Ibero-American Symposium on Computer Graphics - SIACG (2006)
P. Brunet, N. Correia, and G. Baranoski (Editors)

A Mesh-Particle Model for Fluid Animation

N. Suárez and A. Susín

Technical University of Catalonia, Barcelona, Spain

Abstract

We use our adapted versions of the two most used methods in Computer Fluid Animation, Marker and Cell and Smoothed Particle Hydrodynamics, to develop a new method taking advantage of the calculation speed of the first and the great level of detail and controllability of the second. Such a method is very useful in animations with a great volume of fluid where the events needing high-level detail take place on the surface.
Finally, we present some simulation examples made with this new method.

Categories and Subject Descriptors (according to ACM CCS): I.6.5 [Simulation and Modeling]: Modeling Methodologies, I.6.8 [Simulation and Modeling]: Types of Simulation. Animation

1. Introduction

The complexity of fluid behaviour is well known. Generally, all the methods used in Fluid Animation have their advantages and their disadvantages and the use of only one of such methods is not enough to catch the realism of the scene maintaining, at the same time, an acceptable performance and some animation control to modify the behaviour of a fluid when its real simulation does not fit the artistic requirements. These are the main challenges when simulating fluids for video-games, virtual environments or other interactive applications.

In the Fluid Simulation world the reference equations for modeling ordinary events (like liquid streams, liquids moving inside containers and even low speed smoke) are the *Navier-Stokes Equations for Incompressible Viscous Flows*:

$$\begin{cases} \frac{\partial}{\partial t}\vec{u} + (\vec{u}\cdot\nabla)\vec{u} + \nabla p = \frac{1}{Re}\triangle\vec{u} + \vec{g} \\ \text{div } \vec{u} = 0, \end{cases} \tag{1}$$

where the unknowns are $\vec{u}$ and p, velocity and pressure, respectively, Re is the *Reynolds Number*, directly related with the Kinematic Viscosity ($Re = 1/\mu$) and $\vec{g}$ represents de body forces, such as gravity.

From the wide range of approaches to the numeric solution of these equations different methods of Fluid Simulation have arisen, each one, as we said, with its advantages and its disadvantages. Among all of them, we can stand out the following two:

- *Marker and Cell* (MAC).
  Simulation method with an eulerian approach, in which the unknowns are calculated over a mesh of the domain and the fluid position is determined, at every time step, by marker particles. These particles do not have any mass and are moved through the simulation area according to the velocity field.
  It has very good simulation times when the scene does not require high-level details [FM96].
- *Smoothed Particle Hydrodynamics* (SPH).
  It is a lagrangian method in which the fluid is represented by particles, each of them with its own values and associated characteristics, that determine the movement of the fluid.
  This method can achieve high-level detail but means a very important computational effort since the behaviour of each particle depends on the behaviour of the surrounding particles at every time step [Mon92].

Nowadays, MAC and SPH coexist with other important and interesting methods, like the semi-lagrangian one of J. Stam [Sta99]. Unfortunately, this method is very suitable for dealing with smoke but has some troubles when it is directly applied to liquid simulation (it suffers from mass dissipation) and needs some special techniques to be used on the surface [FF01].

Most of these models try to visually improve liquid surface using particles that make possible the simulation of foam and splash (e.g. [TFK*03] or [CCLY02]) but their appearance depends directly on mesh methods values, keeping their associated global problems, like loss of waves because of the tendency to smooth features and lack of control by the animator (usually, unphysical changes are absorbed by mesh methods producing unnatural behaviours).

For this work, we have taken MAC and SPH as our main grounding because both are widely used methods in Fluid Simulation (in their different versions) with very good results and they are, clearly, complementary models: our aim is the creation of a new method combining MAC, in great volume zones where not a too much detailed simulation is needed, and SPH, in zones where it is actually needed. This way, we will reduce the computation time without loosing the level of detail. Moreover, SPH is a very flexible model, in the sense that the fluid behaviour can be manipulated by animators through the forces acting over each particle.

In the two following sections we describe some characteristics of our 3*D* versions of MAC and SPH.

