Towards fluid-particle simulations: CFD - DEM coupling

Davide Fantin

Delft University of Technology

March 21, 2018

Index

Index

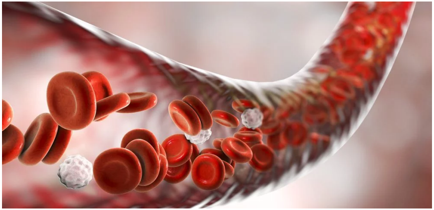
Aim

Study coupling methods for fluid and particle simulation

Highlight pros and cons of different approaches

Develop ideas for improvements in current implementations

Why?



Index

CFD - Fluid Dynamics

Incompressible Navier-Stokes Equations

Conservation of Mass and Momentum

Hypothesis : incompressible flows

$$\begin{cases} \nabla \cdot \mathbf{u} = 0 \\ \rho \left(\dfrac{\partial \mathbf{u}}{\partial t} + \mathbf{u} \cdot \nabla \mathbf{u} \right) = -\nabla p + \mu \Delta \mathbf{u} + \mathbf{f} \end{cases} \tag{1}$$

CFD - Discretization

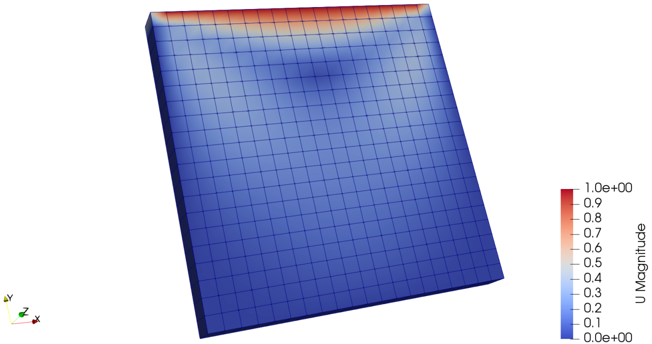
Finite Volume Method
Discretization of the domain in control volumes
Solve equations:

- integrate equation on control volumes
- discretize operators and numerically approximate fluxes
- solve (non)linear systems

OpenFOAM
C++ toolbox, open source, highly customizable
Built-in solvers fot broad set of problems:
examples: basic CFD, (in)compressible flows with RANS and LES, multiphase flows, particle-tracking solvers, combustion, finance

CFD - Example I - icoFoam

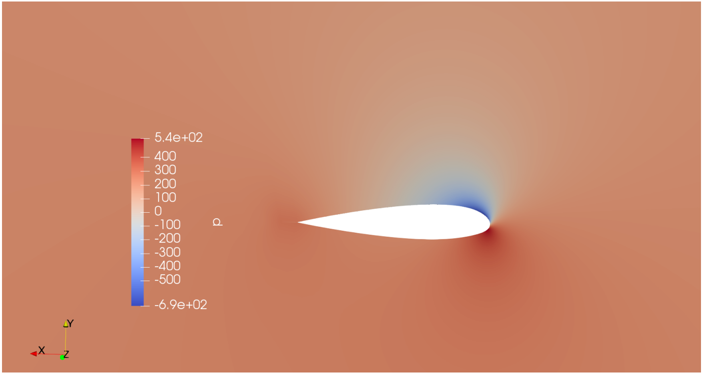
Isothermal, Incompressible, Laminar flow in a 2D square domain



Left, right and down side of the square are walls, at the top the fluid is moved with velocity 1 in the right direction.
Magnitude of velocity is plotted

CFD - Example II - simpleFoam

Incompressible, Turbolent flow in a 2D square domain



Free stream on an airfoil
Magnitude of pressure is plotted.

Index

DEM - Idea

- Calculates trajectory of each particle considering the influences by other particles, walls or other problem-specific forces
- The particle flow is resolved at the particle level
- The motion of a particle consists of a rotational and a translational component ⇒ Newton's Law

$$\begin{aligned} m_i \frac{du_i}{dt} &= F_i \\ I_i \frac{d\omega_i}{dt} &= T_i \end{aligned} \tag{2}$$

DEM - Outline

DEM Procedure

- Initialization:
  choice of force models
  orient particles in space
  assignment of initial position and velocity to particles
- Evaluation of single forces on each particle, according to models
- Evaluation of total forces on each particle
- Numerical integration in time to evaluate the new positions and the velocities of the particles

