Swedish Cement and Concrete Research Institute
Royal Institute of Technology

OpenFOAM Tutorial: Modeling Free Surface Flow using multiphaseInterFoam

Author:
Annika Gram
annika.gram@cbi.se

Tutorial multiphaseInterFoam

1. Introduction

Layered concrete casting in a formwork may be simulated using OpenFOAM. The case presented below is a sameller scale layered simulation with water poured into a cup. The tutorial is an introduction to multiphaseInterFoam simulating water splashing on a settled layer of concrete. The concrete is modeled as a Herschel-Bulkley material (see section 2) and the multiphaseInterFoam solver is picked specifically for its ability to to solve incompressible multiphase transient flow. It is based on the Volume of Fluid, VOF, method for tracking of fluid interface. Every cell holding fluid carries a marker, [1]. The case can be extended, a few examples are shown in the last chapter: changing the HerschelBulkley coefficients, picking a different viscosity model and adding another phase.

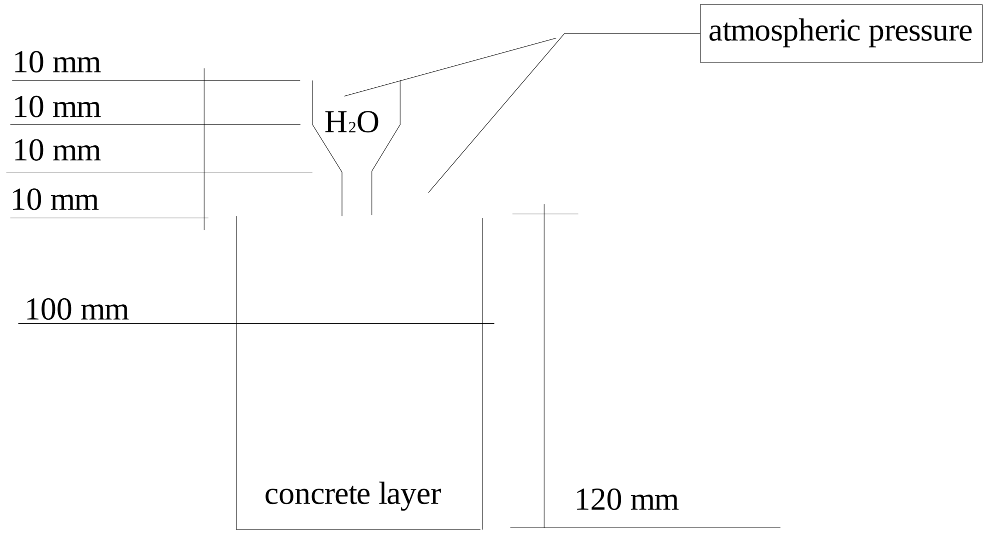
The following is covered in the tutorial:

– Generating of a CFD mesh using **`blockMesh`**
– Defining the geometry of the phases using **`setFields`**
– Solving a case using **`multiphaseInterFoam`**
– Postprocessing with **`paraFoam`**

2. Description of the Case

The geometry for this case is based on the filling bottle case from 2007: Free Surface Tutorial Using InterFoam and rasInterFoam by Hassan Hemida found at: http://www.tfd.chalmers.se/~hani/kurser/OS_CFD_2007/ . However, to keep computation time down, the bottle was shortened. The water container is 20 mm wide, its faucet is 10 mm:

Figure 1.



The material model for the concrete has been selected to be modeled using Herschel-Bulkley. A Herschel-Bulkley material exhibits an initial resistance to flow (elastic part). Once the level of yield stress, τ_0 , has been exceeded, the material flows (elastic part) according to the following stress-shear relation:

$$\tau = \tau_0 + \mu_{pl}\, \gamma^n$$

with μ_{pl} being the plastic viscosity and γ^n the shear rate of the material [2].
Division by shear rate and density of the material renders the kinematic viscosity, $\mu = \tau/(\rho\, \gamma^n)$. As the concrete ages, the level of yield stress τ_0 and plastic viscosity μ_{pl} will get higher, until the concrete stiffens completely, $\tau_0 = \tau_0\,(t,T)$.