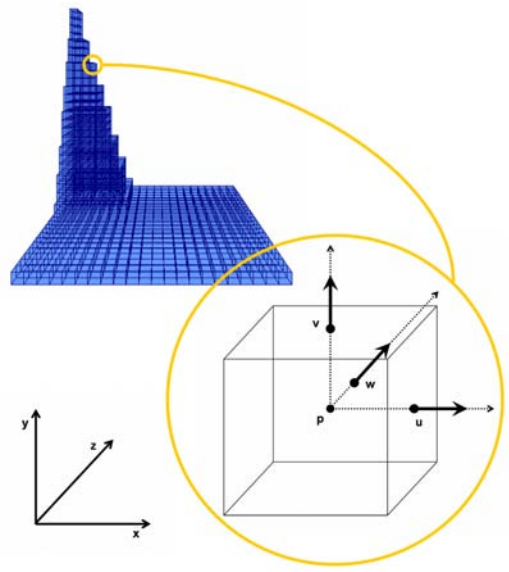

Figure 1: *Staggered Grid.*

2. Marker and Cell Model

This model was first developed in the 1960's by F. Harlow and J. Welch [HW65] and meant a very important step in the world of Fluid Simulation. Since then, it is being used by many authors as grounding to develop their own methods, like N. Foster and D. Metaxas [FM97] or J. Stam [Sta03].

The MAC starting point are the equations in (1) written and simplified for 3 dimensions:

$$\begin{aligned}
&\frac{\partial u}{\partial t}+\frac{\partial p}{\partial x}=\frac{1}{Re}\left(\frac{\partial^2 u}{\partial x^2}+\frac{\partial^2 u}{\partial y^2}+\frac{\partial^2 u}{\partial z^2}\right)-\frac{\partial u^2}{\partial x}-\frac{\partial uv}{\partial y}-\frac{\partial uw}{\partial z}+g_x\\
&\frac{\partial v}{\partial t}+\frac{\partial p}{\partial y}=\frac{1}{Re}\left(\frac{\partial^2 v}{\partial x^2}+\frac{\partial^2 v}{\partial y^2}+\frac{\partial^2 v}{\partial z^2}\right)-\frac{\partial uv}{\partial x}-\frac{\partial v^2}{\partial y}-\frac{\partial vw}{\partial z}+g_y\\
&\frac{\partial w}{\partial t}+\frac{\partial p}{\partial z}=\frac{1}{Re}\left(\frac{\partial^2 w}{\partial x^2}+\frac{\partial^2 w}{\partial y^2}+\frac{\partial^2 w}{\partial z^2}\right)-\frac{\partial uw}{\partial x}-\frac{\partial vw}{\partial y}-\frac{\partial w^2}{\partial z}+g_z\\
&\frac{\partial u}{\partial x}+\frac{\partial v}{\partial y}+\frac{\partial w}{\partial z}=0,
\end{aligned} \tag{2}$$

where u, v and w represent velocities in x, y and z directions, respectively.

For the numerical solution of these equations, the simulation area is voxelized in parallelepipeds with their faces parallel to those of the main domain. Pressure is calculated in the middle of each one of these resulting cells and velocities are calculated in the middle of their right, upper and back faces, obtaining a *Staggered Grid* (see figure 1).

Once we have done this, we introduce the corresponding changes into the equations.

2.1. Space and time discretization

The approximation of the spatial derivatives is made using progressive, centred or regressive finite differences (f.d.), depending on the considered physical magnitude and its derivation order.

Dealing with the advection terms, we use a combination of centred f.d. and *Donnor-Cell Discretization*. This technique gives the MAC method more stability when these terms dominate the motion [GDN98].

In the case of time derivatives, progressive f.d. are used and the values of p are taken in time $t^{(n+1)}$, being $(n+1)$ the new time state, in which we do not know the values of u, v, w or p:

$$\begin{aligned}
u^{(n+1)} &= F^{(n)}-dt\left[\frac{\partial p^{(n+1)}}{\partial x}\right]\\
v^{(n+1)} &= G^{(n)}-dt\left[\frac{\partial p^{(n+1)}}{\partial y}\right]\\
w^{(n+1)} &= H^{(n)}-dt\left[\frac{\partial p^{(n+1)}}{\partial z}\right].
\end{aligned} \tag{3}$$

Thus, we have an implicit method in pressure ($[\cdot]$ means spatial discretization).