DEM - Idea

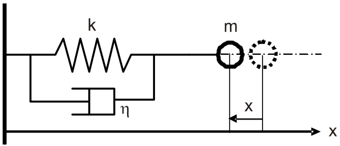
Spherical Particles $\Rightarrow T_i$ is a function of the tangential component of F_i.
How to model F_i?
Exploit linearity: F_i is sum of forces of different nature

- a gravitational component $m_i g$
- particle-particle collision $\sum_{N_p} F_{i,p}$
- particle-wall interactions $\sum_{N_w} F_{i,w}$
- cohesive interactions $\sum_{N_p} F_{i,c}$
- chemical components, ...

$$F_i = m_i g + \sum_{N_p} F_{i,p} + \sum_{N_w} F_{i,w} + \sum_{N_p} F_{i,c} \tag{3}$$

DEM - An example of modeling: contact

Spring-damper system:



$$m\ddot{x} + \eta\dot{x} + kx = 0 \tag{4}$$

Contact force between particle i and j, Hertz-Mindlin theory:

$$F_{nij} = (-k_n\delta_n^{3/2} - \eta_{nj}(\mathbf{v}_i - \mathbf{v}_j)\cdot\mathbf{n})\mathbf{n} \tag{5}$$

$$F_{tij} = \begin{cases} -f|F_{nij}|\mathbf{t} & |F_{tij}| \geq f|F_{nij}| \\ -k_t\delta_t - \eta_{tj}\mathbf{V}_{ct} & \text{else} \end{cases} \tag{6}$$

DEM - An example of modeling: cohesion

Cohesive forces between particle i and j:

Electrostatic $\rightarrow F_{ij} = \dfrac{q_i q_j}{4\pi\varepsilon_0\varepsilon_r l^2}$

Van der Waals: $\rightarrow F_{ij} = \dfrac{A}{12 l^2}\dfrac{D_i D_j}{D_i + D_j}$

Capillary Bridges: $\rightarrow F_{ij} = -2\pi R_{ij}^{*}\gamma\cos\theta$

DEM - Softwares

LIGGGHTS

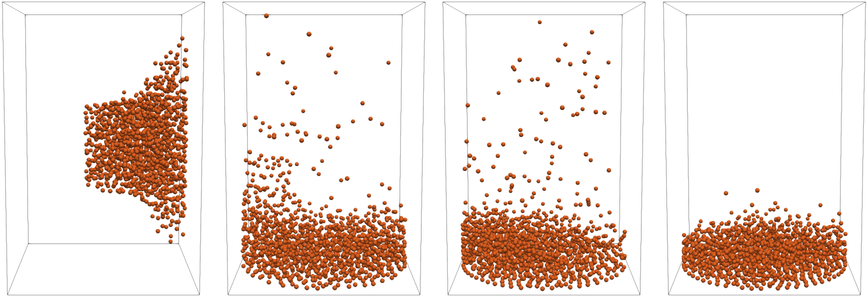
- C++ toolbox
- Supports parallel computing via MPI
- Open Source vs Commercial
- Import and handling of complex geometries

HADES

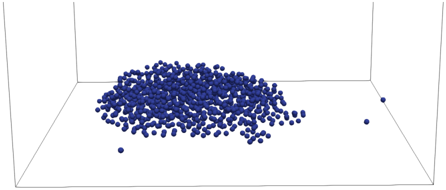
- C++ toolbox
- Based on Jem and Jive
- Open Source
- Geometry input from file available

