GROBDESIGN

Tutorial 4

CFD ANALYSIS OF WATER FALLING ON CYLINDRICAL SURFACE USING INTERFOAM SOLVER

GROBDESIGN PVT LTD, JAIPUR **2018**

TUTORIAL 04

Title: CFD analysis of water falling on cylindrical surface using interfoam solver.

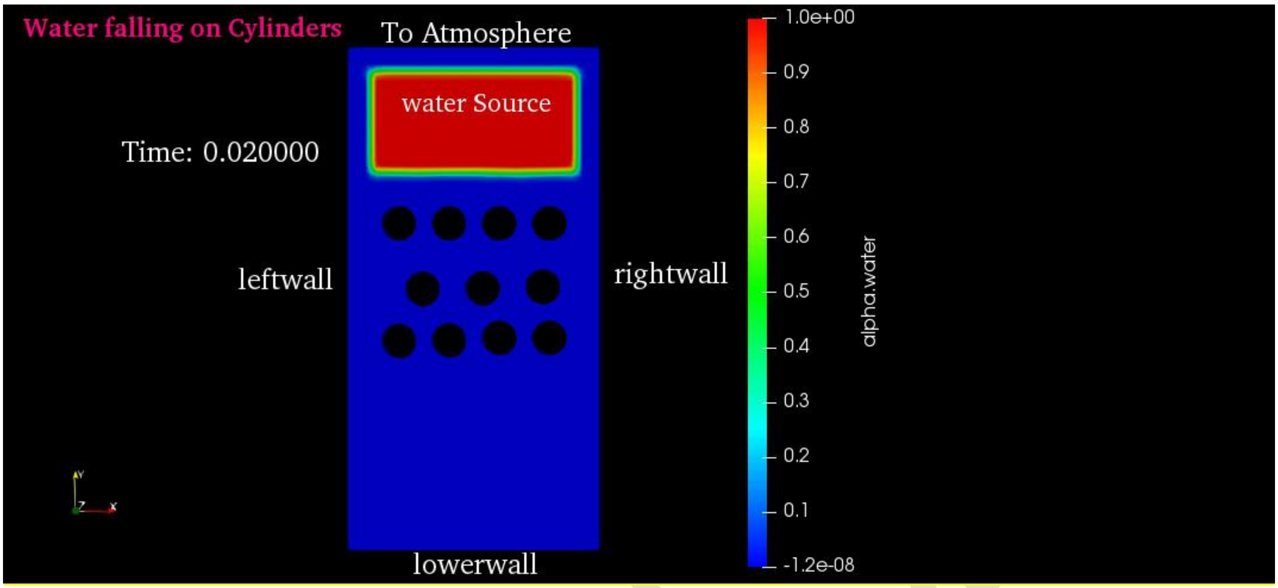


Figure 4.1 (a) Initial volume fraction (VF) of water Contour of free water falling with BCs

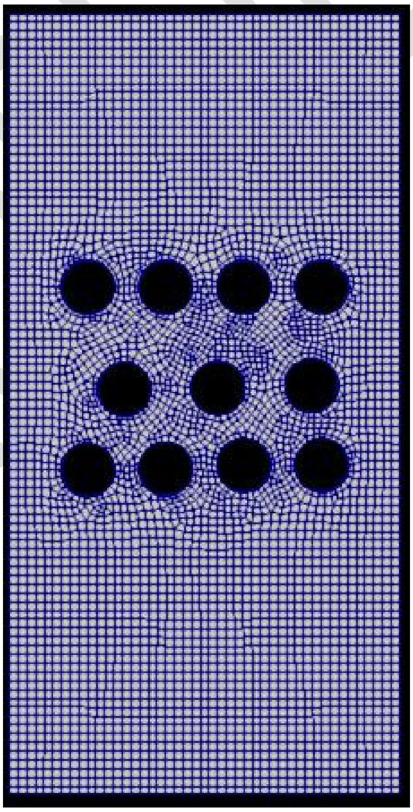


Figure 4.1 (b) Meshed domain of water falling

Problem Identification

In this problem investigation of free water falling using OPENFOAM is proposed for multi phase modeling (free water falling cooling), in which water is flow in free water falling from prescribed location section from top initial condition (see figure), which is assumed at surface of top side wall (See the following figure). Air and water properties are selected from literature available in digital medium. Outlet is at top of the domain which is selected for continuity effect of free water falling system. Some assumptions are applied in this problem like initial room temperature is assumed at constant value for this problem (ISOTHERMAL). Air and water properties are also assumed constant for this problem.

"interFoam" is selected as solver for this problem. Open Foam software is installed on Win 7, provided by **FSD blueCAPE Lda: http://bluecfd.com/**

Note: This tutorial is not endorsed/ supported by provider of software's which are used in this tutorial. This tutorial is made for educational purpose only.

