

Project in TME 050

*A twoPhaseEulerFoam tutorial*

Praveen Prabhu Baila

## Introduction

The twoPhaseEulerFoam solver is described in the present work.

Two incompressible fluid phases with one phase dispersed are solved using this solver. Both the phases are described using the Eulerian conservation equations and thus it is referred as Euler-Euler model. Each of the phases are treated as a continuum in this approach.

The tutorial gives a brief description how to simulate two phase flows using the twoPhaseEulerFoam solver.

In this tutorial, a fluidized bed is simulated. The fluidized bed column is made of a single block of 0.5 X 0.1 X 2.5 m as shown in Figure 1. The block is discretised uniformly with 25 cells in the x-direction, 125 cells in the Z direction and 1 cell in the y-direction to simulate a 2D flow. The column is filled with sand particles of 480 micro- meter diameter and air is blown from the bottom with a velocity of 0.9m/s. There are options for changing the physical properties of the two phases and also specify which fluid is the discrete phase of the two phases.

The numerical simulation generally consist of 3 steps

- the pre-processing
- Solving of discretised equations
- Post processing



Figure 1. Geometry of the column

## 1. Pre-Processing.

The tutorial files for this solver can be found in the *twoPhaseEulerFoam* directory and are copied to the run directory.

```
cp -r $FOAM_TUTORIALS/twoPhaseEulerFoam/bed $FOAM_RUN
cd $FOAM_RUN/bed
```

The files in the tutorials are similar to any other tutorials with 0, system and constant directories. There are different files in the constant directory which are the necessary inputs for this solver.

### 1.1 Mesh

The mesh file can be found in the *constant/polymesh/blockMeshDict* file with dimensions and mesh parameters for the column. A uniform mesh is created for the column without any grading. The mesh is generated in a usual way by typing the *blockMesh* command in the terminal window.

### 1.2 Specifying discrete phase

The solver is implemented for two incompressible fluids 'a' and 'b' and the discrete phase can be specified here. The *constant/interfacialProperties* file is used to specify which of the two phases is to be considered as the discrete phase and also specify the drag model for the two phases to calculate the drag coefficient  $\beta$ .

The *dragPhase* is used to specify which of the two phases is the discrete phase.

When *dragPhase* is set to 'a' as shown below then the phase 'a' is taken as the discrete phase and the other phase as continuous phase

```
dragModela  GidaspowSchillerNaumann;
dragModelb  GidaspowSchillerNaumann;

dragPhase   a;
```

After specifying the discrete phase, the diameter corresponding to discrete phase is obtained from the *transportProperties* file for the calculation of drag forces.

### 1.3 Inlet and outlet boundary conditions

The phase b is taken as the continuous phase and thus in *0/Ub* file the velocity is set as 0.9m/s at the inlet i.e. the bottom boundary. The derived boundary condition *inletOutlet* is used at the top. This switches *Ub* between fixed value and *zeroGradient* depending on the direction of the flux at the each boundary face.

The other fields that are present are alpha - volume fraction or void fraction which is ratio of volume of the discrete phase to that of the control volume with fixed value of 0 at the inlet, Theta – granular temperature, p – pressure, Ua – velocity of discrete phase and also turbulence properties k and epsilon for which the boundary conditions have to be specified.

#### 1.4 Initial and wall boundary Conditions

The fluidized bed is filled with the sand particles and thus the alpha field is initialized to some value in the whole domain. The setFieldsDict dictionary can be used to initialize the discrete phase only in certain region. The pressure is initialized to zero as the fluid is incompressible. Theta represents the granular temperature which indicates the fluctuating motion of the particles. As the velocity of both the phases is zero initially, the value of this variable is also set to zero. The initial and the wall boundary conditions for different variables are given below in the table 1.

variable	Initial and wall boundary 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

Table: 1 Initial and wall boundary conditions

#### 1.5 Physical properties

There are certain files in the *constant/* directory which help in setting different models and properties for simulating two phase flows. The property files are

##### 1. *environmentalProperties* –

The value of the gravity and its direction is specified in this file.

```
g          g          [0 1 -2 0 0] (0 0 -9.81);
```

##### 2. *InterfacialProperties* –

The different models to calculate the drag transfer coefficient for the 2 phases is specified in this file. Also which of the two phases is the discrete phase is set here.

```
dragModela  GidaspowSchillerNaumann;
dragModelb  GidaspowSchillerNaumann;
```

dragPhase a;

The first two lines indicate the different models to calculate the drag transfer coefficient which is further used for the calculation of drag force. When the drag model is read, the member function related to the drag model is called.

