Journal of the Applied Mathematics, Statistics and Informatics (JAMSI), **6** (2010), No. 1

HANDLING SEVERAL FLUIDS IN COMPUTER GRAPHICS SIMULATIONS

ROMAN ĎURIKOVIČ

Abstract

Fluid dynamics governed by Navier-Stokes equations is already solved for decades but the recent trend in computer graphics is to modify the simulation such that it easily controllable for the purpose of computer animation or real time fluid animation. In Eulerian approach, fields must be discretized in space, making these techniques inherently mesh-based. Each mesh cell represents an approximation of the infinitesimal volume at a fixed position in space and during the simulation monitors the evaluation of state variables at that location.

Our approach is an extension of a well known Marker-and-Cell (MAC) fluid simulation method, within each cell we define density and pressure in center and fluid velocities on the walls separately in each 6 directions. To simulate several fluids, we propose the modification of Volume-of-Fluid (VOF) method enabling us to tract the fluid surface and integrate it into the multiphase-fluid approach. A mixture of fluids is treated as a single fluid having variable density and viscosity. This scheme allows two or more fluids having different densities and viscosities to be simulated simultaneously.

We simulate a one-way solid fluid interaction (either solid influences the velocity of the fluid or fluid moves the solid) that requires having fine details in the colliding areas. To cope with this problem we use an unrestricted octree data structure with adaptive mesh refinement technique to enable higher level of detail and solve Navier Stokes equations for multiple fluids. We propose the technique for discretizing the Poisson equation on octree grid. The resulting linear system is symmetric positive definite enabling the use of fast solution methods, while the standard approximation to the Poisson equation on an octree grid results in a non-symmetric linear system which is more computationally challenging. Finally, we show several fluid flow animations with static obstacles, floating objects colliding each other.

ACM Computing Classification System 1998: I.3.5, I.3.7
Additional Key Words and Phrases: volume of fluid, fluid dynamics, one-way coupling, octree, Navier-Stokes, floating objects

1. INTRODUCTION

Interaction between fluid and solid is very common in real world and in computer graphics animation. Common interactions between fluid and solid object can be various objects falling into water, objects floating on the surface or acting as unmoving obstacle for the flow of fluid.

First type of interaction is one-way solid-to-fluid coupling, where the motion of solid is predetermined and is not influenced by the velocity of fluid, but the solid influences the velocity of the fluid so it can splash the water as it falls into water or it can be an unmoving obstacle.

Second type of interaction is one-way fluid-to-solid coupling, where the fluid moves the solid without the solid affecting the fluid. This is the reason why the size of the solid can be from tiny to big object without affecting the motion of the fluid.

Most interesting in the way of simulation and visual effect is the two-way coupling of solids and fluid. It is the 'real world' way of interaction, where the properties of solid, like density, are taken in count. This type of coupling is mathematically a difficult problem.

2. PREVIOUS WORK

The three dimensional Navier-Stokes equations were intoduced to computer graphics comminity by Kass and Miller 1990 [Kass and Miller 1990], Chen and Lobo 1995 [Chen and da Vitoria Lobo 1995] solved the two dimensional Navier-Stokes equations converting them to 3D by using height field based on the pressure. They demonstrated both types of one-way coupling. This kind of coupling became common in many other groups. The full three dimensional Navier-Stokes equations were solved by Foster and Metaxas [Foster and Metaxas 1996; 1997a; 1997b] for both water and smoke. Big steps in efficiency were made introducing the use of semi-Lagrangian numerical techniques by Stam 1999 [Stam 1999]. Another improvemnet of fluid-solid interaction was the introduction of tangential movement of fluid along the obstacles presented by Foster and Fedkiw 2001 [Foster and Fedkiw 2001].

Foster and Metaxas 1996 [Foster and Metaxas 1996] demonstrate one-way fluid-to-solid coupling where solids are treated as massless particles that move freely on the fluids surface.

Yngve et al. 2000 [Yngve et al. 2000] demonstrated two-way coupling of breaking objects and compressible fluids in explosions, however their technique does not apply to incompressible fluids like water. Two-way coupling on regular grid was first presented by Takahashi et al. 2002 [Takahashi et al. 2002]. To approximate solid-to-fluid coupling they set zero Neumann boundary conditions for the pressure at these boundaries. Later they presented another variation at Takahashi et al. 2003 [Takahashi et al. 2003], where they used rigid body solver and fluid solver to achieve solid-fluid coupling. Drawback of this techniques is neglecting the dynamic forces and torques of solid objects.

