MODELING SEVERAL FLUIDS WITH PRESERVED VOLUME ON AN OCTREE DATA STRUCTURE

Roman Ďurikovič
Comenius University

Abstrakt

V článku rozširujeme metódu Marker-and-Cell (MAC) simulácie kvapalín na bunkách. V každej bunke máme zaznamenanú hustotu a tlak v strede bunky, a rýchlosti kvapaliny na každej stene samostatne v 6 smeroch. Táto metóda štandardne pracuje len s jedným typom kvapaliny. Náš prístup rozširuje metódu simulácie na viac kvapalín súčastne a preto sme navrhly modifikáciu Volume-of-Fluid (VOF) metódy, ktorá sleduje povrch kvapalín v bunke. Hodnoty VOF metódy používame na presné sledovanie povrchu kvapaliny, nutnej pre výpočet povrchového napätia, a tiež na vizualizáciu povrchu kvapaliny. Ak sa v jednej bunke nachádzajú rôzne kvapaliny počítame simulácie s jednou kvapalinou, ktorej hustota a viskozita je konštantná v bunke ale mení sa mimo bunky.
Kľúčové slová: dynamika kvapalín, zachovanie objemu, plávajúce objekty, octree

Abstract

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 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 equations on the octree grid yielding to a symmetric positive definite linear system.

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 track the fluid surface and integrate it into the multi phase-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. Values of VOF are actually used to tract the fluid surface for visualization purposes and to calculate the surface tension forces acting on the fluid interfaces.
Keywords: volume of fluid, fluid dynamics, one-way coupling, octree, Navier-Stokes, floating objects
Classification: Computer graphics:Animation:Procedural animation, Physical simulation

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.

The 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 WORKS

The scope of this paper deals with the visual appealing fluid simulations and therefore we commonly refer to the the computer graphics papers. The full three dimensional Navier-Stokes equations were solved by Foster and Metaxas [5] for both water and smoke in computer graphics community. Big steps in efficiency were made introducing the use of semi-Lagrangian numerical techniques by Stam 1999 [7]. Another improvement of fluid-solid interaction was the introduction of tangential movement of fluid along the obstacles presented by Foster and Fedkiw 2001 [3].

Foster and Metaxas 1996 [4] demonstrate one-way fluid-to-solid coupling, where solids are treated as massless particles that move freely on the fluid surface. Yngve et al. 2000 [11] 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 [9], 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 [8]. To approximate solid-to-fluid coupling they set zero Neumann boundary conditions for the pressure at these boundaries.

There are a few researches to handle interaction of two fluids, or fluid and solid objects in computer graphics. Tanaka et al. [10] used VOF and CIP techniques to create fluid animations. They also integrated rigid body simulation into their fluid simulator. Hong and Kim [6] used the VOF method to animate bubbles in a liquid.

Our paper follows the following simulation steps:

1. Setting the time step size Δt.
2. Move the VOF values according to the current velocity field.
3. Solving one-way solid fluid interaction and rigid dynamics of solids.
4. Set boundary velocities on faces of obstacle cells, similarly set Neumann boundary conditions for pressure on obstacle cells.
5. Calculate intermediate velocities using the Navier-Stokes equations without taking account of the pressure term, see Eqs. 1, 2.
6. Calculate pressure field from the intermediate velocity field.
7. Make the intermediate velocity field mass-conserving to get correct velocities.
8. To generate the animation frame construct fluid surface from the VOF values according to Eqs. 3, 4.

3 MOTION

The main contribution towards the goal of this research is to simulate mixtures of fluids, for example a gas and liquid. We represent each fluid by a VOF function, in other words the percentage occupation of a simulation cell. The VOF function designates volume fraction of a cell, where $F_n = 1$ means that the cell is full of n-th fluid and $F_n = 0$ means the cell does not contain the n-th fluid. Assuming the fluids are incompressible, then the spatial distributions of fluids has the following relation: $\langle F\rangle = \sum_n F_n = 1.0$, where F_n is the VOF function of *n*th fluid, while fluid can have different densities and viscosities, ρ_n and ν_n.