The different models that are available for the calculation of interphase drag transfer coefficient constant in the momentum equation. They are

- Ergun
- Gibilaro
- GidaspowErgunWenYu
- GidaspowSchillerNaumann
- SchillerNaumann
- SymlalOBrien
- WenYu

The most of models use values of Reynolds number - Re, phase fraction – alpha, relative velocity between the phases - Ur, drag coefficient - Cd. Also the different models use different expressions for the calculation of Cd depending on the Reynolds number. For example GidaspowSchillerNaumann use the expression for Cd as

$$C_D = \frac{24}{Re} * (1 + 0.15 * (Re)^{0.687})$$

if the Reynolds number  $Re < 1000$  and a constant value of 0.44 if  $Re > 1000$

### 3. *kineticTheoryProperties* –

The different constants and the models for calculation of different terms required for the Euler solver are specified in this file.

The effective stresses in the solid phase resulting from the particle streaming and collisions can be described using gas kinetic theory. If equilibrium is on then the algebraic equation for granular temperature is solved instead of the balance equation. In this approach is valid when the volume fraction of the solid phase stays high and the velocity of the solid phase stays relatively low. The co-efficient of restitution is denoted by e. The frictional stresses must be considered in the solid phase stress when the solids volume fraction is high. The alphaMax value represents the maximum packing limit of the discrete phase. The alphaMinFriction is the value of solid volume fraction when the frictional stresses become important.

Fr, eta, p are the material dependent constants used for the calculation of normal frictional stress.

The frictional shear viscosity is related to the frictional normal stress by the linear law proposed by coulomb given as

$$\mu_f = P_f \sin \phi$$

where  $P_f$  is the frictional normal stress and phi is the angle of internal friction of the particle.

```

equilibrium          on;

e                    e          [ 0 0 0 0 0 0 0 ] 0.9 ;
alphaMax             alphaMax   [ 0 0 0 0 0 0 0 ] 0.6 ;
alphaMinFriction     alphaMinFriction [ 0 0 0 0 0 0 0 ] 0.5 ;
Fr                   Fr          [ 1 -1 -2 0 0 0 0 ] 0.05 ;
eta                  eta         [ 0 0 0 0 0 0 0 ] 2.0 ;
p                    p           [ 0 0 0 0 0 0 0 ] 5.0 ;
phi                  phi         [ 0 0 0 0 0 0 0 ] 25.0 ;

viscosityModel       Syamlal;
conductivityModel    HrenyaSinclair;
granularPressureModel Lun;
frictionalStressModel JohnsonJackson;
radialModel          Gidaspow;

HrenyaSinclairCoeffs
{
    L                 L          [ 0 1 0 0 0 0 0 ] 5.0e-4 ;
}

```

The different models are described for the calculation of solids shear viscosity, conductivity of granular energy, solids pressure co-efficient (modeled as ratio of solids pressure to granular temperature in OpenFoam solver), frictional stress, radial distribution function respectively.

#### 4. *ppProperties* –

The particle-particle interaction force can be activated by setting a value  $g0 > 0$ . When the packing limiter is set to on then the solid volume fraction is checked in every cell and then limits the solids volume fraction to alphaMax, the packing limit value.

```

preAlphaExp          500;
expMax               1.0e+3;
alphaMax              0.6;
g0                   g0 [1 -1 -2 0 0 0 0 ] 1000 ;
packingLimiter       no;

```

### 5. *transportProperties* –

The values of viscosity and density of the two phases and the diameter of the discrete phase are specified using this file. The diameter of the continuous phase is not taken into consideration and any value can be given.

If the drag coefficient has to be computed using both the phases then *dragPhase blended* has to be specified in the *interfacialProperties* file considering diameters of the both phases. The different coefficients in the inter-phase momentum transfer term in the momentum equation are specified in this file. The term *Cvm* represents co-efficient of virtual mass and *Cl* represents lift co-efficient.

If the values of these terms are set to zero then the virtual mass and lift force are neglected.

The last line *alphaAlpha* is not used by the solver and can be removed.

```
// a is the discrete phase and
// b is the continuous phase

// 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;
}

Cvm             Cvm          [0 0 0 0 0]    0.0;
Cl              Cl           [0 0 0 0 0]    0.0;
Ct              Ct           [0 0 0 0 0]    0.0;
alphaAlpha      alphaAlpha   [0 0 0 0 0]    0.0;
```

### 6. *filterProperties* –

The function of this file is that the flux becomes zero at the filter position and the plane specified.

```
filter          off;

// filterNormal has to be in the direction of trapped particles
filterNormal    (0 0 1);
```

```
filterPosition      filterPosition      [0 1 0 0 0]      (0 0 0.02);  
filterThickness     filterThickness     [0 1 0 0 0]      1.0e-6;
```

### 7. *RASproperties* –

The turbulence model and the constants for the particular model can be specified in this file.