Our paper follows the simulation steps according to our implementations, consisting of

(1) Setting the time step size Δt.
(2) Computing rigid dynamics of solids using Eqs. 1, 6.
(3) Setting boundary conditions between solid objects and fluid described in Section 4.
(4) Solving Navier - Stokes equations, see Eqs. 7, 8.
(5) Computation of new VOF values according to Eqs.17, 18.

3. RIGID SOLIDS DYNAMICS

We simplify the movement of solid objects to translation an rotation of center of mass *CM*, where position and rotation matrix is saved in global variables. The solid is represented in the environment as another fluid, but with restrictions in later steps of simulation, when solving Navier-Stokes equations. These restrictions will be explained later in this section. First, we have to calculate forces acting on our solid. We compute them from fluid velocities, known from previous time step, in the fluid cells neighboring the cells with 'solid fluid'. It is easy to find the vector starting at the point of force activity to CM, because we save the CM position. We sum the forces to find the total force F^n and torque τ^n:

$$F^n = \sum_i F_i, \qquad \tau^n = \sum_i r_i \times F_i, \tag{1}$$

where F_i is the force acting on solid and r_i is a vector from point of force activity to CM. This values can be used to integrate the position, velocity, orientation and angular velocity. We do this by solving the following equations for position:

$$r_{CM}^{n+1} = r_{CM}^n + \Delta t v_{CM}^n, \tag{2}$$

where Δt is the time step, r_{CM}^{n+1} is a new position and v_{CM}^n is the velocity of CM

$$v_{CM}^{n+1} = v_{CM}^n + \Delta t \frac{F^n}{M}. \tag{3}$$

M is the mass of solid calculated from volume of 'solid fluid' and the density. The orientation of the solid needed later for visualization of solid is calculated by

$$A^{n+1} = A^n + \Delta t \omega^n A^N. \tag{4}$$

The last integration equation is the computation of angular momentum L_{CM}^{n+1}:

$$L_{CM}^{n+1} = L_{CM}^n + \Delta t \tau^n. \tag{5}$$

We compute the angular velocity as

$$\omega^{n+1} = I L_{CM}^{n+1}, \tag{6}$$

where I is inertia of solid. After solving the dynamics of solid object we set the computed velocities as 'solid fluid' velocities. Now the simulation of fluid can continue, but the 'solid fluid' velocities aren't changed until the next simulation step where we re-compute the solid dynamics again.

4. BOUNDARY CONDITIONS

To prevent fluid to flow into the obstacle, specific boundary conditions must be set. This process contains manually setting up the velocity on faces between fluid and obstacle. The situation is outlined on figure 1.

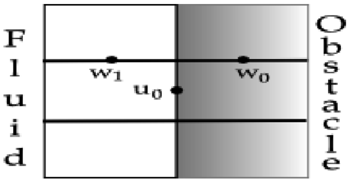

Fig. 1. Boundary conditions set the normal velocity u_0 and tangential velocity w_0 on an obstacle.

First, the normal velocity on boundary face is set to zero, i.e. $u_0 = 0$. pressure in obstacle cell is set the same as in adjacent fluid cell. Those are actually the Neumann boundary conditions.

Later in the simulation process we also set the tangential velocity in obstacle w_0. Depending on the type of slip conditions we have two opportunities: For free-slip boundary we set $w_0 = w_1$ and for no-slip boundary we set $w_0 = -w_1$.

In our simulation there are no empty cells but all cells are occupied by some type of the fluid. Due to this fact empty cells are eliminated and surface cells are cells containing at least two fluids, therefore we do not need to set surface boundaries on fluid/fluid interfaces.

5. NAVIER-STOKES EQUATIONS

The Navier-Stokes equations for incompressible fluids are

$$\frac{\partial u}{\partial t} = -\left(u.\nabla\right)u + \nabla\left(v\nabla u\right) - \frac{1}{\rho}\nabla p + \frac{f}{\rho}, \tag{7}$$

$$\nabla.u = 0, \tag{8}$$

where $u = (u, v, w)$ is velocity vector, ρ is a fluid density, v is fluid viscosity, $f = (f, g, h)$ is the external force, mostly it is only the gravitation force, i.e. $\left(f = h = 0; g = 9.8ms^2\right)$, p is the pressure in the fluid and $\nabla = \left(\frac{\partial}{\partial x}, \frac{\partial}{\partial y}, \frac{\partial}{\partial z}\right)$ is nabla operator. Equation 7 is the law of momentum and Eq. 8 represents the conservation of mass. Left side of Eq. 7 is the change of fluid velocity over time and the right side is the sum of the acting forces on fluids like advection, diffusion, pressure and external forces.

