Frontiers of Discontinuous Numerical Methods and Practical Simulations in Engineering and Disaster Prevention – Chen, Ohnishi, Zheng & Sasaki (Eds)
© 2013 Taylor & Francis Group, London, ISBN: 978-1-138-00110-7

Next generation discontinuous rock mass models: 3-D and rock-fluid interaction

R.G. Mikola
Jacobs Associates, San Francisco, CA, USA
Department of Civil and Environmental Engineering, UC Berkeley, San Francisco, CA, USA

N. Sitar
Edward G. Cahill and John R. Cahill Professor, Department of Civil and Environmental Engineering, UC Berkeley, Berkeley, CA, USA

ABSTRACT: We present a three dimensional fluid-structure coupling between SPH and 3D-DDA for modelling rock-fluid interactions. The Navier-Stokes equation is simulated using the SPH method and the motions of the blocks are tracked by a Lagrangian algorithm based on a newly developed, explicit, 3D-DDA formulation. The coupled model is employed to investigate the water entry of a sliding block and the resulting wave(s). The coupled SPH-DDA algorithm provides a promising computational tool to for modelling a variety of solid-fluid interaction problems in many potential applications in hydraulics, rock mass stability, and in coastal and offshore engineering.

1 GENERAL INSTRUCTIONS

In geotechnical engineering and rock mechanics fluid-solid interaction is often thought of and presented in terms of the influence of seepage through the rock mass and has been historically treated without recourse to discrete body mechanics. However, water flowing over the rock mass or rock impacting on a body of water represent completely different types of problems, which not only require the consideration of discrete body mechanics, but also the consideration of 3-D geometry in order to provide a realistic representation of the phenomena. For example, rock slides and rock falls into confined bodies of water, such as lakes, fjords and dams, have been known to produce large amplitude waves with disastrous consequences such as the 1934 rock slide into Tafjord in Western Norway (Sælevik et al. 2009). There, $1.5 \cdot 10^6$ m^3 of rock plunged into the fjord and produced water run-up heights up to 60 m and resulted in the death of 41 people. Other examples are the Lituya Bay, Alaska, where an earthquake caused a sub-aerial rock slide into Gilbert Inlet on July 8, 1958, yielding a maximum run-up of 524 m, (see Fritz et al., 2001) and the Vaiont reservoir disaster, 1963, where the waves over-topped the dam and claimed 2500 casualties, (see e.g. Semenza and Ghirotti, 2000).

Local mass gravity flows and slumps are believed to be regularly triggered by earthquakes. In some cases, such as for the 1998 Papau NewGuinea event, (Bardet et al., 2003; Lynett et al., 2003), the landslide generated waves are the main hazard. Historic examples of larger slides producing tsunamis include the Shimabara event in, Japan in 1792 and the slide at the Ritter Island Volcano into the sea northeast of New Guinea in 1888, which is the largest lateral collapse of an island volcano to be recorded in historical time, (Ward and Day, 2003). A major future collapse of the Cumbre Vieja volcano at La Palma on the Canary Islands has been suggested, (Ward and Day, 2001).

The complexity of the water-rock mass interaction has been studies using both experimental and analytical methods. Fritz (2002) and Fritz et al. (2003a, b, 2004) performed experiments to study waves created by a deformable landslide in a 2D wave tank. Zweifel et al. (2006) also used experiments to study the non-linearity of impulse waves. Huber and Hager (1997) looked at both 3D and 2D impulse waves. Raichlen and Synolakis (2003) performed experiments with a freely sliding wedge representing a land slide. Liu et al. (2005) used the same type of experiments to validate a numerical model, based on the large-eddy-simulation approach. Recently, Sælevik et al. (2009) performed two-dimensional experiments of wave generation from the possible Åkneset rock slide using solid block modules in a transect with a geometric scaling factor of 1:500.

The numerical simulation approaches used a number of different methods. For example, Harbitz (1992) simulated tsunamis generated by Storegga slides using linear shallow water equations. Jiang and Leblond (1992, 1994), Fine et al. (1998), Thomson et al. (2001), Imamura et al. (2001), Titov and Gonzalez (2001) used nonlinear shallow water approximation to model the slide-water system as a two-layer flow. Lynett and Liu (2002) discussed the limitations of the depth-integrated models with regards to landslide-generated waves, and developed fully nonlinear weakly dispersive model for submarine slides that is capable of simulating waves from relatively deep water to shallow water. Grilli and Watts (2005) derived and validated a two-dimensional fully nonlinear dispersive model that does not have any restrictions on the wave amplitude, wavelength, or landslide depth, and describes the motion of the landslide by the position of its center of mass.

The limitation of these approaches has been the assumption that the slide mass, soil or rock, could be approximated as an equivalent fluid mass or a continuous solid. While this approximation may be adequate and valid in many instances, it is desirable to be able to model the complexity of individual rock blocks interacting with water independently, thus allowing a greater flexibility in the type of phenomena that is modelled.

In this paper we present Discontinuous Deformation Analysis (DDA) coupled with Smoothed Particles Hydrodynamics (SPH) numerical model for the study of rock-fluid interaction in 3-D. Since its introduction by Shi (1993), 2-D DDA has been extensively developed in theory and computer codes, and there has been a significant interest in extending the formulation to 3-D. Shi (2001 a, b) presented the 3-D block matrices such as mass matrix, stiffness matrix, point load matrix, body load matrix, initial stress matrix and fixed point matrix. Herein we present new explicit time integration procedure for the solution of 3D-DDA algorithm in order to reduce the computational effort and memory requirement. A uniform spatial discretization method is utilized to eliminate unnecessary contact computations. The contact resolution is handled by FCP approach (Fast Common Plane, Nezami et al. 2004) and HalfEdge data structure is used to handle the frequent navigation into the topological information associated with polyhedral blocks.