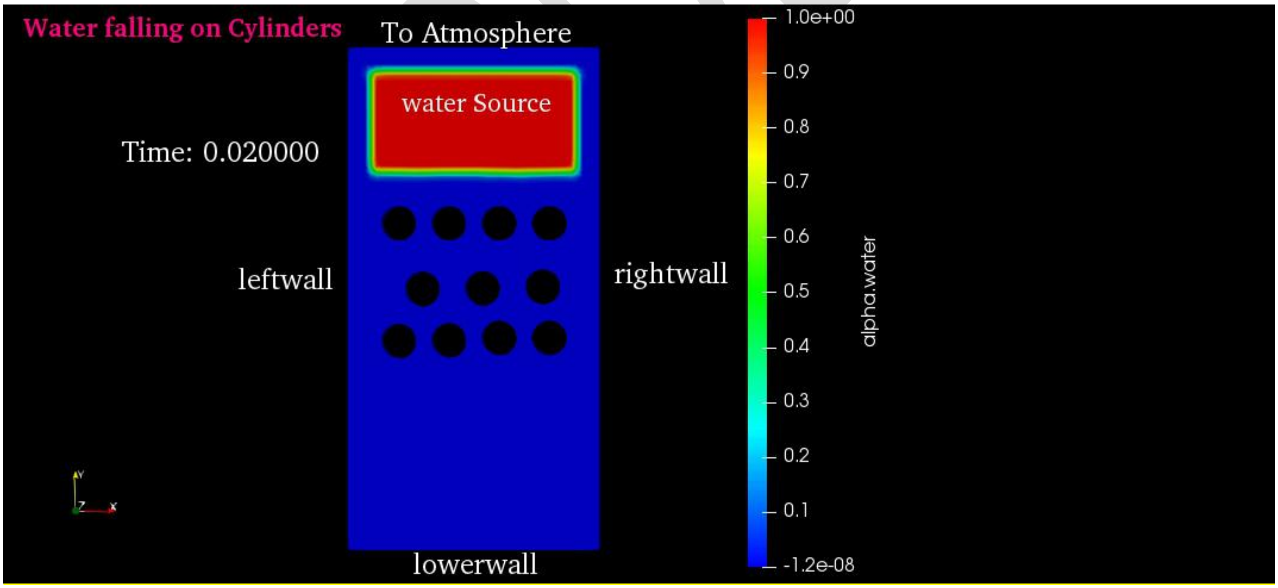


Figure4.2 Different Boundary Conditions

The simulation is solved for ambient conditions at transient flow scheme available in selected solver.

Pre-Processing

This problem is selected as laminar flow problem, so the solver must have capability to solve laminar flow conditions, that's why **"interFoam"** is selected as solver. The very first step is import suitable mesh file in open-foam software. There are various methods available in software to read mesh file, but we use **"fluentMeshToFoam"** command to read *.msh file generated by **Ansys Fluent®** software. It is required to make a folder in home directory (Folder Name **wflow**) of your open-foam software, where important sub-folders are already presented (see the following figure). **At starting only three folders are availble 0, constant and system, which is copied from open foam tutorial. In this case folders are copied from interFoam tutorials.**

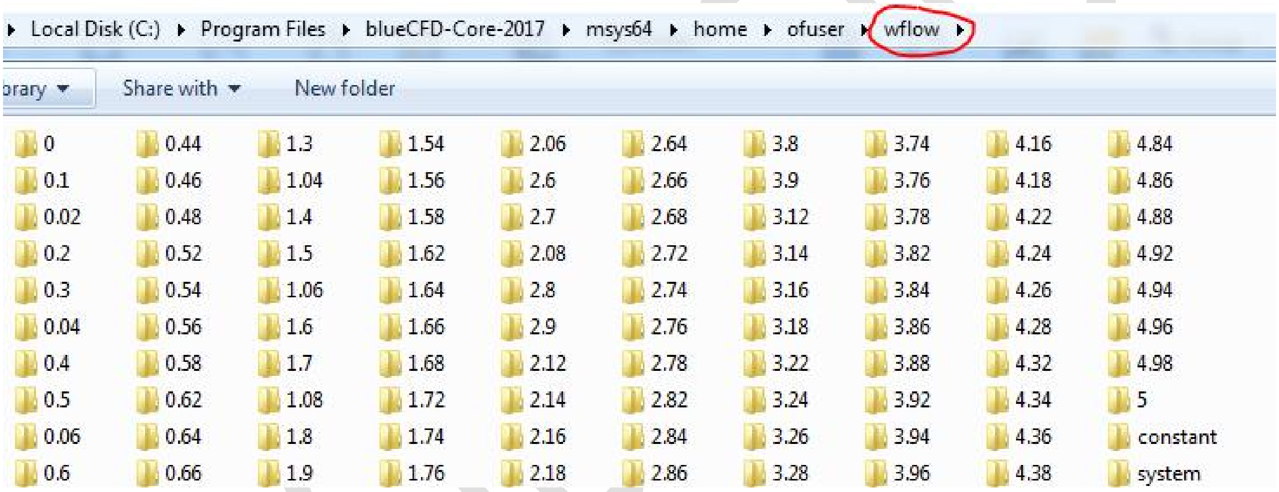


Figure 4.3 Folder in home directory for this tutorial

After desired folder creation, run open-foam software (see the following figure)