3. Case Files

Copy the `splash` case provided from http://www.tfd.chalmers.se/~hani/kurser/OS_CFD_2008 to your working directory. The case files should now be found in:

`$FOAM_RUN/splash`

The case now contains the following floders:

`0`	holding files alphas, alphaair, alphacrete, alphanewcrete, alphawater, pd (specific pressure) and U as well as subdirectory hide for the original alpha files
`constant`	holding files dynamicMeshDict, environmentalProperties and transportProperties as well as subdirectory polyMesh for the files blockMeshDict, boundary, faces, neighbour, owner and points
`system`	holding files controlDict, decomposeParDict, fvSchemes, fvSolution, setFieldsDict

In this tutorial, we will take a closer look at the alpha files, transportProperties, blockMeshDict and setFieldsDict.

4. The Mesh

A fully-structured mesh of the geometry shown in figure 1 will first be created. There is no need to change anything in the `blockMeshDict` file, unless a refinement or change of the mesh is wished for.

5. Case Definition

The definition of the case in this example can be divided into three parts: initial and boundary conditions, material properties and solver controls.

5. Initial and Boundary Conditions

The front and back walls are defined empty, as in a 2D simulation. The inlet patch is in this case atmospheric, water is filled in the funnel on top and pours into the container. The walls of the container are perfectly smooth, holding the fluid in place.

Initial and boundary conditions are specified in the `0` directory.

Table 1. Initial and boundary conditions

Variable	Boundary Conditions	Variable
pd	zeroGradient,	fixedValue and uniform 0 for atmosphere
U	fixedValue uniform (0 0 0)	inletOutlet for atmosphere
alphas	zeroGradient	
alphaair	alphaContactProperties thetaProperties 90 0 0 0 uniform 0	inletOutlet for atmosphere

`thetaProperties` is a heuristical parameter created by Henry Weller, defining the angle between the fluid and the adjacent wall. An alphas field list is created by the `setFieldsDict` in the alpha files when running the **`setFields`** command. This means, these files are altered according to your geometry. In order to be able to retrieve the original files, a complete set of alphas is stored in the `hide` directory, to be copied and used in case the geometry of the case has changed.

5.2 **Material Properties**

Material properties can be found in the `constant` directory.

Table 2. Material properties

File	Initial Conditions
environmentalProperties	gravity
transportProperties	material constants, see below

```
/*--------------------------------*- C++ -*----------------------------------*\
| =========                 |                                                 |
| \\      /  F ield         | OpenFOAM: The Open Source CFD Toolbox           |
|  \\    /   O peration     | Version:  1.5                                   |
|   \\  /    A nd           | Web:      http://www.OpenFOAM.org               |
|    \\/     M anipulation  |                                                 |
\*---------------------------------------------------------------------------*/
FoamFile
{
    version     2.0;
    format      ascii;
    class       dictionary;
    object      transportProperties;
}
// * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * //

phases
(


water
{
    transportModel  Newtonian;
    nu              nu [0 2 -1 0 0 0 0]  1e-6;
    rho             rho [1 -3 0 0 0 0 0] 1000;
}


/*
newcrete
{
    transportModel  HerschelBulkley;
    nu              nu [0 2 -1 0 0 0 0]  1;
    rho             rho [1 -3 0 0 0 0 0] 1000;

    HerschelBulkleyCoeffs
    {
    tau0            tau0 [0 2 -2 0 0 0 0] 0;
       k               k [0 2 -1 0 0 0 0] 0.005;
       n               n [0 0 0 0 0 0 0] 1;
       nu0       nu0 [0 2 -1 0 0 0 0] 10e6;
     }
```

```
}

*/

crete
{
    transportModel   myHerschelBulkley;
    nu              nu [0 2 -1 0 0 0 0]  1;
    rho             rho [1 -3 0 0 0 0 0] 2000;

    HerschelBulkleyCoeffs
    {
    tau0            tau0 [0 2 -2 0 0 0 0] 0.0;
       k               k [0 2 -1 0 0 0 0] 0.001;
       n               n [0 0 0 0 0 0 0] 1;
       nu0       nu0 [0 2 -1 0 0 0 0] 10e5;
    }
}


air
{
        transportModel  Newtonian;
        nu              nu [0 2 -1 0 0 0 0] 1.48e-05;
        rho             rho [1 -3 0 0 0 0 0] 1;
}
);

refPhase air;

sigmas
(
    (air water) 0.07
    (air crete)   0.07
//    (air newcrete)   0.07
    (water crete) 0.07
//    (water newcrete) 0.07
//    (crete newcrete) 0.07
);

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

Material properties for phases crete and water were defined. Constants for 'newcrete' have been prepared. Values for surface tension sigma are all 0.07, same as for water.

5.2 **Solver Controls**

The `setFieldsDict` file can be found in the `system` directory. This is where the area/volume of the placed phases are defined, by creating one or several geometrical boxes to be initially filled with fluid. Running the setFields command renders a numerical value for alphas: 0, 1, 2, 3 ... one value in every cell for each phase included.

The `controlDict` file states startTime and endTime, we will halt the computations at 0.2 s.

6. Solving the Case

The OpenFOAM solver multiphaseInterFoam is used to resolve the two-dimensional incompressible flow. Standing in the case directory, we will first mesh the case, set the initial fields for gamma and run the simulation:

```
blockMesh
checkMesh
setFields
multiphaseInterFoam
```

A series of result files and folders are created inside the case.

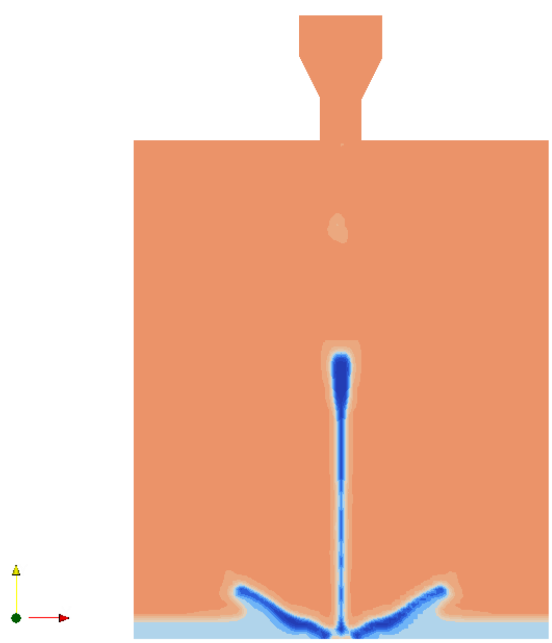
7. Post-processing

The results for this example will be post-processed in Paraview.

```
foamToVTK
paraview
```

Open the case and select the alphas option to view the phases.

Figure 2: Drop of water in the fresh concrete phase



8. Changing the transportProperties

When changing the crete transportModel in the `transportProperties` file to the HerschelBulkley case k = 1 and running the case again, the solution at t = 0.2 s, a somewhat different scenario follows:

```
crete
{
    transportModel  HerschelBulkley;
    nu              nu [0 2 -1 0 0 0 0]  1;
    rho             rho [1 -3 0 0 0 0 0] 2000;

    HerschelBulkleyCoeffs
    {
    tau0            tau0 [0 2 -2 0 0 0 0] 0;
       k               k [0 2 -1 0 0 0 0] 1;
       n               n [0 0 0 0 0 0 0] 1;
       nu0       nu0 [0 2 -1 0 0 0 0] 10e5;
    }
}
```

Figure 3: Water dripping on top of hardened concrete phase

9. Adding another Phase

One may wish to simulate an 'aged' concrete, meaning to change its material properties, or to pour new concrete, named 'newcrete' in the case on top of an older concrete layer.
In order to continue the layering or casting of the concrete in a wall for example, the simulations should run until the phases are settled at the bottom of the cup before filling the funnel with new fluid. The last `alphas` time directory file is now to be copied into a new case. Since the **`setFields`** command will erase all old data, two alpha fields need to be combined:

Sum the alpha `internalField nonuniform List<scalar>` list from the last time directory with the alpha `internalField nonuniform List<scalar>` list from the new fluid filling up the funnel by using for example the **`setFields`** command. The lists may be summed using for example a calculator spreadsheet like Excel. Copy the obtained alpha values into your new alpha `internalField nonuniform List<scalar>` list placed in the 0 directory for the new calculation.

Uncommentthe sigma and wall values for the newcrete material in the `alphaair` file. Uncomment the lines in the `transportProperties` file that were created for newcrete.

Run the case.

References

[1] C. Hirt, B. Nichols, Volume of Fluid (VOF) method for the dynamics of free boundaries, Journal of Computational Physics **39** (1981) 201-225.

[2] C.W.Macosco, Rheology Principles, Measurements and Application, VCH Publishers, 1994.