# Summer Fellowship Report

On

## Analysis of flow behavior when obstructed by moving vane

Submitted by

**Vignesh.S.P**

Under the guidance of

**Prof.Kannan M. Moudgalya**
Chemical Engineering Department
IIT Bombay

Vignesh.S.P

July 10, 2019

# Acknowledgment

The success of this project required a lot of guidance and help from many people and I am extremely privileged to have got this along the completion of this project. I respect and thank Prof.Kannan M. Moudgalya, Chemical Engineering Department and Prof.Shivasubramanian Gopalakrishnan, Mechanical Engineering Department, for providing me an opportunity to do the project work in FOSSEE, IIT Bombay and giving me all the support and guidance which made me complete the project duly.

I'm extremely thankful to my mentors Mr. Sathish Kanniappan and Ms. Deepa Vedartham for their support and mentorship through out the course of completion of the project. I'm thankful to my mentors for providing equipment support which was very helpful.

I'm thankful and fortunate enough to get constant encouragement, support and guidance from all the personnel's from FOSSEE, IIT Bombay.

# Abstract

This case study aims to numerically simulate the behavior of a water while the flow is being obstructed by a moving vane. The mesh motion of the vane is achieved by Overset Grid technique or Chimera Grid technique. This case study explores the functionality and capability of Overset Grid method for dynamic problems available in OpenFOAM.

# Contents

# List of Figures

# List of Tables

# Chapter 1

# Introduction

The behavior of fluid in dynamic systems are very difficult to numerically simulate using traditional methods to handle dynamic systems such as mesh deformation, Arbitrary Moving Interface (AMI), and re-meshing. This case study explores the functionality of overset method in dynamic problems. The numerical simulation of water flow when obstructed by a moving vane at a constant velocity is studied here. The motion of the vane is controlled by overset grid method.

## 1.1 OpenFOAM

OpenFOAM (for "Open-source Field Operation And Manipulation") is a C++ toolbox for the development of customized numerical solvers, and pre/post-processing utilities for the solution of continuum mechanics problems, most prominently including computational fluid dynamics. OpenFOAM is freely available and open source, licensed under the GNU General Public Licence. The licence is designed to offer freedom, in particular it encourages users of the software to make modifications or developments. Overset Grid in OpenFOAM is available in versions released only after 2017. It is being actively developed and several solvers and validation has been done.

## 1.2 Overset Grid Method

Overset gridding refers to the use of multiple disconnected grids to discretize the flow domain. The component grids, which can be any size, type, or shape, need only overlap each other to completely cover the solution domain. Furthermore, a component grid resolving one geometric feature may intersect another geometric feature. As a final preprocessing step (or during the solution, in the case of moving bodies) composite grid assembly software determines which grid points lay outside the flow domain and grid-to-grid connectivity.

The overset approach also enables changing the geometry and grid system locally without requiring regeneration of other grids. This flexibility greatly simplifies design studies as geometry perturbations can easily be added to an existing design and grid system by gridding the new feature and possibly including grids to connect the new feature with the existing grids. Since the baseline grid system is not altered the changes in the flow are more reflective of the change in the geometry and not changes resulting from re-meshing the entire geometry.

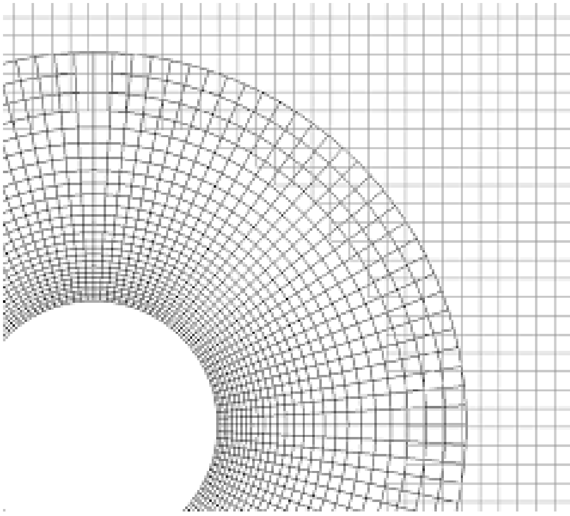

Figure 1.1: Overset Grid of Sphere

# Chapter 2

# Case Setup

## 2.1 Geometry and Mesh

The Geometry of the case study is a slice on a circular pipe of diameter of 0.1m and of length 1m. The vane is of thickness 0.01m and of height 0.1m. The mesh for the fluid domain is made using blockMesh utility with an empty boundary patch named oversetPatch to trigger overset interpolation. The vane is meshed separately and is merged with the mesh of the pipe using 'mergeMeshes' utility. The fluid domain in pipe is meshed coarsely with local size of 0.002m which resulted in 12500 cells and the vane is meshed finely with local size of 0.001m which resulted in 9000 cells.

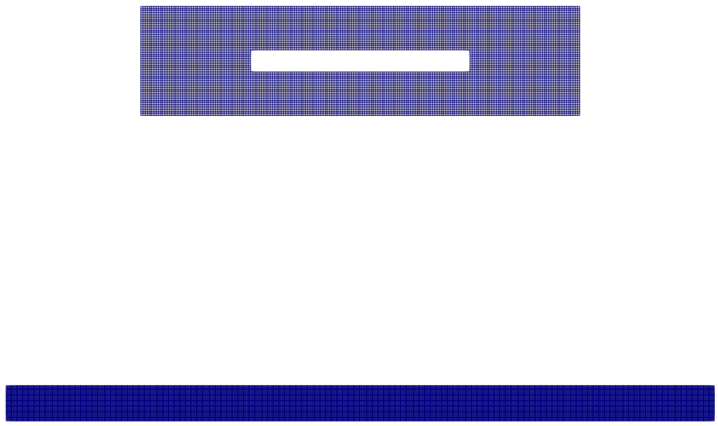

Figure 2.1: Vane Mesh and domain mesh separated

Figure 2.2: Overset grid of vane onto pipe mesh

## 2.2 Boundary conditions

The boundary conditions employed for the computational domain is given in table 2.1. The overset interpolation zones are setup using setFields with zoneID. The fluid domain in pipe is set to zoneID 0 and overset vane mesh is set to zoneID 1.

