OpenFOAM

twoPhaseEulerFoam

Praveen Prabhu Baila

Content

- Introduction
- Fluidized bed tutorial
- Running the solver
- Postprocessing

Introduction

- Used to solve two incompressible fluid phases with one of the phases dispersed
- Continuum approach is used for both the phases with Eulerian form of conservation equations

- Momentum equation

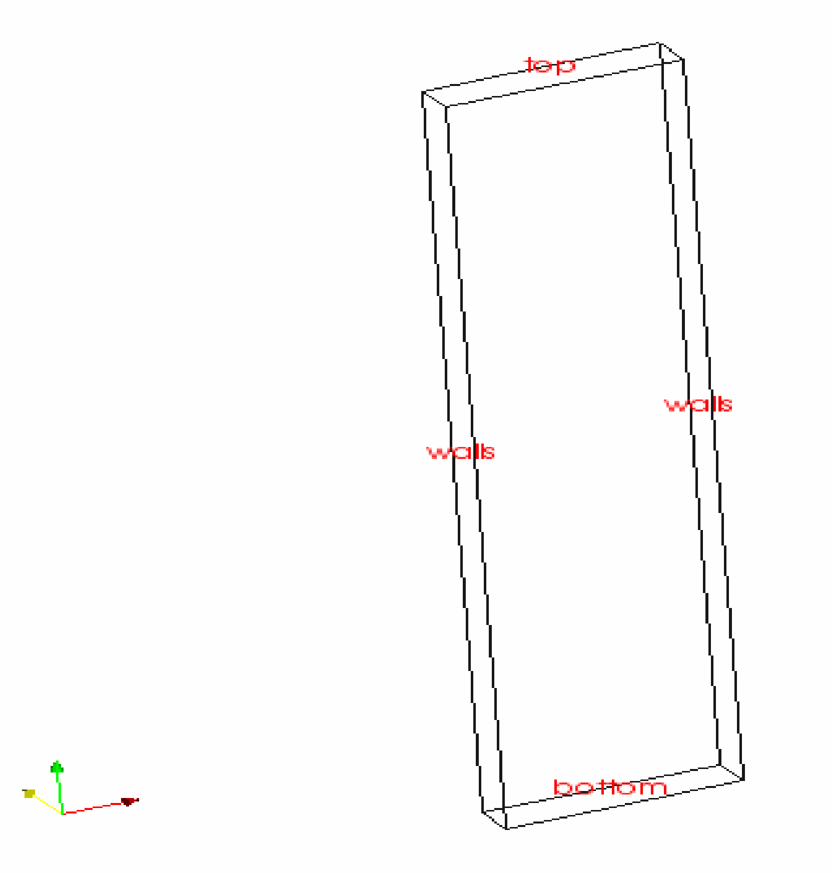
$$\frac{\partial \alpha_\varphi \overline{U}_\varphi}{\partial t} + \nabla \cdot \left(\alpha_\varphi \overline{U}_\varphi \overline{U}_\varphi \right) + \nabla \cdot \left(\alpha_\varphi \overline{R}_\varphi^{eff} \right) = -\frac{\alpha_\varphi}{\rho_\varphi} \nabla \overline{p} + \alpha_\varphi g + \frac{\overline{M}_\varphi}{\rho_\varphi}$$

- Continuity equation

$$\frac{\partial \alpha_\varphi}{\partial t} + \nabla \cdot \left(\overline{U}_\varphi \alpha_\varphi \right) = 0$$

- Inter-phase momentum transfer forces – drag, lift, virtual mass

twoPhaseEulerFoam tutorial



Fluid injected at 0.9 m/s
from the bottom surface

Copy the tutorial into the working directory

cp –r $FOAM_TUTORIALS/twoPhaseEulerFoam/bed $FOAM_RUN

cd run/bed

constant/polymesh – geometry, mesh parameters

Inputs for solver - *constant/*

environmentalProperties	:	Gravity
interfacialProperties	:	Drag model and specify the discrete phase of the 2 phases
kineticTheoryProperties	:	Constants and models for Kinetic Theory of granular flow
ppProperties	:	Specify packing limit and constants for calculation of particle-particle forces
transportProperties	:	Specify viscosity, diameter of particle and density of two phases

Control of the solver, solution methods and schemes can be found in directory *system/*

