Discrete Models for Animating Gas-Liquid and Fluid-Surface Interactions

Sicilia F. Judice Gilson A. Giraldi
National Laboratory of Scientific Computing
Ave Getúlio Vargas, 333, 25651-075, Petrópolis, RJ, Brasil
{sicilia,gilson}@lncc.br

Abstract

The past two decades showed a rapid growing of physically-based modeling of fluids and solid for computer graphics applications. In particular, techniques in the field of Computational Mechanics have been applied for realistic animation of systems that involve gas-fluid and fluid-surface interaction for computer graphics and virtual reality applications. The main goal of our work is the development of a particle based framework to create realistic animations of such systems. Specifically, we model and simulate the gas through a Lattice Gas Cellular Automata (LGCA), the liquid by Smoothed Particle Hydrodynamics (SPH) method and the surface through Mass-Spring systems. LGCAs are discrete models based on point particles that move on a lattice, according to suitable rules in order to mimic a fully molecular dynamics. SPH is a Lagrangian, meshfree method for numerical simulation which is based on particle systems and interpolation theory. Mass-Spring systems may be geometrically represented by regular meshes which nodes are treated like mass points and each edge acts like a spring. When combining these methods (LGCA, SPH and Mass-Spring), we get the advantage of the low computational cost of cellular automata and mass-spring systems and the realistic fluid dynamics inherent in the SPH to develop a new animating framework for computer graphics applications. In this work, we discuss the theoretical elements of our proposal and present some preliminary experimental results.[1]

1. Introduction

Physically-based techniques for the animation of natural elements like fluids (gas or liquids), flood, elastic, plastic and melting objects, among others, have taken the attention of the computer graphics community [25, 36, 11]. In particular, techniques in the field of Computational Fluid Dynamics (CFD) have been applied for fluid animation in applications that involve fluid-fluid and fluid-surface interaction [35, 28, 19, 8, 31].

A common approach in this area relies on top down viewpoints that use 2D/3D mesh based techniques in conjunction with fluid/solid equations [8]. Other possibility is to apply Mass-Spring systems to model the elastic object [19], SPH to simulate fluids [28, 34] and Lattice Gas Cellular Automata (LGCA) techniques to simulate gas systems [21, 3].

Mass-Spring models are well suited to animation due to their flexibility to handle non-rigid solid properties, its easy manipulation and implementation. Besides, Mass-Spring models can be faster then their counterpart in continuous mechanics, and so, more suitable for real time applications specially when GPU capabilities are explored [33]. The LGCAs are *bottom up* discrete models, based on point particles that move on a lattice, according to suitable and local rules, that mimics a fully particle dynamics [17]. These methods are cheaper and more stable than the traditional ones for fluid simulation, because there is no need to solve Partial Differential Equations (PDEs) to obtain a high level of description [3]. SPH is a Lagrangian, meshfree method for numerical simulation which is based on particle systems and interpolation theory.

In this paper we propose a framework for animation of fluid-fluid and fluid-surface interaction based on Mass-Spring systems, LGCA techniques and SPH. We apply a traditional Mass-Spring model described in [33] that can be used for both surface and deformable solid modeling. Among the LGCA models [10] we apply the FHP one which was introduced by Frisch, Hasslacher and Pomeau [18] in 1986. The traditional FHP is a model for two-dimensional fluids that describes the motion of particles traveling in a discrete space and colliding with each other. The space is discretized in a hexagonal lattice. We proposed a three dimensional fluid simulation model based on the FHP and interpolation techniques. The SPH implementation follows the method presented in [34]. Finally, render-

1 Full paper of M.Sc thesis.

ing techniques must be applied to ensure the desired level of realism or visual effect. In this step we apply the photon map [27] method because it is able to easily generate area light sources, color bleeding, soft shadows, indirect illumination and caustics, once it collects illumination information of the scene by a pre-trace from light sources.

The main contributions of this work are the $3D$ fluid simulation model based on the FHP and the development of a particle based framework, combining Mass Spring systems, LGCA techniques and SPH, to create realistic animations of systems that involve gas-fluid and fluid-surface interaction for computer graphics and virtual reality applications.

The paper is organized as follows. Section 2 gives a survey of related works. Section 3 offers some fundamental concepts in Navier-Stokes equations. Section 4 describes the SPH method. Section 5 describes the FHP model and our extension for $3D$. In section 6 we presentes the Mass-Spring model used. The rendering model is discussed on section 7. Section 8 describes the proposed framework and some preliminary results. In section 9 we present the conclusions.

2. Related Works

The main focus of this work is the animation of gas-fluid and fluid-surface interaction. The former can be modeled by a FHP method in which there are two kinds of particles - gas particles and liquid particles [2]. Then, Boltzmann approximation is applied to calculate the surface tension as a function of population density.

In [43] an interface model is derived for two-phase flows with surface tension, density and viscosity differences between the phases. The derivation starts from the balance equations for a sharp interface and uses an ensemble averaging procedure on an atomic scale to obtain a diffuse interface version of the equations [12, 43]. Other approach models the interfacial tension is to detect the interfacial surface (boundary between the gas and liquid phases) and then to compute the interaction forces following some heuristic.

Fluid-surface interaction includes: (1) Representation of the object geometry; (2) The modeling of mechanical behavior of elastic surfaces; (3) A suitable model for fluid simulation; (4) A model for interaction of the flow with the object; (5) Rendering issues.

Bidimensional manifolds can be represented by using implicit surfaces [47], triangulated meshes or subdivision surfaces with local parameterization for representation [22, 42]. The mechanical behavior of deformable surfaces (item (2)) can be described by continuum elasticity models that describes how the objects deform under applied forces. Other possibility is to apply discrete models, based on Mass-Spring systems [19, 33]. In this case, the object geometry is represented by a mesh and its nodes are treated like mass points while each edge acts like a spring connecting two adjacent mass points. It is known that methods that are based on the continuum mechanics are more realistic than their discrete counterparts [44]. However, Mass-Spring models can be faster, and so, more suitable for real time applications [44, 14].