Some fluid motions like turbulence and surface tension, are not included in these two equations, but we assume that their effects are dominated by the above velocity and forces.

In our simulation we solve this equations in their discrete form on MAC grid, therefore we use finite difference method. In the gird cell the velocities are defined on faces of the cell and pressure, viscosity and density in the middle.

To solve N-S equations we use Helmholtz - Hodgeo decomposition of vector field into the potential and gradient component

$$\tilde{u} = u + \nabla q. \tag{9}$$

We get the gradient component by solving the following Poisson equation

$$\nabla.u = \nabla^2 q. \tag{10}$$

To do so we define operator P that projects vector field into his potential component $u = P\left(\tilde{u}\right) = \tilde{u} - \nabla q$. After discretizing the first N-S equation in time domain with time step Δt and application of P operator on both sides of equation we get

$$u^{new} = P\left[u + \Delta t\{-\left(u.\nabla\right)u + v\nabla^2 u + \frac{f}{\rho}\}\right], \tag{11}$$

where the ∇ operator on MAC grid cell with index i, j, k is approximated by

$$\nabla \approx \frac{u_{i+1,j,k} - u_{i,j,k}}{\Delta x} + \frac{v_{i,j+1,k} - v_{i,j,k}}{\Delta y} + \frac{w_{i,j,k+1} - w_{i,j,k}}{\Delta z}. \tag{12}$$

Solving Eq. 11 consists of two steps. First, we compute *guess velocities*

$$\tilde{u} = u + \Delta t\{-(u.\nabla)\,u + v\nabla^2 u + \frac{f}{\rho}\} \tag{13}$$

that gives us the velocities of fluid driven by the advection, diffusion and external forces. Second, we apply the *pressure projection* meaning that after second step the velocity will satisfy the second N-S equation $\nabla.u = 0$:

$$\nabla.u^{new} = \nabla.\left(\tilde{u} - \frac{\Delta t}{\rho}\nabla p\right) = \nabla.\tilde{u} - \frac{\Delta t}{\rho}\nabla^2 p = 0. \tag{14}$$

Therefore, by solving the derived poisson equation

$$\nabla^2.p = \frac{\rho}{\Delta t}\nabla.\tilde{u} \tag{15}$$

we get a new velocity in next simulation time step from

$$u^{new} = \tilde{u} - \frac{\Delta t}{\rho}\nabla p. \tag{16}$$

6. VOF METHOD

As a method of surface tracking we choose VOF method based on step function F, presented by Numata, Durikovic [Durikovic and Numata 2006]. The VOF function designates fraction fluid volume of a cell where 1 means that the cell is full of fluid and 0 means the cell is empty.

The difference of our method is the use of this method for multiple fluids on a cell structure consisting different sizes. Value F is changed by the velocity of fluid on cell faces, to compute it we use donor-acceptor schema. The approximation by finite differences would "blur" the surface of fluid so the sharp profile of surface is lost.

We outline the computing of VOF. Volume of stream that flows from donor to acceptor is $|V_i| = u_i \Delta t \Delta S_{ss}$, where u_i is the normal velocity on the face of cell i and the sign of velocity defines wetter the cell is donor or acceptor. The ΔS_{ss} is the common surface area between donor and acceptor, i.e the smaller area of the two faces. Let us note F_{in} the fluid fraction of n-th fuild in cell i. Volume of fluid n that flows trough face of the cell in time step Δt is

$$|V_n| = \min\left\{F_{in}^{AD}\,|V_i| + CF_{in}, F_{in}^{D}\,|V_D|\right\}, \tag{17}$$

where

$$CF_{in} = \max\left\{\left(1 - F_{in}^{AD}\right)|V_i| - \left(\sum_i F_{in}^{D} - F_{in}^{D}\right)|V_D|\,, 0\right\}. \tag{18}$$

Indexes D and A denote donor and acceptor while double index AD denotes donor or acceptor by the orientation of interface in respect of direction of the flow.