| Boundary Name | U | p |
|---|---|---|
| walls | type uniformFixedValue;<br>uniformValue (0 0 0); | type zeroGradient; |
| vane | type movingWallVelocity;<br>value (0 0 0); | type zeroGradient; |
| inlet | type fixedValue;<br>value uniform(1 0 0); | type zeroGradient; |
| outlet | type zeroGradient; | type fixedValue;<br>value uniform 0; |
| sides and oversetPatch | type overset; | type overset; |

Table 2.1: Boundary conditions for the case study.

## 2.3 Dynamic motion of vane

The dynamic motion of vane is an oscillating linear motion which is defined in dynamicMeshDict file inside constant folder. The details of the setup is as follows. The vane moves in y direction with an amplitude of 0.1m, the speed of the motion of the mesh is governed by the value omega.

The dynamicMeshDict is as follows,

```
dynamicFvMesh          dynamicOversetFvMesh;
solver                 multiSolidBodyMotionSolver;
multiSolidBodyMotionSolverCoeffs
{
    vaneZone
    {
        solidBodyMotionFunction  oscillatingLinearMotion;
        oscillatingLinearMotionCoeffs
        {
            amplitude    (0 -0.1 0);
            omega        2.0;
        }
    }
}
```

## 2.4 Solver and Simulation control

overPimpleDyMFoam solver is used for this simulation. The official definition for this solver is as follows:

Transient solver for incompressible, flow of Newtonian fluids on a moving mesh using the PIMPLE (merged PISO-SIMPLE) algorithm.

The transport properties for case is setup in transportProperties file in constant directory in which the value of density and kinematic viscosity for both air and water are given as well as surface tension value is also given.

| Sl.No. | Description | Value |
|---|---|---|
| 1 | Kinematic viscosity of water | 1e-06 $m^2/s$ |
| 2 | Density of water | 1000 $kg/m^3$ |

Table 2.2: Properties of Fluid

The important control values as given in controlDict is as follows
startTime 0;
endTime 1.5;
deltaT 0.00025;
writeControl adjustableRunTime;
writeInterval 0.005;
maxCo 1;
maxDeltaT 0.001;

# Chapter 3

# Result and analysis

## 3.1 Mesh Motion

The overset mesh of vane is coded to move in a linear oscillating pattern which can be seen in the figure 3.1 at different time steps of the simulation

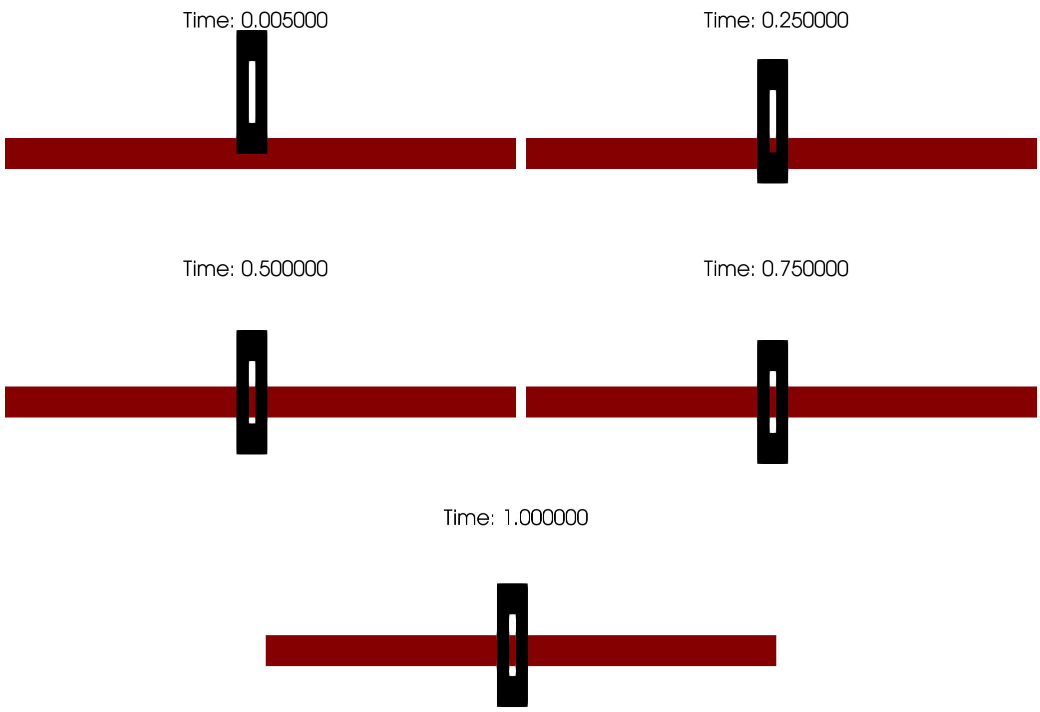


Figure 3.1: Mesh motion

## 3.2 Pressure Field

The pressure field from the simulation are plotted against the length of the pipe for different time steps can be seen in the figure 3.2. The pressure difference on the inlet side and the outlet side of the pipe during closing will rapidly increase as high as 60 $Pa$.