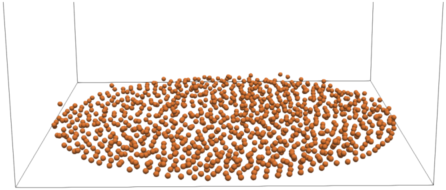
Modular structure

DEM - Example I - LIGGGHTS

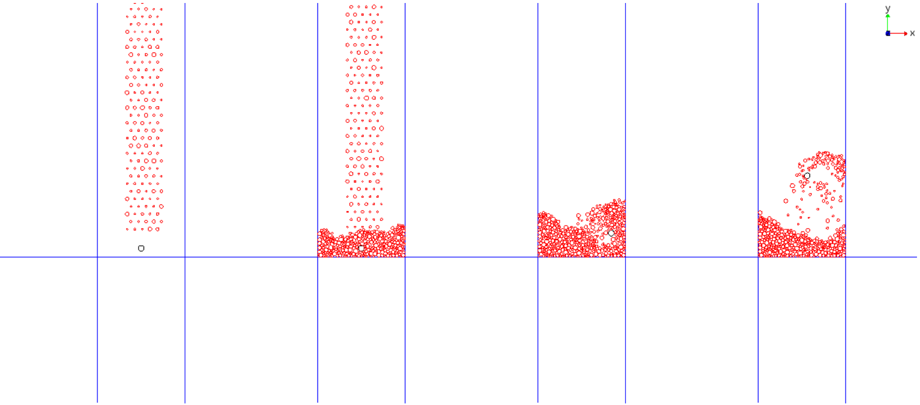


DEM - Example II - LIGGGHTS





DEM - Example III - HADES



Index

Fluid-Particle interactions

Total force fluid exerts on particle is sum of different components

Drag

- Undistrubed flow
- Steady state drag
- Virtual (or added) mass
- Basset term

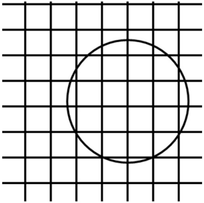
Lift

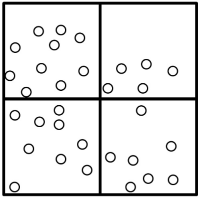
- Magnus lift
- Saffman lift

Coupling CFD - DEM

2 distinct approaches:

- **Resolved Coupling**
  Particles bigger than fluid computational grid
  One particle occupies multiple cells
- **Unresolved Coupling**
  Particles smaller than the computational grid
  Lots of particles occupy one single cell





General issues of Coupling

- Choice of time step Δt
  Necessary to catch collision dynamics and satisfy maximum particle overlap constraint
  DEM-Δt usually at least one order smaller than CFD-Δt
- Contact detection
  How to detect which particles collide every time step in a computationally efficient way?
  Check all possibilities is extremely expensive

How to estimate Δt-s ?

CFD

$$CFL = \Delta t_{CFD} \sum_{i=1}^{n} \frac{u_i}{\Delta x_i}, \tag{7}$$

Measures of how many cells an infinitesimal volume of fluid passes in one time step

$\Rightarrow CFL < 1$ to ensure stability

DEM

Rayleigh time-step: related to the speed of propagation of surface waves in materials that travel near the surface of solids

T_R time necessary for such a wave to travel across the smallest particle

$$T_R = \pi r \frac{\sqrt{\rho / G}}{0.1631\nu + 0.8766} \tag{8}$$

$\Rightarrow \Delta t_{DEM} < T_R$ to ensure stability

How to estimate Δt-s ?

$\boxed{CFD - DEM}$

Model particle relaxation time τ

Large τ is a strong resistance for a particle to adapt to flow motion

Different models for τ : Stokes regime ($Re \ll 1$) $\tau = \dfrac{\rho_p d^2}{18\mu}$

$\Rightarrow \Delta t_{CFD-DEM} < \tau$ to ensure stability

Take into account ALL three different constraints

Contact detection

Neighbour list

A list of potential contacts is built periodically (every N time steps)
Every time-step, list is checked and actual contacts are evaluated
A priori excluding pairs of particles too far away from each other

- Constant time-step Δt
- Maximum particle velocity with magnitude ν_{max}
- Verlet-parameter s

List is valid for N time steps, where:

$$N = \frac{s}{2\nu_{max}\Delta t} \tag{9}$$

Resolved Coupling - Idea

Idea

Add a force term to Navier-Stokes equations, to take into account the influence of solid particles.

$\Leftrightarrow$

- **Fictious Domain Method**
  Firstly, solve fluid equations without considering particles
  Perform a correction of the velocity field of the fluid
- **Immersed Boundary Method**
  Detect fluid areas in which particles lie
  Perform corrections of velocity, pressure and forcing

Resolved Coupling - Fictious Domain - Algorithm

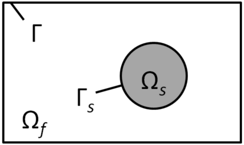
while *(not done)* **do**

- DEM solver: evaluation of positions and velocities of particles)
- Data from DEM solver are passed to CFD solver
- Evaluation of interim velocity field
- Particle tracking: locate cells occupied by each particle
- Correction of velocity in the cells occupied by particles
- Evaluation of fluid forces acting on particles
- Data transfer: CFD solver → DEM solver, for next time step
- Divergence-free correction of velocity field
- Evaluation of other equations (i.e. concentrations, ...)

