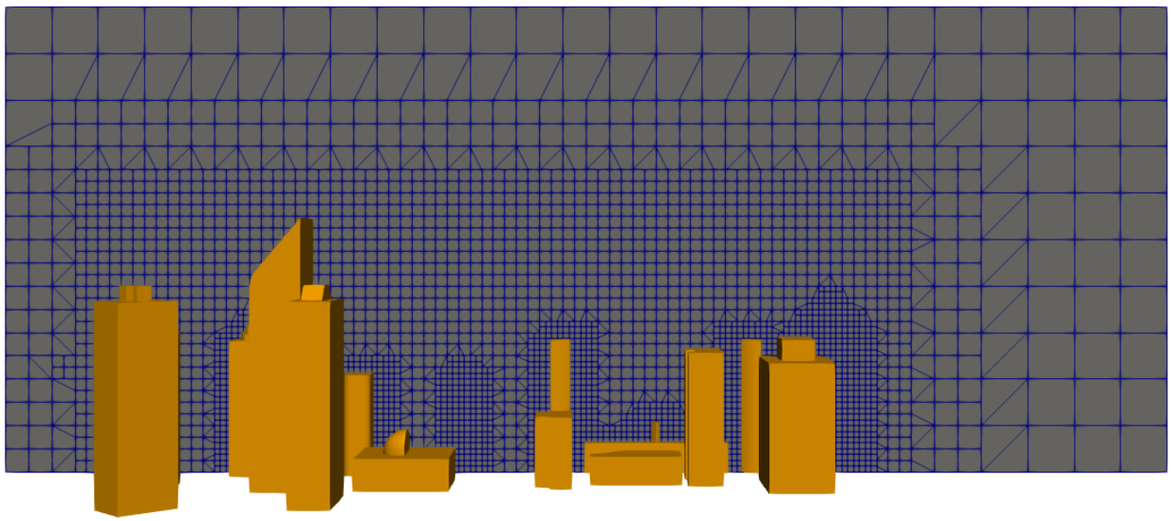
TU Delft

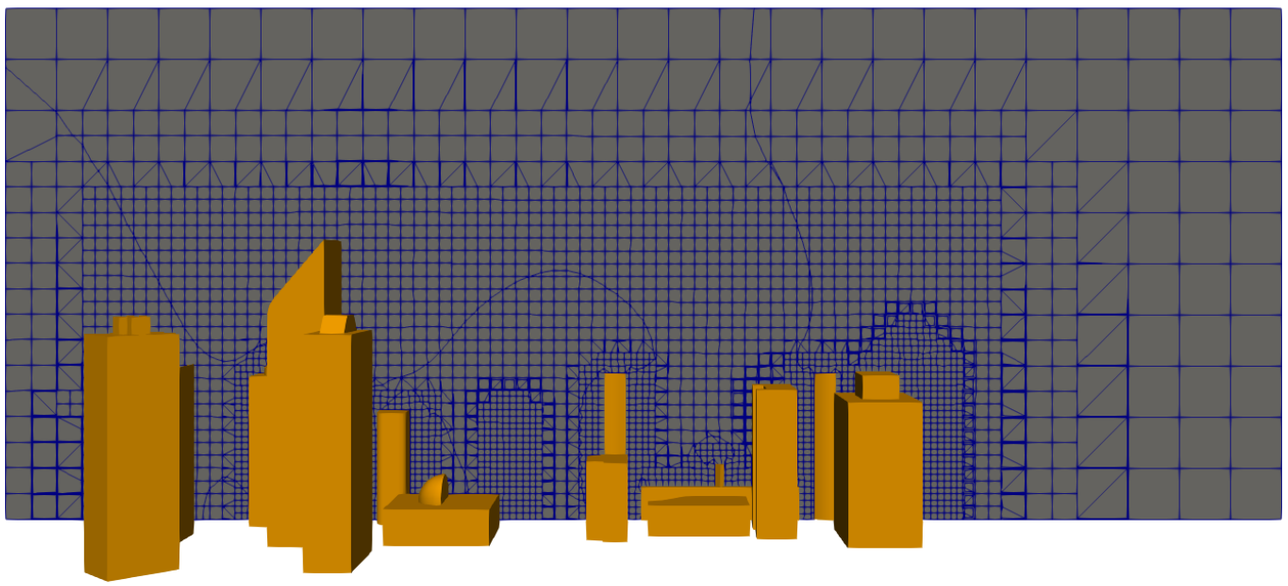
How to run openFOAM test ca



1. surfaceFeatures
2. blockMesh
3. **snappyHexMesh**
4. simpleFoam