The item (3) involves numerous works that can be coarsely classified in non-physically and physically based models [25, 11]. Our work belongs to the later class, which can be subdivided in PDEs and Lattice based techniques [17, 25].

PDEs methods involve continuous fluid equation, like the Navier-Stokes ones, and numerical techniques based on discretization approaches that can be Lagrangian Smoothed Particle Hydrodynamics (SPH) [29], method of characteristics [41], Moving-Particle Semi-Implicit [37] or Eulerian (Finite Element) ones [16].

Lattice based techniques, like HPP, FHP and Lattice Boltzmann methods, work following a different viewpoint [17, 5]. For instance, in the case of HPP and FHP, instead of applying continuous mechanics (and, consequently, PDEs) principles, they model the system as a set of point particles, that move on a lattice, interacting according to suitable and simple rules in order to mimics a fully dynamics [17]. These are *bottom up* approaches in which the macroscopic behavior of the fluid can be recovered by multiscale techniques [17].

Lattice models have a number of advantages over more traditional numerical methods, particularly when fluids mixing and phase transitions occur [38]. The simulation is always performed on a regular grid and can be efficiently implemented on a massively parallel computer. Solid boundaries and multiple fluids can be introduced in a straightforward manner and the simulation is performed equally efficiently, regardless of the complexity of the boundary or interface [7]. In addition there are not numerical stability issues because the evolution follows integer arithmetic. However, system parameterization (viscosity, for example) is a difficult task in such lattice models and they are less realistic than PDE based models.

The item (4), interaction between deformable manifolds and fluids, can be addressed by hybrid methods in which the fluid is a continuum medium, simulated by Navier-Stokes plus SPH or grid based techniques, and the surface is represented as a discrete one [19, 40, 4]. These approaches deal with the specific problem of preventing the leaking of fluid across the polygonal surface [22, 6]. In addition, fluid flows can be simulated on $2D$ manifolds represented by (continuous) subdivision surfaces that have a natural quad patch parametrization [42]. Interaction between Navier-Stokes fluids and digital terrain models is another subclass of fluid-surface interaction [46]. Besides, a hybrid particle and implicit surface approach to simulating water was proposed in [15], which led to the particle level set method of [13].

Finally, visualization and rendering techniques must be applied to ensure the desired level of realism or visual effect. Photo-realistic rendering can properly account through several algorithms including path tracing, bidirectional path tracing [23], Metropolis light transport [45], and photon map [27]. The interested reader is also encouraged to browse interesting reviews in this area [1, 25].

In this paper, the proposed method for gas-liquid interaction is a hybrid one in the sense that the gas is simulated by a discrete system and the fluid is animated by Navier-Stokes plus SPH. On the other hand, the fluid-surface interactions model is a discrete one because the fluid is FHP-based and the surfaces are represented by polygonal meshes, or Mass-Spring systems in the case of deformable manifolds.

3. Navier-Stokes for Fluid Animation

The majority fluid models in computer graphics follow the Eulerian formulation of fluid mechanics that is based on a top down viewpoint of the nature: the fluid is considered as a continuous system subjected to Newton's and conservation Laws as well as state equations connecting the macroscopic variables of pressure P, density ρ and temperature T. So, the mass conservation, also called continuity equation, is given by [24]:

$$\frac{\partial \rho}{\partial t} + \nabla \cdot (\rho \vec{u}) = 0 \tag{1}$$

The linear momentum conservation equation, also called Navier-Stokes, can be obtained by applying the third Newton's Law to a volume element dV of fluid. For incompressible flows it can be written as [24]:

$$\rho \left(\frac{\partial \vec{u}}{\partial t} + \vec{u} \cdot \nabla \vec{u} \right) = -\nabla P + \mathbf{F} + \mu \nabla^2 \vec{u}, \tag{2}$$

$$\nabla \cdot \vec{u} = 0. \tag{3}$$

where $\mathbf{F}$ is an external force field and μ is the viscosity of the fluid. Also, we may need an additional equation for the pressure field. This is a state equation which ties together all of the conservation equations for continuum fluid dynamics and must be chosen to model the appropriate fluid (*i.e.* compressible or incompressible). In the case of liquids, the pressure P is temperature insensitive and can be approximated by $P = P(\rho)$. Morris in [32] proposed an expression that have been used for fluid animation also [34]:

$$P = c^2 \rho, \tag{4}$$

where c is the speed of sound in this fluid [39]. Equations (2)-(4) need initial conditions $(\rho(t=0,x,y,z), \vec{u}(t=0,x,y,z))$ as well as boundary conditions, like the usual no-sleep one: $\vec{u}|_S = 0$.

4. Smooth Particle Hydrodynamics

In this section we follow the references [39, 30]. The two fundamental elements in the Smoothed Particle Hydrodynamics method are an interpolation kernel W and a particle system that represents a discrete version (sample) of the fluid. The kernel estimate of a scalar quantity $A(\mathbf{r})$ is defined by:

$$\langle A(\mathbf{r}) \rangle = \int_{Space} A(\mathbf{r}') W(\mathbf{r} - \mathbf{r}', h) \, d\mathbf{r}', \tag{5}$$

where the function $W(\mathbf{r} - \mathbf{r}', h)$ is an (interpolation) kernel which must satisfies the following properties [30]:

1) Volume conservation:

$$\int_{Space} W(\mathbf{r} - \mathbf{r}', h) \, d\mathbf{r}' = 1, \tag{6}$$

2) The kernel W should satisfy the Dirac delta function in the limit:

$$\lim_{h \to 0} W(\mathbf{r} - \mathbf{r}', h) = \delta(\mathbf{r} - \mathbf{r}'). \tag{7}$$

