

CFD WITH OPENSOURCE SOFTWARE

A COURSE AT CHALMERS UNIVERSITY OF TECHNOLOGY
TAUGHT BY HÅKAN NILSSON

Project work:

A twophaseEulerFoam tutorial

Developed for OpenFOAM-2.3.x

Author:
Sandra BUSCH

Peer reviewed by:
EFSTRATIOS FONIAS
JELENA ANDRIC

Disclaimer: This is a student project work, done as part of a course where OpenFOAM and some other OpenSource software are introduced to the students. Any reader should be aware that it might not be free of errors. Still, it might be useful for someone who would like learn some details similar to the ones presented in the report and in the accompanying files. The material has gone through a review process. The role of the reviewer is to go through the tutorial and make sure that it works, that it is possible to follow, and to some extent correct the writing. The reviewer has no responsibility for the contents.

January 20, 2015

1 Introduction

This tutorial describes how to solve an Euler-Euler multiphase model with the twoPhaseEulerFoam solver. The simulation that will be carried out is a gas-solid flow in a bubbling fluidized bed energy converter. A similar simulation has already been carried out using the commercial software ANSYS/FLUENT and this project thus also aims partly at comparing both softwares and their results.

The fluidized bed is a 2D column with height 1 m, width 0.28m and thickness 0.025 m. Fluidizing gas is blown under the column through a distributor plate. The static bed height is 0.4m with a solids volume fraction of 60 %. Figure 1 shows the basic geometry and the numbering of the vertices used in Section 3.2 ('Mesh'). Parameters that will be used in the set-up are given in Table 1 and the drag models will be those of Schiller-Naumann, Wen-Yu and Syamlal-O'Brien.

Table 1: Parameters used for the simulation

Parameter	- Value
Solid density	- 2500 kg/m^3
Gas density	- 1.225 kg/m^3
Solid diameter	- 275 μm
Initial solid volume fraction	- 40%
Superficial gas velocity	- 0.46m/s
Static bed height	- 0.4m
Maximum solid packing	- 62%
Lift forces	- Negligible

2 Theory

2.1 Governing equations

There are different approaches to numerically model multiphase flows: direct numerical simulation (DNS), Lagrangian particle tracking and multi-fluid modelling. Multi-fluid modelling is used here, where both phases are treated as interpenetrating continua and therefore share the same basic momentum and continuity equations. An overview of relevant equations for multiphase flows is given by the PhD. thesis of Henrik Rusche ¹. Only the basic governing equations are listed here. The continuity equation reads

$$\frac{\partial}{\partial t} \alpha_k \rho_k + \frac{\partial}{\partial x_i} \alpha_k \rho_k U_{k,i} = 0 \quad (1)$$

The momentum equation reads

$$\alpha_k \rho_k \left(\frac{\partial}{\partial t} U_{k,i} + U_{k,j} \frac{\partial}{\partial x_j} U_{k,i} \right) = \alpha_k \rho_k g_i - \alpha_k \frac{P_l}{\partial x_i} - \frac{\partial \tau_{k,ij}}{\partial x_j} + I_{k,i} \quad (2)$$

Here, α_k is the volume fraction and ρ_k the density of the respective fluid. The index k is symbolic for the solid and the fluid phase, e.g. the solid phase fraction may be denoted by α_p , the one of the continuous phase as α_f . The sum of the volume fraction of the different phases is always unity. $\tau_{k,ij}$ denotes the stress tensor and $I_{k,i}$ denotes the momentum transfer between the phases, which may include drag forces, lift forces, turbulent dispersion, etc..

Since drag forces have a major influence on the behaviour of the fluidized bed, those that are used further will be briefly described.

¹Henrik Rusche, Computational Fluid Dynamics of Dispersed Two-Phase Flows at High Phase Fractions, PhD Thesis, Imperial College, London, 2002.