The inputs related to solver control, solution schemes, writing of results can be found in the *system/* directory.

## 2. Running the solver

The mesh for the simulation can be created by typing ***blockMesh*** in the terminal window.

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

The mesh can be checked by typing ***checkMesh*** command to see that there are no errors in the mesh.

The solver can be started and the output is printed into a separate log file.

```
twoPhaseEulerFoam >& log&
```

## 3. Post Processing

The results of the simulation can be viewed in *paraFoam* similar to any other case by loading the geometry and results at the required time. The figure 2 below show the volume fraction in the fluidized bed channel and figure 3 shows the velocity of the continuous phase.

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

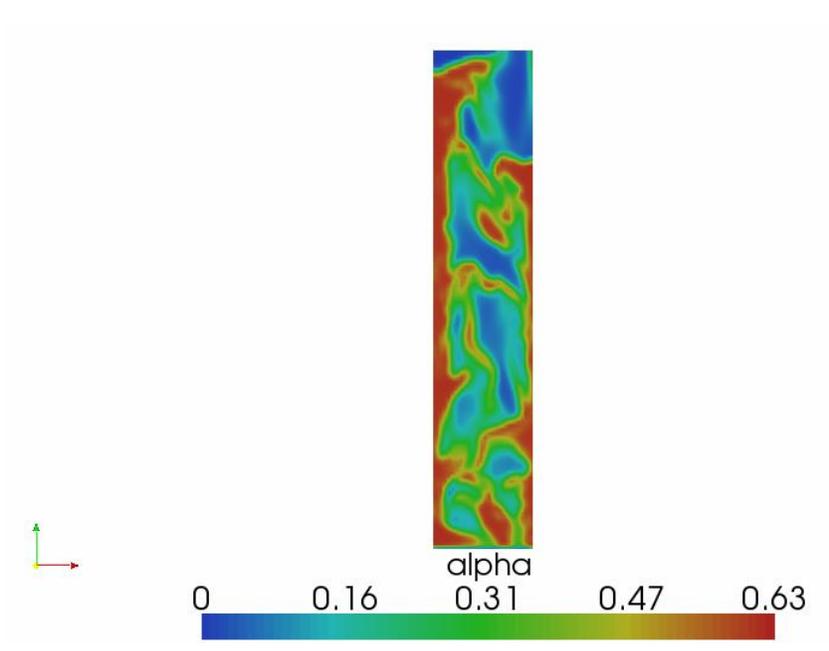


Figure 2. Phase fraction  $\alpha$  at 4 s.

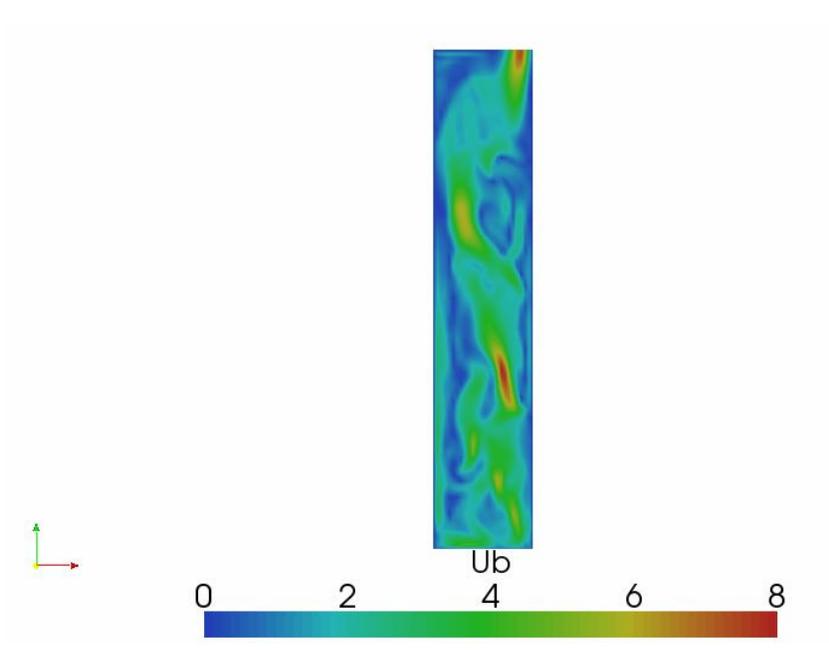


Figure 3. Continuous phase velocity at 4 s

## Two fluid modeling theory

The Eulerian conservation equations are used to describe both the phases in the two-fluid model. Each of the phase is treated as continuum and inter-penetrating each other and is represented by averaged equations. The equations implemented in OpenFOAM solver are given here. The equations for two fluid modeling approach in OpenFOAM are implemented from “Computational Fluid Dynamics of Dispersed Two-Phase flows at high phase fractions” by Henrik Rusche. The averaged inter-phase momentum transfer term accounts for the transfer of momentum between the two phases.