2.2. Pressure calculation

Now, to solve our problem, we only need the pressure values in time $t^{(n+1)}$ for all the cells. In order to achieve them, we

use the expressions in (3) in the last equation in (2), obtaining a *Poisson Equation* for pressures,

$$\frac{\partial^2 p^{(n+1)}}{\partial x^2} + \frac{\partial^2 p^{(n+1)}}{\partial y^2} + \frac{\partial^2 p^{(n+1)}}{\partial z^2} = \frac{1}{dt}\left(\frac{\partial F^{(n)}}{\partial x} + \frac{\partial G^{(n)}}{\partial y} + \frac{\partial H^{(n)}}{\partial z}\right), \tag{4}$$

that we solve with the *Successive Over-Relaxation Method* (SOR) [Had00].

2.3. Solid contact and free surface

The numerical solution of the equations discretized this way needs some values for the different physical magnitudes inside the solid cells. Those values depend on the chosen boundary conditions.

On the other hand, some suitable pressure and velocity values are also needed to simulate the behaviour of the fluid free surface. To assign these values, we should study all the surface cells at every time step. However, this study is not necessary in our hybrid method because our MAC model is running only in the deeper part of the fluid, far from the surface cells. Thus, no surface conditions are needed, avoiding the most complicated feature of MAC.

3. Smoothed Particle Hydrodynamics Model

SPH was originally developed in the 1970's by R. Gingold and J. Monaghan [GM77] and L. Lucy [Luc77], in separated works. It was first conceived to solve compressible astrophysical problems, but because of its mesh-free nature it has been used, and is being used nowadays, to simulate fluids and deformable substances. The recent works of M. Desbrun and M. Cani [DC99], J. Monaghan and A. Kos [MK00], M. Müller et al. [MCG03] or P. Cleary and M. Prakash [CP04] or are good examples.

In SPH, as a Lagrangian approach, the fluid is not considered a continuous material. Instead, it is formed by particles which carried out the fluid properties and their motion is governed by the *Newton's Second Law*, $F = m \cdot a$. Thus, once we know the forces acting over the particles, we know their velocities and positions.

To calculate this forces, an interpolation technique based on kernel functions, $W_h(x)$, is used. These functions, called *Smoothing Kernels*, are an approximation of the Dirac's δ function and describe the behaviour of the characteristics associated to the particle in its neighbourhood.

With this technique, the equations of the fluid model become:

$$\begin{aligned} F_i^{\nabla P} &= -m_i \sum_{j \neq i} m_j \left(\frac{p_i}{\rho_i^2} + \frac{p_j}{\rho_j^2}\right) \nabla_i W_h^{ij} \\ (\nabla \cdot \vec{u})_i &= \frac{1}{\rho_i} \sum_{j \neq i} m_j \left(\vec{u}_i - \vec{u}_j\right) \nabla_i W_h^{ij}, \end{aligned} \tag{5}$$

where $\nabla_i W_h^{ij}$ is the gradient of $W_h(\vec{x}_i - \vec{x}_j)$ taken with respect to $\vec{x}_i = (x_i, y_i, z_i)$, the coordinates of particle i, ρ_i is the density of particle i and m_i, the mass it represents.

The first equation describes the particles movement due to the differences in pressure (viscosity and body forces should be added to this one). The second one allows us to know the density variation: $\dot{\rho}_i = -\rho_i(\nabla \cdot \vec{u})_i$. This form of the continuity equation has good numerical conservation properties and is not affected by free surface or density discontinuities [CP04].

3.1. Equation of State

Although SPH uses a compressible version of the Navier-Stokes Equations, it can be forced to work near the incompressible limit (actually, that is the situation in real fluids) by means of the *State Equation*, that links pressure and density [MK99]:

$$p_i = P_0 \left[\left(\frac{\rho_i}{\rho_0}\right)^7 - 1\right]. \tag{6}$$

Here, ρ_0 is the initial density and P_0 is a constant depending on the fluid.

3.2. Smoothing kernel

For our SPH version we have chosen a kernel with the basic features (normalized integral and $W_h(\vec{x}) \to \delta(\vec{x})$ when h tends to 0), compact support and with a gradient that prevents from the decrease of the repulsion force when two particles are very close (see figure 2), avoiding cluster formation [DC99]:

$$W_h(\vec{x}) = \frac{15}{\pi(4h)^3} \begin{cases} \left(2 - \frac{\|\vec{x}\|}{h}\right)^3 & \text{if} \quad 0 < \|\vec{x}\| \leq 2h \\ 0 & \text{if} \quad 2h < \|\vec{x}\|. \end{cases} \tag{7}$$

The fact of using a compact support kernel reduces the number of particles interacting with a given one. This allows to increase the speed when we look for the neighbour particles creating a storage matrix with the information of the particles in every $2h \times 2h \times 2h$ cell of the simulation domain.

