
This version is available at https://strathprints.strath.ac.uk/59699/

Strathprints is designed to allow users to access the research output of the University of Strathclyde. Unless otherwise explicitly stated on the manuscript, Copyright © and Moral Rights for the papers on this site are retained by the individual authors and/or other copyright owners. Please check the manuscript for details of any other licences that may have been applied. You may not engage in further distribution of the material for any profitmaking activities or any commercial gain. You may freely distribute both the url (https://strathprints.strath.ac.uk/) and the content of this paper for research or private study, educational, or not-for-profit purposes without prior permission or charge.

Any correspondence concerning this service should be sent to the Strathprints administrator: strathprints@strath.ac.uk
A hybrid slurry CFD model: Euler-Euler to Euler-Lagrange (in development)

Alasdair Mackenzie
Outline

- Background, context and motivation to the problem
- Development of hybrid model will be explained
- Test case will be shown

- Tutorial can be found on Chalmers website (end of January): http://www.tfd.chalmers.se/~hani/kurser/OS_CFD_2016/
Background

- Weir group produce equipment for the mining and oil and gas industries
- Erosion is a large problem
- CFD modelling is used to predict erosion = better designs
- Longer pump life = happy customer :)

Weir
Advanced Research Centre

5th United Kingdom & Éire OpenFOAM® User Meeting
Impellers

Before

After

Could be as little as 2 weeks of continuous running for this to happen

5th United Kingdom & Éire OpenFOAM® User Meeting
Problem/Motivation

- Need particle impact data at the wall for erosion modelling
- Fluid/particulate flow simulation is computationally expensive: especially for dense slurries
- Solution to make faster: Combine with two-fluid model

Velocity contours of submerged jet impingement test

Dotted region where particles are necessary for impact data
Geometry and Solvers

- A simple geometry was chosen for solver development
- `reactingTwoPhaseEulerFoam` for Euler-Euler
- `DPMFoam` for Euler-Lagrange
- `OpenFOAM 3.0.x` was used

Geometry shown with sizes in metres

5th United Kingdom & Éire OpenFOAM® User Meeting
Description of Solvers

**Euler-Euler**

- Two fluid model
- Both phases treated as continuum
- Incompressible model: setting in dictionary
- Fast to solve

**Euler-Lagrange**

- Fluid/particle model
- Transient solver for coupled transport of kinematic particle clouds
- Includes the effect of volume fraction of the particles on the continuous phase

**reactingTwoPhaseEulerFoam**

**DPMFoam**

5th United Kingdom & Éire OpenFOAM® User Meeting
Combining the solvers

- A new solver was made based on the EE model
- To have 2 solvers running, 2 regions were created
- To go from fluid to particles, we need a transition
- An outlet/inlet is needed for particle phase, but shouldn't affect the rest of the flow
- Solution…
**Baffles + Regions**

- `createBaffles`: makes internal surface into boundary face
  - *master* and *slave* patch created
- `splitMeshRegions`: Splits mesh into 2 separate regions
- BC’s can now be applied to baffle patches
- `chtMultiRegionFoam`: Inspiration for solving regions sequentially
Interpolation

- **patchToPatchInterpolation:** transfers data between two patches

- All variables are interpolated: U1, U2, p, p_rgh, alpha1, alpha2, k, epsilon, nut, and theta

- After this is implemented, the domain runs as if it was one region, not two: the surface doesn’t affect the flow

- ‘back pressures’ are taken into account by interpolating upstream

*Iterative loop*

- Solve in Region0
  - Interpolate from master patch to slave patch
  - Solve in Region1
  - Interpolate from slave patch to master patch

5th United Kingdom & Éire OpenFOAM® User Meeting
DPMFoam added

- Code from DPMFoam was added to new solver
- Particles injected from slave patch after back interpolation (slave to master)
- Particles are only in region1 (near wall)
- Injection values based on phase 2 from region0 by using a lookup table: kinematicLookupTableInjection
DPMFoam injection

- Modified kinematicLookupTableInjection used to inject particles
- Lookup table is updated every time step (but not read every time step: advice welcome!)
- 1 line = 1 cell (100 cells in this case)
- Values for particle injection are based on new updated values so solver can deal with geometry changes etc. See Lopez’ presentation for more details:


5th United Kingdom & Éire OpenFOAM® User Meeting
DPMFoam injection

- Number of parcels to be injected is calculated from volume flow rate of 2nd phase of fluid.

- Number of parcels/cell = (alpha particles * area of cell * normal velocity component to cell boundary face) / (volume of one particle * number of particles/parcel * number of time-steps/second)
Velocity contours

- 2D slice through Z normal. Particles injected from slave patch
Velocity contours
Comparison

- New solver was compared against standard EL and EE solvers
- Hybrid model is almost double the speed of the EL

Execution time from 0-0.39s: % mass concentration (MC)

<table>
<thead>
<tr>
<th>Model</th>
<th>1% MC</th>
<th>2% MC</th>
</tr>
</thead>
<tbody>
<tr>
<td>Hybrid model</td>
<td>225</td>
<td>298</td>
</tr>
<tr>
<td>Euler-Lagrange</td>
<td>420</td>
<td>585</td>
</tr>
<tr>
<td>Euler-Euler</td>
<td>102</td>
<td>105</td>
</tr>
</tbody>
</table>
Comparison

Hybrid Model particle impacts

Data taken from bottom wall on pipe bend

DPMFoam particle impacts

5th United Kingdom & Éire OpenFOAM® User Meeting
Future work

- Validation of hybrid model: CFD and experimental (PIV)
- Particles to fluid, for after region of interest…
- Move lookupTable to memory?
- Make solver re-read the lookupTable (suggestions welcome)
Conclusion

- Solver should dramatically reduce computational time
- Particle data should still be present near walls, where required
- Enable better design of mining equipment

Worn impeller of slurry pump
Thank you. Questions?

alasdair.mackenzie.100@strath.ac.uk