Implementation of an iterative solution procedure for multi-fluid gas-particle flow models on unstructured grids

Multi-fluid models in OpenFOAM®

Alberto Passalacqua, Rodney O. Fox

Iowa State University - Department of Chemical and Biological Engineering

NETL 2011 Workshop on Multiphase Flow Science
Airport Marriott Station Square, Pittsburgh, PA
August 16^{th} – 18^{th}, 2011

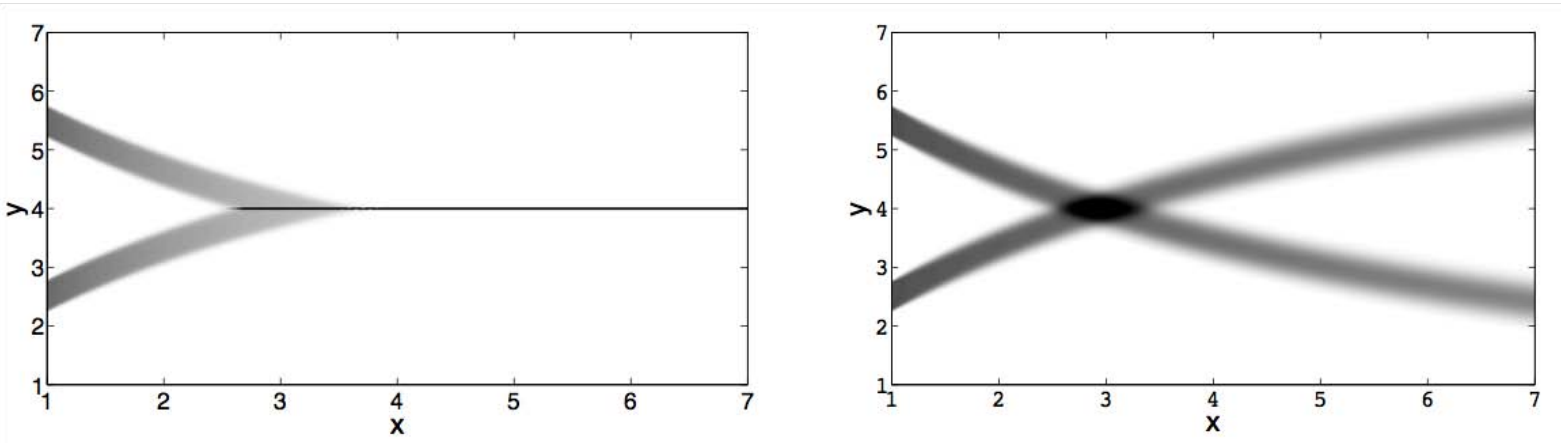
Outline

Outline

Recent advances in Euler-Euler multiphase flow modelling

Quadrature-based moment methods (QBMM)

- Solution of the Boltzmann-Enskog kinetic equation with QBMM
  - QBMM overcome limitations of hydrodynamic (two-fluid) models and:
    - Are valid for arbitrary Knudsnen numbers (Kn $\gg$ 1)
    - Properly describe both collision dominated and collisionless flows
    - Predicting particle trajectory crossing
    - Are not affected by unphysical delta-shocks (unphysical mass accumulation) as the two-fluid model (Desjardin et al., 2008)
    - Rely on a well-posed, hyperbolic, set of equations



Particle trajectory crossing: Two-fluid (left), Lagrangian and QBMM (right)

Recent advances in Euler-Euler multiphase flow modelling

QBMM capabilities

- QMOM (Quadrature method of moments)
  - Solves the Boltzmann-Enskog kinetic equation
  - Calculates the moments of the velocity distribution function
  - The particle velocity is directly calculated from the moments
  - Some difficulty with strongly non-equilibrium flows and low restitution coefficients
- CQMOM (Conditional QMOM)
  - Solution of the generalized kinetic equation
  - Calculates
    - Particle velocity
    - Particle size distribution
  - Robust with arbitrary values of the restitution coefficient
  - Adaptive quadrature:
    - Balances cost and accuracy
    - Manages the case of zero granular temperature
    - Deals with equal abscissas automatically

Recent advances in Euler-Euler multiphase flow modelling

QBMM in the dense limit

- QBMM tend to the hydrodynamic limit for Kn $\rightarrow 0$
- The equation of state of the granular phase becomes, in such a limit, identical to the equation of state derived from the kinetic theory of granular flow

$$p_s = \rho_s \alpha_s \Theta_s + 2\rho_s \alpha_s^2 g_0 \Theta_s (1 + e_s)$$

- Dependency on the radial distribution function g_0: numerical instabilities in the closely packed regions

Observation

The numerical procedure to deal with the dense limit can be developed considering the standard multi-fluid equations, and then extended to QBMM

Motivation

Requirements

- Robust numerical procedure in the limit of closely packed particle phase
- Implicit treatment of the particle pressure

Some considerations on QBMM numerics

- QBMM use the flux-splitting technique to define kinetic fluxes: easy implementation in the co-located grid arrangement
- For consistency, a co-located grid should be used also in the hydrodynamic limit to implement an hybrid QBMM for the whole range of particle concentrations

Objectives

- Develop an iterative procedure for the solution of multi-fluid equations on co-located grids
- Implement the procedure in a general-purpose code: OpenFOAM®

Motivation

Requirements

- Robust numerical procedure in the limit of closely packed particle phase
- Implicit treatment of the particle pressure

Some considerations on QBMM numerics

- QBMM use the flux-splitting technique to define kinetic fluxes: easy implementation in the co-located grid arrangement
- For consistency, a co-located grid should be used also in the hydrodynamic limit to implement an hybrid QBMM for the whole range of particle concentrations

Objectives

- Develop an iterative procedure for the solution of multi-fluid equations on co-located grids
- Implement the procedure in a general-purpose code: OpenFOAM®

Motivation

Requirements

- Robust numerical procedure in the limit of closely packed particle phase
- Implicit treatment of the particle pressure

Some considerations on QBMM numerics

- QBMM use the flux-splitting technique to define kinetic fluxes: easy implementation in the co-located grid arrangement
- For consistency, a co-located grid should be used also in the hydrodynamic limit to implement an hybrid QBMM for the whole range of particle concentrations

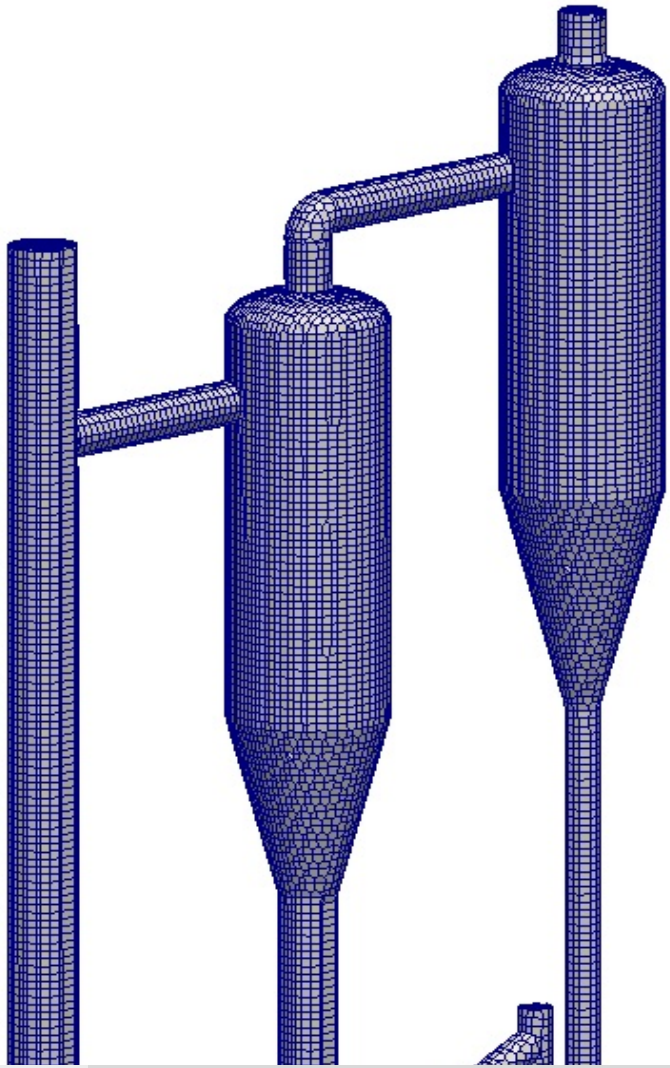
Objectives

- Develop an iterative procedure for the solution of multi-fluid equations on co-located grids
- Implement the procedure in a general-purpose code: OpenFOAM®

Motivation

Why OpenFOAM?