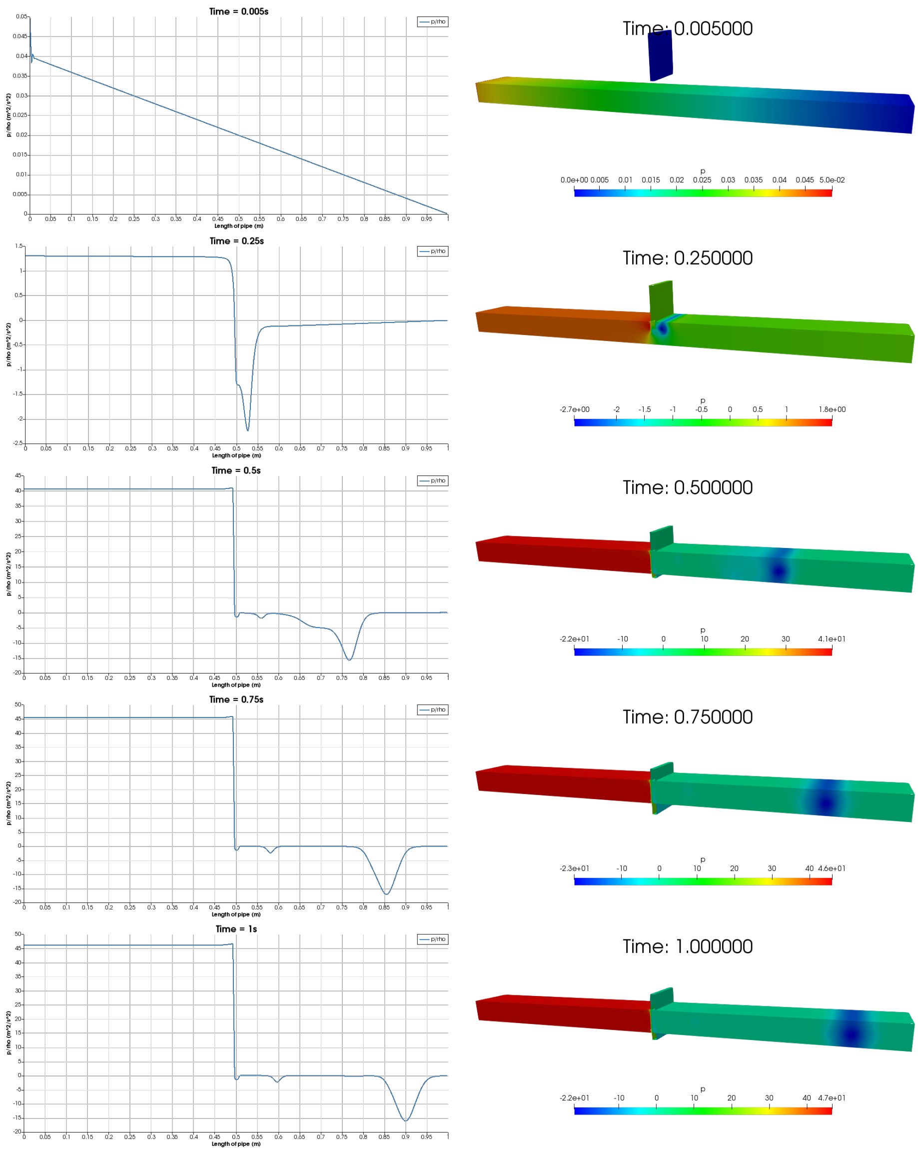


Figure 3.2: Pressure Plot and contours for different time steps

# Chapter 4

# Conclusion

This case study explores the overset grid methodology for handling dynamic systems in CFD in a standard solver for incompressible flow. The setup and simulation of the case is explained in the report and the results are also viewed.

# Reference


- OpenFOAM - Overset grids wiki from cfd-online.com
- Youtube playlist on Overset grid by Wolf Dynamics
- Official Overset Guide by ESI-OpenCFD

# Summer Fellowship Report

On

**Cooling of sphere - Natural and Forced Convection**

Submitted by

**Vignesh.S.P**

Under the guidance of

**Prof.Kannan M. Moudgalya**
Chemical Engineering Department
IIT Bombay

Vignesh.S.P

July 9, 2019

# Acknowledgment

The success of this project required a lot of guidance and help from many people and I am extremely privileged to have got this along the completion of this project. I respect and thank Prof.Kannan M. Moudgalya, Chemical Engineering Department and Prof.Shivasubramanian Gopalakrishnan, Mechanical Engineering Department, for providing me an opportunity to do the project work in FOSSEE, IIT Bombay and giving me all the support and guidance which made me complete the project duly.

I'm extremely thankful to my mentors Mr. Sathish Kanniappan and Ms. Deepa Vedartham for their support and mentorship through out the course of completion of the project. I'm thankful to my mentors for providing equipment support which was very helpful.

I'm thankful and fortunate enough to get constant encouragement, support and guidance from all the personnel's from FOSSEE, IIT Bombay.

# Abstract

This case study aims to numerically simulate the behavior of cooling of a heated sphere in both natural and forced convective environment. The flow behavior in the fluid domain and the temperature in the solid sphere is analysed. Conjugate Heat Transfer between conduction in solid sphere and convection in fluid domain is used to simulated the case study. The thermal source for the simulation is provided by the source terms in the energy equation using fvOptions.