Smoothed Particle Hydrodynamics (SPH), a meshless Lagrangian method, is a method that can capture the complexity of free surface flow with fragmentation and splashes. The SPH technique was conceived by Lucy (1977) and further developed by Gingold and Monaghan (1977) for treating astrophysical problems. Its main advantage is the absence of a computational grid or mesh, since it is a Lagrangian particle based method. This allows the possibility of easily modelling flows with a complex geometry or flows where large deformations or the appearance of a free surface occur.

The interaction model uses SPH to model the fluid and the rigid body solids are modelled using 3-D DDA. However, the general interaction model we propose works with any type of solid model representation as long as the object is represented by a polygonal surface and the fluid by Lagrangian particles.

2 CONTACT DETECTION USING UNIFORM GRID

Contact detection and resolution is the most time-consuming part of DEM/DDA analyses and generally takes about 80% of the overall computational time for particles (Horner et al. 2000). Contact resolution and detection is commonly performed in two consecutive phases (Perkins and Williams 2001), namely, neighbor search and contact detection (Figure 1). Neighbor search phase develops a neighbor list of all potential interacting particles within a neighborhood of the target particle. To speed up the contact detection process a problem region is divided into a number of cells and blocks are “mapped” into cells according to the locations of their vertices. The contact detection is carried out only between potential block vertices contained in each cell (Figure 1). Re-mapping of cells is triggered whenever one block moves outside its original cell space. For example in Figure 1, block A’s box list includes boxes 7, 8, 12, and 13; and box B’s block list includes particles 14, 15, 19, 20, 24 and 25 and block C’s box list includes boxes 4, 5, 9, 10, 14 and 15. These lists are obtained by defining a cuboid bounding volume around each particle and comparing it against the boxes. In this paper the faces of the cuboid bounding volume are

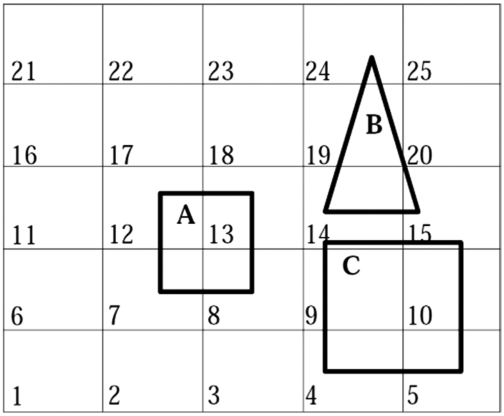

Figure 1. Cell mapping of block (1, 2, 3, …-Cell Number and A, B, C-Blocks).

confined to be parallel to coordinate planes of the global coordinate system.

The performance of the neighbor search algorithm is dependent on the particle shape and the ratio of the box size S to the average bounding box size D_{50} (Nezami et al. 2004). Nezami et al. (2004) suggest that contact search optimal performance correspond to the approximately $S/D_{50} = 1.5$ (Nezami et al. 2004).

3 CONTACT RESOLUTION USING FCP ALGORITHM

After a search for the colliding objects is made for all discrete blocks, the next step is to find the points of intersection on the sides of the home element with the sides of the near element. Common-plane (CP) algorithm was introduced by Cundall (1998) in order to reduce the expensive object-to-object contact detection problem to a less expensive plane-to-object contact problem in DEM. The CP is defined as a plane bisecting the space between the objects. After CP has been located each object is tested separately for contacts with the common-plane. Nezami et al. (2004) proposed the fast common plane (FCP) method in which they improved the original CP algorithm by adding a fast method to identify the right candidates for the plane. In FCP, the number of iterations is significantly reduced by limiting the search space of the CP to a few candidates. In our approach, the FCP method is applied to the contact detection and the common-plane is selected as the reference plane for vertex-to-vertex, vertex-to-edge and edge-to-edge contacts in the 3-D DDA.

The FCP algorithm to find the CP consists of following five steps (Nezami et al. 2004):

i. If there is a CP from the previous time step then use it as the initial guess for the CP in this time step. Otherwise, set the CP as the perpendicular bisector (PB) plane of the line connecting the centroids of the two blocks.

ii. Based on the guess CP, find the closest vertices A and B in two contacting blocks. If more than one pair of closest vertices have the shortest length (i.e., those vertices are equidistant), then any of them can be chosen to proceed with the algorithm.

iii. For the two closest vertices A and B found in Step 2, check all candidate planes of and find the one with the largest gap or smallest overlap. The candidate planes could be on the plane listed below:

   a. Type a: The PB plane of segment AB.
   b. Type b: The plane passing through the midpoint of segment AB parallel to one of the faces of particles A or B. For particle A, only faces which include the vertex A are considered. For particle B, only faces which include the vertex B are considered.
   c. Type c: The plane passing through the midpoint of segment AB parallel to one edge from particle A and one edge from particle B. For particle A, only those edges which share the vertex A are considered. For particle B, only those edges which share the vertex B are considered.
   d. Type d: The plane passing through the midpoint of segment AB parallel to one edge from one of the particles.