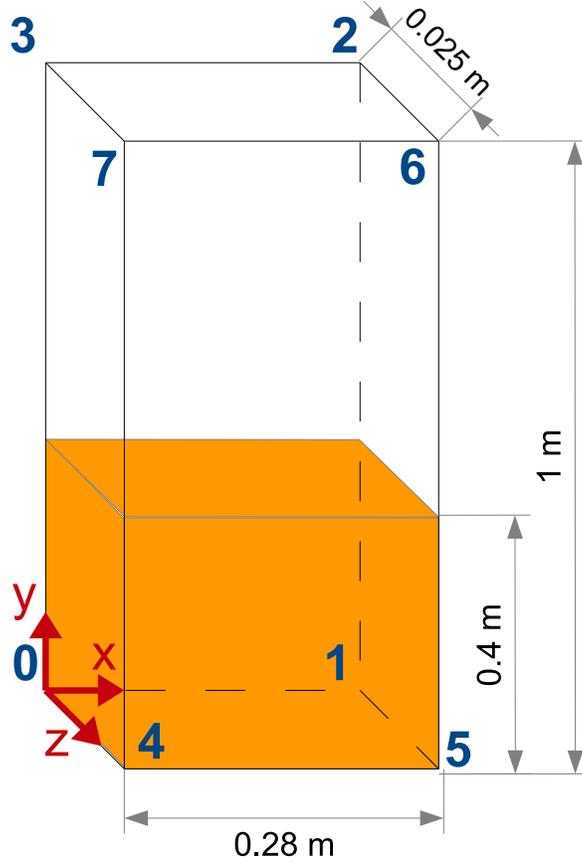


Figure 1: Geometry of the bubbling fluidized bed boiler

2.2 Drag models

The drag force is usually expressed as:

$$F_D = \frac{1}{2} C_D \rho_f A |V_r| V_r \quad (3)$$

where C_D is the drag coefficient, ρ_f is the fluid density, A is the projected area of the particle and V_r the relative velocity between the two phases. The drag coefficient is defined different in each drag model.

For the Syamlal-O'Brien drag model, the drag coefficient is given as:

$$C_D = \left(0.63 + \frac{4.8}{\sqrt{\frac{Re}{V_r}}} \right)^2 \quad (4)$$

where Re is the Reynolds number of the solid phase.

The relative velocity and coefficients are further implemented in OpenFoam as follows:

$$V_r = 0.5 \left(A - 0.06 Re + \sqrt{(0.06 Re)^2 + 0.12 Re (2B - A) + A^2} \right) \quad (5)$$

$$A = \alpha_p^{4.14} \quad (6)$$

$$B = 0.8 \alpha_p^{1.28} \text{ if } \alpha_p \leq 0.85 \quad (7)$$

$$B = \alpha_p^{2.65} \text{ if } \alpha_p > 0.85 \quad (8)$$

where α_p is the solids phase fraction (also denoted with the index 2 in the Open FOAM code). The Schiller-Naumann drag model reads:

$$C_D = \frac{24}{Re} (1 + 0.15Re^{0.687}) \quad (9)$$

And the Wen-Yu drag model reads:

$$C_D = \frac{24}{Re} (1 + 0.15Re^{0.687}) (1 - \alpha_f)^{-1.7} \quad (10)$$

For Reynolds numbers larger than 1000, the drag coefficient is approximately constant: $C_D = 0.44$. It is known from earlier simulations in commercial software that the drag models of Syamlal-O'Brien and Wen-Yu are more suited for fluid-solid simulations like this case and the drag model of Schiller-Naumann is more suited for fluid-gas simulations (e.g. bubbly flows). Thus the drag models of Syamlal-O'Brien and Wen-Yu are expected to be similar, whereas the Schiller-Naumann case is expected to differ from the other two.

3 Pre-processing

This section covers the necessary steps needed to get the twoPhaseEulerFoam case running and also shows where to change different parameters as given in Table 1 and the drag models.

3.1 Getting started

Copy the twoPhaseEulerFoam tutorial for fluidized bed to the run directory. There are actually two different tutorials for fluidized bed, one laminar and one RAS tutorial. Here, the RAS case is considered.

```
cp -r $FOAM_TUTORIALS/multiphase/twoPhaseEulerFoam/RAS/fluidisedBed $FOAM_RUN
cd $FOAM_RUN/fluidisedBed
```

The file structure is similar to other OpenFOAM tutorials where the case directory has a /0 , /constant and /system directory.

