Ninth International Conference on CFD in the Minerals and Process Industries
CSIRO, Melbourne, Australia
10-12 December 2012

SPH Simulation of Packed-beds and Columns Applied to Heap-leaching

Daniel J. BARKER[1*] Gopalkrishnan PARAMESWARAN[1] and Stephen J. NEETHLING[1]

[1] Rio Tinto Centre for Advanced Mineral Recovery, Department of Earth Science and Engineering, Imperial College London, London, SW7 2AZ, UK
*Corresponding author, E-mail address: D.Barker11@imperial.ac.uk

ABSTRACT

Smoothed Particle Hydrodynamics is fast becoming an indispensable tool for simulating complex flows such as those encountered in Heap-leaching. Its meshfree, Lagrangian nature means it is ideally suited to handle multiphase immiscible flows, particularly with a simulation scale on the order of centimetres. We present an overview of the SPH method for simulating such flows and present two novel considerations for handling fluid-solid boundaries in SPH with both geometric objects and faceted meshes. We show results of simulations in 3D through packed-beds of spherical particles and realistic rock shapes.

NOMENCLATURE

d	spatial dimension
h	smoothing length
W	smoothing kernel
P	pressure
$\mathbf{v}$	velocity
ρ	density
ρ_0	rest density
c_s	speed of sound
μ	dynamic viscosity
γ_{kl}	surface tension
g	acceleration due to gravity
L	period length
dx	avg. particle spacing
V	particle volume
m	particle mass
B	Bond number

INTRODUCTION

Heap-leaching (HL), a method for extracting valuable metals from mined ore, accounts for approximately 20% of world copper production (Jergensen, 1999). HL has comparatively low setup and running costs. This means that for ore which is of low grade or for small ore bodies (which are rapidly depleted) HL represents the only economically viable extraction technique.

In HL a reactive *leaching solution* is trickled over ore particles where it percolates down, reacting with the metal sulphides bringing the copper into solution. Once the solution reaches the bottom of the ore pile, an impermeable leaching pad channels the flow into drainage ditches and the copper is extracted using solvent-extraction / electro-winning (SX/EW)

The leaching process is subject to many external factors (e.g. rainfall, ambient temperature) and is itself a complex process. Further, once running only certain parameters can be controlled, namely; leaching solution flow rate, E_h and pH and gas flow rate. For these reasons fully understanding HL provides a massive opportunity to maximise recovery and minimise footprints for mining operations.

Several past works have been undertaken in this direction ranging from computational studies at the whole heap-scale (Cariaga 2005, Leahy 2003, McBride 2005), theoretical work such as (Dixon, 1995) to experiments (Bouffard 2001, Wu 2009).

We use Smoothed Particle Hydrodynamics (SPH) to simulate the type of unsaturated multiphase flows encountered in HL. The flows are simulated at the scale of particles and intra-particle channels which are found in typical industrial heaps; the aim of which is to generate a better understanding of the interplay between flow, hold-up and particle properties such as shape, size and contact-angle.

MODEL DESCRIPTION

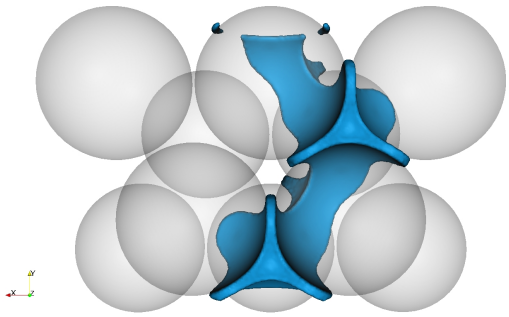

Figure 1: Final state of simulation with B=0.78 showing water held between spherical particles of diameter 2mm.