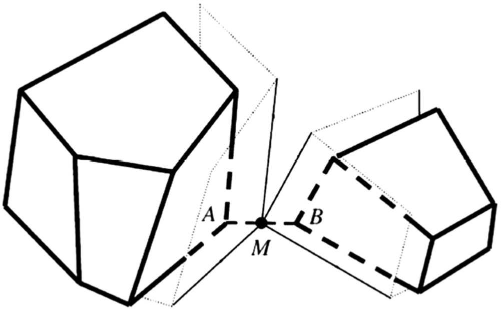

Figure 2. Possible CP for colliding blocks.

iv. If the CP obtained in Step 3 is the same as the one from Step (ii), then it is the correct common plane. Otherwise go to Step (ii). This is an iterative algorithm, with consisting of steps (ii)–(iv) in each iteration. The number of iterations required to find the CP is usually very small and the position of the selected CP is accurate. This is mainly because the iteration is done to locate the two closest vertices, rather than the CP itself.

4 COMPACT DATA STRUCTURE (HALFEDGE)

FCP algorithm needs access to topological information such as faces and edges sharing the same vertices A and B, as just explained. The simplest way of storing the information is a data structure that explicitly stores all topological entities; and all the adjacency relationships among them. This allows very efficient query process, but demands high, sometimes prohibitive, storage space. Moreover, editing tasks may demand high computational efforts because several adjacency relationships have to be updated. In this paper, authors use the compact adjacency-based topological data structure for representing the polyhedral block so called HalfEdge (HE). The HE data structure allows adjacency information to be found in near real time, making it especially useful for FCP algorithm where frequent access is needed to different types of adjacency information and also reduces redundancy.

The main idea of the half-edge is to "split" each edge along its length and store pointers to its previous and next edges, see Figure 3. The full implementation in C++ can be seen in Algorithm 1 (Gustav, 2010).

A face is not defined explicitly instead it can be found by using the next or previous pointers. With this data structure, finding neighboring faces of a vertex is O(1) and for a complete mesh O(n) and the topological queries of the polygon mesh can be performed easily and quickly. The time complexity of these queries is linear in the amount of information gathered and independent of global complexity. For example, iterating over the faces adjacent to a vertex requires the following sequence of steps (Gustav, 2010):

i. Find an edge connected to the given vertex.
ii. Step out on the edge loop and insert the face connected to this edge.
iii. Use edge- > next- > pair- > next to traverse to next face and insert this face into the temporary vector.
iv. Repeat step (iii) until encountering the face inserted in step (i).

Algorithm 1: Half-edge data structure.

```
class Halfedge { // topology
public:
Vertex* vert;
Halfedge* next;
Halfedge* prev;
Halfedge* pair;
Face* left;
};
class Vertex { //geometry
public:
float x, y, z;
Halfedge* edge;
};
class Face {
pulic:
Halfedge* edge;
};
class Polyhedron {
public:
Vertex verts [V];
Face faces [F];
Halfedge edges [3F];
};
```

5 EXPLICIT TIME INTEGRATION SCHEME

Let D_n and D_{n+1} denote the approximation to the values $D(t)$ and $D(t + 1)$ for a time step Δt, respectively. Recall the system of equations Eq. 1 of motion for a DDA system (Shi, 1993):

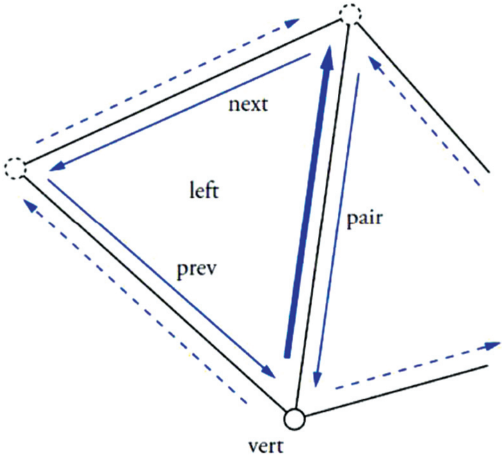

Figure 3. The half-edge data structure as seen from the bold half-edge.

$$M\ddot{D}_{n+1} + C\dot{D}_{n+1} + KD_{n+1} = F_{n+1} \qquad (1)$$

with $D(0)=0, \dot{D}(0)=\dot{D}_0$ as initial boundary conditions. In the above M, C, K are the global mass, damping and stiffness matrices, F is the time dependent applied force vector, and $\ddot{D}, \dot{D}$, D and denote acceleration, velocity and displacement vectors, respectively.

Original DDA time integration scheme adopts the Newmark (1959) approach, which for a single degree of freedom can be written in the following manner:

$$u_{i+1}=u_i+\Delta t\dot{u}_i+\left(\frac{1}{2}-\beta\right)\Delta t^2\ddot{u}_{i+1}+\beta\Delta t^2\ddot{u}_{i+1} \qquad (2)$$

$$\dot{u}_{i+1} = \dot{u} + (1-\gamma)\Delta t\ddot{u}_{i+1} + \gamma\Delta t\ddot{u}_{i+1} \qquad (3)$$

where $\ddot{u}, \dot{u}$, and u are acceleration, velocity, and displacement respectively, Δt is the time step, β and γ are the collocation parameters defining the variation of acceleration over the time step. Unconditional stability of the scheme is assured for $2\beta \geq \gamma \geq 0.5$. DDA integration scheme uses $\beta = 0.5$ and $\gamma = 1$, thus setting the acceleration at the end of the time step to be constant over the time step. This approach is implicit and unconditionally stable. Substituting Eqs. 2 and 3 into Eq. 1 results in the system of equations for solving the dynamic problem:

$$\left(\frac{2}{\Delta t^2}M+\frac{2}{\Delta t}C+K\right)D_{n+1}=F_{n+1}+\left(\frac{2}{\Delta t}M+C\right)\dot{D}_n \qquad (4)$$

The solution of Eq. 4 requires assembling the global mass and stiffness matrices and solving the coupled system of equations using a direct matrix inverse operation or an iterative solver. The global stiffness matrix, *K*, includes the sub-matrix representing deformability of blocks and contacts, with contact matrices as off-diagonal terms.