- Free and open-source software (GPL 3 license) developed and supported by OpenCFD®Ltd (SGI)
- High level syntax through C++ object orientation
- Variety of mesh formats (hex, tet, polyhedral)
  - Arbitrary geometries
  - Automatic hex-dominant meshing
  - Real life applications
- Automatic parallelization
- Not a black-box



The multi-fluid equations

Assumptions

- Phases are incompressible: constant phase material density
- The multi-phase system is isothermal

Fundamental set of incompressible multi-fluid models

- Continuity equation

$$\frac{\partial \alpha_i}{\partial t} + \nabla \cdot (\alpha_i \mathbf{U}_i) = 0$$

- Momentum equation

$$\frac{\partial}{\partial t}(\alpha_i \mathbf{U}_i) + \nabla \cdot (\alpha_i \mathbf{U}_i \otimes \mathbf{U}_i) = \frac{1}{\rho_i}\nabla \cdot \boldsymbol{\tau}_i - \frac{\alpha_i}{\rho_i}\nabla p - \frac{\gamma_i}{\rho_i}\nabla p_i + \alpha_i \mathbf{g} + \sum_{\substack{j=0 \\ j \neq i}}^{N-1} \frac{K_{ij}}{\rho_i}(\mathbf{U}_j - \mathbf{U}_i)$$

The multi-fluid equations

Assumptions

- Phases are incompressible: constant phase material density
- The multi-phase system is isothermal

Fundamental set of incompressible multi-fluid models

- Continuity equation

$$\frac{\partial \alpha_i}{\partial t} + \nabla \cdot (\alpha_i \mathbf{U}_i) = 0$$

- Momentum equation

$$\frac{\partial}{\partial t}(\alpha_i \mathbf{U}_i) + \nabla \cdot (\alpha_i \mathbf{U}_i \otimes \mathbf{U}_i) = \frac{1}{\rho_i}\nabla \cdot \boldsymbol{\tau}_i - \frac{\alpha_i}{\rho_i}\nabla p - \frac{\gamma_i}{\rho_i}\nabla p_i + \alpha_i \mathbf{g} + \sum_{\substack{j=0 \\ j \neq i}}^{N-1} \frac{K_{ij}}{\rho_i}(\mathbf{U}_j - \mathbf{U}_i)$$

Some comment on the momentum equation

$$\frac{\partial}{\partial t}\left(\alpha_i \mathbf{U}_i\right) + \nabla \cdot \left(\alpha_i \mathbf{U}_i \otimes \mathbf{U}_i\right) = \frac{1}{\rho_i}\nabla \cdot \boldsymbol{\tau}_i - \frac{\alpha_i}{\rho_i}\nabla p - \frac{\gamma_i}{\rho_i}\nabla p_i + \alpha_i \mathbf{g} + \sum_{\substack{j=0 \\ j \neq i}}^{\mathrm{N}-1} \frac{K_{ij}}{\rho_i}\left(\mathbf{U}_j - \mathbf{U}_i\right)$$

- It contains the phase fraction α_i in all its terms
- Two pressures:
  - Shared pressure p
  - Phase pressure p_i, only for dispersed phases ($\gamma_i = \alpha_{\mathrm{s},i} / \sum_i \alpha_{\mathrm{s},i}$)
- Momentum exchange term: drag coefficient K_{ij}

Description of the particle phase

Kinetic theory model

- Granular energy equation

$$\frac{3}{2}\left[\frac{\partial}{\partial t}\left(\alpha_s \rho_s \Theta_s\right) + \nabla \cdot \left(\alpha_s \rho_s \mathbf{U}_s \Theta_s\right)\right] = \left(-p_s \mathbf{I} + \boldsymbol{\tau}_s\right) : \nabla \mathbf{U}_s + \nabla \cdot \left(\kappa_s \nabla \Theta_s\right) - \gamma_s + J_{\text{vis}} + J_{\text{slip}}$$

- Equation of state of the dispersed phase

$$p_s = \rho_s \alpha_s \Theta_s + 2\rho_s \alpha_s^2 g_0 \Theta_s \left(1 + e_s\right)$$

- Radial distribution function

$$g_0 = \frac{1}{1 - \left(\frac{\alpha_s}{\alpha_{s,\max}}\right)^{\frac{1}{3}}}$$

Nearly packed flows

Observation

Very dense particle flows are dominated by frictional phenomena:

- The kinetic theory model is insufficient: binary collisions
- Frictional models are used (for example: Syamlal, 1993), replacing the kinetic theory closures for $\alpha_s > \alpha_{s,fr,min} \approx 0.61$:
  - Particle frictional pressure

$$p_f = 10^{25} \left(\alpha_s - \alpha_{s,fr,min}\right)^{10}$$

  - Frictional viscosity

Observation

- This definition of the particle pressure is continuous: removes the singularity
- Other models keep a discontinuous form of the pressure: series expansion

Outline

Singularity when the phase fraction tends to zero

Problem

The phase momentum equation degenerates in an identity when $\alpha_i = 0$

Possible solutions proposed in the literature

Two solutions were proposed in the literature for segregated solvers:

- Solve the equation in non-conservative form (Oliveira and Issa, 2003)
  - Time derivative and convective term are expanded to remove the phase fraction
- Solve the equation in semi-conservative form (Park et al., 2009)
  - Only the time derivative is expanded to remove the phase fraction
- Almost no reference providing details on how to solve the momentum equation in conservative form avoiding this problem

Singularity when the phase fraction tends to zero

Problem

The phase momentum equation degenerates in an identity when $\alpha_i = 0$

Possible solutions proposed in the literature

Two solutions were proposed in the literature for segregated solvers:

- Solve the equation in non-conservative form (Oliveira and Issa, 2003)
  - Time derivative and convective term are expanded to remove the phase fraction
- Solve the equation in semi-conservative form (Park et al., 2009)
  - Only the time derivative is expanded to remove the phase fraction
- Almost no reference providing details on how to solve the momentum equation in conservative form avoiding this problem

Singularity when the phase fraction tends to zero

Observation

The non-conservative form of the momentum equation presents difficulties when strong property transfer (Park et al., 2009) or strong particle pressure gradients are present.

Adopted solution

- Solve the momentum equation in conservative form
- Define a cut-off value and stop solving the momentum equation if the phase fraction falls below that value
  - Mass conservation is ensured, independently from this treatment
  - Momentum conservation errors are negligible if the cut-off is small (low phase fraction)
- Use appropriate numerical techniques to avoid numerical difficulties at phase interfaces

Phase momentum coupling

Problem

The momentum exchange term can lead to numerical instabilities of the iterative procedure, slowing down convergence or making it impossible

Proposed solutions

- Use the partial elimination algorithm - PEA procedure (Spalding, 1983)
- Solve the momentum equations in a coupled fashion (Karema and Lo, 1999; Vasquez and Ivanov, 2000)
- Both the approaches aim at enlarging the stability region of the numerical procedure

Adopted solution

- PEA: we want/need to keep the algorithm purely iterative

Phase momentum coupling

Problem

The momentum exchange term can lead to numerical instabilities of the iterative procedure, slowing down convergence or making it impossible

Proposed solutions

- Use the partial elimination algorithm - PEA procedure (Spalding, 1983)
- Solve the momentum equations in a coupled fashion (Karema and Lo, 1999; Vasquez and Ivanov, 2000)
- Both the approaches aim at enlarging the stability region of the numerical procedure

Adopted solution

- PEA: we want/need to keep the algorithm purely iterative

Phase momentum coupling

Problem

The momentum exchange term can lead to numerical instabilities of the iterative procedure, slowing down convergence or making it impossible

Proposed solutions

- Use the partial elimination algorithm - PEA procedure (Spalding, 1983)
- Solve the momentum equations in a coupled fashion (Karema and Lo, 1999; Vasquez and Ivanov, 2000)
- Both the approaches aim at enlarging the stability region of the numerical procedure