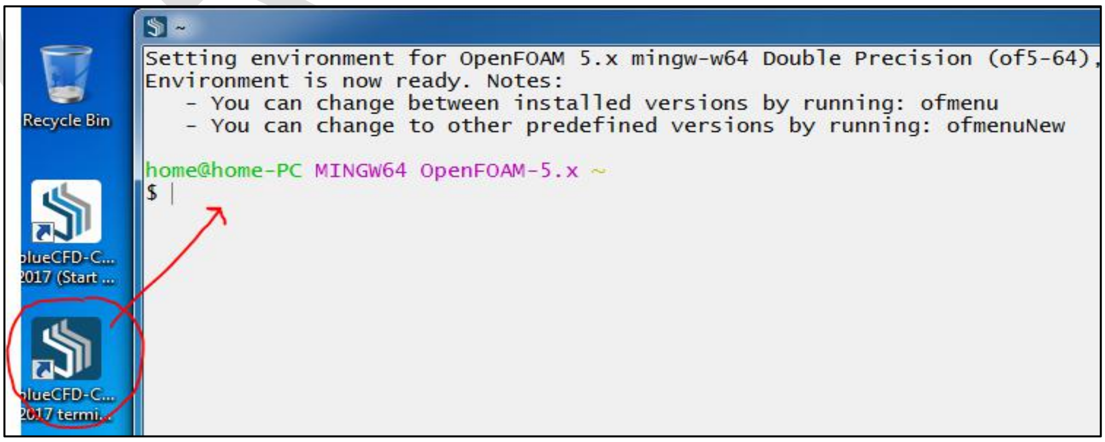


Figure 4.4 Run Open-Foam using blueCFD terminal

Run **"ls"** command in this terminal to verify the user created folder (see the following figure)

```
home@home-PC MINGW64 OpenFOAM-5.x ~
$ ls
AddOns       chamber1          defaultmpi.sh   fire      jitusah      parking01  README.TXT  sourceOF   waterdam
basement1    chamber3D         elbow_tutorial  first     mountBlueCFD pipebend2  room1       spiral     wflow
blueCFD      changedefmpi.bat  ELBOW01         furnace   MultiRegion  platehe    sah         throttle   wflowturb
bubble       damBreak          elbubble        he1       myIcoFoam    react      sah2        vsolver
bubbleColumn datacenter        extendsurf      helical   openflow     reaction   sah3        wallBoiling
```

Figure 4.5 ls command to check the folders in home directory

After verification of user folder (wflow), run **"cd"** command to enter in this folder (see the following figure)

```
home@home-PC MINGW64 OpenFOAM-5.x ~
$ cd wflow
```

Figure 4.6 cd command to enter desired folder

Run **"fluentMeshToFoam"** command in terminal (see the following figure), it must be ensure by user that msh file is available in main folder by run command **"ls"** again, in this tutorial **wflow.msh** file is used (copy from tutorial available in Google drive link)

```
home@home-PC MINGW64 OpenFOAM-5.x ~/wflow
$ ls
0     0.32  0.64  0.96  1.28  1.6   1.92  2.24  2.56  2.88  3.2   3.52  3.84  4.16  4.48  4.8    waterflow.jpg
0.02  0.34  0.66  0.98  1.3   1.62  1.94  2.26  2.58  2.9   3.22  3.54  3.86  4.18  4.5   4.82   waterVF.avi
0.04  0.36  0.68  1     1.32  1.64  1.96  2.28  2.6   2.92  3.24  3.56  3.88  4.2   4.52  4.84   wflow.msh
0.06  0.38  0.7   1.02  1.34  1.66  1.98  2.3   2.62  2.94  3.26  3.58  3.9   4.22  4.54  4.86
0.08  0.4   0.72  1.04  1.36  1.68  2     2.32  2.64  2.96  3.28  3.6   3.92  4.24  4.56  4.88
0.1   0.42  0.74  1.06  1.38  1.7   2.02  2.34  2.66  2.98  3.3   3.62  3.94  4.26  4.58  4.9
0.12  0.44  0.76  1.08  1.4   1.72  2.04  2.36  2.68  3     3.32  3.64  3.96  4.28  4.6   4.92
```

Figure 4.7 ls command to verify msh file

```
home@home-PC MINGW64 OpenFOAM-5.x ~/wflow
$ fluentMeshToFoam wflow.msh
```

Figure 4.8 mesh command to convert msh file in poly mesh file

This command create a sub-folder named polymesh in sub-folder **constant** which we create in starting phase of problem. (see the following figure)

In **polymesh folder** various files are created. These files have geometry information of wflow.msh but our consideration is on **boundary file** (red highlighted in figure)