In the original DDA code (Shi, 1993) the global equations are solved iteratively by repeatedly adding and removing contact springs (penalty values) until each of the contacts converges to a constant state at each time step. This procedure of adding and removing contact springs (penalty values) is known as open-close iterations in the DDA literature (Doolin and Sitar, 2004). If contact convergence is not achieved typically within six iterations, the time step is reduced and the analysis is repeated with the reduced time step. The incremental displacement is restricted also by user-specified displacement limit to enforce infinitesimal displacements. If the incremental displacement is greater than the threshold, Δt is divided by three and the analysis is repeated. Large values of Δt may cause large penetrations at contact points; which results in more iterations to satisfy the penetration threshold. Also, large penetrations result in large contact matrices which can reduce the diagonal dominance of the global stiffness matrix leading to poorly conditioned system of equations.

In the explicit solution procedure presented herein the discrete blocks are integrated explicitly by the central difference method, which gives

$$u_{i+1} = u_i + \Delta t_{i+1}\dot{u}_{i+1} \tag{5}$$

$$\dot{u}_{i+1/2} = \dot{u}_{i-1/2} + \frac{1}{2}(\Delta t_{i+1} + \Delta t_i)\ddot{u}_i \tag{6}$$

where *i*, *i* + 1/2 and *i*, *i* – 1/2 refer to the increment number and mid-increment numbers

$$\ddot{u}_i = M^{-1}(F_i - I_i) \tag{7}$$

where *M* is mass matrix, *F* the applied load vector and *I* is the internal force vector. The equations relating these values to each other are solved locally for each time-step. Moreover, since there is no need to solve a complete system of equations, the incremental calculations for each degree of freedom are done independently at the local level. This uncoupling of the equations of motion is one of the major advantages of explicit integration schemes. In contrast to the implicit time integration scheme, the explicit solution scheme eliminates the need for assembly of global mass or stiffness matrices and inversion of the global matrix. However, computations are conditionally stable, i.e., the time-step size must be smaller than a certain critical value (critical time step, Δt_c) for numerical errors not to grow unbounded. The time increments must satisfy the well-known criterion

$$\Delta t \leq \frac{2}{\omega \mathrm{max}} \tag{8}$$

where ω_{max} is the element maximum eigenvalue.

6 NUMERICAL MODELING OF WATER FLOW

6.1 *Navier-Stokes equations*

The dynamic behaviour of a viscous fluid, like water, is completely described by the so-called Navier-Stokes equations (NSEs). The equations for incompressible fluids are the mass conservation equation and the momentum conservation equation. Many forms of the NSEs appear in the literature. Equations (9) and (10) represent a simplified version for incompressible fluids.

$$\nabla \cdot u = 0 \tag{9}$$

$$\frac{\delta u}{\delta t} + (u.\nabla)u = \frac{1}{\rho}\nabla p + v\nabla^2 u + f \tag{10}$$

where ρ, u, P, v, g are density, velocity, pressure, dynamic viscosity coefficient of the fluid and gravitational acceleration, respectively. The first equation is the incompressibility condition. The second equation is called momentum equation which describes how fluid moves due to the forces.

6.2 *Smoothed Particle Hydrodynamics (SPH) equations*

The SPH is an interpolation method for fluid motion simulation. SPH uses field quantities defined only at discrete particle locations and can be evaluated anywhere in space. SPH distributes quantities in a local neighborhood of the discrete locations using radial symmetrical smoothing kernels. A scalar value A is interpolated at location *r* by a weighted sum of contributions from the particles: In SPH, a physical value at position *x* is calculated as a weighted sum of physical values φ_j of neighbouring particles *j*

$$A_s(X) = \Sigma_j m_j \frac{\varphi_j}{\rho_j} W(X - X_j) \tag{11}$$

where m_j, ρ_j, X_j are the mass, density and position of particle *j*, respectively and *W* is a weight function.

The use of particles instead of a stationary grid simplifies these two equations substantially. First, because the number of particles is constant and each particle has a constant mass, mass conservation is guaranteed and (9) can be omitted completely. Second, the expression $\frac{\delta u}{\delta t} + (u \cdot \nabla)u$ on the left hand side of (10) can be replaced by the substantial derivative $\frac{\delta u}{\delta t}$ Since the particles move with the fluid, the substantial derivative of the velocity field is simply the time derivative of the velocity of the particles meaning that the convective term u• $\nabla.u$ is not needed for particle systems. We regard NSEs as the governing equations, and calculate density, pressure and viscosity force separately using SPH numerical methods. The density of fluid is calculated with Eq. 12 as

$$\rho_j = \Sigma_j m_j W(r_i - r_j, h) \tag{12}$$

Accuracy of the algorithm highly depends on the smoothing kernels. For our implementation we used the following kernel:

$$W(r,h) = \frac{315}{64\pi h^9}\begin{cases}(h^2 - r^2)^3 & (0 \le r \le h)\\ 0 & (r > h)\end{cases} \tag{13}$$

The weight functions used by Muller et al. are also used in this literature (Muller et al., 2003). Instead of an equation described by the SPH rule a modified solution is used for pressure force because it guarantees the symmetry of forces:

$$f_i^{pressure} = -\Sigma_j m_j \frac{p_i - p_j}{2\rho_j} \nabla W_{spiky}(r_i - r_j, h) \tag{14}$$

