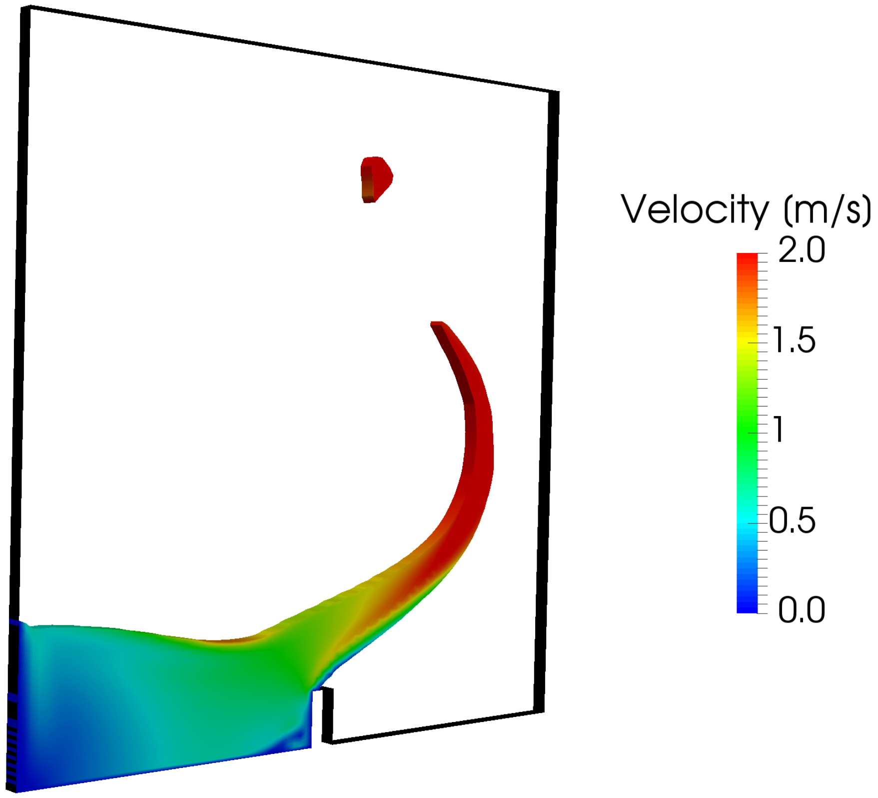
Tutorial Eight

Multiphase



5th edition, Sep. 2019

Compatibility:

- OpenFOAM® 7
- OpenFOAM® v1906

Cover picture from:

- Bahram Haddadi

Contributors:

- Bahram Haddadi
- Clemens Gößnitzer
- Jozsef Nagy
- Vikram Natarajan
- Sylvia Zibuschka
- Yitong Chen

ISBN 978-3-903337-00-8

Publisher: chemical-engineering.at

For more tutorials visit: www.cfd.at

Background

In this tutorial we are going to solve a problem of dam break using the *interFoam* solver. The main feature of this problem is flow of water and air separated by a sharp interface. Before starting, let's cover some of the basics of multiphase flow.

1. Multiphase flow

Multiphase flow is simultaneous flow of materials in different phases. There can be multiple components present in each phase. The common types of multiphase flows are: gas-liquid, gas-solid, liquid-solid, liquid-liquid and three-phase flows.

Multiphase flow can be further categorized based on the visual appearance of the flow into separated, mixed or dispersed flow. In dispersed flow, one phase exist as a continuous fluid, while all other phases act as discontinuous particles flowing through the continuous fluid. In mixed flow regions, dispersed particles as well as semi-continuous interfaces exist together.

So why is multiphase flow important? Multiphase flow is present in many industrial processes, such as bubble columns, absorption, adsorption and stripping columns. Modeling of multiphase flow can help maximizing contact between different phases, hence increasing the efficiency of the process.

2. Modeling approaches

Modeling of multiphase flow can be extremely complex, due to possible flow regime transitions. To simplify the matter, different modeling approaches can be adopted and they generally fall into two categories: lagrangian and Eulerian. In the case of dispersed configuration, Lagrangian approach is more suitable. This involves tracking individual point particles during its movement. The other approach is the Eulerian approach, which observes fluid behavior in a given control volume.

Below we will cover some common modeling approaches of multiphase flow.

2.1. Euler-Euler approach (Multi-fluid model)

All phases are treated as continuous in the Euler-Euler approach. This approach is suitable for separated flows where each phase behaves as a continuum, rather than being discrete. The phases interact through the drag and lift forces acting between them, as well as through heat and mass transfer. The Euler-Euler approach is also capable of modeling dispersed flow, where we are interested in the overall motion of particles rather than tracking individual particles.