Smoothed Particle Hydrodynamics

Originally developed to simulate astrophysical flows (Lucy, 1977), SPH is a meshless, fully Lagrangian numerical method. It can be used to simulate the full 3D Navier-Stokes (NS) equations. For a Newtonian fluid and in its Lagrangian form the NS equation reads

$$\rho \frac{d\mathbf{v}}{dt} = -\nabla P + \mu \nabla^2 \mathbf{v} + \mathbf{g} \qquad (1)$$

and the continuity equation

$$\frac{d\rho}{dt} + \nabla \cdot (\rho \mathbf{v}) = 0 \qquad (2)$$

describes the conservation of mass within the system. SPH works by discretising the system at a set of points or

'particles', but unlike some other methods (e.g FEM) there is no pre-defined topology between the SPH particles. Using the value of a quantity at these particles we can form an approximate interpolation where the contribution from each particle is weighted based upon its distance away:

$$A(\mathbf{x})=\sum_{i=1}^{N} A_i W(|\mathbf{x}-\mathbf{x}_i|,h)V_i \tag{3}$$

Where W is the smoothing kernel. Note that this is not a true interpolation because $A(\mathbf{x}_i)\neq A_i$. It can be shown, see (Liu, 2010), that the gradient of the estimated quantity (3) is given by

$$\nabla A(\mathbf{x})=\sum_{i=1}^{N} A_i \nabla W(|\mathbf{x}-\mathbf{x}_i|,h)V_i \tag{4}$$

Notice how the gradient is transferred from the quantity itself to the smoothing kernel. A similar relation to (4) holds for the SPH Laplacian.

Each SPH particle i represents a small volume of the fluid and as such has the physical properties of mass – m_i, density – ρ_i, pressure – P_i and velocity – $\mathbf{v}_i$. Further, since we are interested in multiphase flows each fluid particle is given a 'colour' indicating to which phase it belongs and the colour function is defined as

$$C_i^k=\begin{cases}1 & i\in k\\ 0 & \text{otherwise}\end{cases} \tag{5}$$

This function indicates whether the particle i is in fluid phase k.

Using these equations it is possible to write SPH estimates for the relevant physical quantities we need to simulate the fluid flow. We follow closely the formulation of Hu and Adams (2006) beginning with

$$\sigma_i=\sum_{j=1}^{N} W(|\mathbf{x}_i-\mathbf{x}_j|,h) \tag{6}$$

where the density is then given by

$$\rho_i=m_i\sigma_i \tag{7}$$

and we have approximately $\sigma_i=1/V_i$. The pressure is directly related to the density through an equation of state; the Tait equation (Becker, 2007). We can then write the SPH estimates of the flow variables. The pressure gradient is given by

$$-\nabla P_i=-\frac{1}{V_i}\sum_{j=1}^{N}\left(\frac{P_i}{\sigma_i^2}+\frac{P_j}{\sigma_j^2}\right)\nabla W_{ij} \tag{8}$$

with

$$W_{ij}=W(|\mathbf{x}_j-\mathbf{x}_i|,h) \tag{9}$$

The viscous term is given by

$$(\mu\nabla^2\mathbf{v})_i=\frac{1}{V_i}\sum_{j=1}^{N}\frac{2\mu^k\mu^l}{\mu^k+\mu^l}\left(\frac{1}{\sigma_i^2}+\frac{1}{\sigma_j^2}\right)\frac{\mathbf{v}_{ij}}{r_{ij}}\frac{\partial W_{ij}}{\partial r_{ij}} \tag{10}$$

where the superscripts k and l indicate the fluid phases for particles i and j respectively. Surface tension can be successfully added to SPH simulations using the continuum surface force (CSF) model. The colour gradient at a particle i in phase k due to neighbouring particles from phase l is

$$\nabla C_i^{kl}=\frac{1}{V_i}\sum_{j\in l}\left(\frac{C_i^l}{\sigma_i^2}+\frac{C_j^l}{\sigma_j^2}\right)\nabla W_{ij} \tag{11}$$

The interface stress between phases k and l ($k\neq l$) is then

$$\Pi_{kl}^{(i)}=\frac{\gamma_{kl}}{|\nabla C_i^{kl}|}\left(\frac{\mathbf{I}}{d}|\nabla C_i^{kl}|-\nabla C_i^{kl}\nabla C_i^{kl}\right) \tag{12}$$

and the total interface stress is the sum over each other phase

$$\Pi_i=\sum_{k\neq l}\Pi_{kl}^{(i)} \tag{13}$$

All simulations carried out in this paper are with a contact angle of θ=30° between the liquid and solid phases. To achieve this, surface tension coefficients are chosen using Young's equation

$$\gamma_{sl}-\gamma_{sg}=\gamma_{lg}\cos(\theta) \tag{14}$$

With l,g and s representing liquid, gas and solid phases respectively. It is worth noting that this SPH formulation exactly conserves linear momentum in the absence of any fixed boundaries.