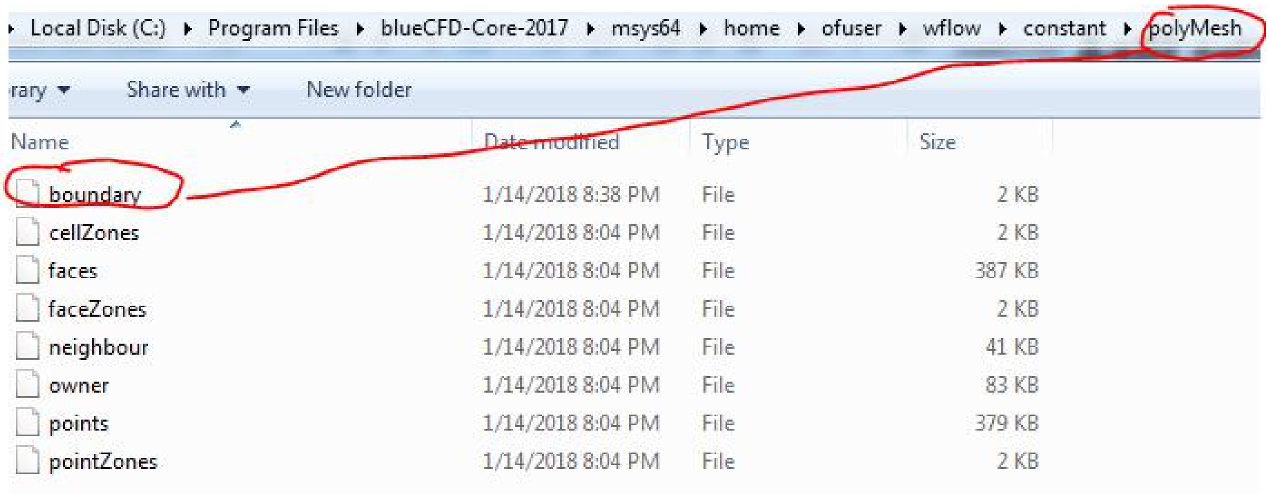


Figure 4.9 Polymesh folder created by **"fluentMeshToFoam"** command

Note down all boundary names created from msh file can be see by open the boundary file available in polymesh folder using **word-pad/notepad** (see the following figure)

```
FoamFile
{
    version     2.0;
    format      ascii;
    class       polyBoundaryMesh;
    location    "constant/polyMesh";
    object      boundary;
}
// * * * * * * * * * * * * * * * * * * * * //

6
(
    leftwall
    {
        type            wall;
        inGroups        1(wall);
        nFaces          89;
        startFace       8328;
    }
    rightwall
    {
        type            wall;
        inGroups        1(wall);
        nFaces          89;
        startFace       8417;
    }
    lowerwall
    {
        type            wall;
        inGroups        1(wall);
        nFaces          45;
        startFace       8506;
    }
    atmosphere
    {
        type            patch;
        nFaces          45;
        startFace       8551;
    }
    balls
    {
        type            wall;
        inGroups        1(wall);
        nFaces          330;
        startFace       8596;
    }
    frontAndBackPlanes
    {
        type            empty;
        inGroups        1(empty);
        nFaces          8636;
        startFace       8926;
    }
)
```

Figure 4.10 Boundary name use in 0 subdirectory

Boundary Conditions

After creation of polymesh in constant folder, its time to create boundary files in directory 0 created by user in starting part of the problem. The boundary files depend on governing equations which will be solved during numerical simulation, for this problem the main variables(**total four variables**) which must be solved are present in the following figure

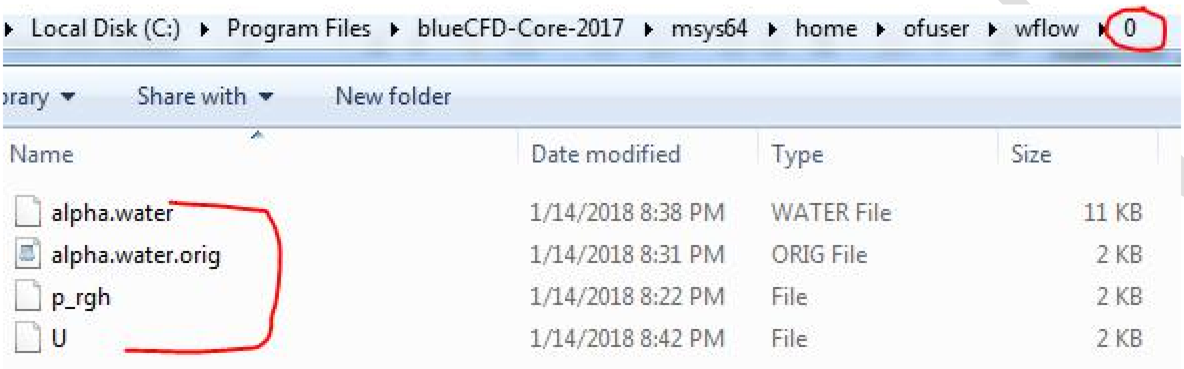


Figure 4.11 Boundary conditions for present problem

There are large number of boundary conditions are available, which must be provide to boundary conditions for every problems, the selection is depend on type of boundary conditions useful for particular problem. In this tutorial some important boundary conditions are used all detail information is provided in last of this tutorial book. All files are created using note-pad-2 (provided in software package)