If we take a sampling of A then the $A(\mathbf{r}')$ in equation (5) will be known only at a discrete set of N points $\mathbf{r}_1, \mathbf{r}_2, ..., \mathbf{r}_N$. Hence, thorough properties (6)-(7) it is possible to show that [39]:

$$\langle A(\mathbf{r}) \rangle = \sum_{j=1}^{N} \frac{m_j}{\rho(\mathbf{r}_j)} A(\mathbf{r}_j) W(\mathbf{r} - \mathbf{r}_j, h). \tag{8}$$

$$\langle \nabla_{\mathbf{r}} A(\mathbf{r}) \rangle = \sum_{j=1}^{N} \frac{m_j}{\rho(\mathbf{r}_j)} A(\mathbf{r}_j) \nabla_{\mathbf{r}} W(\mathbf{r} - \mathbf{r}_j, h). \tag{9}$$

An analogous expression can be obtained by the Laplacian. From equation (9) we can observe that there is no need for a mesh to compute spatial derivatives. With equations (8) and (9), we are ready to write the discrete version of the fluid equations of section 3. The smoothing lengh h is the width of the kernel and defines the distance at which a particle interacts with other particles. It is equivalent to the size of a grid-cell in finite difference methods.

We rewrite the terms of the Navier-Stokes equation (2), using this approach, as:

$$\vec{f}_i^{press} = -\sum_j m_j \frac{p_i + p_j}{2\rho_j} \vec{\nabla} W(\mathbf{r}_i - \mathbf{r}_j, h)$$

$$\vec{f}_i^{visc} = \mu \sum_j m_j \frac{\vec{v_j} - \vec{v_i}}{\rho_j} \vec{\nabla}^2 W(\mathbf{r}_i - \mathbf{r}_j, h)$$

$$\vec{f}_j^{grav} = \rho_j \vec{g_j}$$

where the $\vec{f}_i^{press}$ and $\vec{f}_i^{visc}$ are the pressure and viscosity forces. Only the gravity force $\vec{f}_i^{grav}$ is considered as external force. The density at each particle can be found from the following equation:

$$\rho_i = \sum_{j=1}^{N} \rho_j m_j W(\mathbf{r}_i - \mathbf{r}_j, h).$$

In order to have stability we adopt the following kernels [34]:

$$W_{grav}(r,h) = \frac{315}{64\pi h^9} \begin{cases} (h^2 - r^2)^3, \ 0 \le r \le h, \\ 0, \qquad \text{otherwise} \end{cases}$$

$$W_{press}(r,h) = \frac{15}{\pi h^6} \begin{cases} (h - r)^3, \ \ 0 \le r \le h, \\ 0, \qquad \text{otherwise} \end{cases}$$

$$W_{visc}(r,h) = \frac{15}{2\pi h^3} \begin{cases} -\frac{r^3}{2h^3} + \frac{r^2}{h^2} + \frac{h}{2r} - 1, \ 0 \le r \le h, \\ 0, \qquad \text{otherwise} \end{cases}$$

5. FHP

The FHP was introduced by Frisch, Hasslacher and Pomeau [18] in 1986 and is a model of a two-dimensional fluid. It can be seen as an abstraction, at a microscopic scale, of a fluid. The FHP model describes the motion of particles traveling in a discrete space and colliding with each other. The space is discretized in a hexagonal lattice.

The FHP particles move in discrete time steps, with a velocity of constant modulus, pointing along one of the six directions of the lattice. The dynamics is such that no more than one particle enters the same node at the same time with the same velocity. This restriction is the *exclusion principle*; it ensures that six Boolean variables at each lattice node are always enough to represent the microdynamics.

The velocity modulus is such that, in a time step, each particle travels one lattice spacing and reaches a nearest-neighbor node. When exactly two particles enter the same node with opposite velocities, both of them are deflected by 60 degrees so that the output of the collision is still a zero momentum configuration with two particles. The deflection can occur to the right or to the left, indifferently, as shown in Figure 1. For symmetry reasons, the two possibilities are chosen randomly, with equal probability.

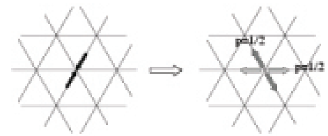


Figure 1. The two-body collision in the FHP. Source [21]

When exactly three particles collide with an angle of 120 degrees between each other, they bounce back to where they come from (so that the momentum after the collision is zero, as it was before the collision). Both two- and three-body collisions are necessary to avoid extra conservation laws. For all other configurations no collision occurs and the particles go through as if they were transparent to each other.

The full microdynamics of the FHP model can be expressed by evolution equations for the occupation numbers defined as the number, $n_i(\mathbf{r}, t)$, of particle entering node $\mathbf{r}$ at time t with a velocity pointing along direction $\vec{c}_i$, where $i = 1, 2, \ldots, 6$ labels the six lattice directions. The numbers n_i can be 0 or 1. We also define the time step as Δ_t and the lattice spacing as Δ_r. Thus, the six possible velocities $\vec{v}_i$ of the particles are related to their directions of motion by

$$\vec{v}_i = \frac{\Delta_r}{\Delta_t} \vec{c}_i. \tag{10}$$

The microdynamics of a LGCA is written as

$$n_i(\mathbf{r} + \Delta_r \vec{c}_i, t + \Delta_t) = n_i(\mathbf{r}, t) + \Omega_i(n(\mathbf{r}, t)) \tag{11}$$

where Ω_i is called the collision term [9].

Now we propose an extension to 3D, using the 2D FHP model explained above. In practice, the system of a given cellular automata rule cannot deal with an infinite lattice, it must be finite and have boundaries [9]. So, first we must define a domain in the three dimensional space, where the gas can evolve. Then, our proposal consists of regularly distribute planes along the x and z axis, like in Figure 2. Each of these planes is a system whose cellular automata rule is the 2D FHP.

Once simulated the two dimensional FHP in each plane independently, we perform a simple interpolation to generate a $2D$ macroscopic flow in each plane. In this step, we add new nodes to the FHP grid in order to complete a rectangular grid em each plane, as pictured in Figure 3.