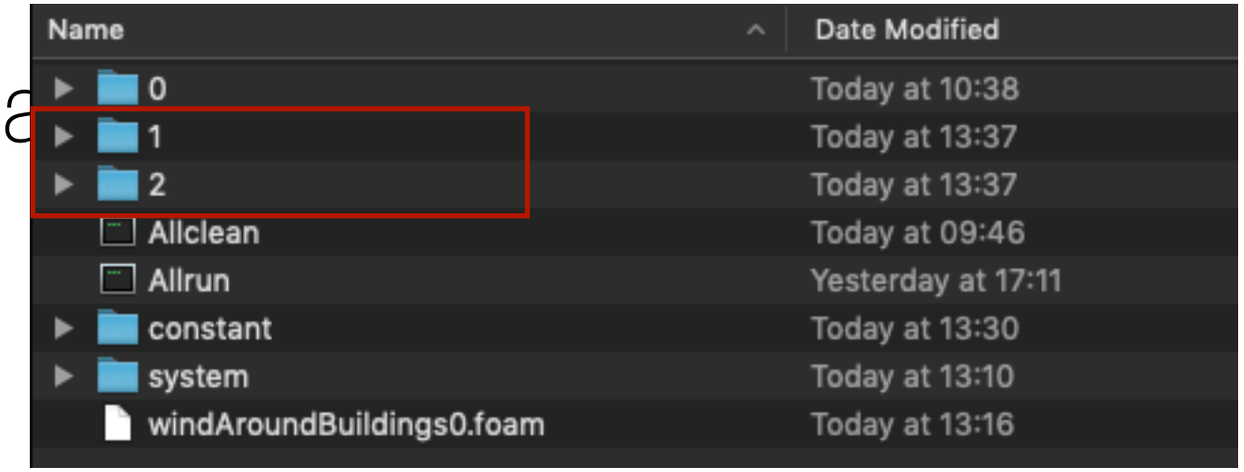
```
#includeEtc "caseDicts/mesh/generation/snappyHexMeshDict.cfg"

1. castellatedMesh on;
2. snap            on;
   addLayers       off;
```