In the Euler-Euler approach, we introduce the concept of phasic volume fractions. These fractions are assumed to be continuous functions of space and time, with their sum equal to one. For each phase, a set of conservation equations for mass, momentum and energy is solved individually; in addition, a transport equation for the volume fraction is solved. Coupling between the phases is achieved through a shared pressure and interphase exchange coefficients.

2.2. Eddy Interaction Model

In the Eddy Interaction Model, each particle interacts with a succession of eddies. The fluid motion of the particle is characterized by three parameters: i) eddy velocity, ii) eddy lifetime, iii) eddy length. It follows the particle-tracking Lagrangian approach.

The eddy lifetime (t_e) and eddy length scale (l_e) are estimated from the local turbulence properties. From the length scale and the particle velocity, one can calculate the eddy transit time (t_c), i.e. the time taken for a particle to cross the eddy. The particle is then assumed to interact with the eddy for a time which is the minimum of the eddy life time and the eddy transit time.

$$t_{int} = \min(t_e, t_c)$$

During that interaction the fluid fluctuating velocity is kept constant and the discrete particle is moved with respect to its equation of motion. Then a new fluctuating fluid velocity is sampled and the process is repeated.

2.3. Volume of Fluid (VOF) method

VOF method belongs to the Eulerian class of modeling approach. It is based on the idea of **fraction function C**. Fraction function indicates whether a chosen phase is present inside the control volume. If C=1, the control volume is completely filled with the chosen phase; if C=0, the control volume is filled with a different phase. A value between 0 and 1 indicates that the interface between phases is present inside the control volume. It is important in VOF method that the flow domain is modeled on a fine grid, i.e. the interface should be resolved.

The focus of the VOF method is to track the interface between phases. To do this, the transport equations are solved for mixture properties, assuming that all field variables are shared between the phases. Then an advection equation for the fraction function C is solved. The discretization of the fraction function equation is crucial for obtaining a sharp interface.

The multiphase flow in this tutorial is analysed using the *interFoam* solver. Here is a brief explanation of the solver below.

3. interFoam solver

interFoam is suitable for solving multiphase flow between 2 incompressible, isothermal immiscible fluids. It is based on the Volume of Fluid (VOF) approach.

interFoam – damBreak

Simulation

Use the interFoam solver to simulate breaking of a dam for 2s.

Objectives

- Understanding how to set viscosity, surface tension and density for two phases

Data processing

See the results in ParaView.

1. Pre-processing

1.1. Copy tutorial

Copy tutorial from the following folder to your working directory:

```
$FOAM_TUTORIALS/multiphase/interFoam/laminar/damBreak/damBreak
```

1.2. 0 directory

In the 0 directory the following files exist:

```
alpha.water.orig  p_rgh  U
```

OpenFOAM® v1906: `alpha.water` also exist!

In the alpha.water.orig and p_rgh files the initial values and also boundary conditions for water phase and also pressure are set. Copy alpha.water.orig to alpha.water (remember: the *.orig files are back up files, and solvers do not use them). E.g. in alpha.water:

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

dimensions      [0 0 0 0 0 0 0];

internalField   uniform 0;