Following the usual definition of statistical mechanics, we compute the macroscopic density in each node (x_i, y_i, z_i) of the plane $x = x_i$ through the expression:

$$\rho_x(x_i, y_i, z_i, t) = \sum_{j \in V} \sum_{k=1}^{6} n_k(x_j, y_j, z_j, t), \tag{12}$$

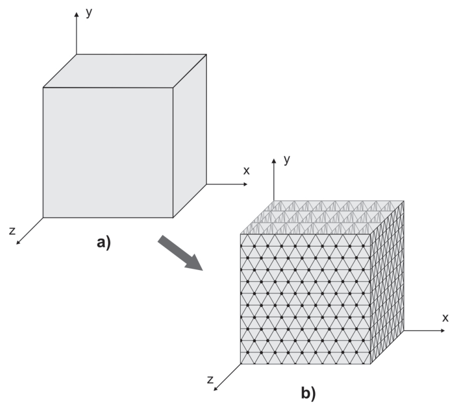


Figure 2. (a) Domain in 3D Space and (b) the distribution of 2D FHP planes.

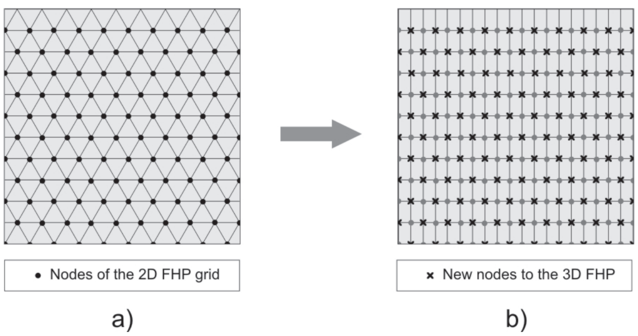


Figure 3. Extend the (a) FHP grid to generate a (b) rectangular mesh.

where V is a neighborhood of point (x_j, y_j, z_j).

An analogous expression can be used for the plane $z = z_i$. Now, we must render a $3D$ macroscopic flow. We shall observe that each node (x_i, y_i, z_i) in Figure 2 belongs to the planes $x = x_i$ and $z = z_i$. So, the $3D$ density (or velocity) can be finally obtained through a simple mean of the corresponding values in the FHP planes, that means:

$$\rho(x_i, y_i, z_i, t) = \frac{\rho_x(x_i, y_i, z_i, t) + \rho_z(x_i, y_i, z_i, t)}{2}. \tag{13}$$

6. Mass-Spring Model

In this section we follow the reference [33]. The Mass-Spring is a discrete model in which surfaces are represented by polygonal meshes. The surface nodes works as masses and the edges defines the linear springs with damping. So, given a particle i with mass m_i and position vector $\mathbf{r}_i$, the force system is composed by the elastic ($\vec{f}_i^{elastic}$), gravitational ($\vec{f}_i^{grav}$) and damping ($\vec{f}_i^{damp}$) forces, defined respectively, by:

$$\vec{f}_i^{elastic} = \sum_{j=1}^{4} k_{ij}\left(l_{ij} - \|\mathbf{r}_i - \mathbf{r}_j\|\right)\frac{(\mathbf{r}_i - \mathbf{r}_j)}{\|\mathbf{r}_i - \mathbf{r}_j\|}, \tag{14}$$

where k_{ij} is the stiffness of the spring linking the nodes $\mathbf{r}_i$ and $\mathbf{r}_j$ and l_{ij} the spring rest length;

$$\vec{f}_i^{grav} = m_i \vec{g}, \tag{15}$$

$$\vec{f}_i^{damp} = \gamma_i \dot{\mathbf{r}}_i, \tag{16}$$

where $\vec{g}$ is the gravity field, γ_i is the damping factor and $counter(i)$ holds the number of particles accumulated in the corresponding position. Following Newton's Laws, we get the following evolution equation:

$$m_i \ddot{\mathbf{r}}_i = \vec{f}_i^{elastic} + \vec{f}_i^{damp} + \vec{f}_i^{grav}, \tag{17}$$

This system of ordinary differential equations can be efficiently solved by the Verlet integration technique [33]:

$$\mathbf{r}_i(t+h) = 2\mathbf{r}_i(t) - \mathbf{r}_i(t-h) + \ddot{\mathbf{r}}_i(t)\, h^2. \tag{18}$$

7. Rendering through Photon Map

Originally developed for global illumination simulation in scenes without participating media [26], Photon Map is a two-pass method where the first pass is the construction of structures to store the light information (photon maps) and the second is rendering using these information. The construction of the photon maps consists of photons emitted from the light sources and traced through the scene using photon tracing. Along the time evolution, if a photon hits a nonspecular surface, it is stored in the photon map.

Jensen [26] proposed the use of two photon maps: a caustics photon map and a global photon map. The caustics photon map stores all photons that have been traced from the light source through a number of specular reflections or transmissions before intersecting a diffuse surface, and the global photon map contains all photons representing indirect illumination on a nonspecular surface. In the rendering pass, the caustics photon map is used to render caustics directly and the global photon map is used to limit the number of reflections traced by the distribution ray tracer and to sample indirect illumination more efficiently.

It is possible to estimate radiance at any given surface position x using the photon map. By locating the n photons with the shortest distance to x it is possible to estimate the photon density around x [27]:

$$L_r(x,\vec{\omega}) \approx \sum_{p=1}^{n} f_r(x,\vec{\omega_p'},\vec{\omega})\frac{\Delta\Phi_p(x,\vec{\omega_p'})}{\pi r^2}, \qquad (19)$$

where f_r is the bidirectional reflectance distribution function, r is the distance to the nth nearest photon and $\Delta\Phi_p$ the flux carried by each photon p in direction $\vec{\omega_p'}$. This approach can be seen as expanding a sphere centered at x until it contains n photons.