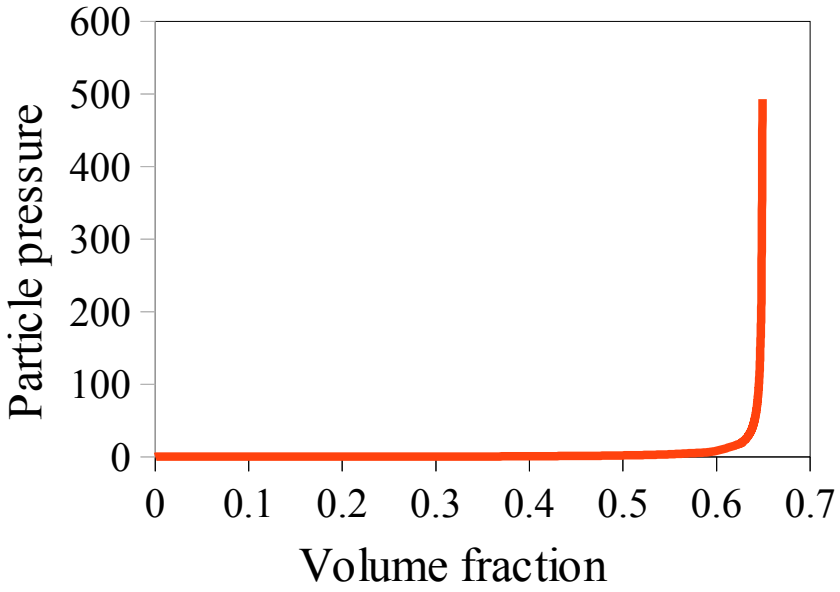
Adopted solution

- PEA: we want/need to keep the algorithm purely iterative

Packing limit enforcement

Problem

The particle packing limit is enforced by the particle pressure term, which diverges when the phase fraction reaches the maximum value. This leads to strong numerical difficulties.



Proposed solutions

- Recast the phase continuity equation into an equation for the particle pressure
- Include the effect of the particle pressure in the phase continuity equation

Equation for the particle pressure

Advantages

It is very efficient in the dense limit, since the pressure equation becomes a simple force balance, independent from the particle pressure term

Disadvantages

- It is problematic in the dilute limit, since it requires the compressibility of the phase ($\partial\alpha_s/\partial p_s$) to be finite
- It requires to find the phase fraction in some way
  - Invert the equation of state
  - Solve the phase continuity equation explicitly again to find the phase fraction

Inclusion of the particle pressure in the continuity equation

Advantages

- It is very efficient in moderately dense flows
- It does not present difficulties in the dilute limit

Disadvantages

- It might slow down the solution in the packed limit
  - Under-relaxation required
- It requires a continuous function for the particle pressure
  - Discontinuous functions can be treated with a power series expansion or similar approach in proximity of the maximum particle phase fraction

Adopted approach

Solve a phase continuity equation, modified to include the effect of the particle pressure

Outline

Rhie-Chow interpolation

The Rhie-Chow interpolation in short

The Rhie and Chow (1983) interpolation is used on co-located grids to avoid the checker-board solutions for the pressure. It consists in a correction to the interpolated face velocity: subtract the difference between the pressure gradient and the interpolated pressure gradient at face centroid (Ferziger and Peric, 2002):

$$u_{i,\mathrm{f}} = \overline{u_{i,\mathrm{f}}} - \Delta V_{\mathrm{f}} \overline{\left(\frac{1}{\mathbb{A}}\right)}_{\mathrm{f}} \left[\left(\frac{\delta p}{\delta x_i}\right)_{\mathrm{f}} - \overline{\left(\frac{\delta p}{\delta x_i}\right)}_{\mathrm{f}} \right]$$

In multiphase codes ...

- Phase fraction gradients and strong transfer terms might lead this approach to fail.
- Strongly spatially-varying terms must be included in the interpolation formula (Karema and Lo, 1999; Zhang and Zhao, 2004).

The Rhie-Chow interpolation in OpenFOAM [1]

Let us consider the single-phase equation (Kärrholm, 2006)

$$\frac{\partial \mathbf{U}}{\partial t} + (\mathbf{U} \cdot \nabla)\,\mathbf{U} - \nabla \cdot (\nu \nabla \mathbf{U}) = -\frac{1}{\rho}\nabla p$$

OpenFOAM implements it as:

```
fvVectorMatrix UEqn
(
    fvm::ddt(U)
  + fvm::div(phi, U)
  - fvm::laplacian(nu, U)
);

solve(UEqn == -fvc::grad(p));
```

- `phi` is the old-time volumetric flux at faces
- OpenFOAM uses Gauss theorem directly

$$\int_V \nabla \cdot (\mathbf{U}\Upsilon)\,\mathrm{d}V = \int_S (\mathbf{U}\Upsilon)_\mathrm{f} \cdot \mathbf{n}\mathrm{d}S \approx \sum_i \mathbf{U}_{\mathrm{f},i}\Upsilon_{\mathrm{f},i} \cdot \mathbf{S}_i = \sum_i \mathbf{U}_{\mathrm{f},i}\phi_i$$

- $\phi = \Upsilon_{\mathrm{f},i} \cdot \mathbf{S}$
- Υ is the velocity held constant, $\mathbf{U}$ is the unknown

The Rhie-Chow interpolation in OpenFOAM [2]

We now derive the pressure equation:

$$\mathbb{A}\mathbf{U} = \mathbb{H} - \nabla p \Leftrightarrow \mathbf{U} = \frac{\mathbb{H}}{\mathbb{A}} - \frac{1}{\mathbb{A}}\nabla p$$

Inserting in the continuity equation, we find the pressure equation

$$\nabla \cdot \left(\frac{\mathbb{H}}{\mathbb{A}}\right) = \nabla \cdot \left(\frac{1}{\mathbb{A}}\nabla p\right)$$

OpenFOAM computes:

- Predicted velocity (H is updated!)

```
U = UEqn.H()/UEqn.A();
```

- Face volumetric flux

```
surfaceScalarField phi =
    fvc::interpolate(U) & mesh.Sf();
```

- Central coefficient

```
volScalarField rUA = 1.0/UEqn.A();
```

- Pressure equation

```
fvScalarMatrix pEqn
(
    fvm::laplacian(rUA, p)
      == fvc::div(phi)
);
pEqn.setReference(pRefCell, pRefValue);
pEqn.solve();
```

- Corrected flux

```
phi -= pEqn.flux();
```

At this time conservation is enforced!

- Corrected velocity

```
U -= rUA*fvc::grad(p);
```

The Rhie-Chow interpolation in OpenFOAM [3]

Comment

No second derivative of p is present explicitly: Gauss theorem. First derivative at cell faces is computed used cell-centre values of p

Summary

- `phi` does not include effects of ∇p when used in the pressure equation
- The central coefficient does not contain the effect of ∇p
- The Laplacian of p uses ∇p on cell faces, which is computed used cell-centre values of p
- In the velocity correction, the pressure gradient is computed from the face values of p (Gauss theorem)

The Rhie-Chow interpolation in OpenFOAM [4]

In multiphase flows...

We have to include the strongly varying terms (gravity, drag, ...) as a function of the phase fraction and the strong coupling terms in the momentum interpolation formula.

Observation

OpenFOAM imposes the continuity constraint on the centroid of cell faces, which is the location where the force balance must be exactly satisfied.