Each simulation cell has several VOF values, the interface between two fluids lies in cells which contain several nonzero VOF values. We treat a flow of a mixture of fluids as a flow of a single fluid which have variable density and viscosity. Then, the motion of the flow is calculated using density and viscosity weighted by VOF values $\langle\rho\rangle = \sum_n(F_n\rho_n)$, $\langle\nu\rangle = \sum_n(F_n\nu_n)$.

In this case, there are two or more fluids, and they are treated as a fluid with variable density $\langle\rho\rangle$ and viscosity $\langle\nu\rangle$. Fluid is expressed by velocity field $\mathbf{u}$, density field ρ and pressure field p. The Navier-Stokes equations describing the multiple flow conservation of momentum then will be

$$\frac{\partial \mathbf{u}}{\partial t} = -(\mathbf{u}\cdot\nabla)\mathbf{u} + \nabla\cdot(\langle\nu\rangle)\nabla\mathbf{u}) - \frac{1}{\langle\rho\rangle}\nabla p + \frac{\mathbf{f}}{\langle\rho\rangle} \tag{1}$$

$$\nabla\cdot\mathbf{u} = 0, \tag{2}$$

Here the viscosity $\langle\nu\rangle$ is placed inside the derivative. Left hand side of the equation is the unsteady acceleration. The first term on the right hand side of equation is the convective acceleration. The other terms following the acceleration are the forces such as the pressure gradient force, the viscosity force and other external forces $\mathbf{f}$. The conservation of mass, Eq. 2, can be written in the other way as

$$\nabla\cdot(\langle\rho\rangle\mathbf{u}) = -\frac{\partial\langle\rho\rangle}{\partial t}.$$

We can interpret the second equation such that the total of mass flow out of a small region is negative of mass change of the region.

3.1 Surface Tension from VOF

Accurate calculation of the surface tension effect needs well-defined fluid interfaces. Although our simulator does not consider actual shape of the interface, a model called continuum surface force (CSF) can be used to approximate the surface force without that information [1]. Method is based on volume force that converges to surface force as the grid size becomes smaller.

We propose the extension of this idea to multiple fluids as well. Surface force, $\mathbf{f}_n^{sv}$, exerted by fluid n based on the VOF value, F_n, is written as

$$\mathbf{f}_n^{sv} = \sigma_n\kappa_n\nabla F_n,$$

where σ_n is surface tension coefficient, and κ_n is curvature of the fluid surface. The normal and curvature of the surface can be calculated from VOF values as $\mathbf{n}_n = \nabla F_n$, $\kappa_n = -(\nabla\cdot\widehat{\mathbf{n}}_n)$, where $\widehat{\mathbf{n}}_n = \mathbf{n}_n/|\mathbf{n}_n|$ represents unit normal vector. Curvature κ_n is defined at center of a cell. However, as normal is a vector quantity, components of $\widehat{\mathbf{n}}_n$ are separately defined on faces of a grid cell, and they must be interpolated to obtain values at other positions. For example, x-component of a normal vector defined on center of right face on grid cell (i,j,k) with the cell size in x direction dx will be

$$n^{nx}_{i+1/2,j,k} = \frac{F_{n\ i+1,j,k} - F_{n\ i,j,k}}{dx},$$

and its y-component can not be calculated directly from VOF but by interpolation of known y-componen normals at the neighboring parallel faces on a grid cell: $n^{ny}_{i+1/2,j,k} = \frac{1}{4}(n^{ny}_{i,j-1/2,k} + n^{ny}_{i+1,j-1/2,k} + n^{ny}_{i,j+1/2,k} + n^{ny}_{i+1,j+1/2,k})$. Now, we are almost ready to calculate the surface force $\mathbf{f}_n^{sv}$ on each cell face. To get the curvature defined on a face between two cells we need to interpolate two cell-centered curvatures. Finally, surface forces for all fluids are accumulated and added to the external forces $\mathbf{f}^{ext}$:

$$\mathbf{f} = \sum_n \mathbf{f}_n^{sv} + \mathbf{f}^{ext}.$$

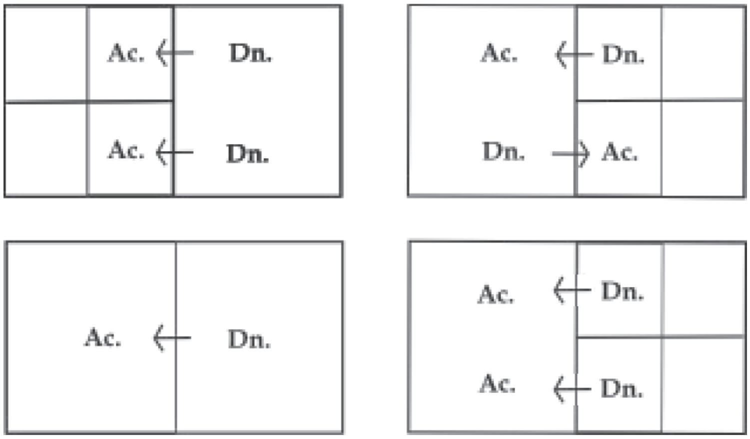

Fig. 1. Donor-Acceptor situations. Left column: donor cell is a single cell and the acceptor cells are on the left side. Right column: a single cell on the right serves as accepter or has a double functionality donnor/acceptor.

4 PROPOSED MULTIPLE FLUID VOF

As a method of surface tracking we choose VOF method based on step function F, presented by Durikovic, Numata [2]. Our previous surface tracking method worked primarily for a regular grid structure. The contribution of our work here is to extend the VOF method for multiple fluids on an adaptive octree cell structure consisting of different cell sizes. The problem we are facing here is that a single face of a bigger cell can tough two smaller cells in octree cell structure. Therefore we introduce the term of common surface area the smaller area of the two faces and calculate the fluid stream passing two smaller faces rather than one big face. 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 detail 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 whether 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

$$|V_n| = \min\left\{F_{in}^{AD}\,|V_i| + CF_{in}, F_{in}^{D}\,|V_D|\right\}, \tag{3}$$

where

$$CF_{in} = \max\{(1 - F_{in}^{AD})\,|V_i| - (\sum_i F_{in}^{D} - F_{in}^{D})\,|V_D|\,, 0\}. \tag{4}$$

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. 3 restrains the cell to give away more fluid it has and the maximum in Eq. 4 assures additive flow CF if volume that should be transferred is larger than accessible. Few examples of donor acceptor situations are shown in Figure 1. A single donor can donate the fluid to a single acceptor, Figure 1 left-bottom, or to two acceptor cells, Figure 1 left-top. Two smaller cells can serve as donor to a single acceptor cell, Figure 1 right-bottom. The mixed situation can happen when a large cell serves both as a donor and acceptor to two smaller neighoring cells, Figure 1 right-top.

It can happen that if some cell has the $F_{in} > 0$ then 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.

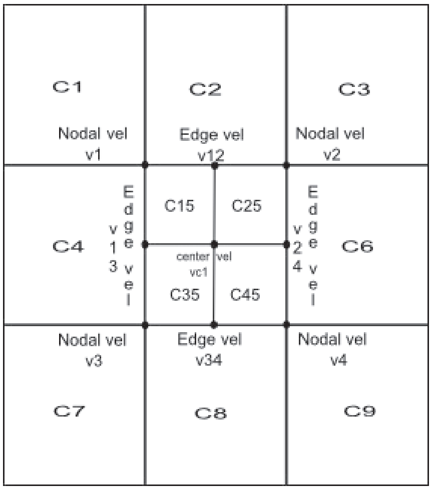

Fig. 2. Velocity computation by trilinear interpolation.

5 OCTREE GRID

In our simulation we propose to use the 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, pressure and VOF values when changing the level of octree.

5.1 Velocity Computation

We show the method of velocity re-computation at the center of $C5$ on a 2D example in Figure 2. 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:

$$c15 = \frac{v1 + v12 + vc1 + v13}{4}. \tag{5}$$

The remaining three sub-face velocities are computed similarly.