3.3. Integration scheme

In order to obtain more accurate results, we use a *Leap-Frog* scheme, where accelerations and positions are calculated with a time lag of $dt/2$ in respect to velocities. Because

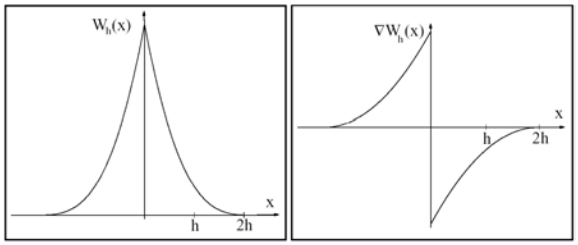

Figure 2: *Smoothing kernel and its gradient in 1D.*

of this, it is very important to calculate velocities $dt/2$ in advance to compute the following acceleration values with the SPH algorithm:

$$\vec{u}^{(n+1+1/2)} = \vec{u}^{(n+1)} + \vec{a}^{(n+1)} dt/2. \tag{8}$$

4. MAC-SPH mixed model

Now, we want to create a method profiting from the advantages of the previous methods and avoiding or minimizing their disadvantages.

Let us suppose that a great volume of liquid suffers some disturbances in such a way that the more important consequences take place on its surface.

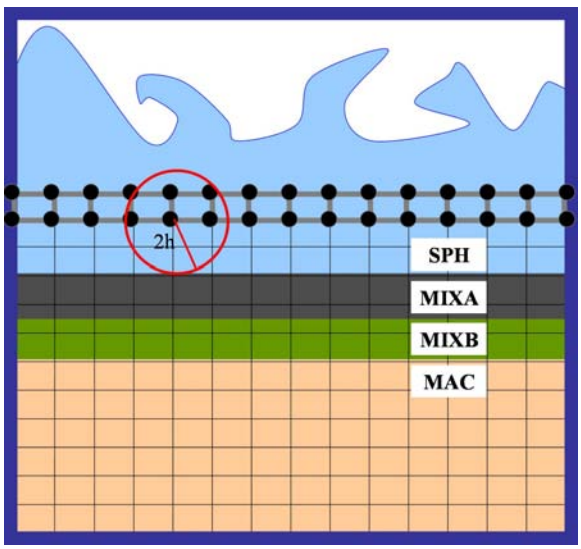

Figure 3: *Qualitative 2D representation of the MAC-SPH model.*

4.1. MAC adaptation

On the one hand, we find that most of the fluid has a smooth and very regular behaviour that can be simulated efficiently with MAC. So, we choose a suitable height and make the necessary adaptations to use a MAC simulation under it.

First of all, calculation of velocity and pressure values on the last of the MAC-mesh rows needs these values on its upper row cells. Using the SPH technique, we can assign them to some nodes (see figure 3),

$$A_i = \sum_{i \neq j} A_j \frac{m_j}{\rho_j} W_h^{ij} \quad (A \equiv u, v, w, p), \tag{9}$$

and then make the corresponding averages to complete the staggered grid values of our new shortened mesh (for the pressure we must observe that MAC-mesh values are in fact p/ρ, whereas for SPH particles are actually p).

To implement the calculation of node values we use the same storage matrix of SPH. In this case, since nodes are fixed, their $2h \times 2h \times 2h$ cells are computed only once at the beginning of the simulation (only SPH information is deleted each time). Besides, they do not take part in actual SPH calculations.

We make sure that all the particles at distance $2h$ around the nodes are SPH particles. This way, all the operations are well defined and we manage to influence MAC values by SPH values, since this influence is transmitted through the mesh.

4.2. SPH adaptation

On the other hand, more accuracy is needed to simulate the surface zone and this can be achieved using SPH.

In order to transmit the information of MAC calculations to the SPH particles located in this zone, we need some kind of "transmitters".