3.2 Mesh

The mesh has to be re-defined in `constant/polyMesh/blockMeshDict` to fit the desired geometry. The cell size is chosen to be roughly about 30 times the particle diameter of the solid phase, which then comes down to an equidistant cell size of $10mm$. The patches remain unchanged. The numbering of the vertices is according to Figure 1. If a finer mesh should be used, it is important that the size of the control volumes is not smaller than the diameter of the particle. In a control volume smaller than the particle diameter the particle would fill the entire control volume ($\alpha_p = 1$) and not allow for e.g. particle-particle interactions. This would also violate the maximum solid packing which is intended to be 62%, thus $\alpha_p \leq 0.62$. In general, the maximum solid packing should not be over around 63% as then the flow would be very dense and the probability of collisions between particles would go to infinity.

```
vertices
(
  (0 0 0)
  (0.28 0 0)
  (0.28 1 0)
  (0 1 0)
  (0 0 0.025)
  (0.28 0 0.025)
  (0.28 1 0.025)
```

```

    (0 1 0.025)
);

blocks
(
    hex (0 1 2 3 4 5 6 7) (28 100 1) simpleGrading (1 1 1)
);

```

BlockMesh can now be run from the case directory.

3.3 Boundary and initial conditions

Definition of static bed height

The static bed height can be defined in `system/setFieldsDict`. Here the volume fraction α can be defined for each part of the fluidized bed. The default setting is $\alpha_{air} = 1$, except for the lower part which can be specified under 'regions' in the data example below. The static bed height is set to $0.4m$, with a solids volume fraction of 40%.

```

defaultFieldValues
(
    volScalarFieldValue alpha.air 1
    volScalarFieldValue alpha.particles 0
);

regions
(
    boxToCell
    {
        box ( 0 0 0 ) ( 0.28 0.4 0.025 );
        fieldValues
        (
            volScalarFieldValue alpha.air 0.60
            volScalarFieldValue alpha.particles 0.40
        );
    }
);

```

It has to be noticed, that `0/alpha.air.org` and `0/alpha.particles.org` are backup files and need to be copied before running `setFields`:

```

cp 0/alpha.air.org 0/alpha.air
cp 0/alpha.particles.org 0/alpha.particles

```

Having done this, `setFields` can be executed from the case folder.

Boundary conditions

The boundary conditions for inlet and outlet can be set in `0/U.air`. Here, the inlet velocity of the air is set and the outlet is defined as pressure outlet:

```

dimensions      [0 1 -1 0 0 0 0];

internalField   uniform (0 0.46 0);

boundaryField
{

```

```

inlet
{
    type            interstitialInletVelocity;
    inletVelocity   uniform (0 0.46 0);
    alpha           alpha.air;
    value           $internalField;
}

outlet
{
    type            pressureInletOutletVelocity;
    phi            phi.air;
    value           $internalField;
}

walls
{
    type            fixedValue;
    value           uniform (0 0 0);
}

frontAndBackPlanes
{
    type            empty;
}
}

```

An overview of all initial settings is given in Table 2.

Since the mesh is equidistant and not refined at the wall, $0/\epsilon$ and $0/k$ do have wall functions at the wall boundary. Furthermore the outlet boundary is defined as `inletOutlet`, in case the flow reverses. This condition switches U and p between `fixedValue` and `zeroGradient` depending on the direction of U .

3.4 Physical properties

Phase properties

The basic properties of both phases are assigned in the respective property files:

`constant/thermophysicalProperties.air`

`constant/thermophysicalProperties.particles`

These are read by `constant/phaseProperties`. Here, the particle diameter and the maximum solid packing (`alphaMax`) can be changed:

```
phases (particles air);
```

```

particles
{
    diameterModel constant;
    constantCoeffs
    {
        d            275e-6;
    }

    alphaMax 0.62;
}