# Contents

# List of Figures

# List of Tables

# Chapter 1

# Introduction

The coupled heat transfer between conduction in solid and convection in fluid is termed as conjugate heat transfer. This methodology is employed here to simulate cooling of sphere under natural convection and forced convection. The comparison between the two modes of convection is done and the results are studied. Heat to the solid zone is given using fvOptions utility which enables user to specify heat

## 1.1 OpenFOAM

OpenFOAM (for "Open-source Field Operation And Manipulation") is a C++ toolbox for the development of customized numerical solvers, and pre/post-processing utilities for the solution of continuum mechanics problems, most prominently including computational fluid dynamics. OpenFOAM is freely available and open source, licensed under the GNU General Public Licence. The licence is designed to offer freedom, in particular it encourages users of the software to make modifications or developments. Overset Grid in OpenFOAM is available in versions released only after 2017. It is being actively developed and several solvers and validation has been done.

## 1.2 Conjugate Heat Transfer

This model, based on a strictly mathematically stated problem, describes the heat transfer between a body and a fluid flowing over or inside it as a result of the interaction of two objects. The physical processes and solutions of the governing equations are considered separately for each object in two subdomains. Matching conditions for these solutions at the interface provide the distributions of temperature and heat flux along the bodyflow interface, eliminating the need for a heat transfer coefficient.
The mass conservation equation is as follows

$$\frac{\partial \rho}{\partial t} + \nabla.(\rho U) = 0$$

where $\rho$ is density of the fluid, $U$ is velocity field.

The momentum conservation equation is as follows

$$\frac{\partial \rho U}{\partial t} + \nabla.(\rho U U) = -\nabla p + \nabla.\rho\nu\tau + F$$

where $U$ is the velocity field, $\rho$ is the density, $p$ is the pressure, $\nu$ is the kinematic viscosity, $\tau$ is the stress tensor and $F$ is the combined sources term.
The energy equation for the fluid is as follows

$$\frac{\partial \rho E}{\partial t} + \nabla.(\rho U E) + \nabla.(U p) = -\nabla.q + \nabla.(\tau.U) + \rho r + \rho g.U$$

where $E$ is the total energy, $r$ is specific heat source, $g$ is acceleration due to gravity.
Then finally the equation governing the heat conduction in solid regions is as follows

$$\frac{\partial \rho h}{\partial t} = \nabla.(\alpha\ grad(h))$$

where $h$ is specific enthalpy, $\rho$ is the density of solid, $\alpha = k/c_p$ is thermal diffusivity and $k$ is thermal conductivity and $c_p$ is specific heat capacity.

The coupling of temperature between solid and fluid regions are done using baffle patches which are shared by both regions. Boundary condition for sharing the temperature information is also given at these baffle patches.

# Chapter 2

# Case Setup

## 2.1 Geometry and Mesh

The geometry of the case study is a rectangular duct in which the spherical solid is placed near the inlet. The sphere is place at (0,0,0) co-ordinate with an diameter of 0.05m in a rectangular duct of 0.5m height and 1m length. Thickness is not considered since the case is simulated in 2D. The sphere encompasses the solid region with structured hexagonal mesh with 4500 cells. The fluid zone is the rectangular duct without the solid zone which is off 4500 cells. This combined mesh is separated into regions using 'splitMeshRegions' utility which will create the necessary baffles (shared patches) between solid and fluid regions. The mesh can be seen in figure 2.1

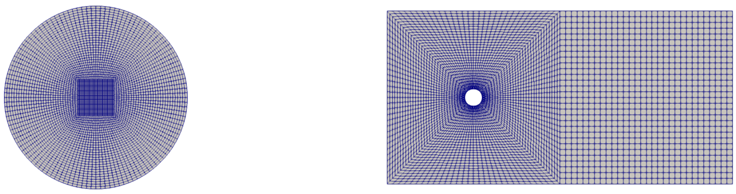

Figure 2.1: Separated mesh regions

## 2.2 Boundary conditions

The boundary conditions employed for the computational domain is given in table 2.1. Boundary condition for natural and forced convection is similar only there is no velocity at the inlet in natural convection. The Solid region has one patch named solidZone_to_fluidZone which will be given same temperature boundary condition as in fluidZone_to_solidZone. Boundary condition for the baffles solidZone_to_fluidZone and fluidZone_to_solidZone are same for other fields.

| Boundary Name | U | p_rgh | T |
|---|---|---|---|
| inlet | type fixedValue; value uniform (0.001 0 0); | type zeroGradient; value uniform 0; | type fixedValue; value uniform 300; |
| outlet | type inletOutlet; inletValue uniform (0 0 0); | type fixedValue; value uniform 0; | type inletOutlet; value uniform 300; |
| 'bottom' and 'top' | type fixedValue; | type fixedFluxPressure; value uniform 0; | type zeroGradient; value uniform 300; |
| 'frontFluid' and 'backFluid' | type empty; | type empty; | type empty; |
| fluidZone_ -to_solidZone | type fixedValue; value uniform (0 0 0); | type fixedFluxPressure; value uniform 0; | type compressible::turbulent -TemperatureCoupled -BaffleMixed; Tnbr T; kappaMethod fluidThermo; value uniform 300; |

Table 2.1: Boundary Conditions for Fluid Region - Forced Convection

## 2.3 Adding Source using fvOptions

The fvOptions functionality in OpenFOAM is flexible framework to add various source terms to the governing equations without the need to rewrite the original source code. The fvOptions framework has been introduced to allow users to select any physics that can be represented as sources or constraints on the governing equations, e.g. porous media, thermal source and body forces. This new fvOptions framework enhances and supercedes the previous run-time selectable sources in version 2.1. The thermal source use in this case is fixedTemperatureConstraint to the solidZone for 1000s starting from 500s of the simulation time step.

```
fixedTemperature
{
    type            fixedTemperatureConstraint;
    active          yes;
    timeStart       500;
    duration        1000;
    selectionMode   cellZone;
    cellZone        heatSource;
    mode            uniform; // uniform or lookup
    temperature     constant 1273;
    // fixed temperature with time [K]
}
```