For pressure computations we use Debrun's spiky kernel (1996):

$$\nabla W_{spiky}(r,h) = \frac{45}{\pi h^6}\begin{cases}\left(\frac{h^2+r^2}{r} - 2h\right)r & (0 \le r \le h, r = |r|)\\ 0 & (r > h)\end{cases} \tag{15}$$

The pressure at particle locations has to be calculated first, which can be computed via the ideal gas equation:

$$p = kr \tag{16}$$

where k is a gas constant that depends on the temperature. A modified version—which we used in our implementation—makes the simulation numerically more stable:

$$p = k(\rho - \rho_0) \tag{17}$$

where ρ_0 is the at-rest density. Applying the SPH rule to the viscosity term also yields to asymmetric forces because the velocity field varies. The idea of symmetrizing the expression is using velocity differences:

$$f_i^{viscosity} = \mu \Sigma_j m_j \frac{v_i - v_j}{\rho_j} \nabla^2 W_{viscosity}\left(r_i - r_j, h\right) \tag{18}$$

In (Muller et al. 2003), Muller designed a kernel for the computation of viscosity forces:

$$\nabla^2 W_{vis\cos ity}(r,h) = \frac{45}{\pi h^6}\begin{cases}(h - r) & (0 \le r \le h, r = |r|)\\ 0 & (r > h)\end{cases} \tag{19}$$

Finally, for the acceleration a_i of a particle i we have

$$a_i = \frac{1}{\rho_i}\left(f_i^{pressure} + f_i^{vis\cos ity} + f_i^{external}\right) \tag{20}$$

where $f_i^{external}$ are external body forces such as gravity forces. We have used a simple Euler integrator in our simulations, which is first order accurate in position and velocity, and can be written as,

$$\begin{aligned} v_i\left(t + \Delta t\right) &= v_i(t) + \Delta t\, a_i(t)\\ x_i(t + \Delta t) &= x_i(t) + \Delta t v_i(t + \Delta t)\end{aligned} \tag{21}$$

where Δt is the time step.

7 COUPLING BETWEEN SPH AND DDA

The coupling algorithm used here is parallel; fluid (SPH) and solid block (DDA) evolutions are calculated explicitly at the same time. In order to couple the SPH and DDA the interaction force between fluid particles and solid blocks needs to be estimated. We choose to employ a fairly standardized "repulsion" force to prevent a particle from penetrating the boundaries. This method was chosen for the ease with which multiple types of boundaries can be implemented. The repulsion force is fairly easily implemented for both "wall boundaries" as well as "solid blocks". The no-penetration condition states that the fluid cannot penetrate the boundary surface. To repel the fluid particles from the boundary we use a penalty-force method:

$$f_i^{boundary} = (K_S d - (v \cdot n) K_D) \cdot n \tag{22}$$

where K_S is the penalty force stiffness and K_D is the damping coefficient for the velocity v of an approaching fluid particle d is the penetrated distance measured normal to the boundary, and n

is the unit-length surface normal. It can be seen from Equation (22) that the penalty force method behaves as a spring-based model, because the more a particle penetrates the boundary the more it is pushed away from the surface.

8 SIMULATIONS

Three examples are presented to demonstrate the newly developed 3-D DDA algorithm. The scenes in the following examples have been rendered with POV-ray, a free code ray tracing rendering program (POV-Ray, 2004).

8.1 *Example 1: Wave maker*

This simulation involves a wave maker in the form of an oscillating piston on the one end of the model, a straight line beach with a slope of 4% and a horizontal section 70 m long between the wave maker and the beach. The SPH simulation used almost 65000 particles and the boundaries as well as wavemaker itself have been simulated using as rigid blocks. Figure 4 shows the propagating waves onto the beach.

8.2 *Example 2: Sliding block*

In this example we simulate waves generated by a rigid wedge sliding into water along an inclined