Timestepping

In the SPH literature the Leap-frog scheme is commonly used for integrating the equations of motion. Here we use a predictor-corrector scheme, with adaptive timestepping to increase computational efficiency.

Boundaries

The handling of solid-fluid boundaries in SPH has been approached in several different ways. The three main ways are by simply making the walls repulsive to the SPH particles (Harada 2007, Müller 2004), by using ghost SPH particles (Monaghan 2009, Ovaysi 2010) and using semi-analytical formulations (Feldman 2010, Ferrand 2012). We take the approach of having reflected ghost particles.

SPH particles within $2h$ of any walls are reflected across the wall and their velocities are also reflected, i.e. $\mathbf{v}_{ref}$ = -$\mathbf{v}$. Reflecting the velocities like this is important to ensure the no-slip boundary conditions at the wall.

Because of the smoothed nature of the velocity field, if the reflected particles have, for instance, zero velocity then the particles at the wall do not see the smoothed velocity field as being exactly zero at the boundary.

Geometric Objects

As a first step towards simulating realistic beds and columns we will show some simulations through packed beds of spherical particles. By geometric objects we mean simple objects such as spheres, circles, torii and cylinders whose surface can be represented mathematically.

Rather than breaking these surfaces up into many flat triangular faces we store in our SPH code the mathematical constants necessary to define them. So for a sphere we would store its centre $\mathbf{x}_c$ and its radius R. This has the advantage that simulations are more robust to changes in resolution (smoothing length).

For example, using the centre and radius of a sphere a reflected particle's position is given by

$$\mathbf{x}_{refl}=\frac{(\mathbf{x}-\mathbf{x}_c)}{|\mathbf{x}-\mathbf{x}_c|}(2R-|\mathbf{x}-\mathbf{x}_c|) \tag{15}$$

This is the same for all particles around the sphere.

However care must be taken when performing this reflection operation. Since the surface in which we are

reflecting is curved the reflected particles necessarily occupy a different volume V' to that of their matching fluid particle – V. An example of this effect can be seen in figure 2; the fluid particles (green) are shown reflected in the circle (blue) to give the reflected particles (red). See how the curved edge of the circle leads to the reflected particles bunching closer together than the average spacing dx.

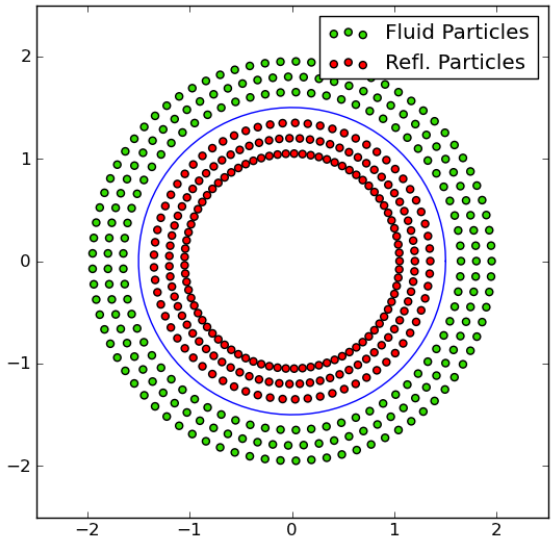

Figure 2: Illustration of how particle spacing is affected by reflection in a circle.

For a circle the ratio of the areas of the reflected and original particle is given by

$$V_{fac}^{\text{circ}} = \frac{R+d}{R-d} \tag{16}$$

where d is the distance between the particles and the circle. It is simple to show that this is the case by considering the geometric construction shown in Figure 3 in which we work from the fact that on the fluid side of the circle (here the inner side) the spacing between particles is dx.