## 2.4 Solver and Simulation control

chtMultiRegionFoam solver is used for this simulation. The official definition for this solver is as follows:

Transient solver for buoyant, turbulent fluid flow and solid heat conduction with conjugate heat transfer between solid and fluid regions

The transport and thermal properties for case is setup in thermophysicalProperties file in region directory within constant directory. The fluid properties are as follows

```
mixture
{
    specie
    {
        molWeight       18;
    }
    equationOfState
    {
        rho             1000;
    }
    thermodynamics
    {
        Cp              4181;
        Hf              0;
    }
    transport
    {
        mu              959e-6;
        Pr              6.62;
    }
}
```

The solid Zone properties are as follows

```
mixture
{
    specie
    {
        molWeight   50;
    }

    transport
    {
        kappa   80;
    }
```

```
    thermodynamics
    {
        Hf          0;
        Cp          450;
    }

    equationOfState
    {
        rho         8000;
    }
}
```

The important control values as given in controlDict is as follows

```
application         chtMultiRegionFoam;
startFrom           latestTime;
startTime           0.001;
stopAt              endTime;
endTime             4000;
deltaT              0.001;
writeControl        adjustableRunTime;
writeInterval       20;
purgeWrite          0;
writeFormat         ascii;
writePrecision      8;
writeCompression    off;
timeFormat          general;
timePrecision       6;
runTimeModifiable   yes;
maxCo               0.6;
maxDi               10.0;// Maximum diffusion number
adjustTimeStep      yes;
```

# Chapter 3

# Result and analysis

## 3.1 Temperature Plot

The temperature plot against the length of the duct at different time steps for both natural and forced convection will show cooling behavior. From the plots we can surely see the effect of large temperature reduction in forced convection in comparison to natural convection as in figure 3.1 and 3.2

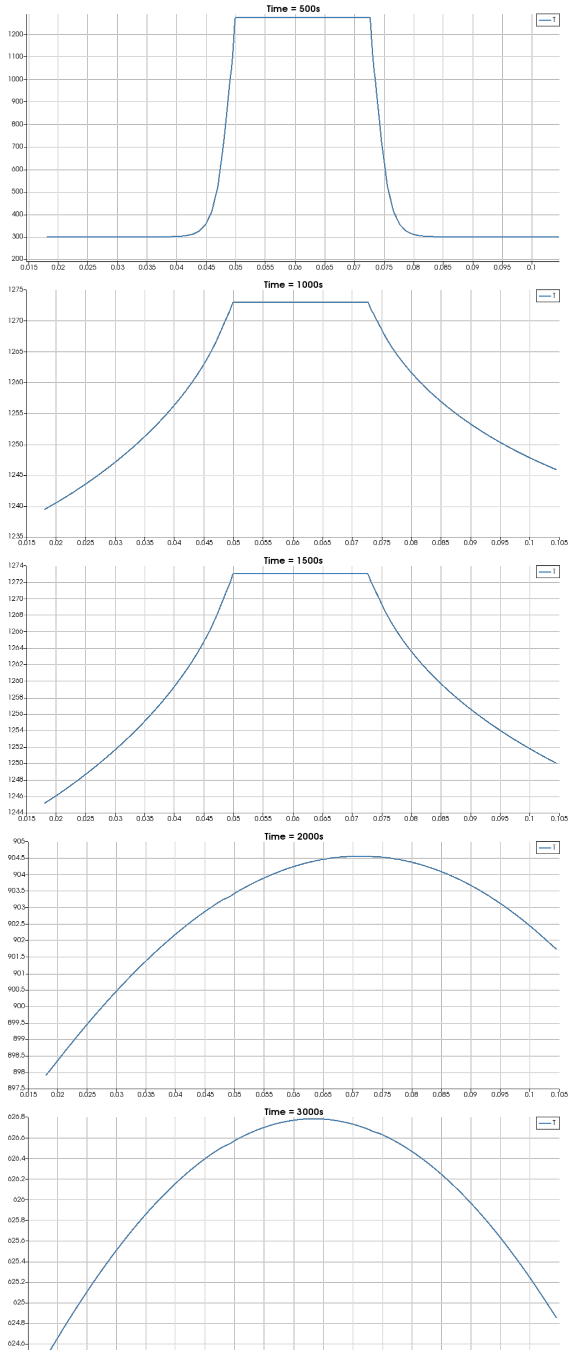


Figure 3.1: Temperature plot of solid zone at different time steps - Natural Convection