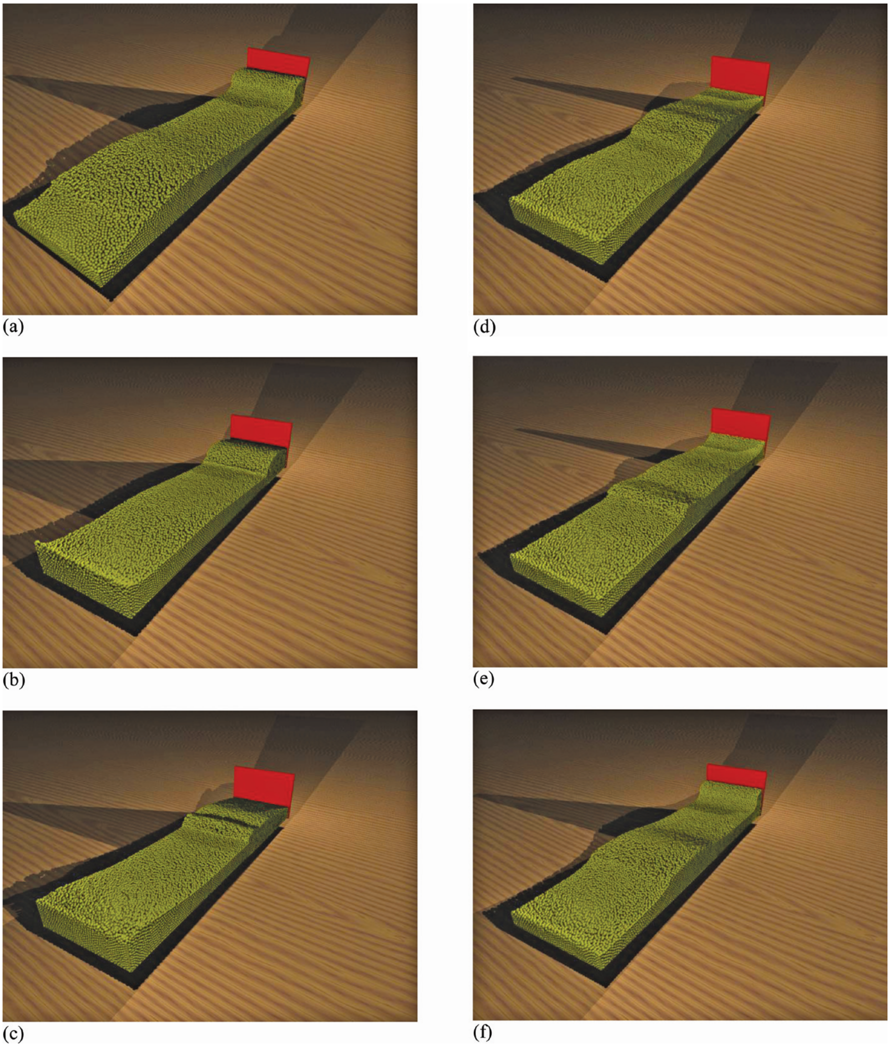

Figure 4. Particles and rigid block configuration for the wavemaker.

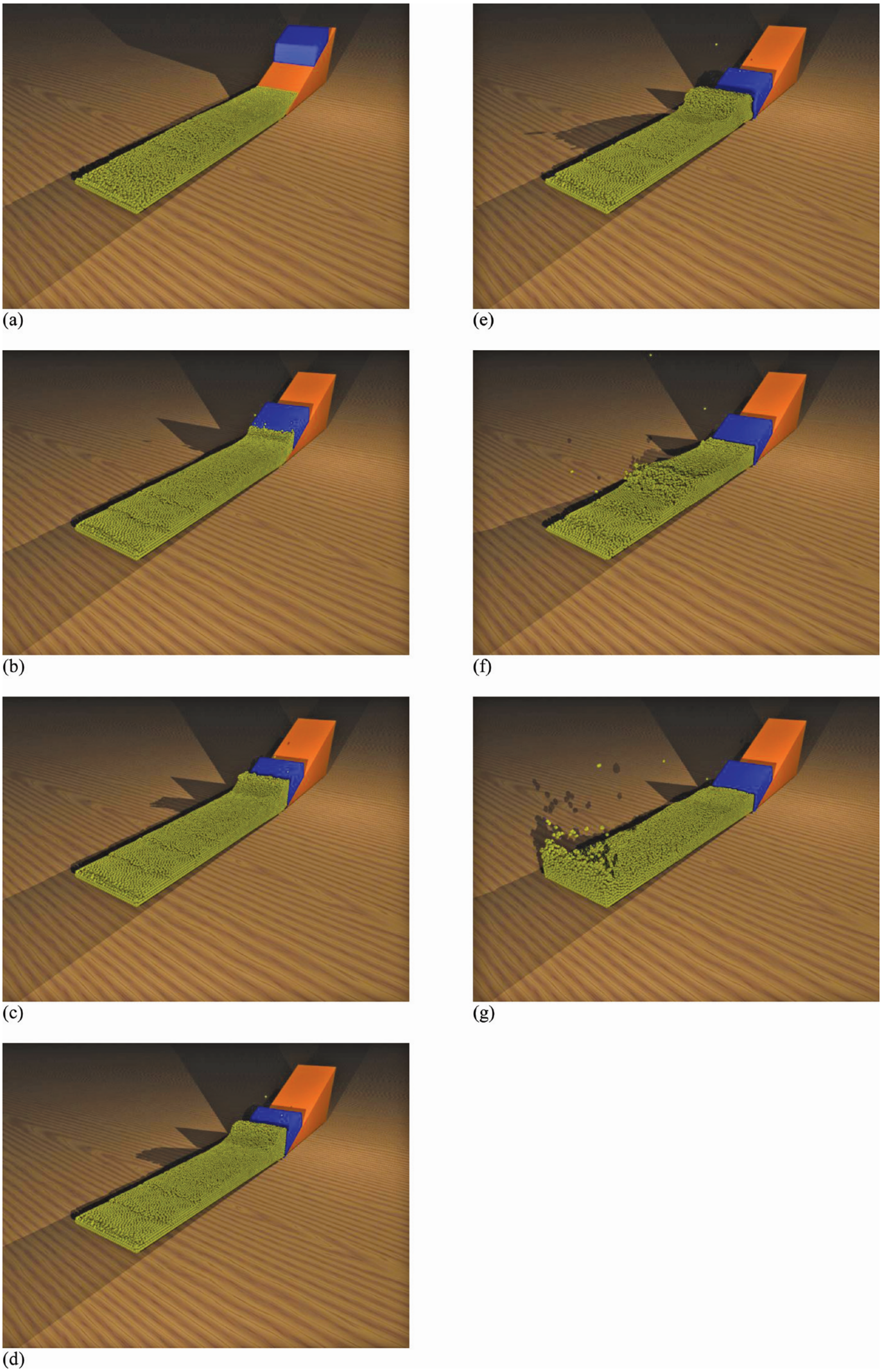

Figure 5. Particles and rigid blocks configuration for the rigid wedge sliding down a plane inclined 25°on the horizontal at different time steps.

plane. In this simulation water waves were generated by allowing a wedge shape block to freely slide down a plane inclined 25° on the horizontal. The density of the wedge assumed to be 2500 kg/m^3. The SPH simulation used almost 25000 particles and the boundaries as well as sliding block have been simulated as rigid blocks. Particles configuration due to sliding of the rigid wedge is presented at different times in Figure 5.