- SPH particles can not interact directly with MAC marker particles because they need acceleration, mass, pressure, density, etc., as well as their position and velocity values. To solve this problem, we could assign such values by means of the MAC field to those marker particles situated at distance $2h$ from the SPH ones, taking into account that:
  - As we saw in section 4.1, mesh pressure values are actually p/ρ.
  - To use the SPH method we need the $dt/2$ advanced velocities for all the particles involved and this forces us to make first the MAC calculation and then to estimate the "advanced" values of the marker particles (in fact, now they are delayed) using the new velocities $\vec{u}^{(n+1)}$:

$$\begin{aligned} \vec{a}^{(n)} &= \frac{\vec{u}^{(n+1)} - \vec{u}^{(n)}}{dt} \\ \vec{u}^{(n+1/2)} &= \vec{u}^{(n)} + \vec{a}^{(n)} dt/2. \end{aligned} \tag{10}$$

However, marker particles with such a treatment do not maintain the distances between themselves as SPH particles do, causing instabilities when these particles become SPH (entering into the SPH zone). Let us denote MIXB this kind of particles.

- To avoid these instabilities, we create MIXA, a new level of particles between SPH and MIXB, $2h$ high and in such a way that their movement (at least near the SPH particles) does respect this distance.
  The idea is to move them, with a linear combination of both simulation methods, being stronger the SPH behaviour as we approach the SPH/MIXA limit, l.
  Let us define $\beta_i = (l - y_i)/2h \in [0,1]$, being y_i the vertical position of the MIXA particle i. We want the new velocity of i to be:

$$\vec{u}^{(n+1)} = \beta\, \vec{u}^{(n+1)}_{MAC} + (1-\beta)\, \vec{u}^{(n+1)}_{SPH}, \qquad (11)$$

  what can be carried out actualizing the SPH part of the combination at the end of each MAC-SPH mixed model iteration and the MAC part in the next iteration, after the MAC calculation and before the SPH one.

We have to be careful with the actualization of pressure and density for MIXA and MIXB particles, that must be done just after the SPH calculation (their old values are needed in it). Although we have given them the treatment in (11), there is the possibility of calculating one of them this way and then, the other by means of the state equation (6). This could obtain better results and still must be studied.

To summarize we have 4 kinds of particles (see figure 3): SPH, MIXA (moved with a linear combination of both methods), MIXB (MAC marker particles that, besides, carry the information needed to calculate the SPH part of the MIXA ones) and MAC. Moreover, we must respect the distances between the different levels of particles, as well as the calculation order of the MAC/SPH contributions, to obtain a good interaction between the two simulation models.

4.3. Results

Figure 4 shows some snapshots of a drop falling into a receptacle, where the fluid is initially at rest. We see particles, velocities and both together. The total number of particles in this example is $70,883$ and, initially, $39,673$ of them need the SPH treatment (SPH and MIXA).

In figure 6 we see some images of the evolution of this example. The smooth surface is a point based surface rendered with POV-Ray (www.povray.org), adapting the ray-surface intersections according to the method of I. Wald [WS05].

Figure 5 is made up of some snapshots of an externally manipulated example. At the beginning, the fluid is at rest. Then, two cubes are raised by means of forces directly applied to the corresponding particles and stopped some time later. During the fall, pressure and density evolution makes the cubes change into spheroids. In this case, the total number of particles is $68,662$ and, initially, $37,452$ of them need the SPH treatment.

The general behaviour of the fluid is very natural (in Fluid Animation we look for a fast fluid-like behaviour although not physically exact) and the improvement in simulation speed is quite noticeable: although the running times are not competitive because the algorithm is not totally optimized (e.g. improvements can be done in the neighbours search and the Poisson Equation solution) and it was executed in an ordinary laptop, in the first example there is a saving of 32% in averaged iteration running time compared with a pure SPH execution and a saving of 37% in the second example.