Note: We copy these files from tutorial available in open-foam program files (see the following figure)

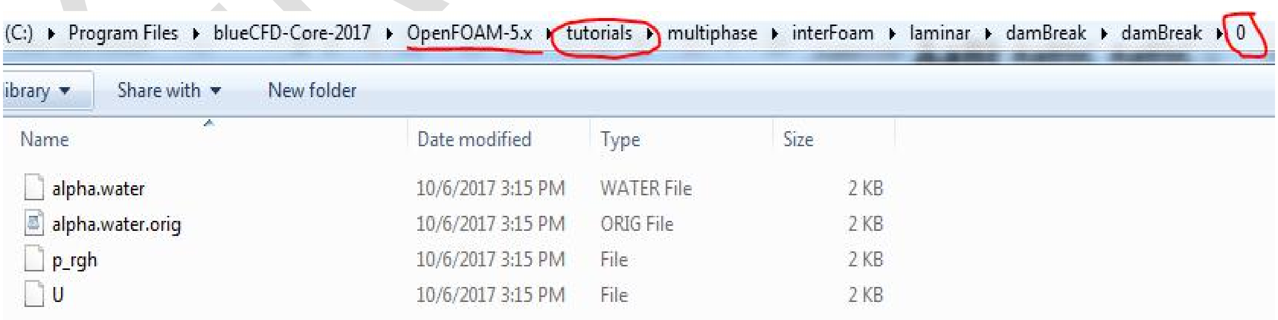


Figure 4.12 Copy pee-generated files for present problem

Now edit these boundary files as per given details for all boundary conditions:

Alpha.water.orig boundary condition for free water falling problem

```
/*--------------------------------*- C++ -*----------------------------------*\
| =========                 |                                                 |
| \\      /  F ield         | OpenFOAM: The Open Source CFD Toolbox           |
|  \\    /   O peration     | Version:  5                                     |
|   \\  /    A nd           | Web:      www.OpenFOAM.org                      |
|    \\/     M anipulation  |                                                 |
\*---------------------------------------------------------------------------*/
FoamFile
{
    version     2.0;
    format      ascii;
    class       volScalarField;
    object      alpha.water;
}
// * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * //

dimensions      [0 0 0 0 0 0 0];

internalField   uniform 0;

boundaryField
{
    leftwall
    {
        type            zeroGradient;
    }

    rightwall
    {
```

```
        type                zeroGradient;
    }

    lowerwall
    {
        type                zeroGradient;
    }
    balls
    {
        type                zeroGradient;
    }

    atmosphere
    {
        type                inletOutlet;
        inletValue          uniform 0;
        value               uniform 0;
    }

    frontAndBackPlanes
    {
        type                empty;
    }
}

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

P_rgh boundary condition for free water falling problem

```
/*--------------------------------*- C++ -*----------------------------------*\
| =========                 |                                                 |
| \\      /  F ield         | OpenFOAM: The Open Source CFD Toolbox           |
|  \\    /   O peration     | Version:  5                                     |
|   \\  /    A nd           | Web:      www.OpenFOAM.org                      |
|    \\/     M anipulation  |                                                 |
\*---------------------------------------------------------------------------*/
FoamFile
{
    version     2.0;
    format      ascii;
    class       volScalarField;
    object      p_rgh;
}
// * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * //

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

internalField   uniform 0;

boundaryField
{
    leftwall
    {
        type            fixedFluxPressure;
        value           uniform 0;
    }

    rightwall
```

```
    {
        type            fixedFluxPressure;
        value           uniform 0;
    }

    lowerwall
    {
        type            fixedFluxPressure;
        value           uniform 0;
    }
    balls
    {
        type            fixedFluxPressure;
        value           uniform 0;
    }

    atmosphere
    {
        type            totalPressure;
        p0              uniform 0;
    }

    frontAndBackPlanes
    {
        type            empty;
    }
}

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

U boundary condition for free water falling problem

```
/*--------------------------------*- C++ -*----------------------------------*\
| =========                 |                                                 |
| \\      /  F ield         | OpenFOAM: The Open Source CFD Toolbox           |
|  \\    /   O peration     | Version:  5                                     |
|   \\  /    A nd           | Web:      www.OpenFOAM.org                      |
|    \\/     M anipulation  |                                                 |
\*---------------------------------------------------------------------------*/
FoamFile
{
    version     2.0;
    format      ascii;
    class       volVectorField;
    location    "0";
    object      U;
}
// * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * //

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

internalField   uniform (0 0 0);

boundaryField
{
    leftwall

    {
        type            noSlip;
    }
```

```
rightwall

{
    type            noSlip;
}

lowerwall

{
    type            noSlip;
}

balls

{
    type            noSlip;
}

atmosphere

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

frontAndbackPlanes

{
    type            empty;
}
```

```
}


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

Fluid Flow Properties (Transport and Turbulence)

These properties are present in directory constant where polymesh is available, the process is same like previous section. For simplicity copy these files from tutorial available in open-foam program files path for selected solver.

Source code for g