9 CONCLUSIONS AND FUTURE WORK

We introduce a three dimensional numerical model coupling the SPH method and 3D-DDA for modeling fluid-discrete solid body interaction problems. The coupling algorithm is very efficient when dealing with fluid-structure interaction problems in the presence of a free-surface and is relatively simple to implement. The ability of SPH to fragment and reconnect interfaces presents a great opportunity when modeling impacts of solids on fluids, and vice versa. The result of the example computations show that coupled SPH and DDA can be used to simulate dynamic fluid discrete block interactions in a variety of settings. Future developments need to concentrate on developing highly computationally efficient and optimized algorithms in order to tackle the full scope of complex problems of interest. We intend to further accelerate the simulation by using GPU based methods, using faster neighbor search algorithm as well as using predictive corrective incompressible method proposed by Solenthaler et al. (2009).

ACKNOWLEDGMENT

This research was supported in part by funding from the Edward G. Cahill and John R. Cahill Chair and Jacobs Associates generously provided the first author with the opportunity to pursue the research.

REFERENCES

Bardet, J.-P., Synolakis, C.E., Davies, H.L., Imamura, F. and Okal, E.A., 2003. Landslide tsunamis: recent findings and research directions. Pure and Applied Geophysics 160: 1793–1809.

Beyabanaki, S.A.R., Mikola, R.G. and Hatami, K. 2008. Three-dimensional discontinuous deformation analysis (3-DDDA) using a new contact resolution algorithm. Comp. Geotech. 35:346–56.

Cundall, P.A. 1988. Formulation of a three-dimensional distinct element model. Part I: a scheme to detect and represent contacts in a system composed of many polyhedral blocks. Int J. Rock Mech. Min. Sci. & Geomech. Abstr. 25(3): 107–16.

Desbrun, M. and Cani, M.P. 1996. Smoothed particles: A new paradigm for animating highly deformable bodies. InComputer Animation and Simulation '96 (Proceedings of EG Workshop on Animation and Simulation), pages 61–76. Springer-Verlag, Aug.

Doolin, D.M. and Sitar, N. 2004. Time integration in discontinuous deformation analysis DDA. ASCE Journal of Engineering Mechanics, 130:249–258.

Fritz, H.M., 2002. Initial phase of landslide generated impulse waves. PhD thesis, Swiss Federal Institute of Technology Zurich.

Fritz, H.M., Hager, W.H. and Minor, H.E., 2001. Lituya bay case: rockslide impact and wave run-up. Science of Tsunami Hazards 19(1): 1–67.

Fritz, H.M., Hager, W.H. and Minor, H.E., 2003a. Landslide generated impulse waves. 1. instantaneous flow fields. Experiments in Fluids 35: 505–519.

Fritz, H.M., Hager, W.H. and Minor, H.E., 2003b. Landslide generated impulse waves. 2. hydrodynamic impact craters. Experiments in Fluids 35: 520–532.

Fritz, H.M., Hager, W.H. and Minor, H.E., 2004. Near field characteristics of landslide generated impulse waves. Journal of Waterway, Port, Coastal and Ocean Engineering 287–302.

Fine, I., Rabinovich, A., Kulikov, E., Thomson, R. and Bornhold, B. 1998. Numerical modelling of landslide-generated tsunamis with application to the Skagway Harbor tsunami of November 3, 1994. In Proc. Int. Conf. on Tsunamis, Paris, pp. 211–223.

Grayeli, R. and Hatami, K. 2008. Implementation of the finite element method in the three-dimensional discontinuous deformation analysis (3D-DDA), International Journal for Numerical and Analytical Methods in Geomechanics, 32: 1883–1902.

Grilli, S. and Watts, P. 2005. Tsunami generation by submarine mass failure. I: Modeling, experimental validation, and sensitivity analysis, J. Waterw. Port. Coast. Ocean. Eng. 131(6), 283–297, DOI 10.1061/(ASCE)0733–950X(2005)131:6(283).

Gingold, R.A. and Monaghan, J.J. 1977. Smoothed particle hydrodynamics: theory and application to non-spherical stars Mon. Not. R. Astron. Soc. 181: 375–89.

Gustav, G, 2010. The Half-edge data structure, http://www.gustavgahm.com/wp-content/upload/moa/Gustav_Gahm_The_Half_edge_mesh_data_structure.pdf

Harbits, C. 1992. Model simulations of tsunamis generated by the Storegga Slides, Marine Geol. 105, 1–21.

Horner, D.A. and Carrillo, A., et al. 2000. Very large scale coupled discrete element-finite element modeling for simulating excavation mechanics. Fourteenth Engineering Mechanics Conference, American Society of Civil Engineerings, Austin, Texas.

Hoek, E. and Bray, J. 1974. Rock slope engineering. London: The Institution of Mining and Metallurgy; p. 309.

Jang, H.I. and Lee, C.I. 2002. Development of a three-dimensional discontinuous deformation analysis technique and its application to toppling failure. In: Hatzor Y.H., editor. Proceedings of the 5th international conference on analysis of discontinuous deformation (ICADD-5), Abingdon, p. 225–9.

Jiang, Q.H. and Yeung, M.R. 2004. A model of point-to-face contact for three dimensional discontinuous deformation analysis. Rock Mech. Rock Eng. 37:95–116.

Jiang, L. and Leblond, P. 1992. The coupling of a submarine slide and the surface waves which it generates, J. Geophys. Res. 97(C8), 12,731–12, 744.

Jiang, L. and Leblond, P. 1994. Three-dimensional modeling of tsunami generation due to a submarine mudslide, J. Phys. Oceanogr. 24(3), 559–572.