TU Delft

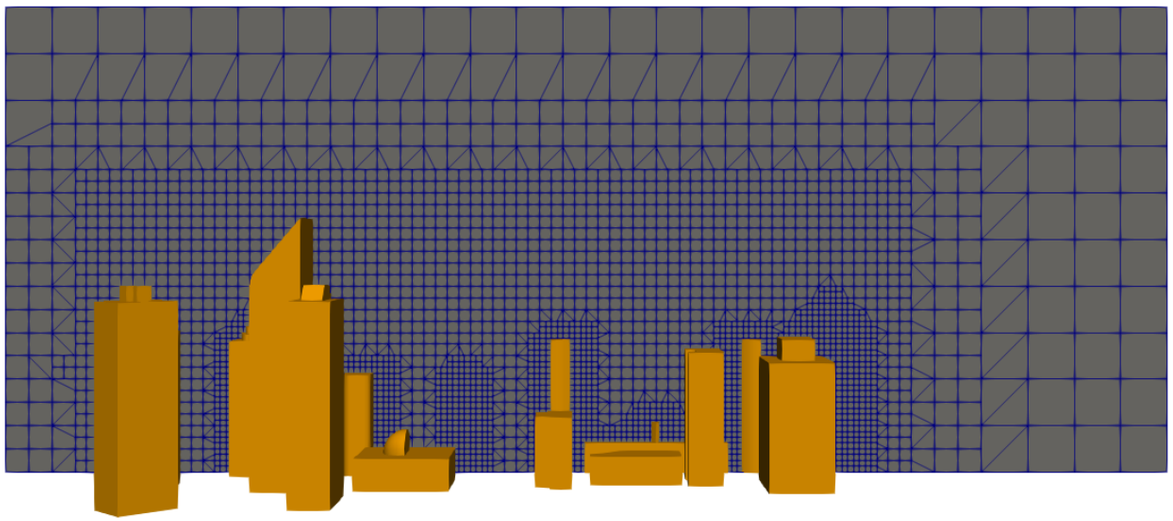
How to run openFOAM test ca

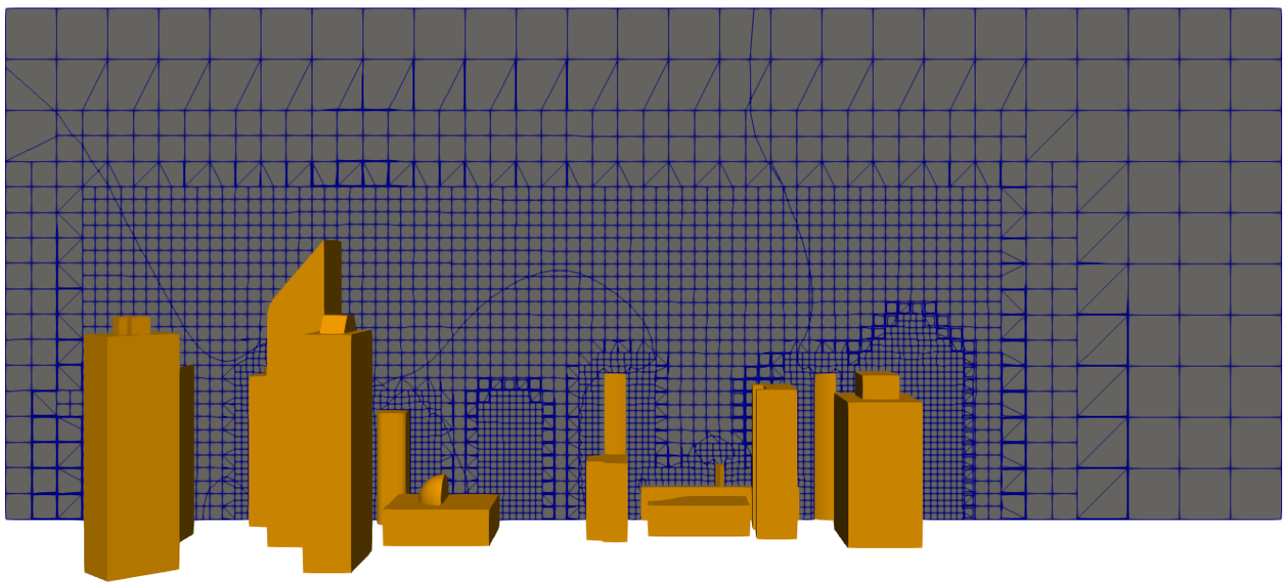


1. surfaceFeatures
2. blockMesh
3. **snappyHexMesh**
4. simpleFoam

```
#includeEtc "caseDicts/mesh/generation/snappyHexMeshDict.cfg"

1. castellatedMesh on;
2. snap            on;
   addLayers       off;
```