From equation (16) it is possible to generalise to arbitrary shapes in 2D and 3D. Consider an arbitrary shape in 2D, then at the closest point (the point across which we reflect) we have a defined radius of curvature which becomes R in equation (16). We can rewrite this in terms of the curvature $\kappa = 1/R$ yielding

$$V_{fac}^{2D} = \frac{1+\kappa d}{1-\kappa d} \tag{17}$$

In 3D we have two *principle curvatures* which are orthogonal to one and other – κ_1 and κ_2. The volume factor in 3D is the product of the two 2D volume factors which would be obtained by circles with curvatures κ_1 and κ_2 respectively:

$$V_{fac}^{3D} = \frac{(1+\kappa_1 d)}{(1-\kappa_1 d)}\frac{(1+\kappa_2 d)}{(1-\kappa_2 d)} \tag{18}$$

The volume (area) factor is shown for a number of shapes in Table 1. In the table R and d have their usual meanings and the torus is defined by the two radii R_1 (wheel radius) and R_2<R_1 (tube radius), with the poloidal coordinate being θ where θ=0 is defined to be the inside ring.

Shape	Dim.	V_{fac}
Circle	2	$(R+d) / (R-d)$
Sphere	3	$((R+d) / (R-d))^2$
Cylinder	3	$(R+d) / (R-d)$
Torus	3	$\left(\frac{R_2+d}{R_2-d}\right)\left(\frac{R_1-R_2-d}{R_1-R_2+d}\right)\cos(\theta)$

Table 1: Volume factors for reflecting in various shapes.

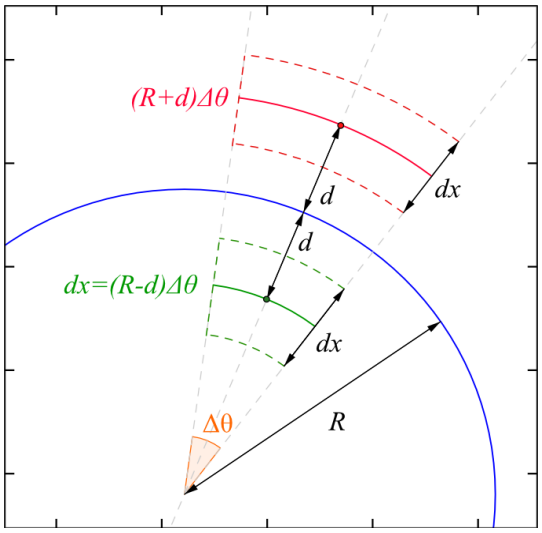

Figure 3: Geometric construction used for calculating the volume factor associated with reflecting across an edge with radius of curvature R.

A modified version of equation (6) is used to ensure that $\sigma_i = 1/V_i$ holds near curved boundaries;

$$\sigma_i = \frac{1}{V_{fac,i}} \sum_{j=1}^{N} W_{ij} V_{fac,j} \tag{19}$$

The representation of solids as mathematical objects is practical because it is very memory efficient plus the resulting simulations tend to be stable. Spheres and cylinders are good for modelling lab experiments where idealised geometries are sometimes used (Bouffard 2001), however if we are to move towards realistic geometries we must use *faceted geometries*.

Faceted Objects

Possibly the most common way of representing 3D objects on computers is by breaking the object up into a series of triangles (and sometimes quadrilaterals). The resulting triangles are called the *facets* of the object. Note since the triangles represent a 2D surface of a 3D object they must be locally manifold meaning at each edge exactly two triangles must meet. Additionally the surface must be orientable (closed).

Unlike the geometric objects discussed above, the triangles of a faceted geometry are flat meaning the curvature of the individual facets is identically zero; this means we no longer need to worry about the volume factors when reflecting particles in them. Unfortunately faceted geometries suffer a different issue which must be remedied.

Edges

Where two triangles meet at an edge the angle between them is in general not 180°. If ignored this can

cause either (i) overlapping regions of reflection or (ii) gaps in the reflected particle field, as shown in figure 4.

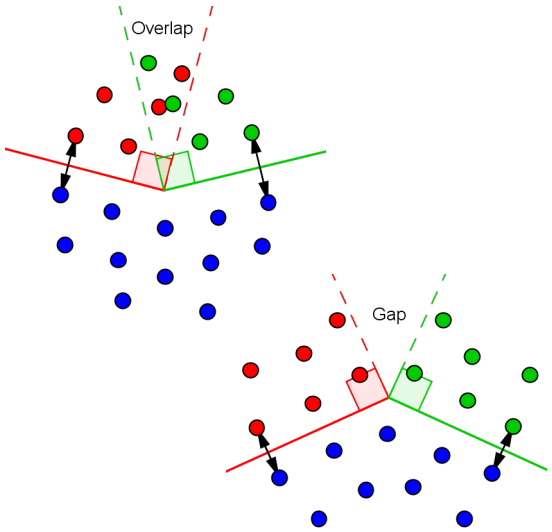

Figure 4: In general there will be either an overlap or a gap in reflection where two triangles meet at an edge (vertex in 2D).