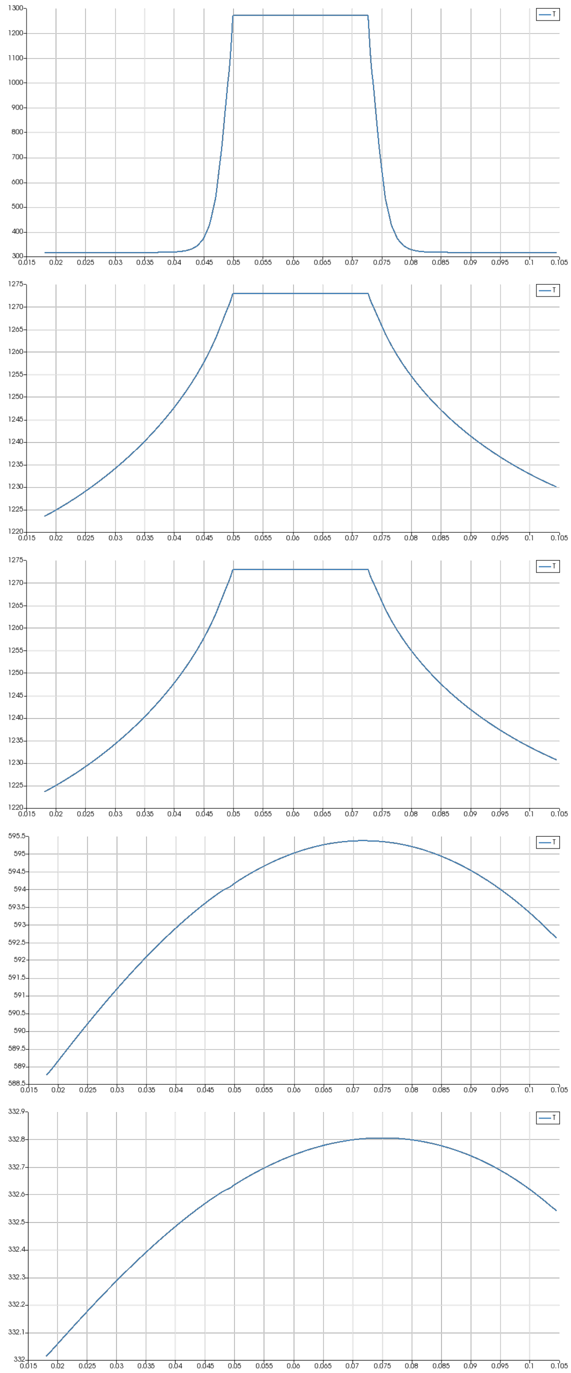

Figure 3.2: Temperature plot of solid zone at different time steps - Forced Convection

# Chapter 4

# Conclusion

This case study explores the conjugate heat transfer between solids and fluids. By using fvOptions utility, thermal source for simulation is setup without having to recompile the source file of the solver. Comparison between natural and forced convection is studied.

# Reference


- chtMultiRegionFoam solver - OpenFOAMwiki.net
- Tutorial on conjugate heat transfer in OpenFOAM
- Implementing chtMultiRegionFoam Solver for Electric Welding in OpenFoam 1.6.x

# Summer Fellowship Report

On

## Simulation of Water Droplet Impact on Hydrophilic and Hydrophobic surfaces

Submitted by

**Vignesh.S.P**

Under the guidance of

**Prof.Kannan M. Moudgalya**
Chemical Engineering Department
IIT Bombay

Vignesh.S.P

July 7, 2019

# Acknowledgment

The success of this project required a lot of guidance and help from many people and I am extremely privileged to have got this along the completion of this project. I respect and thank Prof.Kannan M. Moudgalya, Chemical Engineering Department and Prof.Shivasubramanian Gopalakrishnan, Mechanical Engineering Department, for providing me an opportunity to do the project work in FOSSEE, IIT Bombay and giving me all the support and guidance which made me complete the project duly.

I'm extremely thankful to my mentors Mr. Sathish Kanniappan and Ms. Deepa Vedartham for their support and mentorship through out the course of completion of the project. I'm thankful to my mentors for providing equipment support which was very helpful.

I'm thankful and fortunate enough to get constant encouragement, support and guidance from all the personnel's from FOSSEE, IIT Bombay.

# Abstract

This case study aims to numerically simulate the behavior of a water droplet falling onto a hydrophobic and hydrophilic surface from an arbitrary height of 5 cm. Volume of Fraction method (VOF) is employed to handle multi-phase interaction between water and air. Adaptive Mesh refinement is being used to get better accurate behavior of water Droplets after impact. The hydrophilic and hydrophobic behavior is governed by the contact angle of the surface.

# Contents

# List of Figures

# List of Tables

# Chapter 1

# Introduction

Hydrophilic and Hydrophobic surface are widely used in food industries, air-crafts, automobiles, etc. Impact of a water droplet on these surfaces will produce different behaviors. This behavior is studied and simulated using OpenFOAM. Multi-phase model is employed to handle the water and air interface. Since water droplets on impact will separate into very small droplets, adaptive mesh refinement is employed to refine the mesh during simulation to capture small droplets. The fluid domain is meshed and appropriate solver is chosen along with boundary and initial conditions and after the simulation is done, results are obtained and studied.

## 1.1 OpenFOAM

OpenFOAM (for "Open-source Field Operation And Manipulation") is a C++ toolbox for the development of customized numerical solvers, and pre/post-processing utilities for the solution of continuum mechanics problems, most prominently including computational fluid dynamics. OpenFOAM is freely available and open source, licensed under the GNU General Public Licence. The licence is designed to offer freedom, in particular it encourages users of the software to make modifications or developments.

## 1.2 Multiphase Model

Multiphase model, employed for this case study is Volume of Fluid (VoF) Model. In computational fluid dynamics, the volume of fluid (VOF) method is a free-surface modelling technique, i.e. a numerical technique for tracking and locating the free surface or fluidfluid interface. It employs a VoF equation along with continuity and momentum which solves for the volume fraction of the fluid. From the volume fraction, density is calculated.

The continuity equation for incompressible flow is as follows

$$\nabla.U = 0$$

The momentum equation for incompressible flow is as follows

$$\frac{\partial \rho U}{\partial t} + \nabla.(\rho U U) = -\nabla p + \nabla.\rho\nu[2S] + F$$

where $U$ is the effective velocity, $\rho$ is the density, $p$ is the pressure, $\nu$ is the kinematic viscosity, $[S]$ is the stress tensor and $F$ is the combined sources term. The average values are calculated from the volume fraction $\alpha$

The Volume of fluid equation is as follows

$$\frac{\partial \alpha}{\partial t} + \nabla.(\alpha U) + \nabla.[U_r\alpha(1-\alpha)] = 0$$

where $\alpha$ is volume fraction of fluid. $U_r$ is the compression velocity which is given by $U_r = U_l - U_g$ and $U$ is the effective velocity obtained by weighted average $U = \alpha U_l + (1-\alpha)U_g$ where the subscripts $l$ and $g$ denotes liquid and gaseous phase respectively. The weighted average values of density $\rho$ and kinematic viscosity $\nu$ is calculated from the equations below.

$$\rho = \rho_l\alpha + \rho_g(1-\alpha)$$

$$\nu = \nu_l\alpha + \nu_g(1-\alpha)$$

The calculated weighted average values of density and kinematic viscosity are substituted in momentum equation and continuity equation to solve for the field variables.