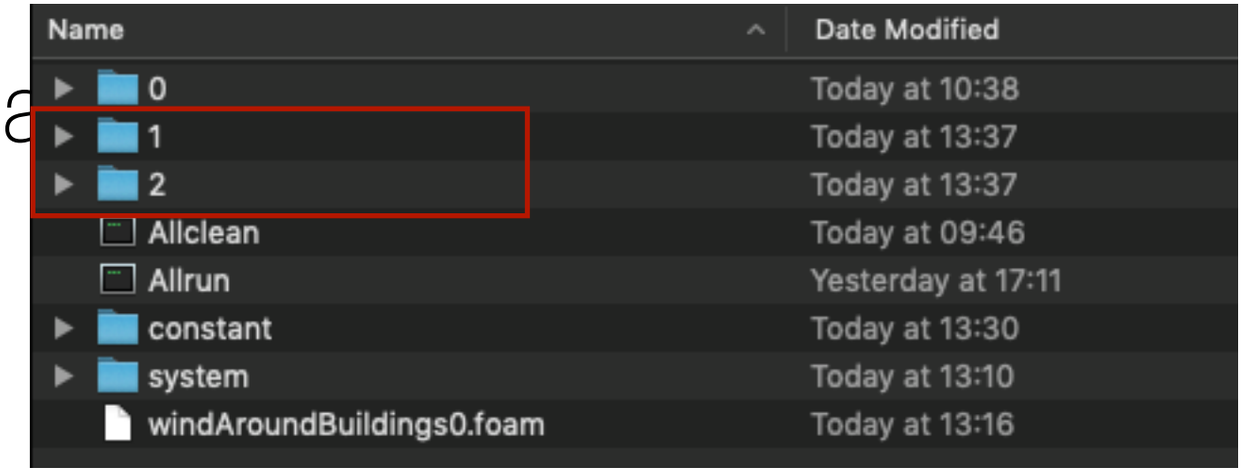
1. Castellation





TU Delft

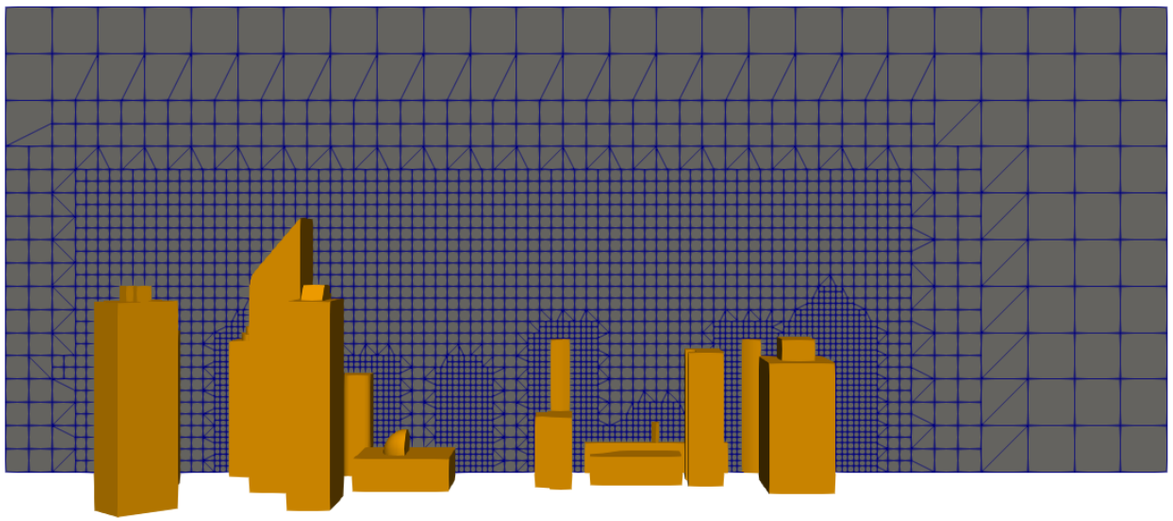
How to run openFOAM test ca



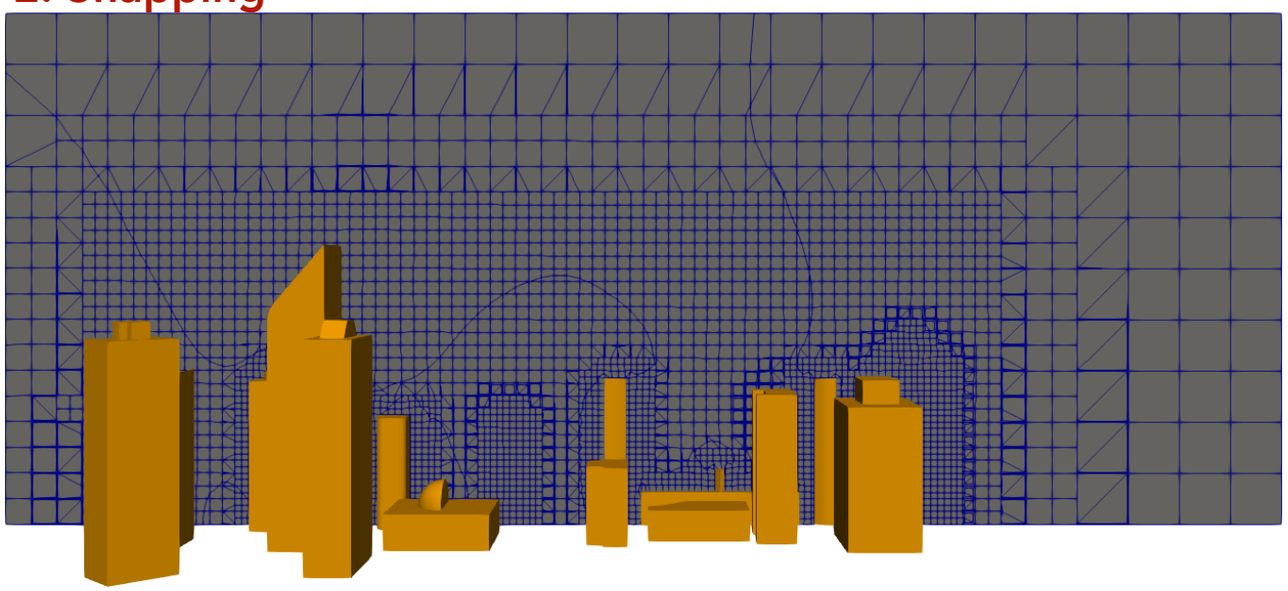
1. surfaceFeatures
2. blockMesh
3. **snappyHexMesh**
4. simpleFoam

```