Case (i) occurs when the angle between outward normals $\mathbf{n}_1$ and $\mathbf{n}_2$ is >180º and will lead to overestimation of the density near such an edge. Case (ii) occurs when the angle between $\mathbf{n}_1$ and $\mathbf{n}_2$ is <180º and will cause an underestimation of the density. Case (ii) can become a serious issue as an underestimated density can allow some particles to leak from the simulation domain. A similar issue can occur where triangles meet at vertices.

To overcome these issues, our SPH code was modified to detect when reflecting a particle near an edge and if necessary reflect an additional particle across the edge into the resulting gap (case ii) or not reflect an otherwise reflected particle (case i). To detect if a reflected particle would lie in the region of interest we can perform a relatively straightforward calculation.

Consider the edge *e* with endpoints $\mathbf{v}_1$ and $\mathbf{v}_2$, and a particle at position $\mathbf{x}$. The nearest point to $\mathbf{x}$ which lies on *e* is given by

$$\mathbf{x}_{np} = \mathbf{v}_1 + \lambda(\mathbf{v}_2 - \mathbf{v}_1) \tag{20}$$

with

$$\lambda = \frac{-(\mathbf{v}_1 - \mathbf{x})\cdot(\mathbf{v}_2 - \mathbf{v}_1)}{|\mathbf{v}_2 - \mathbf{v}_1|^2} \tag{21}$$

Now consider the cross product of $(\mathbf{x}-\mathbf{x}_{np})$ and the face normals; $(\mathbf{x}-\mathbf{x}_{np})\times\mathbf{n}_1$ and $(\mathbf{x}-\mathbf{x}_{np})\times\mathbf{n}_2$. Both of these vectors must be parallel or antiparallel to the edge vector $(\mathbf{v}_1-\mathbf{v}_2)$. The two face normals in combination with the edge vector each define two planes. If we are between the two planes then only one of $(\mathbf{x}-\mathbf{x}_{np})\times\mathbf{n}_1$ and $(\mathbf{x}-\mathbf{x}_{np})\times\mathbf{n}_2$ will lie parallel to the edge vector, and the other will be antiparallel. This means that $((\mathbf{x}-\mathbf{x}_{np})\times\mathbf{n}_1)\cdot(\mathbf{v}_2-\mathbf{v}_1)$ and $((\mathbf{x}-\mathbf{x}_{np})\times\mathbf{n}_2)\cdot(\mathbf{v}_2-\mathbf{v}_1)$ must have opposite signs. If they are either both positive or both negative we do not lie in the overlap between the two facets.

If the edge is convex (from the fluids perspective) then reflecting over both walls will lead to an overestimated density (case i) therefore we only reflect if the original particle is the correct side of the plane formed by the vectors $(\mathbf{n}_1+\mathbf{n}_2)$ and $(\mathbf{v}_2-\mathbf{v}_1)$, i.e. the bisecting plane of the edge.

If the edge is convex and we lie in the region of interest then the particle is reflected according to $\mathbf{x}' = 2\mathbf{x}_{np}-\mathbf{x}$.

Vertices

Treating concave vertices in 3D is more complicated than edges since we can have an arbitrary number of faces meeting at any vertex and in general there will be a mixture of concave and convex edges.

Each pair of adjoining faces that meet at a vertex define a plane in which both normals lie. If all the edges were concave then we could simply check that we were on the correct side of this plane for all pairs of faces around a vertex, i.e. that for all pairs $((\mathbf{x}-\mathbf{x}_v)\times\mathbf{n}_1)\cdot\mathbf{n}_2$ has the same sign, where $\mathbf{x}_v$ is the vertex location.