# Chapter 2

# Case Setup

## 2.1 Geometry and Mesh

The Geometry of the fluid domain is a simple cube of side length 0.1m. The bottom side of the cube is named as walls, the surface for which hydrophilic or hydrophobic boundary condition will be given, all other sides atmosphere. The mesh is generated using blockMesh utility of OpenFOAM, with local cell size 0.005m with simple grading on all sides. The mesh is coarse initially but will be refined during simulation according to the volume fraction of water.

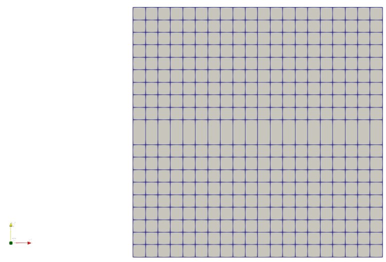

Figure 2.1: Initial Mesh

## 2.2 Boundary conditions

The boundary conditions employed for the computational domain is given in table 2.1. The alpha water boundary condition for the walls govern the hydrophilic and hydrophobic nature of the surface. For hydrophilic theta0 is 45$^o$ and for hydrophobic theta0 is 135$^o$.
The water droplet is given as an initial condition using setFields. Region used to set water level is as follows

```
sphereToCell
  {
  origin (0.0 0.0 0.0);
  radius 0.01;
  centre (0.05 0.05 0.05);
  fieldValues ( volScalarFieldValue alpha.water 1 ;
  }
```

## 2.3 Adaptive Mesh Refinement

In numerical analysis, adaptive mesh refinement (AMR) is a method of adapting the accuracy of a solution within certain sensitive regions of simulation, dynamically and during the time the solution is being calculated, refining the mesh. Adaptive Mesh Refinement is implemented in OpenFOAM in dynamicMeshDict in constant directory.

The dynamicMeshDict is as follows,

```
  dynamicFvMesh dynamicRefineFvMesh;
  dynamicRefineFvMeshCoeffs {
      refineInterval 1;
      field alpha.water;
      lowerRefineLevel 0.1;
      upperRefineLevel 0.9;
      unrefineLevel 0.005;
      nBufferLayers 1;
      maxRefinement 3;
      maxCells 1000000;
      dumpLevel true;
      correctFluxes ((phi none) (nHatf none) (rhoPhi none) (ghf none));
  }
```

| **Boundary** | **U** | **p_rgh** | **alpha.water** |
|---|---|---|---|
| **atmosphere** | type inletOutlet;<br>value (0 0 0); | type fixedValue;<br>value uniform 0; | type inletOutlet;<br>value uniform 0; |
| **walls** | type noSlip; | type fixedFluxPressure;<br>Value uniform 0; | type<br>constantAlphaContactAngle;<br>theta0 45;<br>limit gradient;<br>value uniform 0; |

Table 2.1: Boundary Conditions for Hydrophilic Surface

| Sl.No. | Description | Value |
|---|---|---|
| 1 | Kinematic viscosity of water | 1e-06 $m^2/s$ |
| 2 | Density of water | 1000 $kg/m^3$ |
| 3 | Surface Tension | 0.072 N/m |
| 4 | Kinematic viscosity of air | 1.48e-05 $m^2/s$ |
| 5 | Density of air | 1 $kg/m^3$ |

Table 2.2: Properties of Fluids

## 2.4 Solver and Simulation control

interFoam solver is used for this simulation. The official definition for this solver is as follows:

Solver for 2 incompressible, isothermal immiscible fluids using a VOF (Volume of Fluid) phase-fraction interface capturing approach. The VOF model can model two or more immiscible fluids by solving a single set of momentum equations and tracking the volume fraction of each of fluids throughout the domain. Typical applications include the prediction of jet breakup, the motion of large bubbles in a liquid, the motion of liquid after a dam break, and the steady or transient tracking of any liquid-gas interface. Simulation control values are given in controlDict in system folder of the case folder. The time step is adjustable with respect to courant number as well alpha courant number.

The transport properties for case is setup in transportProperties file in constant directory in which the value of density and kinematic viscosity for both air and water are given as well as surface tension value is also given.

The important control values as given in controlDict is as follows

```
startTime 0;
endTime 1;
deltaT 0.000001;
writeControl adjustableRunTime;
writeInterval 0.005;
maxCo 1;
maxAlphaCo 1;
maxDeltaT 0.001;
```

# Chapter 3

# Result and analysis

## 3.1 Mesh Refinement

The mesh is refined at each time interval based on the volume fraction of water (alpha.water) field value where the values are above 0.9 and less than 0.1, upto a level of 3. The refining of mesh during simulation of hydrophilic surface can be seen in the figure 3.1. The same can be seen for hydrophobic surface also.