end

Resolved Coupling - Equations

$$
\begin{cases}
\rho\left(\dfrac{\partial \mathbf{u}}{\partial t} + \mathbf{u}\cdot\nabla\mathbf{u}\right) &= -\nabla p + \mu\Delta\mathbf{u} + \mathbf{f} & \text{in } \Omega_f \\
\nabla\cdot\mathbf{u} &= 0 & \text{in } \Omega_f \\
\mathbf{u} &= \mathbf{u}_\Gamma & \text{on } \Gamma \\
\mathbf{u}(x, t=0) &= \mathbf{u}_0(x) & \text{in } \Omega_f \\
\mathbf{u} &= \mathbf{u}_i & \text{on } \Gamma_s \\
\boldsymbol{\sigma}\cdot\hat{\mathbf{n}} &= \mathbf{t}_\Gamma & \text{on } \Gamma_s.
\end{cases}
\tag{10}
$$



Resolved Coupling - Correction of velocity field

1. Interim velocity field

Solve the Navier-Stokes equation

Use Finite Volume method with the PISO algorithm (*Pressure-Implicit with Splitting of Operators*).

2. New velocity field

Interim velocity is corrected in the particle areas imposing the velocity obtained from the DEM calculations.

Equivalent to adding a force term **f** to the Navier-Stokes:

$$\mathbf{f} = \rho \frac{\partial}{\partial t}(\tilde{\mathbf{u}} - \hat{\mathbf{u}}) \tag{11}$$

Resolved Coupling - Correction of velocity field

3. Final velocity field

New velocity field is not divergence-free.

Apply a correction operator: define a corrected field $\mathbf{u}$

$$\mathbf{u} = \tilde{\mathbf{u}} - \nabla\phi \tag{12}$$

ϕ is an unknown scalar field

$\mathbf{u}$ is forced to satisfy the divergence-free constraint.

Apply divergence operator to (12) $\Rightarrow$ Poisson equation for ϕ:

$$\Delta\phi = \nabla \cdot \tilde{\mathbf{u}} \tag{13}$$

4. Correction of pressure field

Pressure obtained from first step is corrected using ϕ.

Resolved Coupling - Issues

Resolved Coupling is a Direct Numerical Simulations (DNS) method.
⇒ Very precise results require high resolution of the fluid mesh in the area of the particles.
⇒ Huge computational costs, even for small problems.

Possible remedies:

- Dynamic local mesh refinement
- Parallelization

Unresolved Coupling - Idea

Idea
Add a force term to Navier-Stokes equations, to take into account the influence of solid particles (same idea as Resolved Coupling)

BUT
Lots of particles in a discretized CFD cell
$\Rightarrow$ Average Navier-Stokes equation in a cell, introducing

$$\alpha = \text{fluid fraction}$$

Force is added to locally averaged Navier-Stokes

Unresolved Coupling - Algorithm

while *(not done)* **do**

  DEM solver: evaluation of positions and velocities of particles
  Data transfer: DEM solver → CFD solver
  Particle tracking: for each particle determine the cell in which it lies
  Determine the particle volume fraction and a mean particle velocity for each CFD cell
  Evaluation of fluid forces from particle volume fraction, for each particle
  Evaluation of exchange terms, for each cell
  Evaluation of fluid velocity (considering particle volume fraction and exchange terms), for each cell
  Data transfer: CFD solver → DEM solver, for next time step
  Evaluation of other equations (i.e. concentrations, ...)

end

Unresolved Coupling - Equations

α = fluid fraction

$$\begin{cases} \rho\left(\frac{\partial(\alpha\mathbf{u})}{\partial t} + \nabla\cdot(\alpha\mathbf{u}\mathbf{u})\right) = -\alpha\nabla p + \mu\Delta\mathbf{u} + R_{pf} \\ \frac{\partial\alpha}{\partial t} + \nabla\cdot(\alpha\mathbf{u}) = 0. \end{cases} \tag{14}$$

$$R_{pf} = K_{pf}(\mathbf{u} - \mathbf{u}_p) \tag{15}$$

Usually drag forces are dominant $\Rightarrow$

$$K_{pf} = \frac{\sum_i f_{d,i}}{V} \tag{16}$$

Unresolved Coupling - Issues

Calculation of Lagrangian properties from Eulerian data
Necessary for:

1. Solid volume fraction α_s.
2. Solid phase velocity $\mathbf{u}_p$.
3. Fluidparticle interaction force R_{pf}.

Procedure: Coarse graining (or Averaging)
Different approaches available

Unresolved Coupling - Issues

- Particle centroid method (PCM)
  Sum over all particle volumes in each cell to get cell-based α_s
  Easy to implement but unstable: possible $\alpha_s > 1$, ...
- Divided particle volume method (DPVM)
- Statistical Kernel method
  Volume of each particle in entire domain
  Weight functions $h(\mathbf{x}) \Rightarrow$ At location $\mathbf{x}$: superposition
- **Diffusion-based coarse graining**
  Use PCM and take it as initial value for a diffusion equation
  Easy to implement, stable results with just 3 pseudo-time steps τ

Diffusion-based coarse graining

Example: Particle volume fraction

$$\begin{cases} \dfrac{\partial \alpha_s}{\partial \tau} - \Delta \alpha_s & = 0 \\ \alpha_s(\mathbf{x}, 0) & = \alpha_s^0(\mathbf{x}) \qquad \leftarrow \text{from PCM} \end{cases} \tag{17}$$

Integration grid: same as CFD equation
Just 3 integrations steps are enough

Time-steps obtained from Gaussian analogy (Statistical Kernel Method)
Gaussian bandwidth: $b = \sqrt{4T}$
Impose $b = 6d_p \Rightarrow$ Get T

Simultaneous Resolved-Unresolved

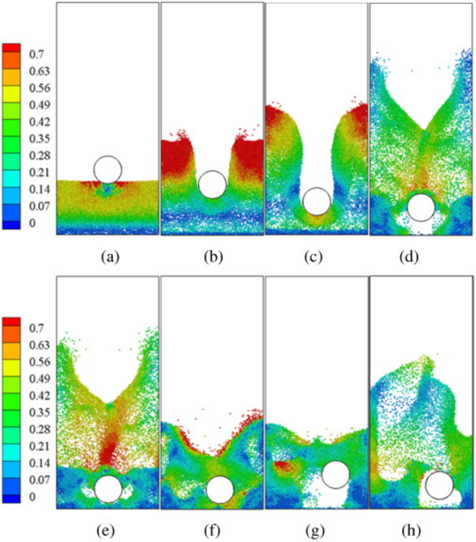
How to handle simultaneous presence of large and small particles?

First attempts of stable numerical methods for handling both cases

Nov 2017: Paper on coupling on ANSYS/Fluent platforms

1. Perform Unresolved, considering influence of fluid flow, large particles and walls. DEM until CFD time is reached ($\Delta t_{DEM} \ll \Delta t_{CFD}$)
2. Perform Resolved, considering influence of fluid flow, small particles and walls

Simultaneous Resolved-Unresolved



Coupling softwares available

- `DPMFoam`
  Library provided by the standard distribution of OpenFOAM.
  Not DEM, but DPM (clouds of particles). Not used
- `sediFoam`
  Couples OpenFOAM 2.4.0 with LAMMPS 1-Fed-2014.
  Focus: simulation of sediment transport and fluidized beds.
  Diffusion-based coarse graining implemented
  Stable for particles sizes at most two times the CFD cell size.
- CFDEM
  Set of libraries, couples OpenFOAM-5.x and LIGGGHTS-3.8.0
  Most complete implementation of the coupling.
  Resolved: Fictious Domain Method
  Unresolved: Diffusion-coarse graining
- `coupledPimpleFoam`
  Internal library for OpenFOAM, expansion of the standard `PimpleFoam`
  Simple adding term to the Navier Stokes equation for forces arising from the presence of particles: unresolved.

Index

Possible future directions ?

- Improvement of `pimpleCoupledFoam`:
  Unresolved: implement a diffusion-based coarse graining procedure
  Resolved: implement a fictious domain method or PISOIB
  Current problems:
  1. DEM update is performed at every CFD step.
     Not efficient since the Δt_{DEM} needs to be small
  2. PCM is adopted as interpolation procedure. Huge gradients in the exchange terms.
  3. Particle volume fraction α_s is not calculated. Various drag models require it
- Implementation of a coupling from scratch OpenFOAM-HADES
  Extensive coding. Necassary a preliminary study on the actual feasibility
- Theoretical research on optimal Δt_{DEM}, Δt_{CFD}, $\Delta t_{Coupling}$,