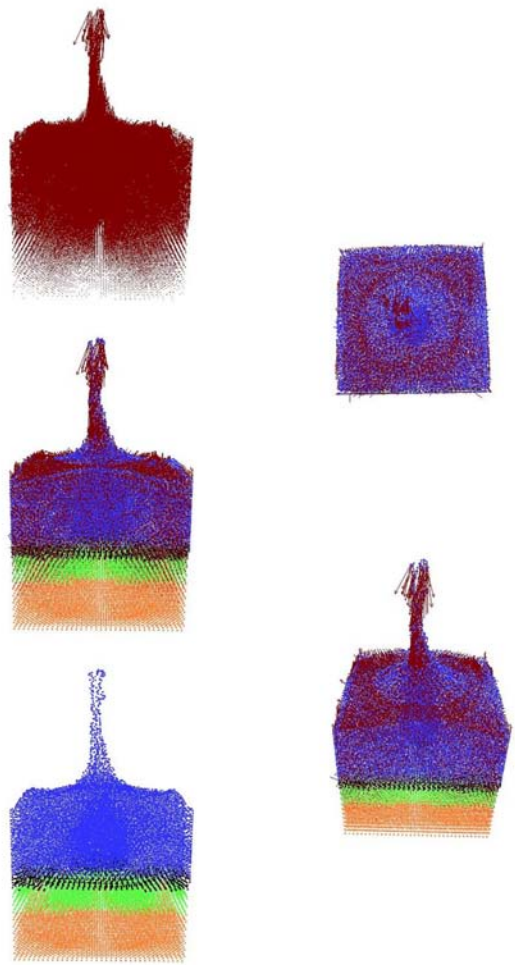

Figure 4: *Snapshots of falling fluid drop. Particles and velocities are represented.*

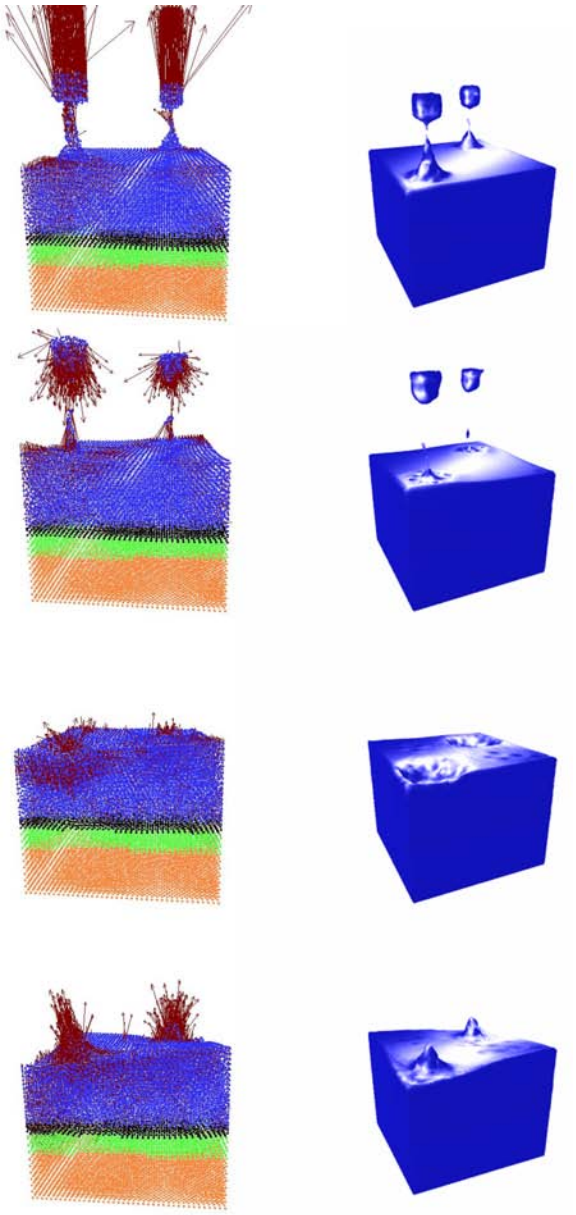

Figure 5: *Example directly modified by the animator. Two cubes are raised and stopped some time later to then let them fall.*

5. Conclusions and future work

We have presented a new method where MAC and SPH interact with the aim of minimizing the computational effort in problems where different parts of the domain have very different characteristics, allowing at the same time some external manipulation of the scenes.

In examples in which a great volume of fluid needs to be present but there is not any important feature needing high-level detail in it the results can be noticed immediately, achieving, in some cases, up to 40% of saving in averaged iteration running time (regarding the pure SPH model). Nevertheless, associated limitations to the original methods are kept, like the coupling of grid resolution and simulation of turbulent features of the MAC.

On the other hand, making the height for the interaction layer being dependent on some fluid parameters, like the fluid quantity or velocity, avoids its choice in advance and can extend the amount of problems to be treated with this method, for example, to problems in which containers are filled or emptied. Furthermore, if the detection of the superficial zones to be treated with SPH could be done automatically, the method would be more flexible and could be very useful for a wider range of problems. The inclusion of moving solid objects is also an issue for further research.