```
/*--------------------------------*- C++ -*----------------------------------*\
| =========                 |                                                 |
| \\      /  F ield         | OpenFOAM: The Open Source CFD Toolbox           |
|  \\    /   O peration     | Version:  5                                     |
|   \\  /    A nd           | Web:      www.OpenFOAM.org                      |
|    \\/     M anipulation  |                                                 |
\*---------------------------------------------------------------------------*/
FoamFile
{
    version     2.0;
    format      ascii;
    class       uniformDimensionedVectorField;
    location    "constant";
    object      g;
}
// * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * //
dimensions      [0 1 -2 0 0 0 0];
value           (0 -9.81 0);
// ************************************************************************* //
```

Source code for Transport Properties

```
/*--------------------------------*- C++ -*----------------------------------*\
| =========                 |                                                 |
| \\      /  F ield         | OpenFOAM: The Open Source CFD Toolbox           |
|  \\    /   O peration     | Version:  5                                     |
|   \\  /    A nd           | Web:      www.OpenFOAM.org                      |
|    \\/     M anipulation  |                                                 |
\*---------------------------------------------------------------------------*/
FoamFile
{
    version     2.0;
    format      ascii;
    class       dictionary;
    location    "constant";
    object      transportProperties;
}
// * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * //

phases (water air);

water
{
    transportModel  Newtonian;
    nu              1e-06;
    rho             1000;
}

air
{
    transportModel  Newtonian;
```

```
    nu              1.48e-05;

    rho             1;

}

sigma           0.07;

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

Source code for Turbulence Properties

```
/*--------------------------------*- C++ -*----------------------------------*\
| =========                 |                                                 |
| \\      /  F ield         | OpenFOAM: The Open Source CFD Toolbox           |
|  \\    /   O peration     | Version:  5                                     |
|   \\  /    A nd           | Web:      www.OpenFOAM.org                      |
|    \\/     M anipulation  |                                                 |
\*---------------------------------------------------------------------------*/
FoamFile
{
    version     2.0;
    format      ascii;
    class       dictionary;
    location    "constant";
    object      turbulenceProperties;
}
// * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * //

simulationType  laminar;


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

Time/Solver (Input-Output)

Like other CFD software, Open-Foam is also required time size as per requirement of problem. This task is fulfill by creating **controlDict** file in directory **“system”**

```
/*--------------------------------*- C++ -*----------------------------------*\
| =========                 |                                                 |
| \\      /  F ield         | OpenFOAM: The Open Source CFD Toolbox           |
|  \\    /   O peration     | Version:  5                                     |
|   \\  /    A nd           | Web:      www.OpenFOAM.org                      |
|    \\/     M anipulation  |                                                 |
\*---------------------------------------------------------------------------*/
FoamFile
{
    version     2.0;
    format      ascii;
    class       dictionary;
    location    "system";
    object      controlDict;
}
// * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * //

application     interFoam;

startFrom       latestTime;

startTime       0;

stopAt          endTime;

endTime         5;
```

```
deltaT              0.0005;

writeControl      adjustableRunTime;

writeInterval     0.02;

purgeWrite         0;

writeFormat        ascii;

writePrecision    6;

writeCompression uncompressed;

timeFormat         general;

timePrecision      6;

runTimeModifiable yes;

adjustTimeStep    yes;

maxCo               1;
maxAlphaCo          1;

maxDeltaT           1;