The minimum in Eq. 17 restrain the cell to give away more fluid it has and the maximum in Eq. 18 assures additive flow CF if volume that should be transferred is larger then accessible. Few examples of donor acceptor situations are shown in figure 2.

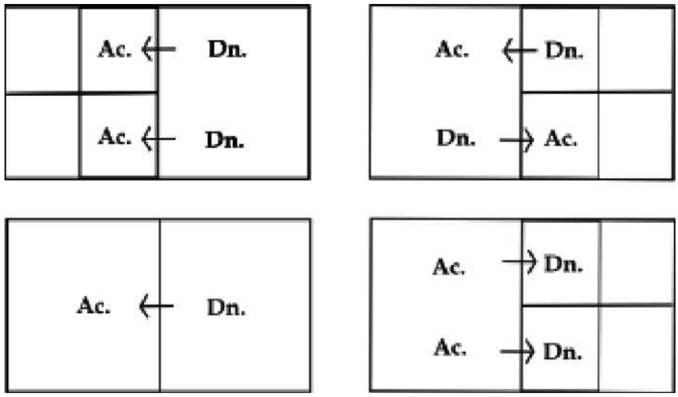

Fig. 2. Donor-Acceptor situations.

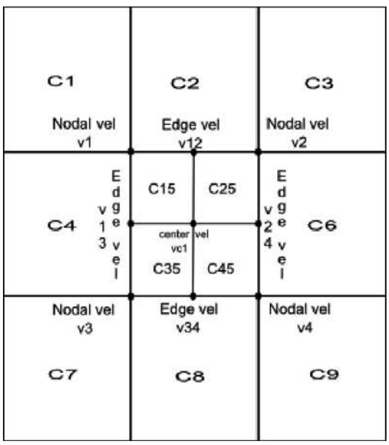

Fig. 3. Velocity computation by trilinear interpolation.

It can happen that some cell has the $F_{in} > 0$ than we proportionally distribute excess fluid back to donor cells. The whole process runs for all cells in volume from top to bottom and should be repeated till no change in F_{in} values occurs.

7. REWRITING THE METHODS ON OCTREE STRUCTURE

In our simulation we use octree data structure. Advantages of this approach are less computational demands at preservation of level of details. We use the ability of adaptive mesh refinement (AMR) in visually pleasant sections. Disadvantage is the necessity of recomputing the velocity, density and pressure and VOF values when changing the level of octree.

7.1 Velocity Computation

We show the method of velocity re-computation at the center of $C5$ on a 2D example in figure 3. The cell $C5$ was split by octree method into four equal sub-cells.

First, we compute the nodal velocities on $C5$ which are $v_1, \ldots, v_4$. Nodal velocity is an average of velocities on faces that are incident on the node. Next, we compute edge velocities v_{12}, v_{13}, v_{24}, v_{34} as an average of adjacent nodal velocities. Average of edge velocities gives us central velocity v_{c1} at face $C5$. The final velocities of new sub-faces of face $C5$ are average of their nodal velocities:

$$c15 = \frac{v1 + v12 + vc1 + v13}{4}. \tag{19}$$

The remaining three sub-face velocities are computed analogically.

Other way around, when combining 4 faces into a single face by octree method we just average their velocities.

7.2 Density and Pressure Computation

Pressure and density are constants defined at the cell center. When dividing a single cell into sun-cells we just set the respective values of sub-cell for pressure and density equal to the parent cell. When combining cells into a parent cell we set the pressure and density parameters as average of child cells.

7.3 VOF Computation

When combining child cells indexed $i = 1, \ldots, 8$ to a parent cell the final VOF value is given as the weighted average of child VOF values

$$F_n = \frac{\sum_{i=1}^{8} F_{i_n} |V_i|}{|V|}. \tag{20}$$

Dividing of parent cell into cells indexed $i = 1, \ldots, 8$ is a little more complex problem because we need to be sure there won't be any holes in the fluid after dividing. The value of VOF of the child cell is

$$F_{i_n} = min\left(1.0, 8F_n \frac{\sum_{neigh(i)} F_{s_n}}{\sum_{j=1}^{8}\left(\sum_{neigh(j)} F_{s_n}\right)}\right), \tag{21}$$

where n is the index of the fluid, i is the index of child cell and s is the index of neighbor of child cell. Neighbor of cell is every cell that has common face with child cell i and one neighbor cell with common node. None of these neighbors is another child cell.