Khan, M.S., 2010. Investigation of discontinuous deformation analysis for application in jointed rock masses, Ph.D. Dissertation, University of Toronto.

Liu, J. and Kong, X. 2003. Development of three-dimensional discontinuous deformation analyses. In: Lu M, editor. Proceedings of the 6th international conference on analysis of discontinuous deformation (ICADD-6), Norway, p. 45–54.

Liu, J., Kong, X. and Lin, G. 2004. Formulation of the three-dimensional discontinuous deformation analysis method. Acta. Mech. Sinica. 20(3): 270–82. June.

Lucy, L.B. 1977. A numerical approach to the testing of the fission hypothesis Astron. J. 82: 1013–24.

Lynett, P. and Liu, P.F. 2002. A numerical study of submarine landslide generated waves and runup, Proc. Roy. Soc. London A 458: 2885–2910.

Lynett, P.J., Borrero, J.C., Liu, P.L.-F., Synolakis, C.E., 2003. Field survey and numerical simulations: a review of the 1998 Papua New Guinea tsunami. Pure and Applied Geophysics 160: 2119–2146.

Muller, M., Charypar, D., and Gross, M. 2003. Particle based fluid simulation for interactive applications. In SCA '03:Proceedings of the 2003 ACM SIGGRAPH/ Eurographics symposium on Computer animation, Eurographics Association, Airela-Ville, Switzerland, Switzerland, 154–159.

Nezami, E., Hashash, Y.M.A., Zhao, D. and Ghaboussi J, 2004. A fast contact detection algorithm for 3-D discrete element method. Comput. Geotech. 31(7): 575–87.

Newmark, NM. 1959. A Method of Computation for Structural Dynamics", ASCE Journal of the Engineering Mechanics Division, Vol. 85 No. EM3.

Perkins, E. and Williams, J.R. 2001. A fast contact detection algorithm insensitive to object sizes. Engineering Computations 18(1–2): 48–62.

Persistence of Vision Pty. Ltd. 2004. Persistence of Vision Raytracer (Version 3.6), http://www.povray.org/download/

Raichlen, F. and Synolakis, C., 2003. Runup from three dimensional sliding mass. Long Waves Symposium, Briggs, M., Koutitas. Ch. (Eds).

Semenza, E. and Ghirotti, M., 2000. History of the 1963 Vaiont slide: the importance of geological factors. Bulletin of EngineeringGeology and the Environment 59(2): 87–97.

Shi, G.H. 1993. Block system modeling by discontinuous deformation analysis. Southampton UK and Boston USA: Computational Mechanics Publications.

Shi, G.H. 2001a. Three-dimensional discontinuous deformation analysis. In: Bicenic N, editor. Proceedings of the forth international conference on analysis of discontinuous deformation (ICADD-4), Glasgow, Scotland, June 6–8, p. 1–21.

Shi, G.H. 2001b. Three-dimensional discontinuous deformation analysis. In: Ellsworth et al., editors. Proceedings of the 38th US rock mechanics symposium, Washington DC, July 7–10, p. 1421–8.

Solenthaler, B. and Pajarola, R. 2009. Predictive-corrective incompressible sph. In SIGGRAPH '09: ACM SIGGRAPH 2009 papers, pages 1–6, New York, NY, USA, ACM.

Saelevik, G. Jensen, A. and Pedersen, G. Experimental investigation of impact generated tsunami; related to a potential rock slide, Western Norway, Volume 56, Issue 9, September 2009, Pages 897–906.

Thomson, R.E., Rabinovich, A.B., Kulikov, E.A. and Fine, 2001. I.V., and Bornhold, B.D., On numerical simulation of the landslide-generated tsunami of November 3, 1994 in Skagway Harbor, Alaska. In: Tsunami Research at the End of a Critical Decade (Hebenstreit, G.T., ed.), (Kluwer 2001), pp. 243–282.

Titov, V. and Gonzalez, F. 2001. Numerical study of the source of the July 17, 1998 PNG tsunami. In: Tsunami Research at the End of a Critical Decade, (Hebenstreit, G.T., ed.), (Kluwer 2001), pp. 197–207.

Ward, S.N. and Day, S. 2003. Ritter Island volcano—lateral collapse and the tsunami of 1888. Geophysical Journal International 154(3): 891–902.

Wu, J.H., Ohnishi, Y., Shi, G.H. and Nishiyama, S. 2005. Theory of three-dimensional discontinuous deformation analysis and its application to a slope toppling at Amatoribashi, Japan. International Journal of Geomechanics 5(3): 179–195.

Wu, J.H., Ohnishi, Y., Shi, G.H. and Nishiyama S. 2005. Theory of three dimensional discontinuous deformation analysis and its application to a slope toppling at Amatoribashi, Japan. Int J Geomech179–95.

Wu, J.H., Juang, C.H. and Lin, H.M. 2005. Vertex-to-face contact searching algorithm for three-dimensional frictionless contact problems. Int. J Numer. Methods. Eng. 63(6): 876–97.

Yeung, M.R., Jiang, Q.H. and Sun, N. 2007. A model of edge-to-edge contact for three-dimensional discontinuous deformation analysis. Comput. Geotech. 34(3): 175–86.

Yeung, M.R., Jiang, Q.H. and Sun, N. 2003. Validation of block theory and three dimensional discontinuous deformation analysis as wedge stability analysis methods. Int. J. Rock Mech. Min. Sci. 40(2): 265–75.

Zweifel, A., Hager, W.H. and Minor, H.E., 2006. Plane impulse waves in reservoirs. Journal of Waterway, Port, Coastal and Ocean Engineering 358–368.