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

decomposeParDict is not required in this problem so no need to discuss here

Numerical Schemes

The numerical schemes are defined in file **fvSchemes** available in directory **system** created by user in starting phase of problem.

```
/*--------------------------------*- C++ -*----------------------------------*\
| =========                 |                                                 |
| \\      /  F ield         | OpenFOAM: The Open Source CFD Toolbox           |
|  \\    /   O peration     | Version:  5                                     |
|   \\  /    A nd           | Web:      www.OpenFOAM.org                      |
|    \\/     M anipulation  |                                                 |
\*---------------------------------------------------------------------------*/
FoamFile
{
    version     2.0;
    format      ascii;
    class       dictionary;
    location    "system";
    object      fvSchemes;
}
// * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * //

ddtSchemes
{
    default         Euler;
}

gradSchemes
{
    default         Gauss linear;
}
```

```
divSchemes
{
    div(rhoPhi,U)    Gauss linearUpwind grad(U);
    div(phi,alpha)    Gauss vanLeer;
    div(phirb,alpha) Gauss linear;
    div(((rho*nuEff)*dev2(T(grad(U))))) Gauss linear;
}

laplacianSchemes
{
    default              Gauss linear corrected;
}

interpolationSchemes
{
    default              linear;
}

snGradSchemes
{
    default              corrected;
}

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

Solution Control

Solution is controlled by making file **fvSolution** in directory **system**

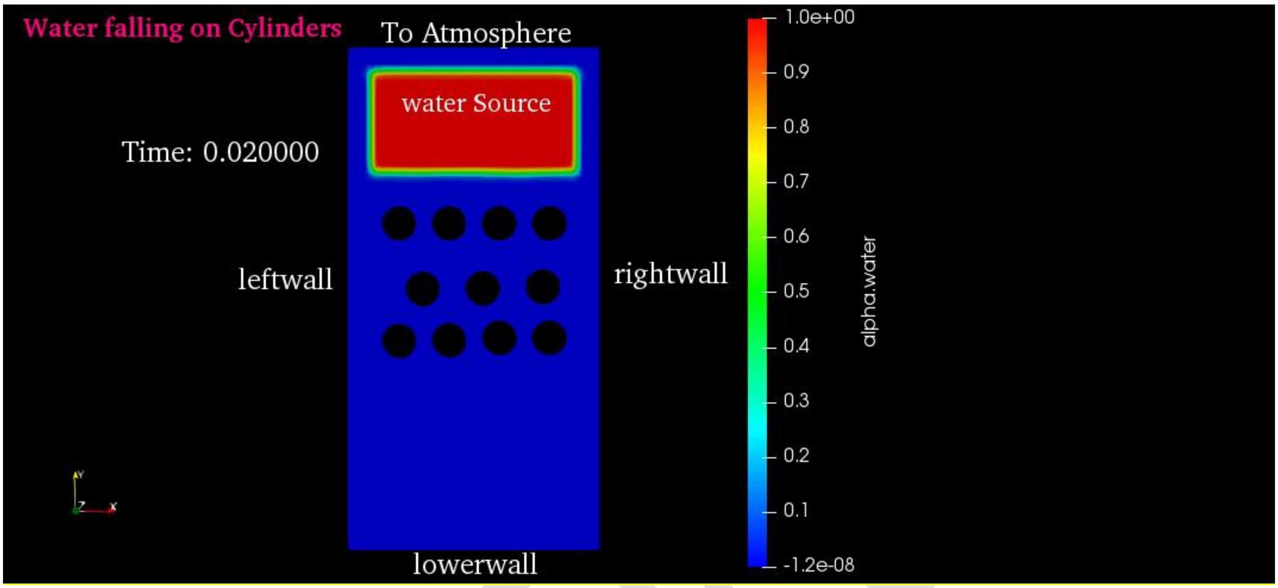
```
/*--------------------------------*- C++ -*----------------------------------*\
| =========                 |                                                 |
| \\      /  F ield         | OpenFOAM: The Open Source CFD Toolbox           |
|  \\    /   O peration     | Version:  5                                     |
|   \\  /    A nd           | Web:      www.OpenFOAM.org                      |
|    \\/     M anipulation  |                                                 |
\*---------------------------------------------------------------------------*/
FoamFile
{
    version     2.0;
    format      ascii;
    class       dictionary;
    location    "system";
    object      fvSolution;
}
// * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * //

solvers
{
    "alpha.water.*"
    {
        nAlphaCorr      2;
        nAlphaSubCycles 1;
        cAlpha          1;

        MULESCorr       yes;
        nLimiterIter    5;
```

```
        solver          smoothSolver;
        smoother        symGaussSeidel;
        tolerance       1e-8;
        relTol          0;
    }

    "pcorr.*"
    {
        solver          PCG;
        preconditioner  DIC;
        tolerance       1e-5;
        relTol          0;
    }

    p_rgh
    {
        solver          PCG;
        preconditioner  DIC;
        tolerance       1e-07;
        relTol          0.05;
    }

    p_rghFinal
    {
        $p_rgh;
        relTol          0;
    }

    U
    {
```

```
            solver          smoothSolver;
            smoother        symGaussSeidel;
            tolerance       1e-06;
            relTol          0;
        }
}

PIMPLE
{
        momentumPredictor     no;
        nOuterCorrectors      1;
        nCorrectors           3;
        nNonOrthogonalCorrectors 0;
}

relaxationFactors
{
        equations
        {
            ".*" 1;
        }
}

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

setFieldsDict

This command is used for initialize the secondary phase (in this problem water is secondary phase) in fluid domain. As seen in figure, water is initially at top of the domain, so a boxtocell command is used to set water volume fraction to 1 using this command.




```
/*--------------------------------*- C++ -*----------------------------------*\
| =========                 |                                                 |
| \\      /  F ield         | OpenFOAM: The Open Source CFD Toolbox           |
|  \\    /   O peration     | Version:  5                                     |
|   \\  /    A nd           | Web:      www.OpenFOAM.org                      |
|    \\/     M anipulation  |                                                 |
\*---------------------------------------------------------------------------*/
FoamFile
{
    version     2.0;
    format      ascii;
    class       dictionary;
    location    "system";
    object      setFieldsDict;
```

```
}
// * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * //

defaultFieldValues
(
    volScalarFieldValue alpha.water 0
);

regions
(
    boxToCell
    {
        box (0.05 0.75 -1) (0.45 0.95 1);
        fieldValues
        (
            volScalarFieldValue alpha.water 1
        );
    }
);

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

Run this command using

cp 0/alpha.water.orig 0/alpha.water