#includeEtc "caseDicts/mesh/generation/snappyHexMeshDict.cfg"

1. castellatedMesh on;
2. snap            on;
   addLayers       off;
```

1. Castellation



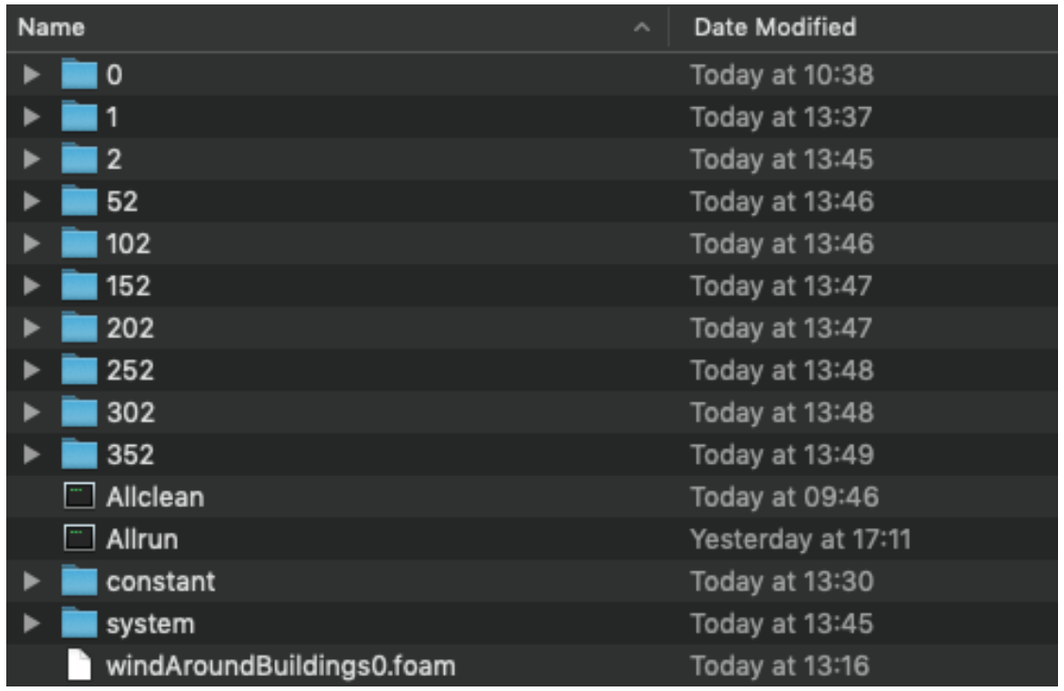
2. Snapping



TU Delft

How to run openFOAM test case:

1. surfaceFeatures
2. blockMesh
3. snappyHexMesh
4. **simpleFoam**




```
SIMPLE
{
    residualControl
    {
        p               1e-4;
        U               1e-4;
        "(k|omega|epsilon)" 1e-4;
    }
    nNonOrthogonalCorrectors 0;
    pRefCell        0;
    pRefValue       0;

}
```

```
Time = 399

smoothSolver:  Solving for Ux, Initial residual = 0.000374315, Final residual = 2.13748e-05, No Iterations 2
smoothSolver:  Solving for Uy, Initial residual = 0.00149301, Final residual = 8.74602e-05, No Iterations 2
smoothSolver:  Solving for Uz, Initial residual = 0.00168795, Final residual = 0.000102205, No Iterations 2
GAMG:  Solving for p, Initial residual = 0.0132326, Final residual = 0.000431114, No Iterations 2
time step continuity errors : sum local = 1.12228e-05, global = 1.23819e-06, cumulative = 0.00416481
smoothSolver:  Solving for epsilon, Initial residual = 0.00286634, Final residual = 0.000159461, No Iterations 1
smoothSolver:  Solving for k, Initial residual = 0.00206614, Final residual = 0.000165122, No Iterations 1
ExecutionTime = 241.25 s  ClockTime = 247 s

Time = 400

smoothSolver:  Solving for Ux, Initial residual = 0.000373325, Final residual = 2.13718e-05, No Iterations 2
smoothSolver:  Solving for Uy, Initial residual = 0.00148155, Final residual = 8.70915e-05, No Iterations 2
smoothSolver:  Solving for Uz, Initial residual = 0.00167385, Final residual = 0.000101258, No Iterations 2
GAMG:  Solving for p, Initial residual = 0.0132477, Final residual = 0.000427785, No Iterations 2
time step continuity errors : sum local = 1.11348e-05, global = 1.15567e-06, cumulative = 0.00416596
smoothSolver:  Solving for epsilon, Initial residual = 0.00288552, Final residual = 0.000160044, No Iterations 1
smoothSolver:  Solving for k, Initial residual = 0.00205397, Final residual = 0.000165605, No Iterations 1
ExecutionTime = 241.83 s  ClockTime = 248 s
```