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

5.2 Density and Pressure

Pressure and density are constants defined at the cell center. When dividing a single cell into sub-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.

5.3 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

$$F_n = \frac{\sum_{i=1}^{8} F_{i_n} |V_i|}{|V|}. \quad (6)$$

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 within the fluid after cell dividing. The value of VOF of the child cell is

$$F_{i_n} = min\left(1.0, 8F_n \frac{\sum_{neigh(i)} F_{s_n}}{\sum_{j=1}^{8}\left(\sum_{neigh(j)} F_{s_n}\right)}\right), \quad (7)$$

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.

6 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 grid cells lying on the object surface.

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: $F_{Gi} = M_s \cdot g$, where M_s is the mass of the object s and g is the gravitational acceleration.

Buoyancy force is calculated as: $F_{Bi} = gV_sD_f$, where V_s is the volume of object s. D_f is temporary density computed every simulation step as

$$D_f = \frac{\sum_i d_i \cdot V_i}{\sum_i V_i}, \quad (8)$$

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 contain points representing the object from the velocities of fluid and object known from previous simulation step. Resistance force is $F_{Ri} = \frac{1}{2}CS\rho\, v^2$, 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.

7 RESULTS

The results of our simulation are stored in series of xml files. One xml file for each simulation step and one PovRay [2] file, where the positions of obstacles are defined are combined to form the scene file that is rendered by PovRay raytracing program. Figure 3 shows interaction of two fluids air and water with static obstacle. The second example in Figure 4 shows the water floating down the stairs in air fluid. Following tables show initial parameters and sizes of the scenes.

Bubble in a fluid [# of cells]		
	Start	End
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	End
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	Tension
air	0.1 $[kg/m^3]$	1 $[Pa \cdot s]$	0 $[N/m]$
water	10 $[kg/m^3]$	0.1 $[Pa \cdot s]$	35 $[N/m]$

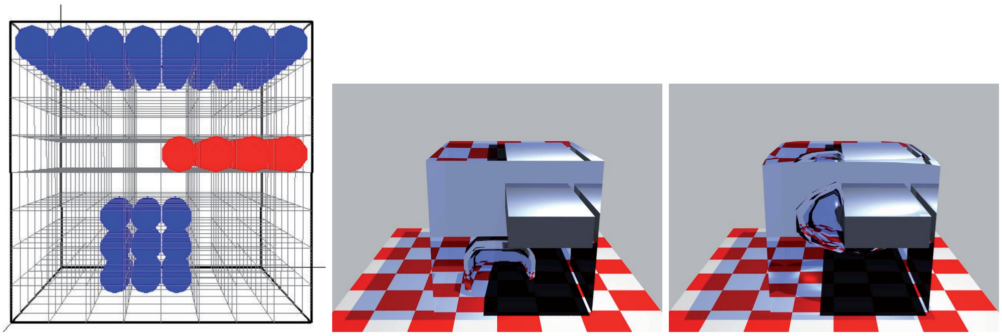

Fig. 3. Bubble. Left: the grid cells occupied by obstacle and fluids, the rest images are the key-frames from animation. There are two fluids used in the simulation air and the water.

Figure 5 shows interaction of two fluids air and water with a flowing cylinder. Cylinder movement is fully influenced by the velocity of two fluids.

8 CONCLUSION

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

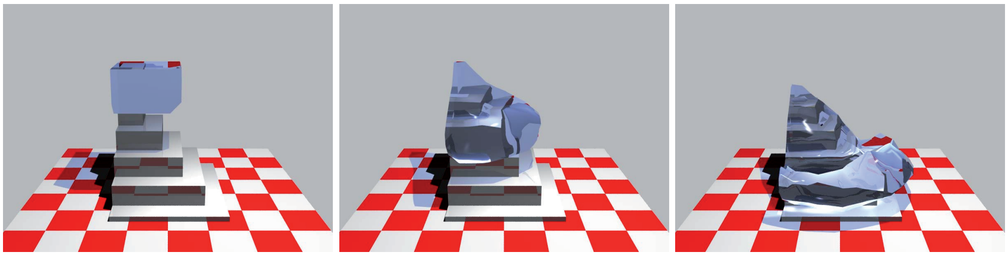

Fig. 4. Fluid falling down the stairs. There are two fluids used in the simulation air (was not rendered) and the water.