```

Table 2: Parameters used for the simulation

Filename	-	Function	-	Case
0/alpha.air	-	Volume fraction of gaseous phase	-	copy from 0/alpha.air.org, generated through setFields
0/alpha.particles	-	Volume fraction of solid phase	-	copy from 0/alpha.air.org, generated through setFields
0/epsilon	-	Turbulent dissipation	-	set to uniform 10, inletOutlet at Outlet, wall treatment
0/k	-	Turbulent kinetic energy	-	set to uniform 1, inletOutlet at Outlet, wall treatment
0/nut.air	-	Turbulent eddy viscosity of the gaseous phase	-	calculated
0/nut.particles	-	Turbulent eddy viscosity of the solid phase	-	calculated
0/p	-	Pressure	-	set to uniform 1e5
0/T.air	-	Temperature of air	-	set to uniform 600 and uniform 300 at inlet and outlet
0/T.particles	-	Temperature of solids phase	-	set to uniform 600
0/Theta.particles	-	Granular temperature of solids phase	-	calculated
0/U.air	-	Velocity of the gaseous phase	-	prescribed inlet velocity and pressure outlet, see code above
0/U.particles	-	Velocity of the solid phase	-	no solids velocity

This is also where the drag models can be defined. In this case, the drag model of WenYu is chosen:

```
drag
(
  (particles in air)
  {
    type          SyamlalOBrien;
    residualAlpha 1e-6;
    residualRe    1e-3;
    swarmCorrection
    {
      type        none;
    }
  }
);
```

A choice from 11 different drag models is possible in the twoPhaseEulerFoam:

```
Ergun
Gibilaro
GidaspowErgunWenYu
GidaspowSchillerNaumann
Lain
SchillerNaumann
SyamlalOBrien
TomiyamaAnalytic
TomiyamaCorrelated
WenYu
segregated
```

Information about the different drag models can be found in their respective files:

```
$FOAM_APP/solvers/multiphase/twoPhaseEulerFoam/interfacialModels/dragModels
```

Thermophysical properties

The thermophysical properties have to be set for each phase, in `constant/thermophysicalProperties.air` and `constant/thermophysicalProperties.particles`, respectively. Also, the density of the particles is set here:

```
mixture
{
  specie
  {
    nMoles      1;
    molWeight   100;
  }
  equationOfState
  {
    rho        2500;
  }
  thermodynamics
  {
    Cp         6000;
    Hf         0;
  }
  transport
  {
    mu         0;
    Pr         1;
  }
}
```

Turbulence properties

The turbulence modelling can be either turned off or turned on in `constant/turbulenceProperties.air` and `constant/turbulenceProperties.particles` respectively by setting it to off or on:

```
RAS
{
  RASModel kineticTheory;

  turbulence      on;
  printCoeffs    on;
```

Otherwise no further changes were made compared to the original case. If the case was chosen from the laminar fluidized bed tutorial, the simulation type is set to `laminar` in `constant/turbulenceProperties.air`.

Other properties

Specification of the direction and value of the gravity in `constant/g`:

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

3.5 Running the code

Having prepared the case, as described in Section 3, the solver can now be run by typing in the case directory:

```
twoPhaseEulerFoam >& log &
```

This will generate an output directory for each time step in the case directory as well as a log file.

3.6 Post processing with ParaFoam

ParaFoam can be run from the case directory by typing

```
paraFoam
```

The case can then be opened in paraView. The most interesting is to visualize the volume fraction of the particle phase, Figure 2 shows the bed after 20 seconds of simulation for all three drag models. Interesting observations are the volume fraction of the solid phase and the bed height. In Figure 3, the bed is shown at different times in the beginning of the simulation with the Schiller-Naumann and Syamlal-O'Brien drag models. Whereas in Figure 3a the two phases are already starting to mix at 2.4 seconds of simulation time, the Schiller-Naumann drag model shown in Figure 3b clearly shows symmetrical behaviour. As can be seen in 3c, the symmetry also exists at the beginning of the Syamlal-O'Brien drag model but breaks up much earlier than for the Schiller-Naumann drag model. Since all drag models show a symmetrical behaviour in the beginning of the simulation, a certain time is needed until this symmetry breaks. This time varies for different drag models as has been shown and has to be taken into account in these simulations.