Approach

- The involved terms are treated as source terms in the momentum equation, using values reconstructed from the face centroids.
- The actual solution of the exact force balance is performed with the solution of the pressure equation
- Cell-centred values of the velocity are corrected from face fluxes

Outline

Derivation of the momentum predictor

For simplicity, we consider two phases under the following hypotheses

- Incompressible
- The momentum exchange is dominated by the drag force
- We derive
  - The velocity predictors, using the PEA algorithm
  - The flux predictors and correctors
  - The pressure equation based on the total continuity
  - The modified phase continuity equation

Velocity predictors

The semi-discrete momentum equations are:

$$\mathbb{A}_\mathrm{s}\mathbf{U}_\mathrm{s} = \mathbb{H}_\mathrm{s} - \frac{\alpha_\mathrm{s}}{\rho_\mathrm{s}}\nabla p - \frac{1}{\rho_\mathrm{s}}\nabla p_\mathrm{s} + \alpha_\mathrm{s}\mathbf{g} + \frac{K_\mathrm{sg}}{\rho_\mathrm{s}}\left(\mathbf{U}_\mathrm{g} - \mathbf{U}_\mathrm{s}\right),$$

and

$$\mathbb{A}_\mathrm{g}\mathbf{U}_\mathrm{g} = \mathbb{H}_\mathrm{g} - \frac{\alpha_\mathrm{g}}{\rho_\mathrm{g}}\nabla p + \alpha_\mathrm{g}\mathbf{g} + \frac{K_\mathrm{sg}}{\rho_\mathrm{g}}\left(\mathbf{U}_\mathrm{s} - \mathbf{U}_\mathrm{g}\right).$$

Collecting the terms containing the unknown velocity in each of these equations, defining the quantities

$$\lambda_\mathrm{s} = \frac{1}{\mathbb{A}_\mathrm{s} + \frac{K_\mathrm{sg}}{\rho_\mathrm{s}}}, \qquad \lambda_\mathrm{g} = \frac{1}{\mathbb{A}_\mathrm{g} + \frac{K_\mathrm{sg}}{\rho_\mathrm{g}}},$$

$$\mathbb{H}^*_\mathrm{s} = \mathbb{H}_\mathrm{s} - \frac{\alpha_\mathrm{s}}{\rho_\mathrm{s}}\nabla p - \frac{1}{\rho_\mathrm{s}}\nabla p_\mathrm{s} + \alpha_\mathrm{s}\mathbf{g},$$

$$\mathbb{H}^*_\mathrm{g} = \mathbb{H}_\mathrm{g} - \frac{\alpha_\mathrm{g}}{\rho_\mathrm{g}}\nabla p + \alpha_\mathrm{g}\mathbf{g}.$$

Velocity predictors

$$\mathbf{U}_\mathrm{s} = \lambda_\mathrm{s}\mathbb{H}^*_\mathrm{s} + \lambda_\mathrm{s}\frac{K_\mathrm{sg}}{\rho_\mathrm{s}}\mathbf{U}_\mathrm{g}, \qquad \mathbf{U}_\mathrm{g} = \lambda_\mathrm{g}\mathbb{H}^*_\mathrm{g} + \lambda_\mathrm{g}\frac{K_\mathrm{sg}}{\rho_\mathrm{g}}\mathbf{U}_\mathrm{s}.$$

If the quantities

$$\xi_\mathrm{s} = \frac{1}{\mathbb{A}_\mathrm{s} - \lambda_\mathrm{g}\dfrac{K^2_\mathrm{sg}}{\rho_\mathrm{s}\rho_\mathrm{g}} + \dfrac{K_\mathrm{sg}}{\rho_\mathrm{s}}},$$

$$\xi_\mathrm{g} = \frac{1}{\mathbb{A}_\mathrm{g} - \lambda_\mathrm{s}\dfrac{K^2_\mathrm{sg}}{\rho_\mathrm{s}\rho_\mathrm{g}} + \dfrac{K_\mathrm{sg}}{\rho_\mathrm{g}}}$$

are introduced, such intermediate velocities are given by

$$\mathbf{U}_\mathrm{s} = \xi_\mathrm{s}\mathbb{H}^*_\mathrm{s} + \xi_\mathrm{s}\frac{K_\mathrm{sg}}{\rho_\mathrm{s}}\lambda_\mathrm{g}\mathbb{H}^*_\mathrm{g},$$

$$\mathbf{U}_\mathrm{g} = \xi_\mathrm{g}\mathbb{H}^*_\mathrm{g} + \xi_\mathrm{g}\frac{K_\mathrm{sg}}{\rho_\mathrm{g}}\lambda_\mathrm{s}\mathbb{H}^*_\mathrm{s}.$$

Velocity predictors

Substituting back and collecting the coefficients of the same terms, we obtain

$$\mathbf{U}_\mathrm{s} = \xi_\mathrm{s}\left[\mathbb{H}_\mathrm{s} + \frac{K_\mathrm{sg}\lambda_\mathrm{g}\mathbb{H}_\mathrm{g}}{\rho_\mathrm{s}} - \left(\frac{\alpha_\mathrm{s}}{\rho_\mathrm{s}} + \frac{K_\mathrm{sg}\lambda_\mathrm{g}\alpha_\mathrm{g}}{\rho_\mathrm{s}\rho_\mathrm{g}}\right)\nabla p - \frac{1}{\rho_\mathrm{s}}\nabla p_\mathrm{s} + \left(\alpha_\mathrm{s} + \frac{K_\mathrm{sg}\lambda_\mathrm{g}\alpha_\mathrm{g}}{\rho_\mathrm{s}}\right)\mathbf{g}\right],$$

and

$$\mathbf{U}_\mathrm{g} = \xi_\mathrm{g}\left[\mathbb{H}_\mathrm{g} + \frac{K_\mathrm{sg}\lambda_\mathrm{s}\mathbb{H}_\mathrm{s}}{\rho_\mathrm{g}} - \left(\frac{\alpha_\mathrm{g}}{\rho_\mathrm{g}} + \frac{K_\mathrm{sg}\lambda_\mathrm{s}\alpha_\mathrm{s}}{\rho_\mathrm{s}\rho_\mathrm{g}}\right)\nabla p - \frac{K_\mathrm{sg}\lambda_\mathrm{s}}{\rho_\mathrm{s}\rho_\mathrm{g}}\nabla p_\mathrm{s} + \left(\alpha_\mathrm{g} + \frac{K_\mathrm{sg}\lambda_\mathrm{s}\alpha_\mathrm{s}}{\rho_\mathrm{g}}\right)\mathbf{g}\right],$$

which represent the velocity predictors, obtained following the partial elimination algorithm.

Face volumetric flux

Interpolating the velocity predictors on cell faces we obtain the face volumetric flux:

$$\varphi_\mathrm{s} = \xi_{\mathrm{s,f}} \left(\mathbb{H}_\mathrm{s} + \frac{K_\mathrm{sg}\lambda_\mathrm{g}\mathbb{H}_\mathrm{g}}{\rho_\mathrm{s}} \right)_\mathrm{f} \cdot \mathbf{S} - \xi_{\mathrm{s,f}} \left(\frac{\alpha_\mathrm{s}}{\rho_\mathrm{s}} + \frac{K_\mathrm{sg}\lambda_\mathrm{g}\alpha_\mathrm{g}}{\rho_\mathrm{s}\rho_\mathrm{g}} \right)_\mathrm{f} |\mathbf{S}| \nabla^\perp p - \xi_{\mathrm{s,f}} \left(\frac{G(\alpha_\mathrm{s})}{\rho_\mathrm{s}} \right)_\mathrm{f} |\mathbf{S}| \nabla^\perp \alpha_\mathrm{s} + \xi_{\mathrm{s,f}} \left(\alpha_\mathrm{s} + \frac{K_\mathrm{sg}\lambda_\mathrm{g}\alpha_\mathrm{g}}{\rho_\mathrm{s}} \right)_\mathrm{f} \mathbf{g} \cdot \mathbf{S},$$

$$\varphi_\mathrm{g} = \xi_{\mathrm{g,f}} \left(\mathbb{H}_\mathrm{g} + \frac{K_\mathrm{sg}\lambda_\mathrm{s}\mathbb{H}_\mathrm{s}}{\rho_\mathrm{g}} \right)_\mathrm{f} \cdot \mathbf{S} - \xi_{\mathrm{g,f}} \left(\frac{\alpha_\mathrm{g}}{\rho_\mathrm{g}} + \frac{K_\mathrm{sg}\lambda_\mathrm{s}\alpha_\mathrm{s}}{\rho_\mathrm{s}\rho_\mathrm{g}} \right)_\mathrm{f} |\mathbf{S}| \nabla^\perp p - \xi_{\mathrm{g,f}} \left(\frac{K_\mathrm{sg}\lambda_\mathrm{s}G(\alpha_\mathrm{s})}{\rho_\mathrm{s}\rho_\mathrm{g}} \right)_\mathrm{f} |\mathbf{S}| \nabla^\perp \alpha_\mathrm{s} + \xi_{\mathrm{g,f}} \left(\alpha_\mathrm{g} + \frac{K_\mathrm{sg}\lambda_\mathrm{s}\alpha_\mathrm{s}}{\rho_\mathrm{g}} \right)_\mathrm{f} \mathbf{g} \cdot \mathbf{S}.$$

Pressure equation

The volumetric conservation is imposed by solving

$$\nabla \cdot \varphi = \nabla \cdot \left(\alpha_\mathrm{g} \varphi_\mathrm{g} + \alpha_\mathrm{s} \varphi_\mathrm{s} \right) = 0,$$

which gives

$$\nabla \cdot \left\{ \left[\alpha_{\mathrm{g,f}} \xi_{\mathrm{g,f}} \left(\frac{\alpha_\mathrm{g}}{\rho_\mathrm{g}} + \frac{K_\mathrm{sg} \lambda_\mathrm{s} \alpha_\mathrm{s}}{\rho_\mathrm{s} \rho_\mathrm{g}} \right)_\mathrm{f} + \alpha_{\mathrm{s,f}} \xi_{\mathrm{s,f}} \left(\frac{\alpha_\mathrm{s}}{\rho_\mathrm{s}} + \frac{K_\mathrm{sg} \lambda_\mathrm{g} \alpha_\mathrm{g}}{\rho_\mathrm{s} \rho_\mathrm{g}} \right)_\mathrm{f} \right] |\mathbf{S}| \nabla p \right\} = \nabla \cdot \varphi^o,$$

where φ^o is the total volumetric flux from which the contribution of the pressure gradient is removed.