Summarizing, our MAC-SPH mixed model is a new simulation method that preserves part of the calculation speed of the MAC model and the high-level detail simulation of SPH on the surface, without losing the possibility of being used by animators to modify the fluid behaviour that SPH has. All these characteristics can turn it a very useful tool for the Fluid Simulation for the Computer Animation world.

Acknowledgements

Work partially supported by the Rovira i Virgili University and CICYT TIN-2004-08065-C02-01. Surface visualization was developed in close collaboration with the Visualization and Multimedia Lab, Department of Informatics, University of Zürich.

References

[CCLY02] Chiu, C., Chuang, J., Lin, C., Yu, J., *Modeling Highly-Deformable Liquid*, International Computer Symposium, 2002.

[CP04] Cleary, P., Prakash, M., *Discret-Element Modelling and Smoothed Particle Hydrodynamics: Potential in the Environmental Sciences*, Phil. Trans. R. Soc. Lond. A (2004) 362, 2003-2030, 2004.

[DC99] Desbrun, M., Cani, M., *Space-Time Adaptive Simulation of Highly Deformable Substances*, Technical Report 3839, INRIA, 1999.

[EMF02] Enright, D., Marschner, S., Fedkiw, R., *Animation and Rendering of Complex Water Surfaces*, Proceed. of the 29th Annual Conference on Computer Graphics and Interactive Techniques, 736-744, ACM Press, 2002.

[FF01] Foster, N., Fedkiw, R., *Practical Animation of Liquids*, ACM SIGGRAPH 2001, 15-22, 2001.

[FM96] Foster, N., Metaxas, D., *Realistic Animation of Liquids*, Graphical Models and Image Processing 58, 471-483, 1996.

[FM97] Foster, N., Metaxas, D., *Controlling Fluid Animation*, Computer Graphics International 97, 178-188, 1997.

[GM77] Gingold, R., Monaghan, J., *Smoothed Particle Hydrodynamics: Theory and Application to Non-Spherical Stars*, Royal Astronomical Society, Monthly Notices 181, 375-389, 1977.

[GDN98] Griebel, M., Dornseifer, T., Neunhoeffer, T., *Numerical Simulation in Fluid Dynamics-A Practical Introduction*, SIAM, 1998.

[Had00] Hadjidimos, A., *Successive Overrelaxation (SOR) and Related Methods*, Journal of Computational and Applied Mathematics 123, 177-199, 2000.

[HW65] Harlow, F., Welch, J., *Numerical Calculation of Time-Dependent Viscous Incompressible Flow of Fluid with a Free Surface*, The Physics of Fluids 8, 2182-2189, 1965.

[Luc77] Lucy, L., *A Numerical Approach to the Testing of the Fission Hypothesis*, The Astronomical Journal, vol 82, num 12, 1977.

[Mon92] Monaghan, J., *Smoothed Particle Hydrodynamics*, Annu. Rev. Astron. Astrophys. 30, 543-74, 1992.

[MK99] Monaghan, J., Kos, A., *Solitary Waves on a Cretan Beach*, Journal of Waterway, Port, Coastal and Ocean Eng., 1999.

[MK00] Monaghan, J., Kos, A., *Scott Russell's Wave Generator*, Physics of Fluids, vol 12, num 3, 2000.

[MCG03] Müller, M., Charypar, D., Gross, M., *Particle-Based Fluid Simulation for Interactive Applications*, Proc. Eurographics/ SIGGRAPH Sym., 154-159, 2003.

[MST*04] Müller, M. et al., *Interaction of Fluids with Deformable Solids*, Journal of Computer Animation and Virtual Worlds, vol 15, num 3-4, 159-171, 2004.

[PTB*03] Premože, S. et al., *Particle-Based Simulation of Fluids*, EUROGRAPHICS 2003, vol 22, num 3, 2003.

[Sta99] Stam, J., *Stable Fluids*, ACM SIGGRAPH 99, 121-128, 1999.

[Sta03] Stam, J., *Real-Time Fluid Dynamics for Games*, Proceedings of the Game Developer Conference, March 2003, 2003.