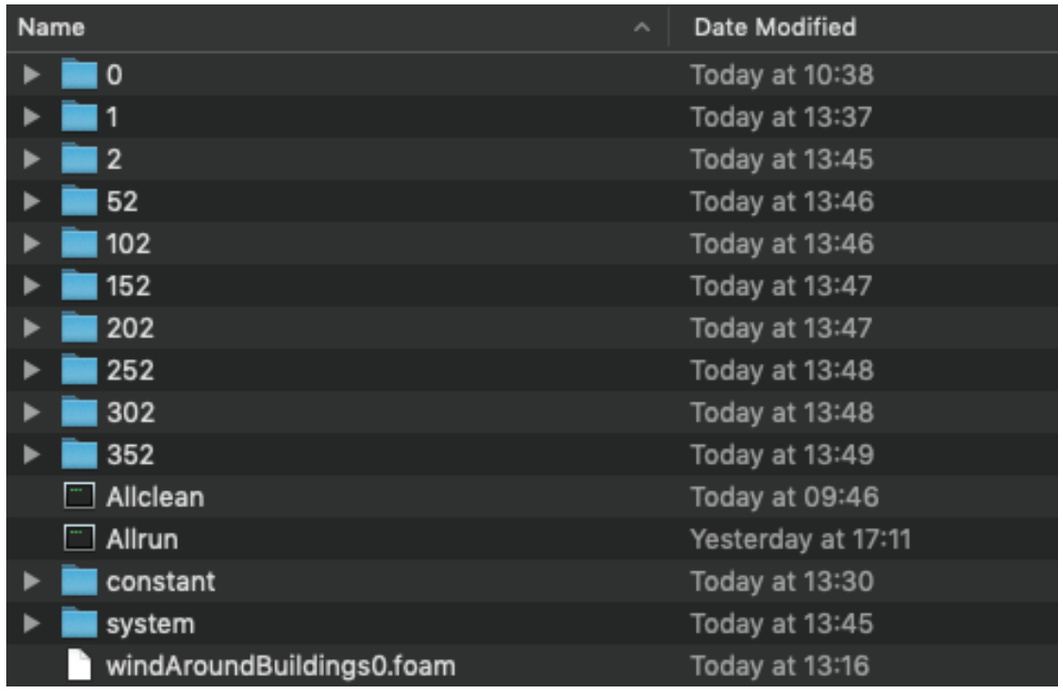
TU Delft

How to run openFOAM test case:

1. surfaceFeatures
2. blockMesh
3. snappyHexMesh
4. **simpleFoam**

```
SIMPLE
{
    residualControl
    {
        p               1e-4;
        U               1e-4;
        "(k|omega|epsilon)" 1e-4;
    }
    nNonOrthogonalCorrectors 0;
    pRefCell        0;
    pRefValue       0;

}
```

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

application     simpleFoam;

startFrom       latestTime;

startTime       0;

stopAt          endTime;

endTime         400;

deltaT          1;

writeControl    timeStep;

writeInterval   50;
```




```
Time = 399

smoothSolver:  Solving for Ux, Initial residual = 0.000374315, Final residual = 2.13748e-05, No Iterations 2
smoothSolver:  Solving for Uy, Initial residual = 0.00149301, Final residual = 8.74602e-05, No Iterations 2
smoothSolver:  Solving for Uz, Initial residual = 0.00168795, Final residual = 0.000102205, No Iterations 2
GAMG:  Solving for p, Initial residual = 0.0132326, Final residual = 0.000431114, No Iterations 2
time step continuity errors : sum local = 1.12228e-05, global = 1.23819e-06, cumulative = 0.00416481
smoothSolver:  Solving for epsilon, Initial residual = 0.00286634, Final residual = 0.000159461, No Iterations 1
smoothSolver:  Solving for k, Initial residual = 0.00206614, Final residual = 0.000165122, No Iterations 1
ExecutionTime = 241.25 s  ClockTime = 247 s