In [27], Jensen proposed an extension of the photon map method to be able to use in scenes with participating media, where photons can be scattered and absorbed by the media. To efficiently render the medium, it is necessary to store information about these scattering events. This storage of the photons occurs explicitly in the volume, given several advantages: the photons can be concentrated where necessary to represent intense illumination, the media do not have to be discretized and anisotropic scattering can be handled by storing the incoming direction of each photon.

The relationship between the density of the photons and the illumination is different on surfaces and in volumes. Then, the photons must be separated when the photon map is queried for information about the incoming flux. A separate volume photon map for the photons that are scattered in participating media was introduced in [27], and it is used to compute the illumination inside a participating medium while the global photon map is used, as before, to compute the illumination on surfaces.

Just like the original method, the first pass consists of building the photon maps using photon tracing. When a photon is traced within a participating medium, it can either pass unaffected through the medium, or an interaction can occurs (be scattered or absorbed). If the photon interacts with the medium, and does not come directly from a light source, it is stored in the photon map. The cumulative probability density function, $F(x)$, expressing the probability of a photon interacting with a participating medium at position x is:

$$F(x) = 1 - \tau(x_s, x) = 1 - e^{-\int_{x_s}^{x} \kappa(\xi)d\xi}, \qquad (20)$$

where x_s is the point at which the photon enters the medium. The transmittance $\tau(x_s, x)$ is computed using ray marching.

There is difference when calculating the density in the participating media. The density on a surface is computed using the projected area and the density in a medium is computed using the full volume, as shown in Figure 4. Once done the storage pass, we can use the photons stored in the volume photon map to compute an estimate of the inscattered radiance:

$$L_i(x,\vec{\omega}) \approx \frac{1}{\sigma(x)} \sum_{p=1}^{n} f(x,\vec{\omega_p'},\vec{\omega})\frac{\Delta\Phi_p(x,\vec{\omega_p'})}{\frac{4}{3}\pi r^3}, \qquad (21)$$

where Φ is the in-scattered flux. Using (21) we can compute a radiance estimate at any given point inside a participating medium.

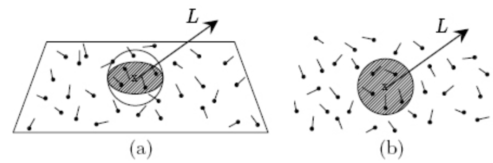


Figure 4. Radiance estimate for: (a) surfaces and (b) volumes. Source [27]

8. Proposal and Experimental Results

In this section we firstly present some results for the three-dimensional FHP rendered with a volume rendering technique, the single scattering method, available in the PBRT library (http://www.pbrt.org/). Then, we show two-dimensional results obtained by our team and discuss their extensions to $3D$ in the context of this work. The corresponding videos can be found in: `http://www.lncc.br/~sicilia/assuntosAfins_pesquisa.htm`.

8.1. Three-Dimensional FHP

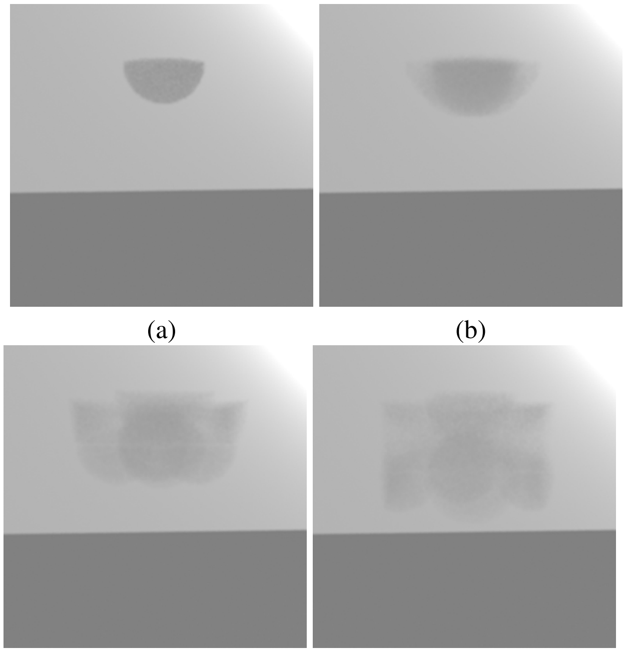
Firstly, we highlight the simplicity and low computational cost of the FHP. Figure 5 shows an initial configuration with a volume of gas in the domain (semi-sphere in the top of the box).

The initialization is a very simple process: Firstly, the grid nodes inside the sphere are detected. Then, for each node detected, the algorithm randomly chose the quantity of particles and its directions.

Figure 6 shows a similar system, with the same initialization, but now using a solid sphere out of the fluid domain. It is used only to highlight transparency effects.

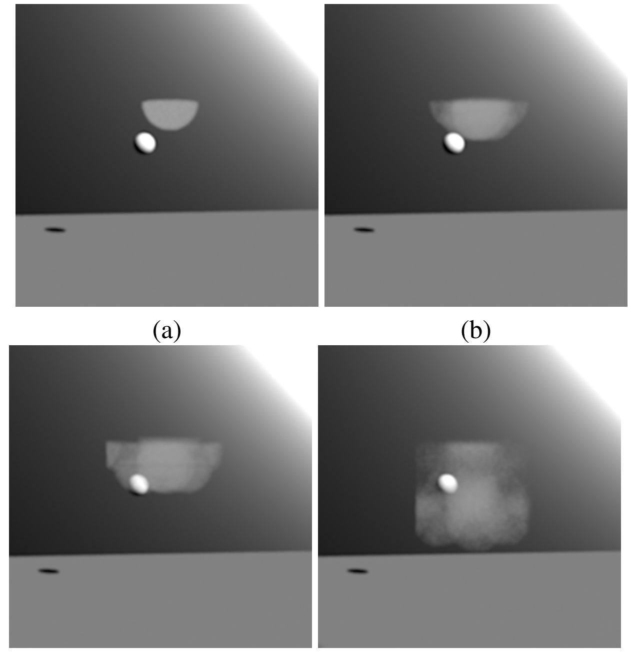
8.2. Gas-Fluid and Fluid-Surface Interactions

In [20] we combine the FHP and SPH to animate $2D$ two-phase systems composed by a gas simulated through FHP and a liquid modeled by SPH. The first point is how to model the interactions between the two phases in the interfacial area. For simplicity, in [20] we proceed as follows: given a point $\vec{r}$ in the interfacial area at a time t, we take a neighborhood and compute the particles mean velocity $\vec{u}_m$ through an expression similar to equation (12). Then, we set



(a) (b) (c) (d)

Figure 5. Configurations of the volume gas: (a) at initial step; (b) after 10 steps; (c) after 20 steps; and (d) after 50 steps of simulation.