The averaged momentum and continuity equations for each phase  $\varphi$  can be written as

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

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

Where the subscript  $\varphi$  denotes the phase,  $\alpha$  is the phase fraction,  $\bar{R}_{\varphi}^{eff}$  the combined Reynolds (turbulent) and viscous stress,  $\bar{M}_{\varphi}$  is the averaged inter-phase momentum transfer term.

Combining the second equation for the two phases when  $\varphi = a$  and  $\varphi = b$  yields the volumetric continuity equation and can be formulated as a implicit equation for pressure

The inter-phase momentum transfer can be calculated by adding the forces acting on the Dispersed Phase particles. The drag, lift and the virtual mass forces are considered as the main contribution. The other forces such as Basset or history forces are neglected.

The volumetric continuity equation is

$$\nabla \cdot \bar{U} = 0$$

where

$$\bar{U} = \alpha_a \bar{U}_a + \alpha_b \bar{U}_b.$$

This equation is recast into a pressure equation

$$\left[ \nabla \cdot \left( \left( \alpha_{af} \left( \frac{1}{\rho_a (\mathcal{A}_a)_D} \right)_f + \alpha_{bf} \left( \frac{1}{\rho_b (\mathcal{A}_b)_D} \right)_f \right) \nabla [\bar{p}] \right) \right] = \nabla \cdot (\alpha_{af} \phi_a^* + \alpha_{bf} \phi_b^*)$$

The phase continuity equation solved is

$$\frac{\partial \alpha_a}{\partial t} + \nabla \cdot (\bar{U} \alpha_a) + \nabla \cdot (\bar{U}_r \alpha_a (1 - \alpha_a)) = 0$$

where  $\bar{U}$  is as given above and  $\bar{U}_r$  is the relative velocity between the phases.

Granular viscosities and pressure are derived from Kinetic theory of granular flow. In this approach, the thermodynamic temperature is replaced by granular flow temperature which is the measure of fluctuating velocity of the particles. Closure of the solid phase momentum equation requires a description for the solid-phase stress. The granular temperature (Theta) equation is found in kineticTheoryModel.C file

$$\frac{3}{2} \left[ \frac{\partial}{\partial t} (\alpha_s \rho_s \Theta) + \nabla \cdot (\alpha_s \rho_s \Theta v_s) \right] = \left( -\nabla P_s \bar{I} + \bar{\tau}_s \right) : \nabla v_s - \nabla \cdot (k_s \nabla \Theta) - \gamma_s - J_s$$

Where the first term on the right hand side represents the creation of fluctuating energy due to shear in the particle phase, the second term represents the diffusion of fluctuating energy along gradients in  $\Theta$ ,  $\gamma_s$  represents the dissipation due to inelastic particle-particle collisions and  $J_s$  represents the dissipation or creation of granular energy resulting from the working of the fluctuating force exerted by the gas through the fluctuating velocity of the particles.

When *equilibrium* is *on* in the kinetictheoryProperties then the algebraic equation is used instead of the balance equation as given below which is implemented in the solver

$$\Theta = \left( \frac{- (K_1 \alpha_s + \rho_s) \text{tr}(\overline{\overline{D_s}}) + \sqrt{(K_1 \alpha_s + \rho_s)^2 \text{tr}^2(\overline{\overline{D_s}}) + 4K_4 \alpha_s \left[ 2K_3 \text{tr}(\overline{\overline{D_s}}^2) + K_2 \text{tr}^2(\overline{\overline{D_s}}) \right]}}{2\alpha_s K_4} \right)^2$$

Where  $\overline{\overline{D_s}}$  is the solids strain tensor rate and  $K_1$ - $K_4$  are calculated with expressions.

The theory related to kinetic granular flow concepts and the expressions present in the kineticTheoryModel.C are referred to “Derivation, Implementation and validation of Computer Simulation Models for Gas-Solid Fluidized Beds” by Berend van Wachem.