Dispersed phase continuity equation

We write the phase continuity equation so that the particle pressure term appears explicitly

$$\frac{\partial \alpha_\mathrm{s}}{\partial t} + \nabla \cdot (\alpha_{\mathrm{s,f}} \varphi_\mathrm{s}^*) - \nabla \cdot \left[\alpha_{\mathrm{s,f}} \xi_{\mathrm{s,f}} \left(\frac{G(\alpha_\mathrm{s})}{\rho_\mathrm{s}} \right)_\mathrm{f} |\mathbf{S}| \nabla^\perp \alpha_\mathrm{s} \right] = 0,$$

where φ_s^* is the s-phase flux, from which the contribution of the particle pressure is removed

$$\varphi_\mathrm{s}^* = \xi_{\mathrm{s,f}} \left(\mathbb{H}_\mathrm{s} + \frac{K_\mathrm{sg} \lambda_\mathrm{g} \mathbb{H}_\mathrm{g}}{\rho_\mathrm{s}} \right)_\mathrm{f} \cdot \mathbf{S} - \xi_{\mathrm{s,f}} \left(\frac{\alpha_\mathrm{s}}{\rho_\mathrm{s}} + \frac{K_\mathrm{sg} \lambda_\mathrm{g} \alpha_\mathrm{g}}{\rho_\mathrm{s} \rho_\mathrm{g}} \right)_\mathrm{f} |\mathbf{S}| \nabla^\perp p + \xi_{\mathrm{s,f}} \left(\alpha_\mathrm{s} + \frac{K_\mathrm{sg} \lambda_\mathrm{g} \alpha_\mathrm{g}}{\rho_\mathrm{s}} \right)_\mathrm{f} \mathbf{g} \cdot \mathbf{S},$$

$$\nabla p_\mathrm{s} = G(\alpha_\mathrm{s}) \nabla \alpha_\mathrm{s},$$

being

$$G(\alpha_\mathrm{s}) = \frac{\partial p_\mathrm{s}}{\partial \alpha_\mathrm{s}}.$$

Outline

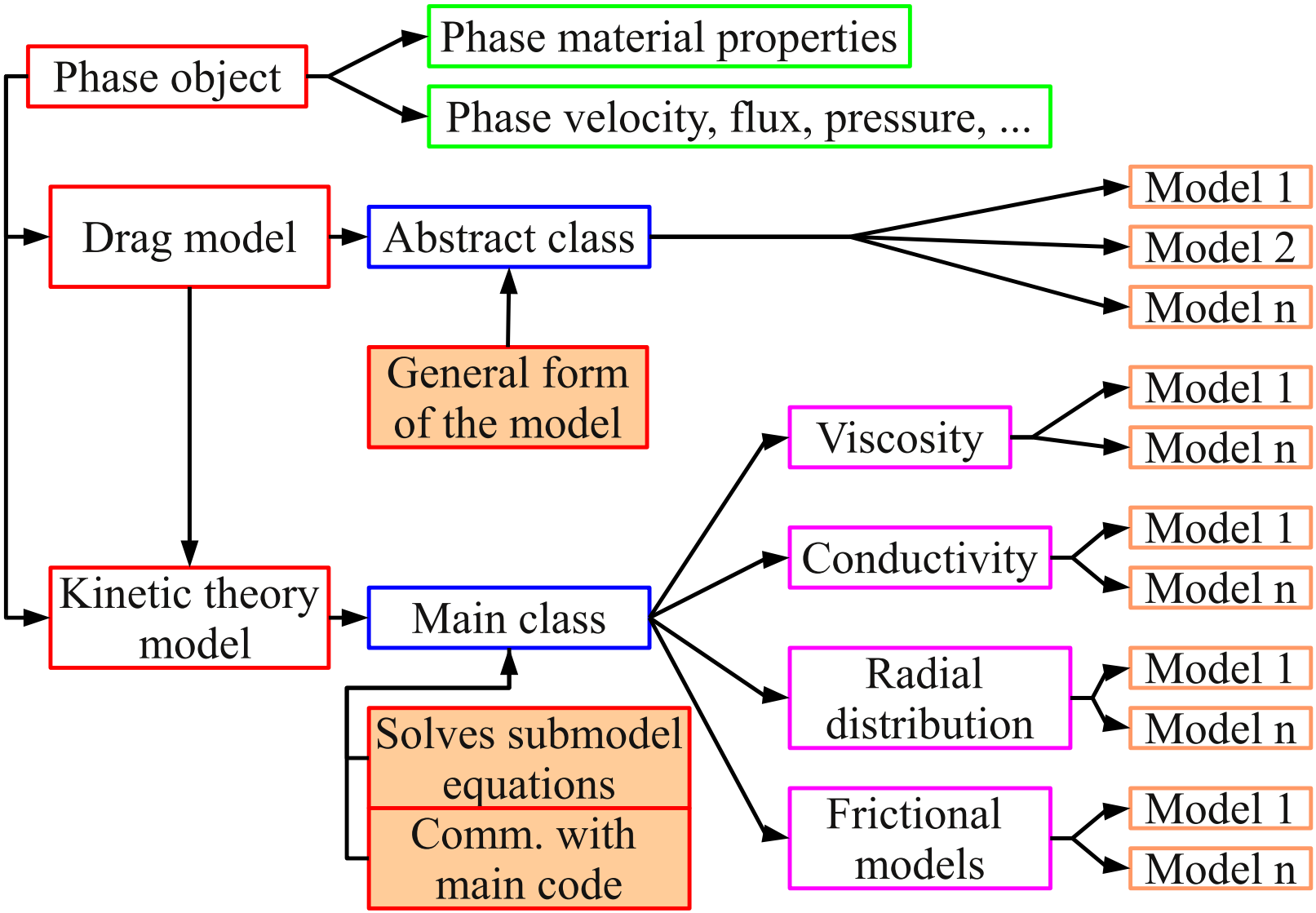
The iterative solution procedure

Following the experience on iterative compressible solvers, we implemented the solution procedure as follows:

1. Solve the dispersed phase continuity equations
2. Compute the continuous phase fraction as $1 - \sum \alpha_s$
3. Update momentum transfer coefficients
4. Solve the granular energy equation
5. Solve the phase momentum equations
6. Solve the pressure equation
7. Correct the phase fluxes
8. Correct the phase velocities (flux reconstruction)

These steps are repeated until a specified convergence criterion, based on total and phase continuity is satisfied.