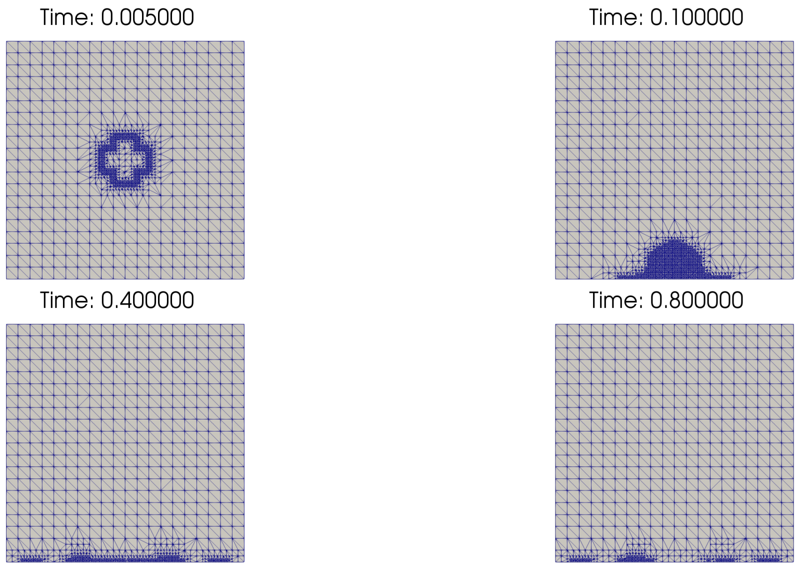


Figure 3.1: Mesh Refinement at different time steps

## 3.2 Volume Fraction of Water

The result of field alpha, which is the volume fraction of water, will show us the behavior of water droplet impacting on both the hydrophilic and hydrophobic surface.

On the Hydrophilic surface, the droplet seems to be adhering to the surface. This happens due to the value of contact angle being less than 90°. On the hydrophobic surface, the droplet seems to flow without much adherence. This may be due to molecular structure. This behavior can be seen from the contour plot of volume fraction of water as seen in Figure 3.2.

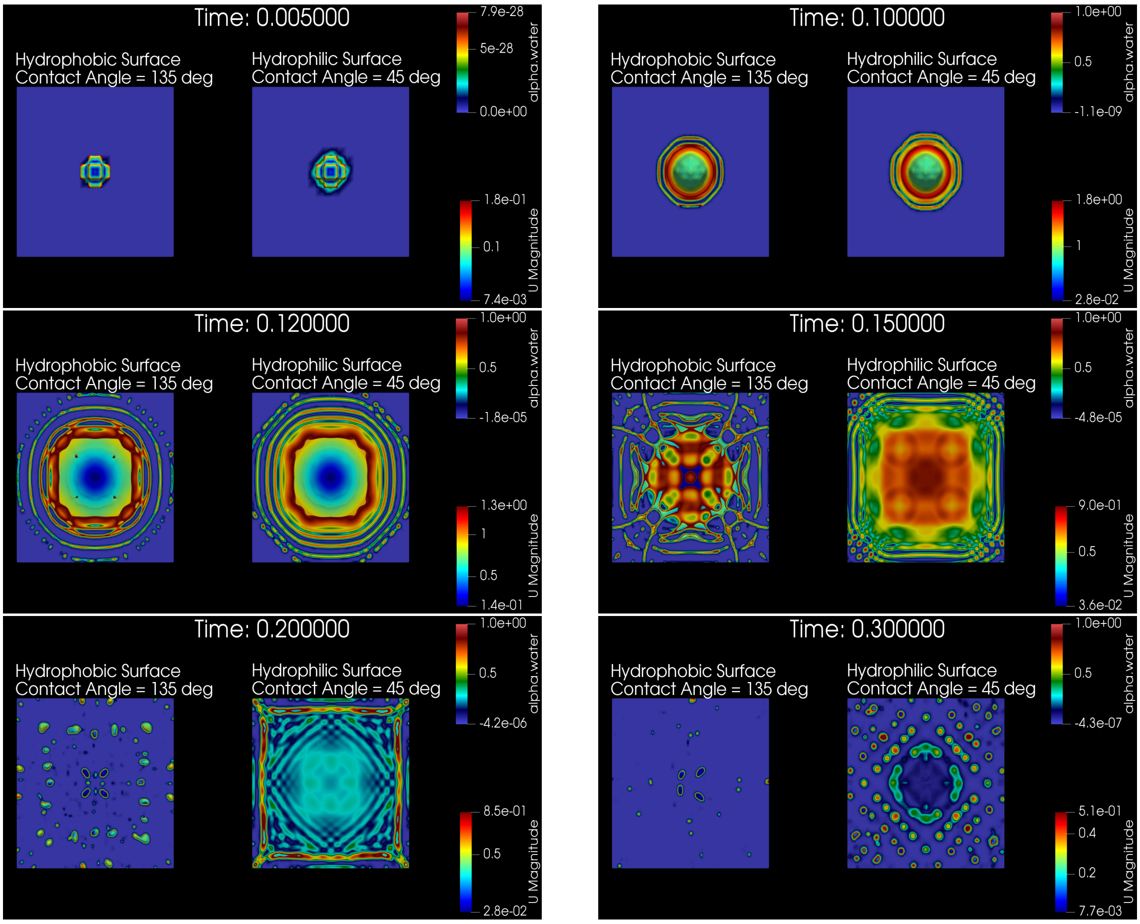


Figure 3.2: alpha field (volume fraction of water) at different time steps

# Chapter 4

# Conclusion

This project has been done to analyse the behavior of water droplet on hydrophilic and hydrophobic surfaces and the results from the simulation are of expected behavior. This case study explores the capability of OpenFOAM for multiphase modelling combined with adapting mesh refinement.

# Reference


- OpenFOAM User Guide
- Tutorial Case for Mesh Refinement by Tobias Holzmann
- OpenFOAM wikipedia page
- interFoam explanation from openfoamwiki.net
- How to post-process multiphase results - openfoamwiki.net
- "Description and utilization of interFoam multiphase solver" by Santiago Mrquez Damin-Final Work-Computational Fluid Dynamics