To account for the mixture of concave and convex edges we find the plane to which our particle is nearest then check if it is on the correct side of it by looking at the sign of $((\mathbf{x}-\mathbf{x}_v)\times\mathbf{n}_1)\cdot\mathbf{n}_2$. Because of the non commutativity of the cross product we must always check the pairs in the same order, thus we loop around the vertex clockwise as seen from the non-fluid side of the surface.

RESULTS

As mentioned previously all simulations here are carried out with a liquid-solid contact angle of 30º.

Spherical Packing

The first case we consider is a regular periodic packing of spherical beads. This idealised situation is use in lab based studies of leaching hydrodynamics, see (Bouffard, 2001) or (Ilankoon, 2012).

In Figure 5 we show results from an unsaturated simulation of water flowing through the regular packing at intervals of 0.1s. In this simulation the spheres have a radius of 10mm, the liquid saturation is 8.5% of the void space and we use periodic boundary conditions in all 3 dimensions.

Note that both the water and air phases are simulated but only the water is shown. The total number of SPH particles in the domain was ~3.9 million. The simulation took 14 days and was carried out using MPI on 192 (2.66GHz) cores.

The surface is reconstructed from the particle representation by calculating the function

$$\varphi(\mathbf{x}) = \sum_{i=1}^{N} W(|\mathbf{x}_i - \mathbf{x}|)\, C_i^l$$

on a regular grid then applying the marching cubes algorithm to extract the iso-surface representing the liquid-gas interface.

A quantity of interest for systems such these is the dimensionless Bond number; $B = \rho g L^2/\gamma$. The Bond number is the ratio of inertial force (due to gravity) and capillary force; it indicates whether the system's behaviour is dominated by inertia (B>>1) or capillary forces (B<1).

The simulation is in the transition region between inertia dominated and capillary dominated regimes; B=78. The system shows behaviours characteristic of

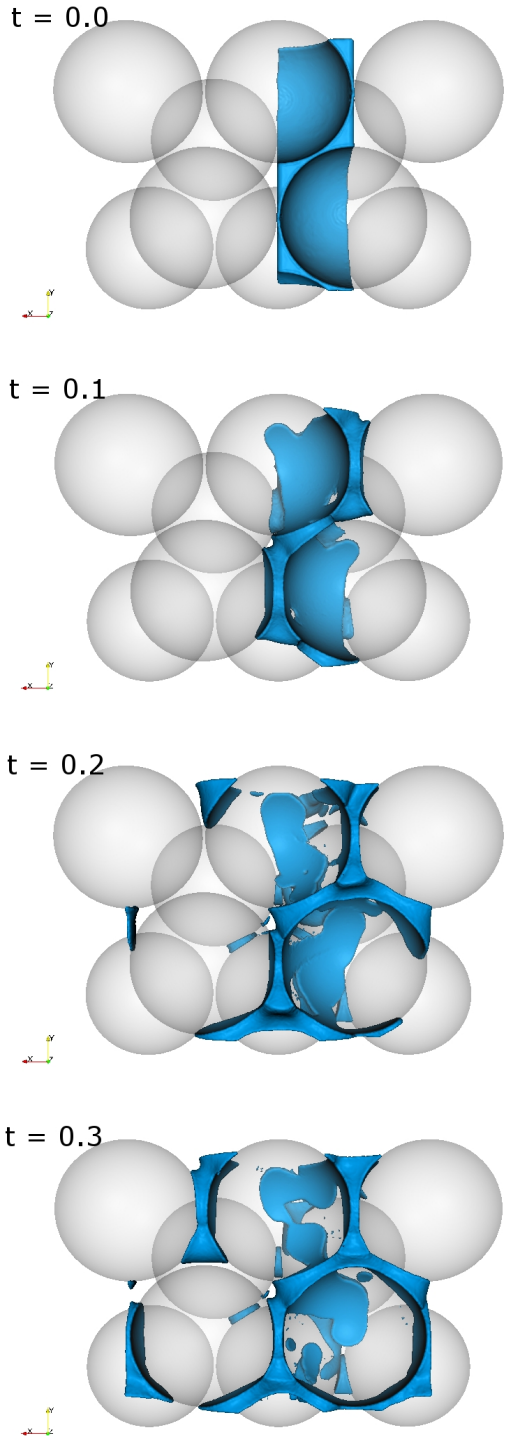

Figure 5: Time series images from simulation of water in a packed bed of spherical particles of diameter 20mm. The Bond number is 78.

intermediate Bond numbers. The water's surface is mostly held at the points where the spheres meet as can be seen in the final still (t=0.3s) but the water itself continues to flow downwards through the voids leading to a quasi stationary state.