Some implementation details - Objects

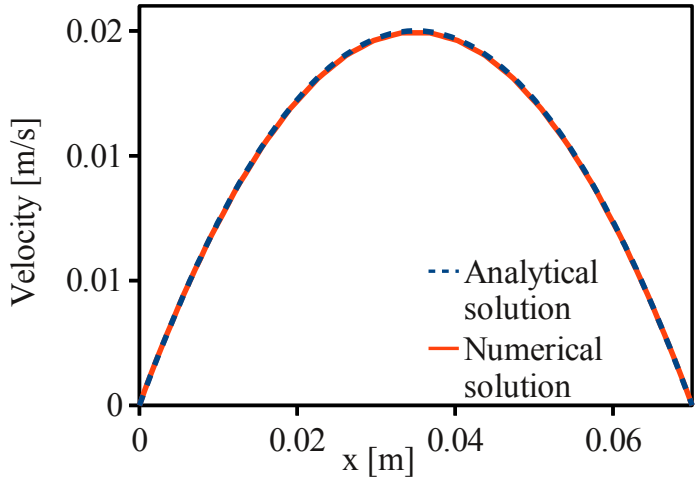


Outline

Single-phase flow

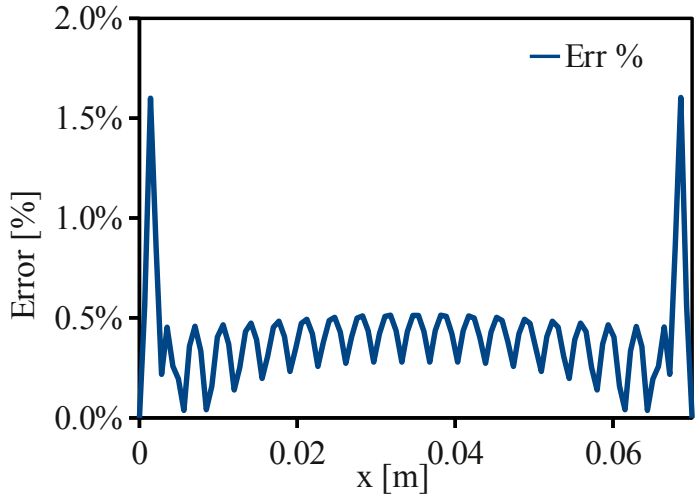
Configuration

- Flow between parallel plates
- Only fluid phase (air)
- No external forces
- Dimensions
  - D = 0.07 m
  - L = 1.0 m
- $|\mathbf{U}_g|_y = 0.01$ m/s
- 40 x 400 computational cells



Observation

The algorithm correctly degenerates into a single-phase solver when the dispersed phase disappears

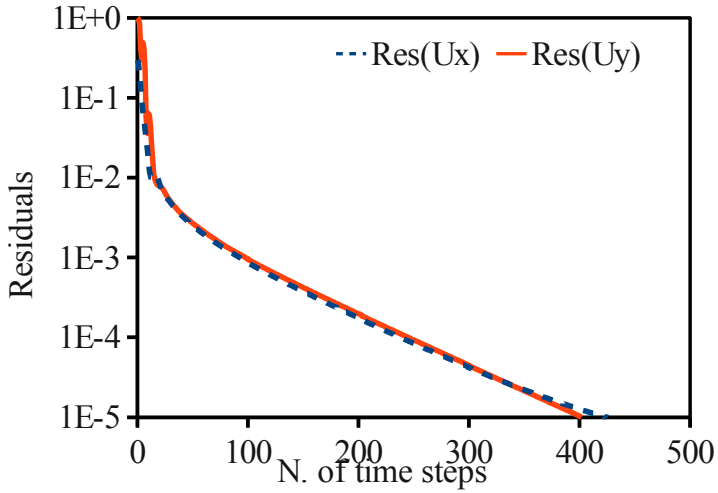


Single-phase flow

Residual convergence

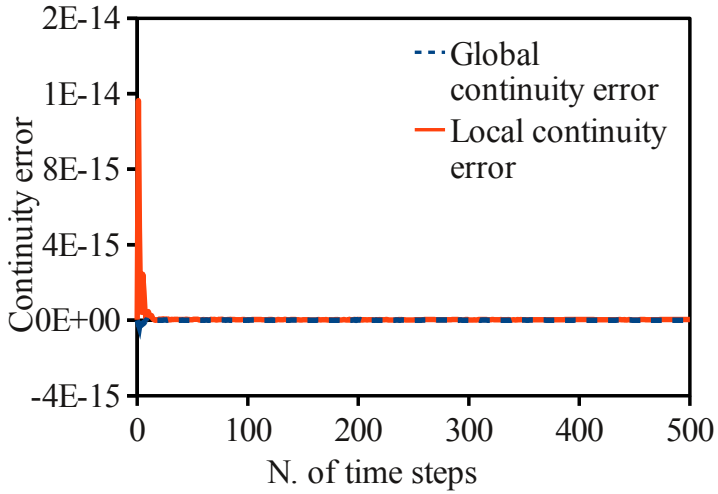
Momentum residuals reduce:

- Below 1.0 x 10^{-3} in 100 iterations
- Below 1.0 x 10^{-5} in less than 450 iterations



Conservation error

Conservation errors are of the same order of magnitude of machine precision (< 1.0 x 10^{-15})



Settling suspension

Geometry

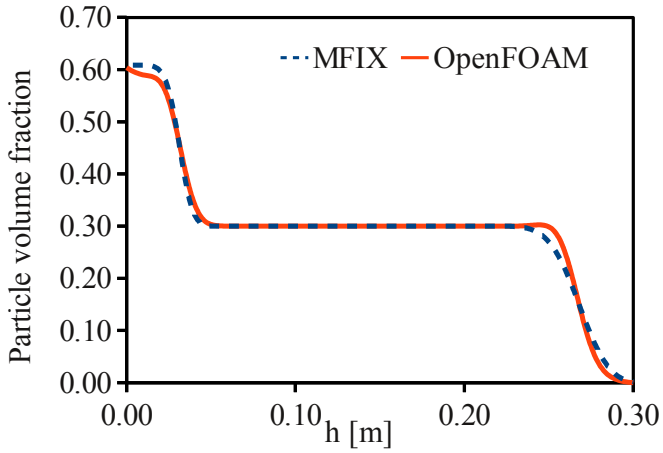
- H = 0.05 m
- D = 0.3 m

Properties

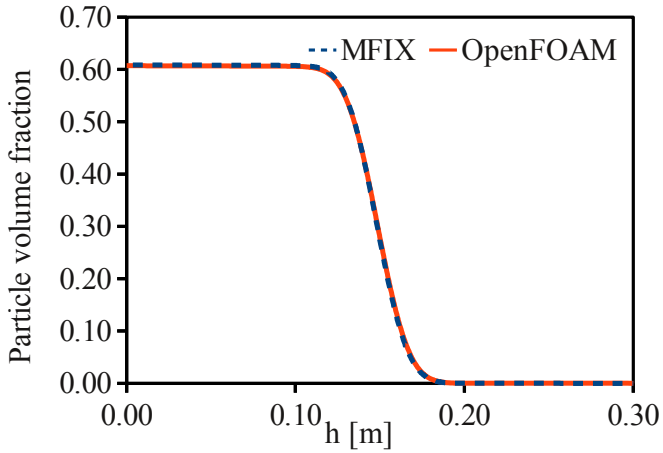
- $\rho_s = 2000\ \mathrm{kg/m^3}$
- $d_p = 400\ \mu\mathrm{m}$
- $\alpha_{s,0} = 0.3$

Numerics

- Grid: 8 x 40 cells
- Second order upwind
- Adaptive Δt



t = 0.1s



t = 8.0s

Bubbling fluidised bed with uniform gas feed

Geometry

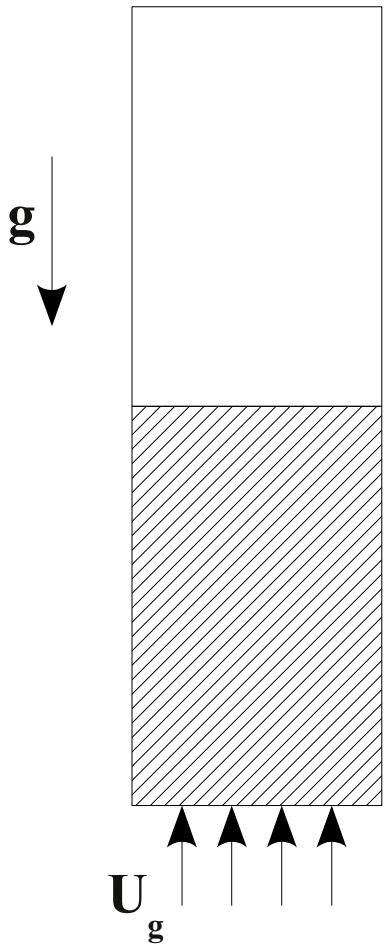
- H = 1 m, D = 0.138 m

Properties

- $\rho_s = 2000\ \mathrm{kg/m^3}$
- $d_p = 350\ \mu\mathrm{m}$, $e = 0.8$
- $\alpha_{s,0} = 0.58$
- $|\mathbf{U}_g|_y = 0.54$ m/s

Numerics

- Grid: 14 x 100 cells
- Second order upwind
- Adaptive Δt



Parmentier et al. (2008)