boundaryField
{
    leftWall
    {
        type            zeroGradient;
    }

    rightWall
    {
        type            zeroGradient;
    }

    lowerWall
    {
        type            zeroGradient;
    }

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

    defaultFaces
    {
        type            empty;
    }
}
// * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * *//
```

Note: The `inletOutlet` *and the* `outletInlet` *boundary conditions are used when the flow direction is not known. In fact, these are derived types and are a combination of two different boundary types.*

- `inletOutlet`*: When the flux direction is toward the outside of the domain, it works like a zeroGradient boundary condition and when the flux is toward inside the domain it is like a fixedValue boundary condition.*

- `outletInlet`*: This is the other way around, if the flux direction is toward outside the domain, it works like a fixedValue boundary condition and when the flux is toward inside the domain, it is like a zeroGradient boundary condition.*

E.g. if the velocity field outlet is set as `inletOutlet` *and the* `inletValue` *is set to* `(0 0 0)`*, it avoids backflow at the outlet! The "*`inletValue`*" or "*`outletValue`*" are values for* `fixedValue` *type of these boundary conditions and "*`value`*" is a dummy entery for OpenFOAM® for finding the variable type. Using* `(0 0 0)`*, OpenFOAM® understands that the variable is a vector.*

1.3. constant directory

In the *transportProperties* file the properties of two phases can be set under each phase sub-dictionary, e.g. `water` or `air`:

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

phases (water air);

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

air
{
    transportModel  Newtonian;
    nu              1.48e-05;
    rho             1;
}

sigma           0.07;

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

In both phases the coefficients for different models of viscosity are given, e.g. `nu` and `rho`. Depending on which model is selected, the coefficients from the corresponding sub-dictionary are read. The selected model is `Newtonian`, only the `nu` coefficient is used.

`sigma` is the surface tension between two phases, in this example it is the surface tension between air and water.

Checking the g file, the gravitational field and also its direction are defined, it is `9.81` m/s^2 in the negative y direction.

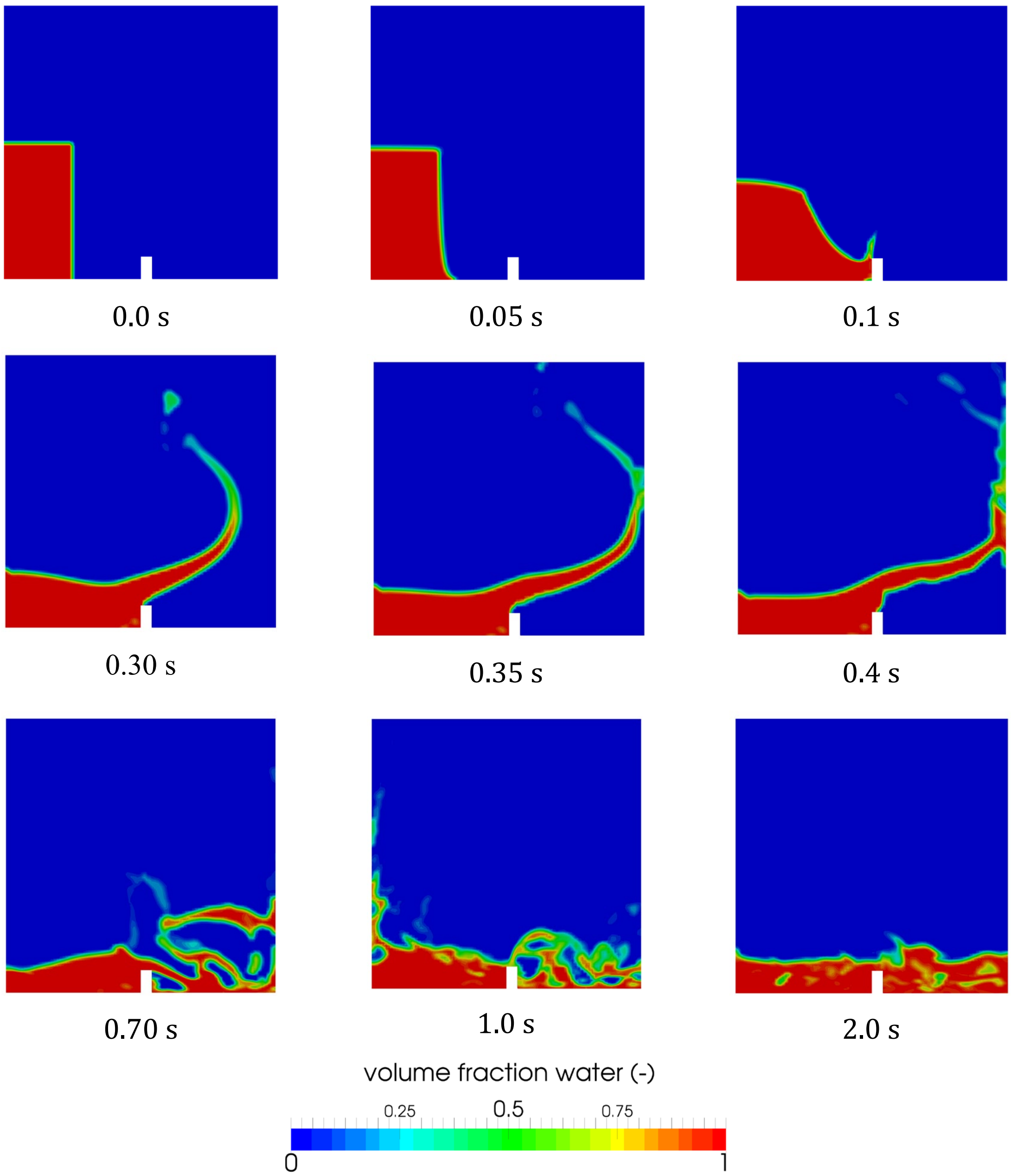
```
// * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * *//


dimensions      [0 1 -2 0 0 0 0];
value           ( 0 -9.81 0 );

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

2. Running simulation

```
>blockMesh
>setFields
>interFoam
```

3. Post-processing

The simulation results are as follows (these are not the results for the original mesh, but a 2x refined mesh):



Contours of the water volume fraction at different time steps

ISBN 978-3-903337-00-8
9 783903 337008