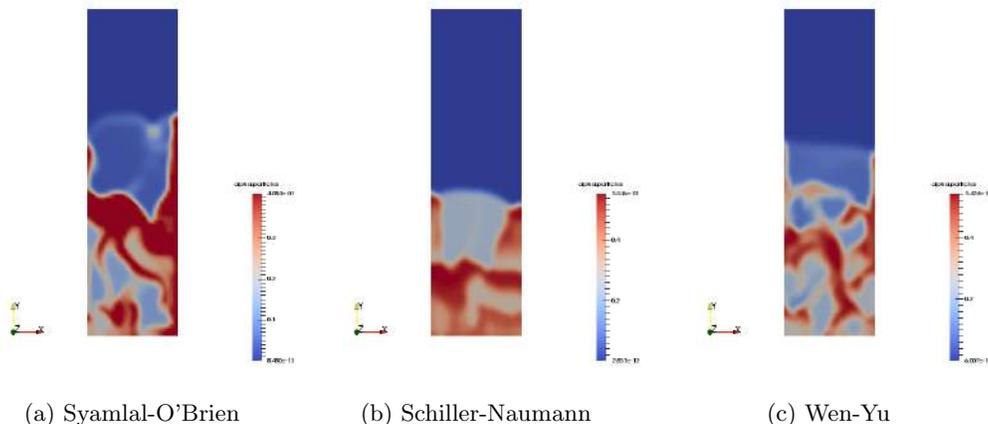


Figure 2: The bed for different drag models after 20 seconds of simulation. The legend gives the volume fraction of the solid phase

Figures 4 to 6 show the averaged particle volume fraction for each drag model at heights of $y = 0.1m$, $y = 0.3m$ and $y = 0.5m$ at a simulation time of 20 seconds. Experimental data of an averaged volume fraction may be available only for a certain height in the fluidized bed boiler and this is a way of comparing the results to experimental results. Also, the different drag models may be compared in this way and by using more planes in y -direction the bed height could be approximated.

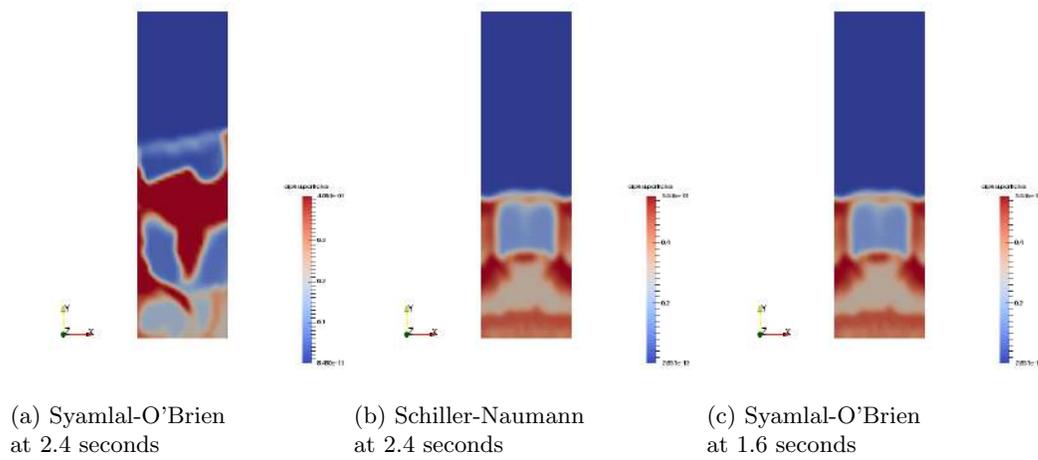


Figure 3: The bed for two drag models at the beginning of the simulation. The legend gives the volume fraction of the solid phase