8. IMPLEMENTATION

In our simulation, we simplify the movement of solid objects to translation and rotation of the centre of the mass CM, with position and orientation matrix saved as global variables. Due to the grid representation of fluid with known velocities at grid points we represent the floating object as a set of points lying on the object surface. The placement density of points is set by the maximal possible division of the octree structure.

First, we have to calculate the forces acting on our solid in a grid cell i. For floating objects, we have three possible types of forces which can act on the object: gravitation force, buoyancy force, and resistance force.

Gravitation force is computed by the following equation:

$$F_{Gi} = M_s \cdot g, \tag{22}$$

where M_s is the mass of the object s and g is the gravitational acceleration.

Buoyancy force is calculated as:

$$F_{Bi} = gV_sD_f, \tag{23}$$

where V_s is the volume of object s. D_f is temporary density computed every simulation step as

$$D_f = \frac{\sum_i d_i \cdot V_i}{\sum_i V_i}, \tag{24}$$

where i is the index of fluid cell that contains at least one of the points representing the floating object. V_i is the volume of the cell i, and the d_i is the density of the fluid cell.

Resistance force is calculated in the fluid cells that contains points representing the object from the velocities of fluid and object known from previous simulation step. Resistance force is

$$F_{Ri} = \frac{1}{2}CS\rho\ v^2, \tag{25}$$

where S is the surface of the floating object, C is the resistance constant depending on the shape of the object, ρ is the density of fluid in the cell i, and v is the relative velocity of the object to water. The relative velocity is the difference between the fluid velocity and the object velocity.

It is easy to find the vector from the point of force activity to the CM, because we save the CM position. The vectors of gravitation and buoyancy are contra-directional to that direction, while gravitation force acts downwards on the z axis, and buoyancy force acts upward on the z axis. We sum the forces to find the total force and torque. After solving the dynamics of the solid object, as explained in the previous section, we update the position of floating object and all of its surface reference points.

9. RESULTS

The results of our simulation are stored in series of xml files. One xml file for each simulation step and one PovRay [Durikovic and Numata 2006] file, where the positions of obstacles are defined are combined to form the scene file that is rendered by PovRay raytracing program. Figure 4 shows interaction of two fluids air and water with static obstacle. The second example in Figure 5 shows the water floating down the stairs in air fluid. Following tables show initial parameters and sized of the scenes.

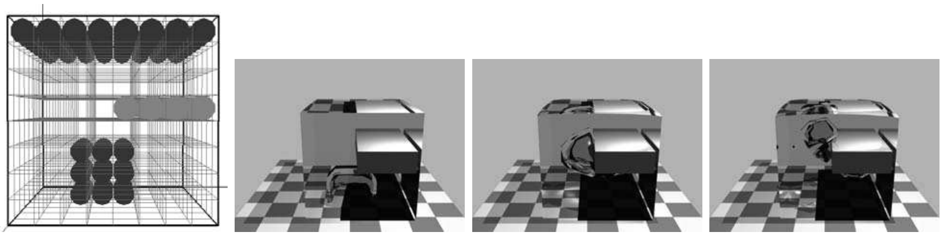

Fig. 4. Bubble. Top: the grid cells occupied by obstacle and fluids, rest are the key-frames from animation.

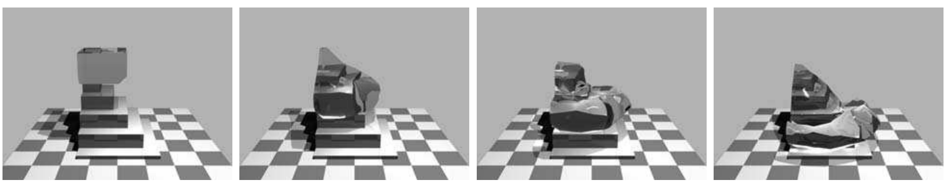

Fig. 5. Fluid falling down the stairs.

Bubble in a fluid		
# of cells	**Start of scene**	**End of scene**
total	512	512
air fluid (bubble)	91(27)	91(3)
water fluid	389	389
obstacles	32	32
# of time steps	50	
time of computation	8 [sec]	

Fluid falling down the stairs		
# of cells	**Start of scene**	**End of scene**
total	512	512
air fluid	48	48
water fluid	374	374
obstacles	90	90
# of time steps	70	
time of computation	10 [sec]	

Number of cells occupied by air and water fluid stays the same at the start and the end, that shows conservation of volume in this case. For both scenes fluids with properties shown in following table were used.