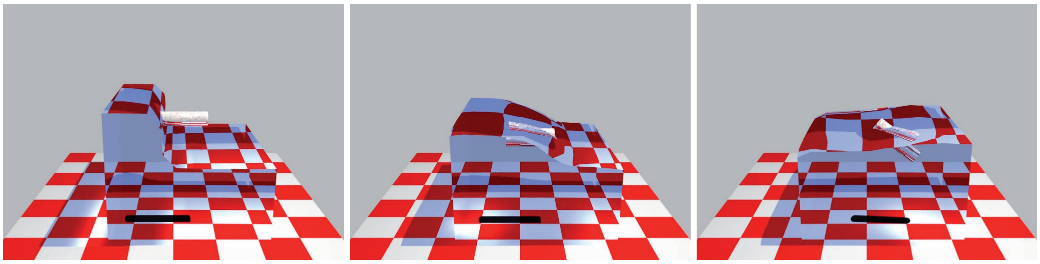

Fig. 5. Floating object in two fluids mixture. There are two fluids used in the simulation air (was not rendered) and the water.

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

References

[1] J. U. Brackbill, D. B. Kothe, and C. Zemach. A continuum method for modeling surface tension. *J. Comput. Phys.*, 100(2):335–354, 1992.

[2] Roman Durikovic and Katsuhiro Numata. Preserving the volume of fluid using multi-phase flow approach. In *Proceedings of the International Conference on Information Visualisation*, pages 757–760, Los Alamitos, CA, USA, 2006. IEEE Computer Society.

[3] Nick Foster and Ron Fedkiw. Practical animation of liquids. In *Proceedings of SIGGRAPH 2001, Computer Graphics, Annual Conference Series*, pages 23–30, ACM SIGGRAPH, August 2001.

[4] Nick Foster and Dimitris Metaxas. Realistic animation of liquids. *Graphical Models and Image Processing*, 58(5):471–483, 1996.

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

[6] Jeong-Mo Hong and Chang-Hun Kim. Animation of bubbles in liquid. *Computer Graphics Forum*, 22(3):253–253, 2003.

[7] Jos Stam. Stable fluids. In Alyn Rockwood, editor, *Siggraph 1999, Computer Graphics Proceedings*, pages 121–128, Los Angeles, 1999. Addison Wesley Longman.

[8] Tsunemi Takahashi, Hiroko Fujii, Atsushi Kunimatsu, Kazuhiro Hiwada, Takahiro Saito, Ken Tanaka, and Heihachi Ueki. Realistic animation of fluid with splash and foam. *Computer Graphics Forum*, 22(3):391–400, 2003.

[9] Tsunemi Takahashi, Heihachi Ueki, Atsushi Kunimatsu, and Hiroko Fujii. The simulation of fluid-rigid body interaction. In *SIGGRAPH '02: ACM SIGGRAPH 2002 conference abstracts and applications*, pages 266–266, New York, NY, USA, 2002. ACM.

[10] Ken Tanaka, Heihachi Ueki, and Atsushi Kunimatsu. The cubic interpolated level set method for realistic fluid animation. In *SIGGRAPH '03: Proceedings of the SIGGRAPH 2003 conference on Sketches & applications*, pages 1–1, New York, NY, USA, 2003. ACM Press.

[11] G.D. Yngve, J.F. O'Brien, and J.K. Hodgins. Animating explosions. In *Proceedings of SIGGRAPH 2000, Computer Graphics, Annual Conference Series*, pages 29–36, ACM SIGGRAPH, July 2000.

Prof. RNDr. Roman Ďurikovič, PhD.
Dept. of Applied Informatics,
Faculty of Mathematics, Physics and Informatics,
Comenius University,
842 48 Bratislava, Slovak Republic;
`http://www.sccg.sk/~durikovic`
`e-mail:roman.durikovic@fmph.uniba.sk`