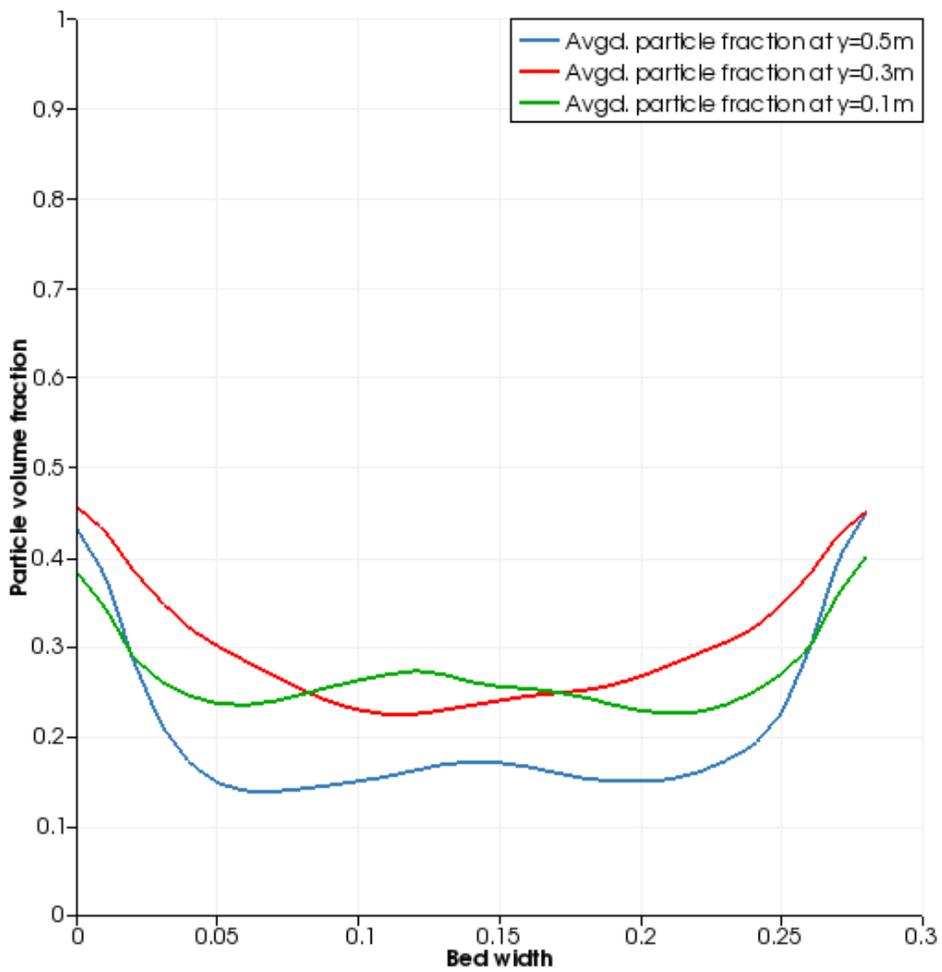


Figure 4: Averaged volume fraction of the particulate phase using the Syamlal-O'Brien drag model