Figure 1 (on the first page) shows a still of the final state from a simulation with particle radii of 1mm. Again both air and water are simulated, and the simulation was run on 192 cores. For this simulation the total number of SPH particles was 1.2 million.

When the particle radii are 1mm, the corresponding Bond number is 0.78 meaning the system is in the capillary dominated regime. The water is entirely held between the spheres reaching a stationary steady-state in which the shape is dictated by the minimum surface area; thereby minimising the surface energy. For comparison the mean velocity in the direction of gravity at t=0.3s for the B=78 simulation is -0.10ms^{-1} as opposed to just -0.01ms^{-1} at long times for the B=0.78 case, both simulations were started with the same initial fluid configuration (up to a scale factor).

The differing computational demands of the two simulations are due to the different scales at which they are run. The second simulation is a factor of 10 smaller meaning the capillary forces are far more dominant and the simulation quickly settlesl into its final state, because the shape of the resulting interface is stationary it was found that a lower resolution can still capture the details of the behaviour.

Rock Packing

The next simulation we show is a simulation of flow over a loose packing of rocks, this demonstrates a simulation which utilises faceted geometries. The rock geometry was obtained experimentally from X-ray CT then packed into a 10cm x 10cm x 10cm box using DEM. The simulation was run on 120 cores, had 1.6 million particles and took 5 days to run.

The fluid properties used for this simulation are not representative of industrial flows, the density ratio between the heaver and lighter phase was set to 4 and the viscosity was set to $50\mu_{water}$, for reasons of computational speed. The complex geometry along with the non-periodic nature of the packing leads to complex flow patterns which require impact the timestep of the simulaiton. Further, the timestep in interfacial SPH is limited in part by the density ratio of the fluid phases, a realistic density difference in this case was found to be prohibitive.

Figure 6 is a time sequence of stills from this simulation, the fluid forms a wetting layer over the rocks it contacts and other flow structures such as the Plateu-Rayleigh instability are observed. This would be expected and further validation against experimental results will of course be necessary before strong conclusions can be drawn from these simulations.

CONCLUSIONS

Following an introduction of SPH and the handling of surface tension using the CSF model we examined the handling of boundaries using reflected ghost particles. In particular we considered the novel aspect of reflecting across a curving edge or surface; showing an illustrative example. We have detailed how to suitably handle this in SPH to avoid artefacts near such boundaries through the introduction of a new *volume factor* which accounts for the curvature in both two and three dimensions.

The spherical packing simulations in both the gravity dominated and capillary dominated regimes indicate that this SPH formalism is likely capable of handling the sort of complex multi-phase flows we are interested in for HL; effects of surface tension, hold-up and superficial velocity were seen.

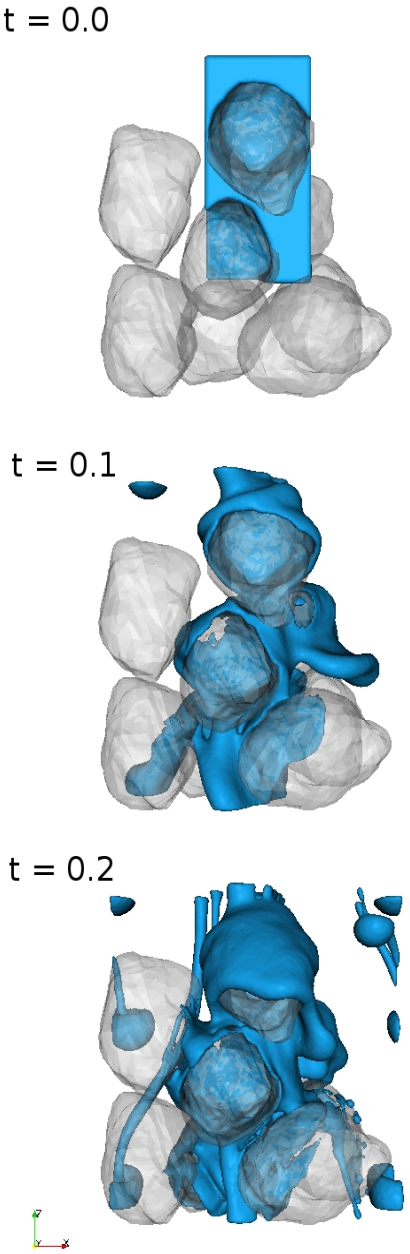

Figure 6: Sequence of stills from a simulation of flow over a rock packing.