Bubbling fluidized bed with uniform gas feed

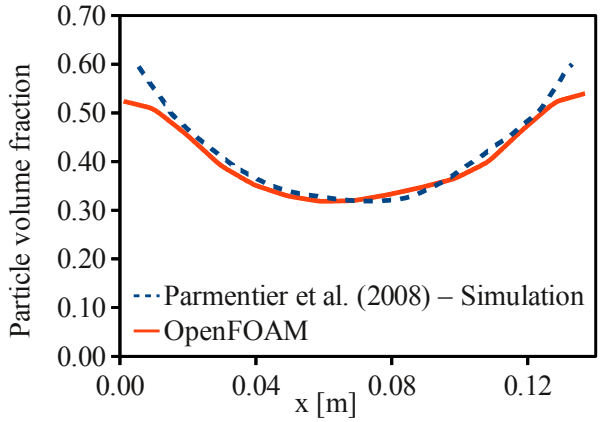
Code verification

Comparison with the numerical results of Parmentier et al. (2008)

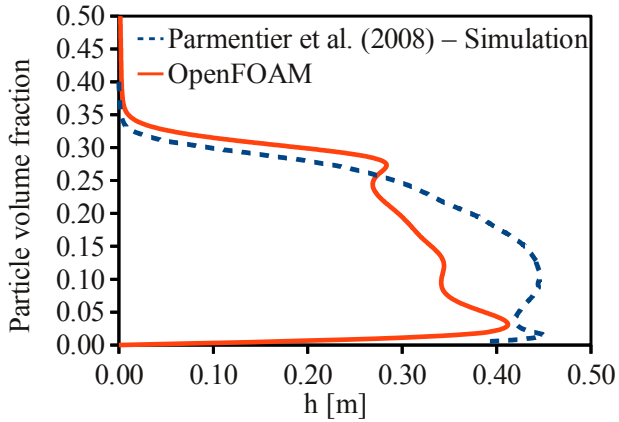
Discussion

- Radial phase fraction profile match the numerical prediction from the literature (Effect of BC)
- Bed expansion predicted in agreement with literature results

Radial particle phase fraction



Axial particle phase fraction



Bubbling fluidized bed with uniform gas feed

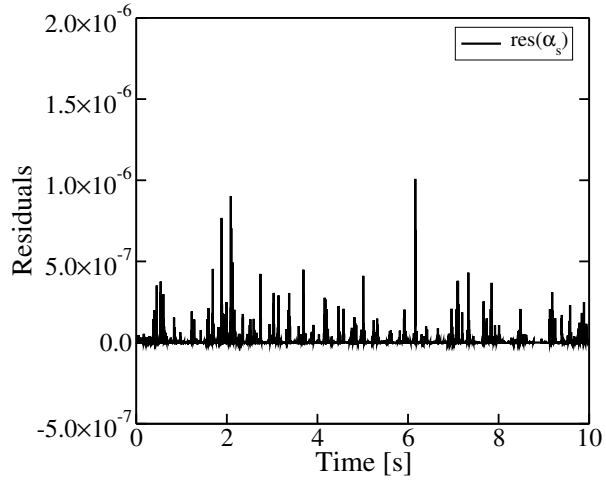
Residual convergence

- The initial residuals of the phase fraction equation are always below 1.0 x 10^{-6} at the end of each time step: good conservation of the dispersed phase
- The residuals of all other variables are required to converge below 1.0 x 10^{-3}

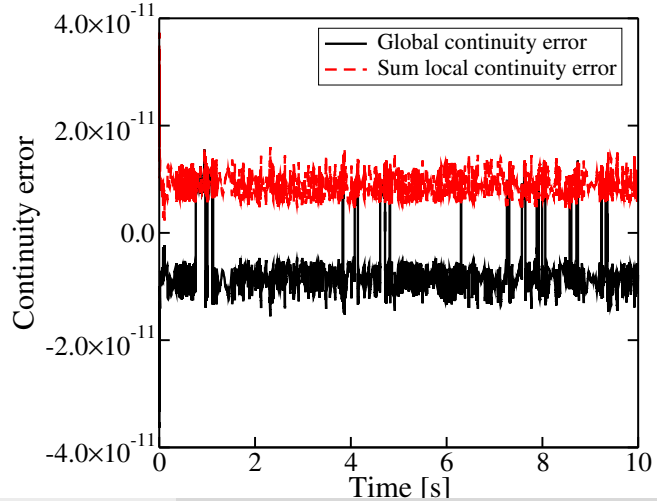
Conservation error

The total continuity error is of the order of 1.0 x 10^{-11} at each time step

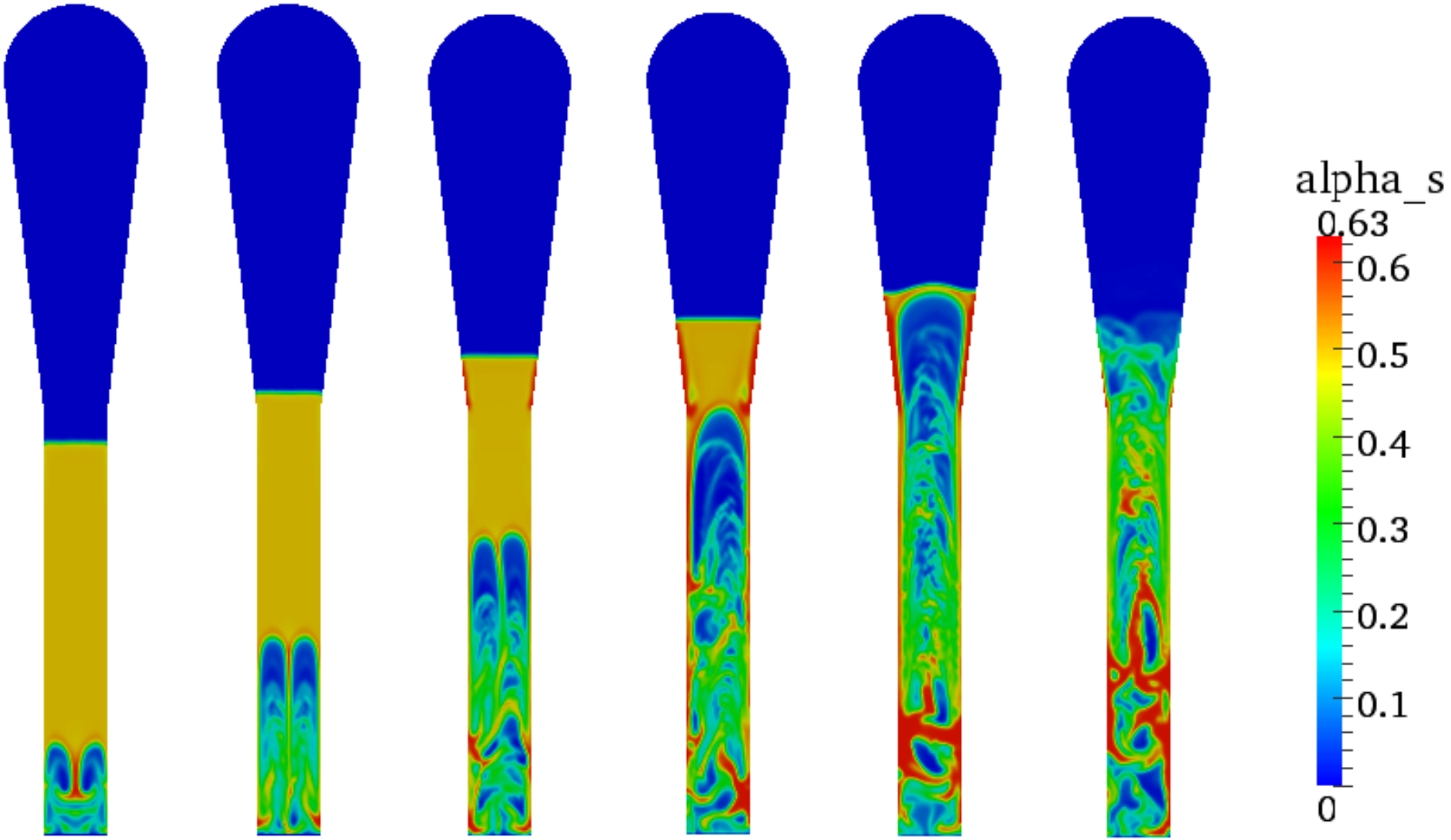
Phase fraction initial res. at final iteration