(a) (b) (c) (d)

Figure 6. Configurations of the volume gas, with a solid sphere out of the system: (a) at initial step; (b) after 10 steps; (c) after 20 steps; and (d) after 70 steps of simulation.

the interaction force as:

$$\vec{F}_{int} = \tau \vec{u}_m(\mathbf{r}, t), \tag{22}$$

where τ is force scale parameter. This approach is more intuitive than the other ones [17] and allows the generation of interesting visual effects. Figure 7 shows examples of gas-liquid systems with the forces in the interfacial area given by expression (22). This figure shows a stream of 1000 particles with vertical macroscopic velocity and the liquid (blue) just before interaction.

The extension of this methodology to $3D$ depends on the following steps: (1) Three-dimensional FHP and SPH; (b) A model for the surface area and surface tension; (b) A suitable rendering technique.

Our team have implemented the SPH $3D$ presented in [34] and the three-dimensional FHP was described in section 5. The interfacial surface/tension model will also follows that reference. The interfacial surface is defined through the smoothed color field given by:

$$c_s(\mathbf{r}) = \sum_j m_j \frac{1}{\rho_j} W(\mathbf{r} - \mathbf{r}_j, h),$$

where m_j is the mass and ρ_j is the density of the particle j and W is an interpolation kernel of size is h [30]. A point belongs to the interface between the fluid and the gas if:

$$\|\nabla c_s\| > T, \tag{23}$$

where T is a pre-defined threshold. Following such approach the force distribution in the interfacial can be computed through the expression [43]:

$$M = \sigma \nabla^2 c_s \nabla c_s. \tag{24}$$

Finally, we must considerer a suitable rendering technique. Our choice is the photon map method because it provides effects like color bleeding, soft shadows, indirect illumination and caustics, which is very important in scenes with participating media. Firstly, we will use the implementation available in the PBRT library. Next we will implement a customized version of the photon map to get performance for real time applications. We could try to simulate both the phases through a Lattice Gas Model. However, a known problem of such approach is that no mathematical understanding is gained of which parameters lead to desired behaviors. Thus, the use of Navier-Stokes for the liquid modeling aims to allow standard ways to control the system.

The interaction between fluids and surfaces will be developed in two aspects: (1) Interaction between gas and a deformable surface. (2) Precipitation in terrain models.

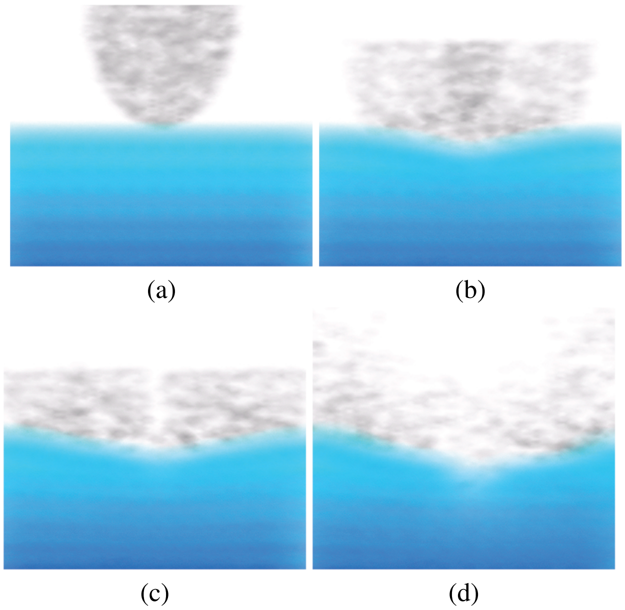


Figure 7. Configurations of liquid-gas: (a) at initial step; (b) after 15 steps; (c) after 25 steps; and (d) after 50 steps of simulation. Source [20]

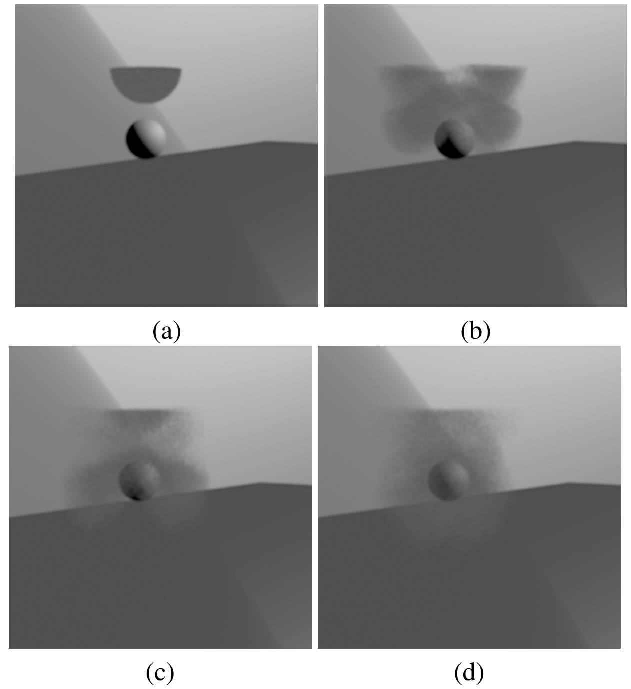


Figure 8. Configurations of a rigid sphere immersed in 3D FHP: (a) at initial step; (b) after 40 steps; (c) after 80 steps; and (d) after 90 steps of simulation.