The handling of complex geometries has received some attention in the past (Harada, 2007) but most SPH simulations are performed with very coarse or simplified features and make no mention of how corners are handled. We presented the geometric considerations necessary to avoid under- or overestimating the density near general edges and showed an example of a successful simulation through a packing with a complex geometry.

REFERENCES

Cariaga, E., Concha, F., and Sepulveda, M. (2005). Flow through porous media with applications to heap leaching of copper ores. *Chemical Engineering Journal*, 111(**2-3**), 151–165.

Leahy, M. J., Schwarz, M. P., and Davidson, M. R. (2003). An Air Sparging CFD Model for Heap Bioleaching of Copper-Sulphide. *Third International Conference on CFD in the Minerals and Process Industries* (pp. 581–586).

Mcbride, D., Cross, M., Croft, N., Bennett, C., and Gebhardt, J. (2006). Computational modelling of variably saturated flow in porous media with complex three-dimensional geometries. *International Journal for Numerical Methods in Fluids*, **50**, 1085–1117.

Dixon, D. (1995). Improved methods for the design of multistage leaching systems. *Hydrometallurgy*, 39(**1-3**), 337–351.

Bouffard, S. C. and Dixon, D. G. (2001). Investigative study into the hydrodynamics of heap leaching processes. *Metallurgical and Materials Transactions B*, 32(B), 763-776.

Wu, a, Yin, S., Qin, W., Liu, J., and Qiu, G. (2009). The effect of preferential flow on extraction and surface morphology of copper sulphides during heap leaching. *Hydrometallurgy*, 95(**1-2**), 76–81.

Lucy, L. B. (1977). A numerical approach to the testing of the fission hypothesis. *The Astronomical Journal*, 82(**12)**, 1013-1024.

Liu, M. B. and Liu, G. R. (2010). Smoothed Particle Hydrodynamics (SPH): an Overview and Recent Developments. *Archives of Computational Methods in Engineering*, 17(**1**), 25-76.

Hu, X. Y. and Adams, N. A. (2006). A multi-phase SPH method for macroscopic and mesoscopic flows. *Journal of Computational Physics*, 213(**2**), 844-861.

Harada, T., Koshizuka, S. and Kawaguchi, Y. (2007). Smoothed particle hydrodynamics in complex shapes. *Proc of spring conference on computer graphics* (pp. 26–28).

Müller, M., Schirm, S., & Teschner, M. (2004). Interactive blood simulation for virtual surgery based on smoothed particle hydrodynamics. *Technology and health care official journal of the European Society for Engineering and Medicine*, 12(**1**), 25-31. IOS Press.

Monaghan, J. J. and Kajtar, J. B. (2009). SPH particle boundary forces for arbitrary boundaries. *Computer Physics Communications*, 180(**10**), 1811-1820.

Ovaysi, S. and Piri, M. (2010). Direct pore-level modeling of incompressible fluid flow in porous media. *Journal of Computational Physics*, 229(**19**), 7456-7476.

Feldman, J. and Bonet, J. (2010). Dynamic refinement and boundary contact forces in SPH with applications in fluid flow problems. *International Journal for Numerical Methods in Engineering*, 72(B), 295-324.

Ferrand, M., Laurence, D. R., Rogers, B. D., Violeau, D., and Kassiotis, C. (2012). Unified semi-analytical wall boundary conditions for inviscid , laminar or turbulent flows in the meshless SPH method. *International Journal for Numerical Methods in Fluids*. doi:10.1002/fld

Ilankoon, I. M. S. K. and Neethling, S. J. (2012). Hysteresis in unsaturated flow in packed beds and heaps. *Min. Eng*. 35, 1-8

Becker M. and Teschner M. (2007) Weakly compressible SPH for free surface flows. *Eurographics* 1-8