```
home@home-PC MINGW64 OpenFOAM-5.x ~/wflow
$ cp 0/alpha.water.orig 0/alpha.water
```

After run this command, alpha.water is created in 0 folder

Solver Running

After created the all desired source files for present problem only last step is to run the solver by provide the command “” (see the following figure)

Figure 4.13 command for run the solver

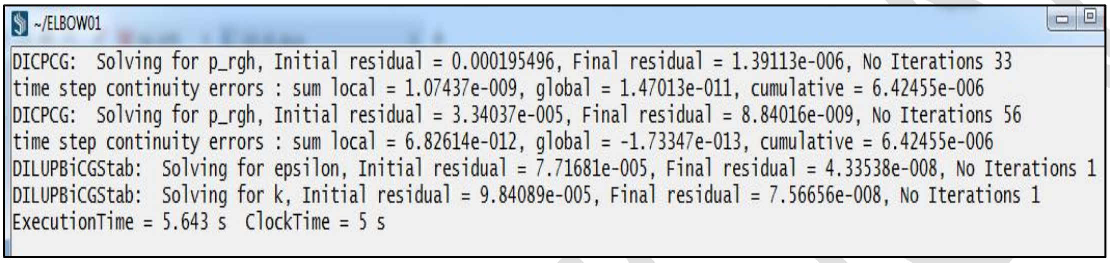


Figure 4.14 Solver running in open-foam code

After successful ending of solver running condition, various sub-folders are created in user directory free water falling for different time conditions

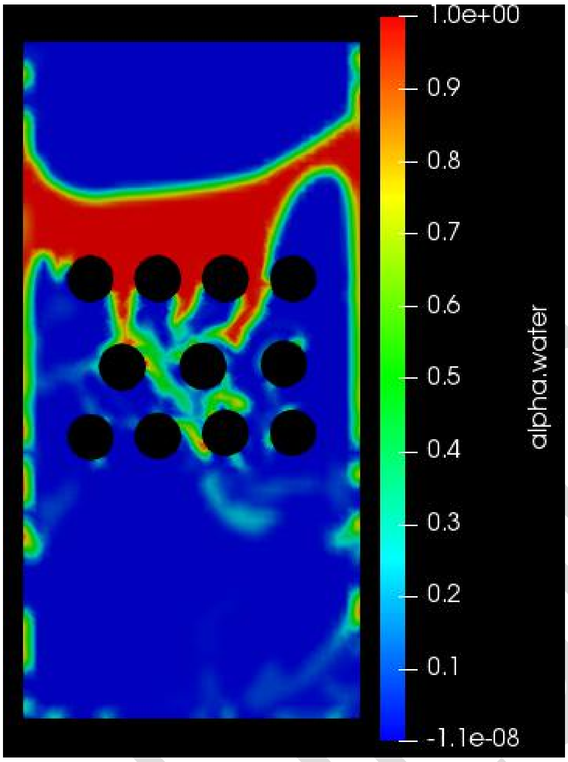
Post-Processing

In this step solved problem is analysis by using para-foam software available with open-foam software. To run this software run command “para-foam” in terminal of open-foam software. It must take care that terminal selected free water falling folder before run this command.

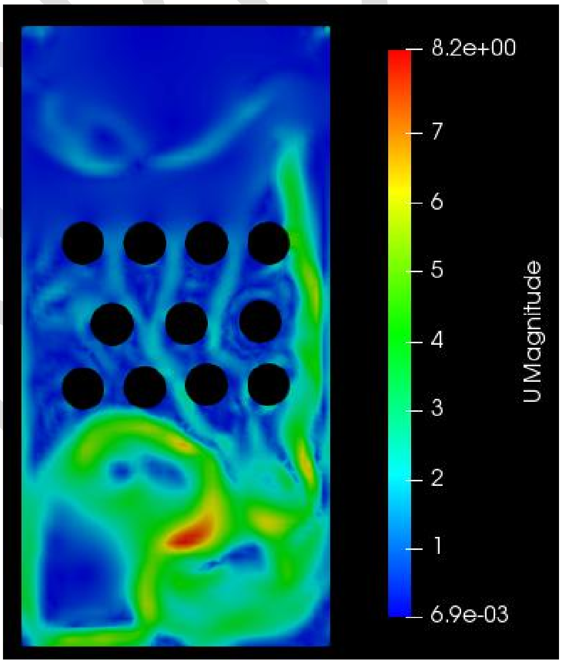
Figure 4.15 para-foam for post-processing of problem

Para-view is very large software for post-processing of CFD simulation files so in this tutorial we skip this software at this stage. Only some important results are present here.

Water Volume Fraction Contours



Velocity Contours



This offering is not approved or endorsed by OpenCFD Limited, producer and distributor of the OpenFOAM software and owner of the OPENFOAM® and OpenCFD® trade marks.

END