In both cases, the fluid will be simulated by the three-dimensional FHP and the surface by a polygonal mesh. The first point is the collision detection. The system must be able to detect the collision between the particles of the FHP model and the surface. Once the surface is immersed in the FHP framework we just perform the rasterization of the surface in the FHP lattice and mark the obtained nodes (set a flag to 1). So, when a FHP particle reaches such a node of a rigid suface, it is just a matter of reflecting the particle following the FHP rules stated in section 5. The Figure 8 shows an example of this method for a sphere immersed in the three dimensional fluid simulation framework proposed in section 5.

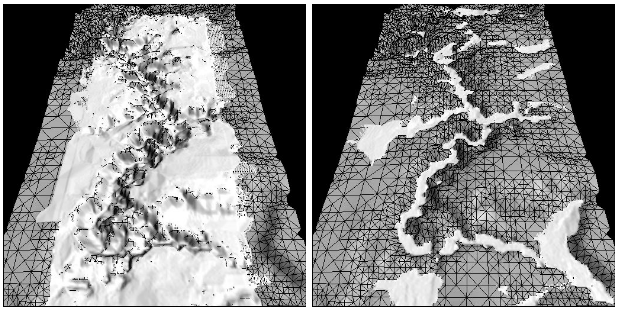
If we have a deformable surface modeled by the Mass-Spring system described in section 6 then, for each node of the surface, we apply expression (22) to compute the interaction force and add this force in expression (17). After each time step of the Mass-Spring algorithm, we must rebuild the rasterization of the surface in the FHP lattice.

Finally, we aim to explore the three dimensional FHP model in the framework developed by our team for surface flow animation in digital terrain models [3]. The surface flow simulation follows a particle model, inspired in the LGCA technique, described in [3]. The basic data structures of the model are a polygonal representation of the surface and a regular lattice with nodes $(i, j) \in L \times L$, where $L \subset \mathbb{N}$. Particles move according to the terrain surface topography and the fluid configuration nearby. There is a counter in each lattice node used to keep the number of particles in the corresponding (i, j) position. A node may contain more then one particle in this model. The Figure 9 shows some snapshots of a simulation using this technique.

In this work we will embed the framework described in [3] in the $3D$ FHP framework (section 5) in order to simulate precipitation over terrains. The terrain surface is considered as a rigid one, so, the procedure for detection of collision between the particles of the FHP model and the terrain surface follows the idea also explained above. Given a lattice node (i, j), we take a neighborhood of the corresponding surface point (i, j, z) and check the fluid density computed in the $3D$ FHP model. This quantity is used to quantify the precipitation value in the (i, j, z) point of the terrain.

9. Conclusions and Future Works

The main goal of our work is the development of a computational framework to create realistic animations of three dimensional systems involving gas-liquid and fluid-surface interactions. In this paper we propose to combine FHP, SPH and Mass-Spring systems for simulation tasks and photon



(a) (b)

Figure 9. Configurations of simulation of flow in a terrain model: (a) at initial step; and (b) after 2500 steps of simulation. Source [3]

map realistic rendering. We discuss the theoretical elements of our proposal and present some experimental results. We do believe that the computational efficiency of the applied methods will generate a useful framework for real time applications, like games and virtual reality.

References

[1] N. Adabala and S. Manohar. Techniques for realistic visualization of fluids: A survey. *Comput. Graph. Forum*, 2002.

[2] C. Adler, D. d'Humières, and D. H. Rothman. Surface tension and interface fluctuations in immiscible lattic gases. *J. Phys. I*, 1994.

[3] B. Barcellos, G. A. Giraldi, R. L. Silva, A. L. Apolinário, and P. S. S. Rodrigues. Surface flow animation in digital terrain models. In *SVR 2007 - IX Symposium on Virtual an Augmented Reality*, 2007.

[4] C. Batty, F. Bertails, and R. Bridson. A fast variational framework for accurate solid-fluid coupling. *ACM Trans. Graph.*, 2007.

[5] R. Benzi, S. Succi, and M. Vergassola. The lattice boltzmann equation: Theory and applications. *Physics Reports*, 1992.

[6] R. Bridson, R. Fedkiw, and J. Anderson. Robust treatment of collisions, contact and friction for cloth animation. In *SIGGRAPH '02*, 2002.

[7] J. M. Buick, W. J. Easson, and C. A. Greated. Numerical simulation of internal gravity waves using a lattice gas model. *Int. J. Numer. Meth. fluids*, 1998.

[8] N. Chentanez, T. G. Goktekin, B. E. Feldman, and J. F. O'Brien. Simultaneous coupling of fluids and deformable bodies. In *SCA '06: Proc. of the 2006 ACM SIGGRAPH/Eurographics*, 2006.

[9] B. Chopard and M. Droz. *Cellular Automata Modeling of Physical Systems*. Cambridge University Press, 1998.

[10] B. Chopard, A. Dupuis, A. Masselot, and P. Luthi. Cellular automata and lattice boltzmann techniques: An approach to model and simulate complex systems. *Advances in Complex Systems*, 2002.

[11] O. Deusen, D. S. Ebert, R. Fedkiw, F. K. Musgrave, P. Prusinkiewicz, D. Roble, J. Stam, and J. Tessendorf. The elements of nature: Interactive and realistic techniques. In *SIGGRAPH '04: ACM SIGGRAPH 2004 Course Notes*, 2004.

[12] D. A. Drew. Mathematical modeling of two-phase. *Annu. Rev. Fluid Mech.*, 1983.

[13] D. Enright, S. Marschner, and R. Fedkiw. Animation and rendering of complex water surfaces. *ACM Trans. Graph.*, 2002.

[14] O. Etzmuss, J. Gross, and W. Strasser. Deriving a particle system from continuum mechanics for the animation of deformable objects. *IEEE Trans. on Vis. and Comp. Graph.*, 2003.