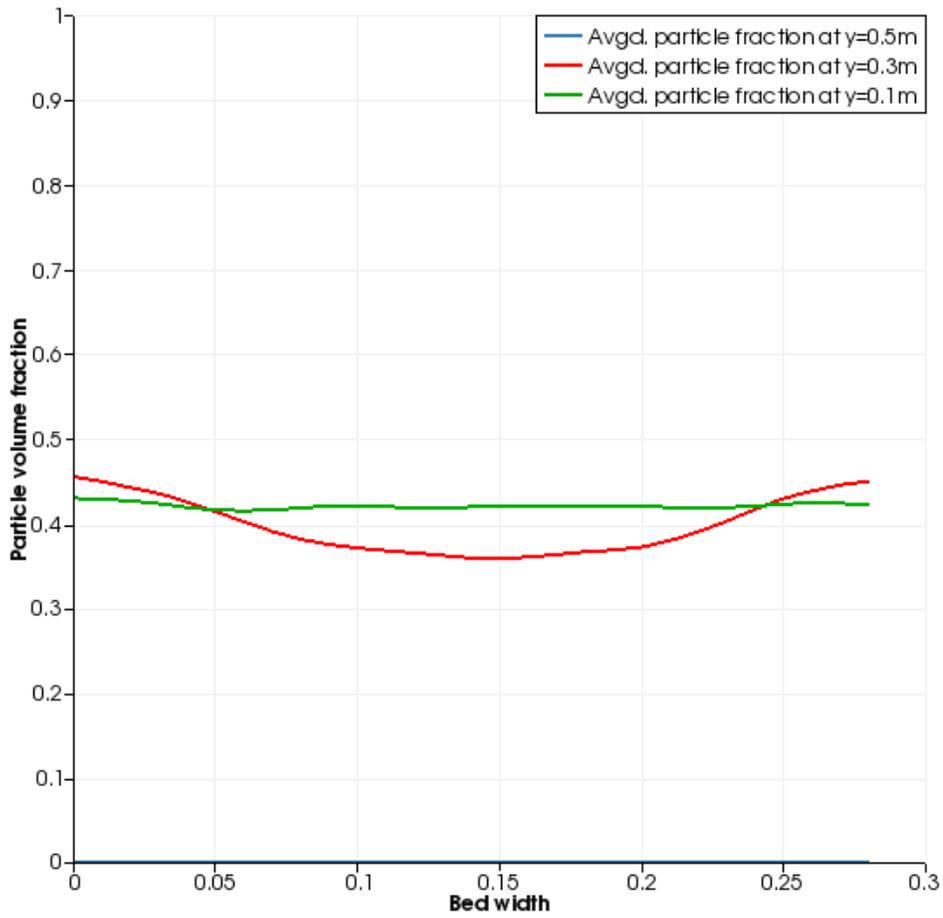


Figure 5: Averaged volume fraction of the particulate phase using the Schiller-Naumann drag model

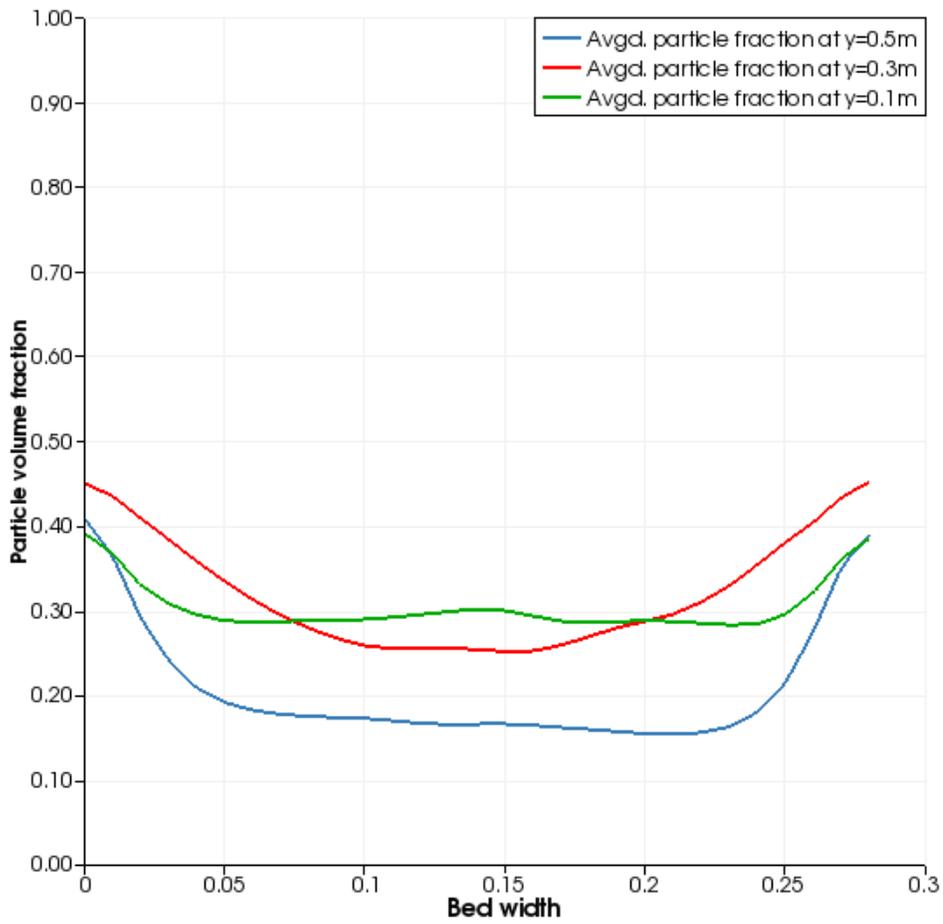


Figure 6: Averaged volume fraction of the particulate phase using the Wen-Yu drag model