Total continuity errors



An example of real-life application: pilot plant reactor



Outline

Parallel speedup

Test cases

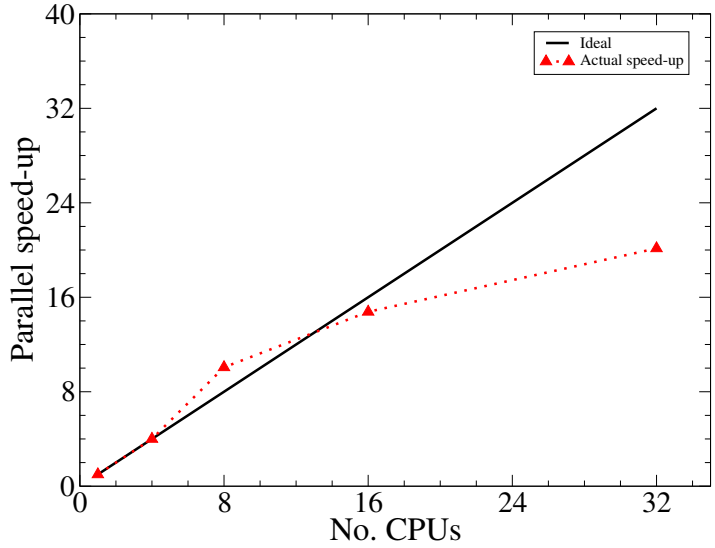
- Settling suspension
- Vertical cylindrical pipe
- Uniform initial conditions

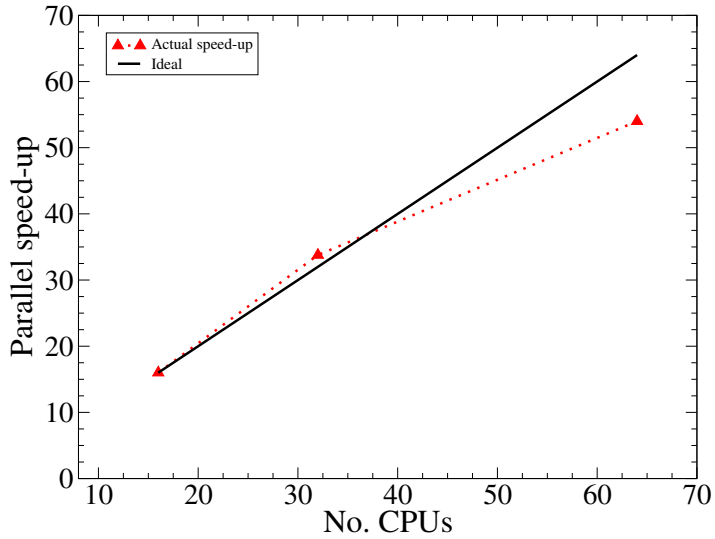
Meshes

- Medium case: 650,000 cells
- Large case: 2,000,000 cells
- Fully hexahedral

Computational system

- Lightningsmp
- Multi-processor multi-core nodes





Outline

Conclusions and future work

Conclusions

- An interative procedure to solve the gas-particle multi-fluid equations on unstructured grids has been implemented in OpenFOAM
- The procedure was verified and provided results in agreement with published results or analytical solutions
- The simulation code was successfully used to model a pilot-plant reactor

Future work

- Integration with QBMM
- Description of a particle size distribution

Acknowledgments

- David Huckaby, Dr. Madhava Syamlal, Dr. Sofiane Benyahia at DOE/NETL
- Dr. Sreekanth Pannala at Oak Ridge National Laboratory
- Henry Weller at OpenCFD Ltd (SGI).
- Juho Peltola at VTT

Disclaimer

Thanks for your attention!

Questions?

References I

J. H Ferziger and M. Peric. *Computational methods for fluid dynamics*. Springer, 2002.

H. Karema and S. Lo. Efficiency of interphase coupling algorithms in fluidized bed conditions. *Computers and Fluids*, 28:323 – 360, 1999.

Y. T. Makkawi, P. C. Wright, and R. Ocone. The effect of friction and inter-particle cohesive forces on the hydrodynamics of gas-solid flow: A comparative analysis of theoretical predictions and experiments. *Powder Technology*, (163):69 – 79, 2006.

J. P. Oliveira and R. I. Issa. Numerical aspects of an algorithm for Eulerian simulation of two-phase flows. *International Journal for Numerical Methods in Fluids*, 43:1177 – 1198, 2003.

References II

I. K. Park, H. K. Cho, H. Y. Yoon, and J. J. Jeong. Numerical effects of the semi-conservative form of momentum equations for multi-dimensional two-phase flows. *Nuclear Engineering and Design*, (239):2365 – 2371, 2009.

J. F. Parmentier, O. Simonin, and O. Delsart. A numerical studiy of fluidization behavior of Geldart B, A/B and A particles using an Eulerian multifluid modeling approach. 2008.

C. M. Rhie and W. L. Chow. A numerical study of the turbulent flow past an isolated airfoil with trailing edge separation. *AIAA Journal*, (21):1525 – 1532, 1983.

D. B. Spalding. Developments in the IPSA procedure for numerical computation of multiphase-flow phenomena with interphase slip, unequal temperatures, etc. In T. M. Shih, editor, *Numerical methodologies in heat transfer*, pages 421 – 436. Hemisphere, 1983.

References III

S. A. Vasquez and V. A. Ivanov. A phase coupled method for solving multiphase problems on unstructured meshes. In *Proceedings of ASME FEDSM'00: ASME 2000 Fluids Engineering Summer Meeting*, Boston, June 2000.

S. Zhang and X. Zhao. General formulations for Rhie-Chow Interpolation. In *2004 ASME Heat Transfer/Fluids Engineering Summer Conference*, number HT-FED04-56274, pages 567 – 573, Charlotte, North Carolina, USA, 2004.

Ensuring volume fraction boundness

Problem

The particle phase volume fraction has to be physically bounded between zero and one. However numerically small negative values or values slightly above one might appear, and destabilize or destroy the numerical solution.

Proposed solutions

- Use first order interpolation schemes to interpolate the phase fraction when defining convective fluxes (Diffusive!)
- Solve for all the phase fractions instead than computing the last one as $1 - \sum \alpha_d$ and re-scale each phase fraction until they are bounded between zero and one and convergence is achieved
- Re-formulate the phase continuity equation so that its solution is bounded (non-linearity appears)

Ensuring volume fraction boundness

Problem

The particle phase volume fraction has to be physically bounded between zero and one. However numerically small negative values or values slightly above one might appear, and destabilize or destroy the numerical solution.

Proposed solutions

- Use first order interpolation schemes to interpolate the phase fraction when defining convective fluxes (Diffusive!)
- Solve for all the phase fractions instead than computing the last one as $1 - \sum \alpha_d$ and re-scale each phase fraction until they are bounded between zero and one and convergence is achieved
- Re-formulate the phase continuity equation so that its solution is bounded (non-linearity appears)

Volume fraction boundness

We consider the total flux and the relative velocity flux:

$$\varphi = \alpha_{\mathrm{g,f}}\varphi_{\mathrm{g}} + \alpha_{\mathrm{s,f}}\varphi_{\mathrm{s}},$$

$$\varphi_{\mathrm{r,s}} = \varphi_{\mathrm{s}} - \varphi_{\mathrm{g}}.$$

The phase flux is then re-written as

$$\varphi_{\mathrm{s}} = \varphi + \alpha_{\mathrm{g,f}}\varphi_{\mathrm{r,s}},$$

which leads, after writing the term containing the particle pressure explicitly, to the modified form of the phase continuity equation

$$\frac{\partial \alpha_{\mathrm{s}}}{\partial t} + \nabla \cdot \left(\alpha_{\mathrm{s}}\varphi^{*}\right) + \nabla \cdot \left(\alpha_{\mathrm{g}}\alpha_{\mathrm{s}}\varphi^{*}_{\mathrm{r,s}}\right) - \nabla \cdot \left[\alpha_{\mathrm{s,f}}\xi_{\mathrm{s,f}}\left(\frac{G(\alpha_{\mathrm{s}})}{\rho_{\mathrm{s}}}\right)_{\mathrm{f}}|\mathbf{S}|\nabla^{\perp}\alpha_{\mathrm{s}}\right] = 0.$$