[TFK*03] Takahashi, T. et al., *Realistic Animation of Fluid with Splash and Foam*, EUROGRAPHICS 2003, vol 22, num 3, 2003.

[WS05] Wald, I., Seidel, H., *Interactive Ray Tracing of Point-Based Models*, Eurographics Symposium on Point-Based Graphics 2005, 1-8, 2005.

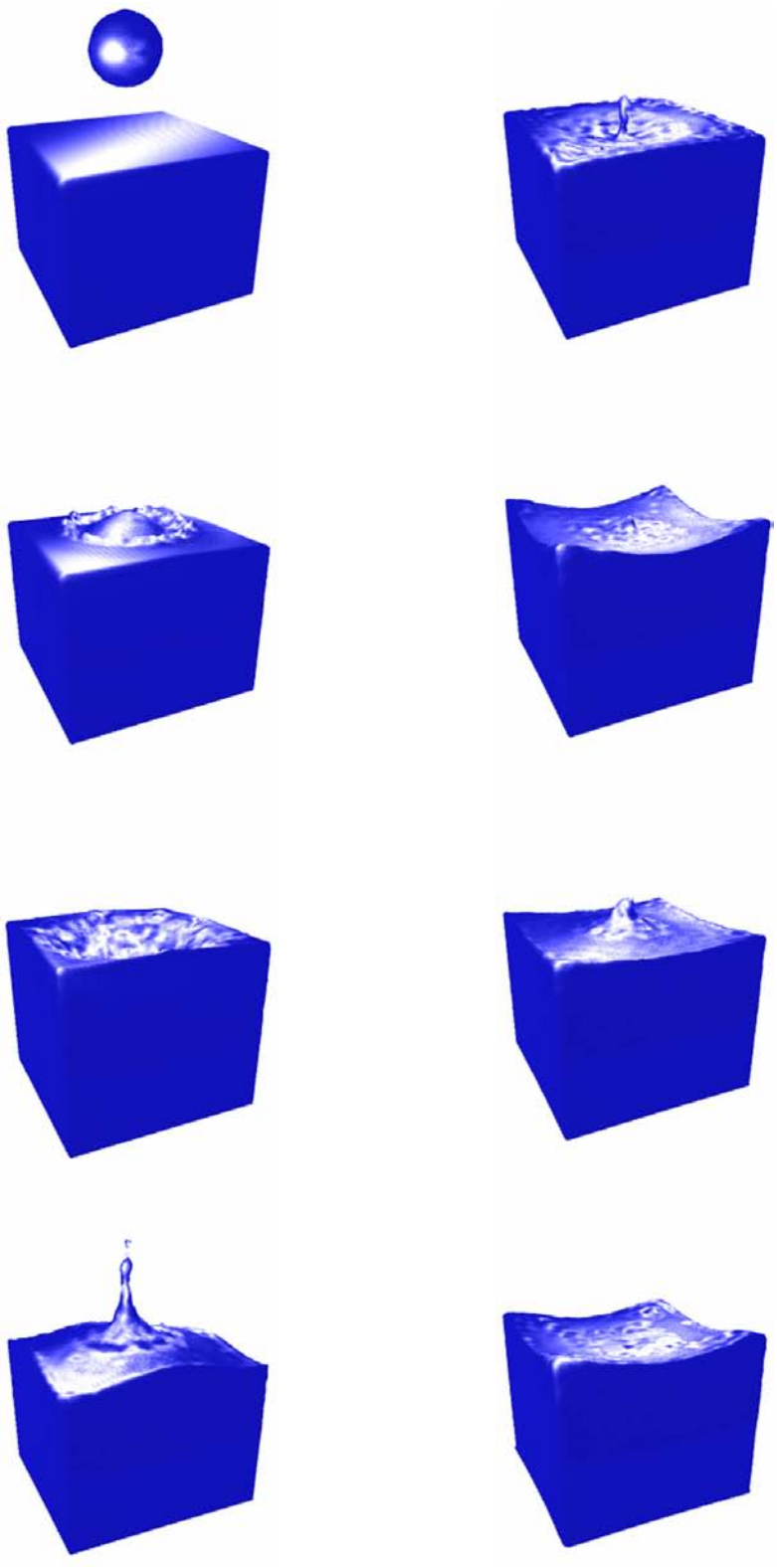

Figure 6: *Evolution of a falling fluid drop.*