Time = 400

smoothSolver:  Solving for Ux, Initial residual = 0.000373325, Final residual = 2.13718e-05, No Iterations 2
smoothSolver:  Solving for Uy, Initial residual = 0.00148155, Final residual = 8.70915e-05, No Iterations 2
smoothSolver:  Solving for Uz, Initial residual = 0.00167385, Final residual = 0.000101258, No Iterations 2
GAMG:  Solving for p, Initial residual = 0.0132477, Final residual = 0.000427785, No Iterations 2
time step continuity errors : sum local = 1.11348e-05, global = 1.15567e-06, cumulative = 0.00416596
smoothSolver:  Solving for epsilon, Initial residual = 0.00288552, Final residual = 0.000160044, No Iterations 1
smoothSolver:  Solving for k, Initial residual = 0.00205397, Final residual = 0.000165605, No Iterations 1
ExecutionTime = 241.83 s  ClockTime = 248 s
```

How to run openFOAM test case:

1. surfaceFeatures
2. blockMesh
3. snappyHexMesh
4. **simpleFoam IN PARALLEL**

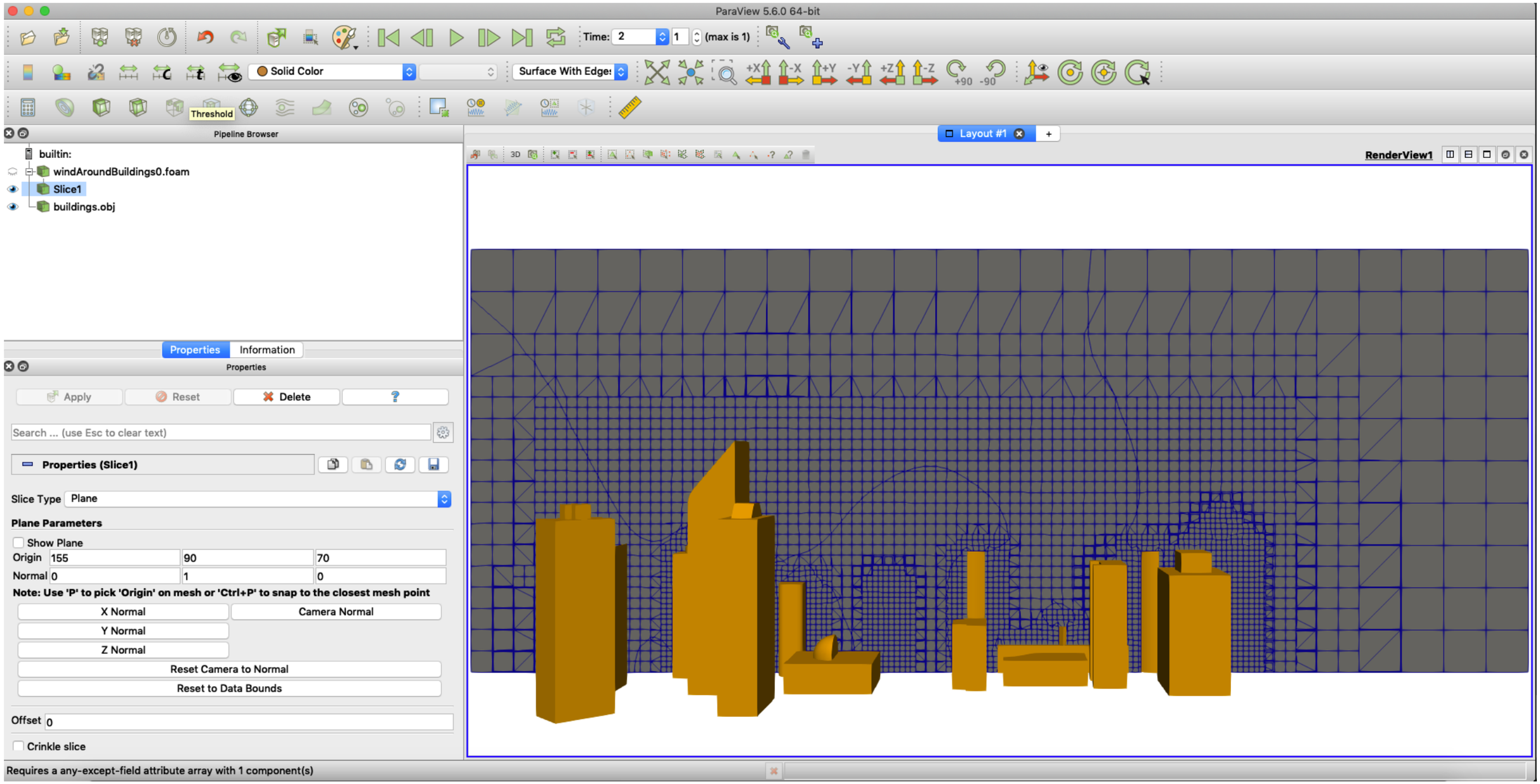
```
cp $FOAM_ETC/caseDicts/preProcessing/decomposeParDict system/
```

```
mpirun -np 4 simpleFoam -parallel >& log.SF &
```

```
tail -f log.SF
```

```
reconstructPar -latestTime
```

- Paraview:



https://www.paraview.org/Wiki/images/6/6e/ParaViewTutorial54.pdf

TU Delft

Activities before next class

- Go to: https://www.paraview.org/Wiki/images/6/6e/ParaViewTutorial54.pdf
- Make the different plots suggested in the tutorial
- Run the case switching off turbulence
- Run the case increasing the **v** (viscosity) to honey level (2000) or oil level (56.2)
- What happens if you change endTime in controlDict