Specifying the discrete phase – *constant/InterfacialProperties*

dragModela GidaspowSchillerNaumann;

dragModelb GidaspowSchillerNaumann;

dragPhase a;

In this tutorial

Phase a - discrete phase

Phase b - continuous phase

transportProperties for the phases

```
// sand
phasea
{
    rho        rho      [1 -3 0 0 0]        2640;
    nu         nu       [0 2 -1 0 0]        1.0e-6;
    d          d        [0 1 0 0 0 0 0]     480.0e-6;
}

// air
phaseb
{
    rho        rho      [1 -3 0 0 0]        1.28;
    nu         nu       [0 2 -1 0 0]        1.328e-5;
    d          d        [0 1 0 0 0 0 0]     1.0;
}
```

Initial condition 0/

variable	Initial conditions
alpha	internalField uniform 0.33, zeroGradient at walls
p	internalField uniform 0, zeroGradient at walls
Ua	internalField uniform 0, walls fixedvalue uniform (0 0 0)
Ub	internalField uniform 0, walls fixedvalue uniform (0 0 0)
Theta	internalField uniform 0, zeroGradient at walls
k	internalField uniform 1.0, zeroGradient at walls
epsilon	internalField uniform 10.0, zeroGradient at walls

Boundary condition Ub,

```
boundaryField
{
    bottom
    {
        type            fixedValue;
        value            uniform (0 0 0.9);
    }
```

The geometry is meshed using blockMesh command

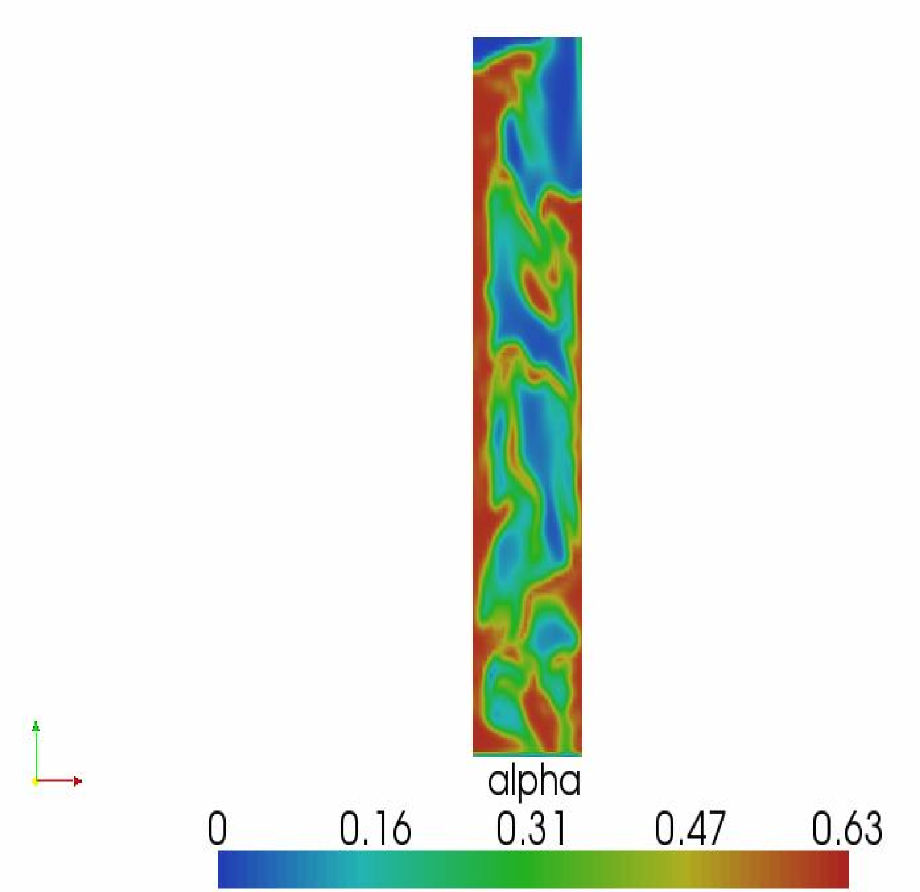
The twoPhaseEulerFoamSolver is started

```
cd $FOAM_RUN/bed
blockMesh
twoPhaseEulerFoam
```

PostProcessing

The mesh and the results are loaded in ParaFoam

```
cd $FOAM_RUN/bed
paraFoam
```



Thank You