[15] N. Foster and R. Fedkiw. Practical animation of liquids. In *SIGGRAPH '01*, 2001.

[16] N. Foster and D. Metaxas. Modeling the motion of a hot, turbulent gas. In *SIGGRAPH '97: Proceedings of the 24th annual conference on Computer graphics and interactive techniques*, 1997.

[17] U. Frisch, D. D'Humières, B. Hasslacher, P. Lallemand, Y. Pomeau, and J.-P. Rivet. Lattice gas hydrodynamics in two and three dimension. *Complex Systems*, 1987.

[18] U. Frisch, B. Hasslacher, and Y. Pomeau. Lattice-gas automata for the navier-stokes equation. *Phys. Rev.*, 1986.

[19] O. GENEVAUX, A. HABIBI, and J.-M. DISCHLER. Simulating fluid-solid interaction. In *Graphics Interface*, 2003.

[20] G. Giraldi, A. Xavier, A. A. Jr, A. Neto, and P. Rodrigues. Animation of gas-liquid systems through lattice gas cellular automata and smoothed particle hydrodynamics. In *Proceedings of the CNMAC 2005*, 2005.

[21] G. A. Giraldi, A. V. Xavier, A. L. Apolinario, and P. S. Rodrigues. Lattice gas cellular automata for computational fluid animation. Technical report, National Laboratory of Scientific Computing, 2005.

[22] E. Guendelman, A. Selle, F. Losasso, and R. Fedkiw. Coupling water and smoke to thin deformable and rigid shells. In *SIGGRAPH '05*, 2005.

[23] P. S. Heckbert. Adaptive radiosity textures for bidirectional ray tracing. In *SIGGRAPH '90*, 1990.

[24] C. Hirsch. *Numerical Computation of Internal and External Flows: Fundamentals of Numerical Discretization*. John Wiley & Sons, 1988.

[25] A. Iglesias. Computer graphics for water modeling and rendering: a survey. *Future Gener. Comput. Syst.*, 2004.

[26] H. W. Jensen. Global illumination using photon maps. In *Proceedings of the eurographics workshop on Rendering techniques '96*, 1996.

[27] H. W. Jensen and P. H. Christensen. Efficient simulation of light transport in scences with participating media using photon maps. In *SIGGRAPH '98*, 1998.

[28] R. Keiser, B. Adams, D. Gasser, P. Bazzi, P. Dutre, and M. Gross. A unified lagrangian approach to solid-fluid animation. In *Point-Based Graphics, 2005. Eurographics/IEEE VGTC Symposium Proceedings*, 2005.

[29] G. R. Liu and M. B. Liu. *Smoothed Particle Hydrodynamics : a Meshfree Particle Method*. World Scientific, 2003.

[30] M. Liu, G. Liu, and K. Lamb. Constructing smoothing functions in smoothed particle hydrodynamics with applications. *Journal of Computational and Applied Mathematics*, 2003.

[31] M. Müller, R. Keiser, A. Nealen, M. Pauly, M. Gross, and M. Alexa. Point based animation of elastic, plastic and melting objects. In *SCA '04: Proceedings of the 2004 ACM SIGGRAPH/Eurographics symposium on Computer animation*, 2004.

[32] J. Morris, P. Fox, and Y. Zhu. Modeling low reynolds number incompressible flows using sph. *JOURNAL OF COMPUTATIONAL PHYSICS*, 1997.

[33] J. Mosegaard and T. S. Sorensen. Gpu accelerated surgical simulators for complex morphology. In *VR '05: Proceedings of the 2005 IEEE Conference 2005 on Virtual Reality*, 2005.

[34] M. Müller, D. Charypar, and M. Gross. Particle-based fluid simulation for interactive applications. In *Proceedings of ACM SIGGRAPH symposium on Computer animation*, 2003.

[35] M. Müller, S. Schirm, M. Teschner, B. Heidelberger, and M. Gross. Interaction of fluids with deformable solids: Research articles. *Comput. Animat. Virtual Worlds*, 2004.

[36] A. Nealen, M. Muller, R. Keiser, E. Boxerman, and M. Carlson. Physically based deformable models in computer graphics. In *Eurographics 2005, State of The Art Report (STAR)*, 2005.

[37] S. Premoze, T. Tasdizen, J. Bigler, A. Lefohn, and R. Whitaker. Particle-based simulation of fluids. In *EUROGRAPHICS*, 2003.

[38] D. H. Rothman and S. Zaleski. Lattice-gas models of phase separation: Interface, phase transition and multiphase flows. *Rev. Mod. Phys*, 1994.

[39] B. Schlatter. A pedagogical tool using smoothed particle hydrodynamics to model fluid flow past a system of cylinders. Master's thesis, Oregon State University, 1989.

[40] B. Solenthaler, J. Schlafli, and R. Pajarola. A unified particle model for fluid-solid interactions. *Comput. Animat. Virtual Worlds*, 2007.

[41] J. Stam. Stable fluids. In *Siggraph 1999*, 1999.

[42] J. Stam. Flows on surfaces of arbitrary topology. In *SIGGRAPH '03*, 2003.

[43] Y. Sun and C. Beckermann. Diffuse interface modeling of two-pahse flows based on averaging: Mass and momentum equations. *Physica D*, 2004.

[44] A. VanGelder and J. Wilhelms. Simulation of elastic membranes and soft tissue with triangulated spring meshes. Technical report, Univ. of Calif. at Santa Cruz, 1997.

[45] E. Veach and L. J. Guibas. Metropolis light transport. In *SIGGRAPH '97*, 1997.

[46] Z. Ye, D. Maidment, and D. McKinney. Map-based surface and subsurface flow simulation models: An object-oriented and gis approach. Technical report, Center for Research in Water Resources, University of Texas at Austin, 1996.

[47] Y. Zhu and R. Bridson. Animating sand as a fluid. *ACM Trans. Graph.*, 2005.