Fluids			
Fluid	**Density**	**Viscosity**	**Surface tension**
air fluid	0.1 $[kg/m^3]$	1 $[Pa \cdot s]$	0 $[N/m]$
water fluid	10 $[kg/m^3]$	0.1 $[Pa \cdot s]$	35 $[N/m]$

Figure 6 shows interaction of two fluids air and water with a flowing cylinder. Cylinder movement is fully influenced by the velocity of two fluids.

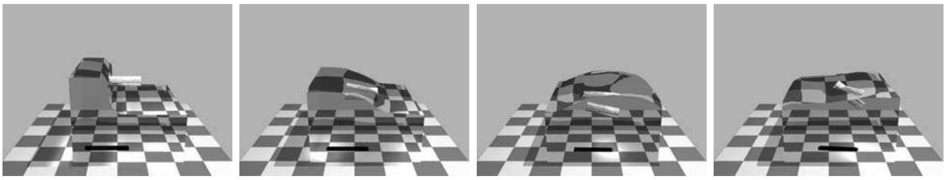

Fig. 6. Floating object in two fluids mixture.

10. CONCLUSION

In this paper we presented simulation of multiple fluid on adaptive grid particularly the octree structure. We have showed how to rewrite the update equation for the velocity, density, pressure and VOF values during the cell subdivision or cells gluing.

Our results show that the volume of fluid is preserved in each simulation step, i.e. volume is not lost.

Acknowledgment

This research was partially supported by a VEGA 1/0662/09 2009-2011 project a Scientific grant from Ministry of Education of Slovak Republic and Slovak Academy of Science.

REFERENCES

Chen, J. and da Vitoria Lobo, N. 1995. Toward interactive-rate simulation of fluids with moving obstacles using navier-stokes equations. In *Graphical Models and Image Processing.* Vol. 57. ACM SIGGRAPH, 107–116.

Durikovic, R. and Numata, K. 2006. Preserving the volume of fluid using multi-phase flow approach. *Information Visualisation, International Conference on 0*, 757–760.

Foster, N. and Fedkiw, R. 2001. Practical animation of liquids. In *Proceedings of SIGGRAPH 2001, Computer Graphics, Annual Conference Series*. ACM SIGGRAPH, 23–30.

Foster, N. and Metaxas, D. 1996. Realistic animation of liquids. *Graphical Models and Image Processing 58,* 5, 471–483.

Foster, N. and Metaxas, D. 1997a. Controlling fluid animation. In *In Proceedings of Computer Graphics International*. 178–188.

Foster, N. and Metaxas, D. 1997b. Modeling the motion of a hot, turbulent gas. In *SIGGRAPH '97: Proceedings of the 24th annual conference on Computer graphics and interactive techniques*. ACM Press/Addison-Wesley Publishing Co., New York, NY, USA, 181–188.

Kass, M. and Miller, G. 1990. Rapid, stable fluid dynamics for computer graphics. In *Proceedings of SIGGRAPH 1990, Computer Graphics, Annual Conference Series*. Vol. 24. ACM SIGGRAPH, 49–58.

Stam, J. 1999. Stable fluids. In *Siggraph 1999, Computer Graphics Proceedings*, A. Rockwood, Ed. Addison Wesley Longman, Los Angeles, 121–128.

Takahashi, T., Fujii, H., Kunimatsu, A., Hiwada, K., Saito, T., Tanaka, K., and Ueki, H. 2003. Realistic animation of fluid with splash and foam. *Comput. Graph. Forum 22,* 3, 391–400.

Takahashi, T., Ueki, H., Kunimatsu, A., and Fujii, H. 2002. The simulation of fluid-rigid body interaction. In *SIGGRAPH '02: ACM SIGGRAPH 2002 conference abstracts and applications.* ACM, New York, NY, USA, 266–266.

Yngve, G., O'Brien, J., and Hodgins, J. 2000. Animating explosions. In *Proceedings of SIGGRAPH 2000, Computer Graphics, Annual Conference Series*. ACM SIGGRAPH, 29–36.

SIMULATING MULTI-PHASE VISCOUS FLUIDS USING PARTICLES

Roman Ďurikovič
Faculty of Mathematics, Physics and Informatics,
Comenius University,
842 48 Bratislava, Slovakia
email: roman.durikovic@fmph.uniba.sk

Received March 2009