

Animation of Gas-Liquid Systems Through Lattice Gas Cellular Automata and Smoothed Particle Hydrodynamics

Gilson A. Giraldi, Adilson V. Xavier,
Antonio L. Apolinario Jr, Algemiro A. S. Neto,
Paulo S. Rodrigues.

National Laboratory of Scientific Computing
Ave Getúlio Vargas, 333, 25651-075, Petrópolis, RJ, Brasil
e-mail: {gilson,adilson,alopes,algemiro,pssr}@lncc.br

Abstract

The past two decades showed a rapid growing of physically-based modeling of fluids for computer graphics applications. Techniques in the field of Computational Fluid Dynamics (CFD) have been applied for realistic fluid animation for virtual surgery simulators, computer games and visual effects. In this paper we focus on gas-liquid modeling for computer graphics applications. The main goal of our work is the development of a technique to create realistic animations of gas-liquid systems through computational simulation. Specifically, we model and simulate the gas through a Lattice Gas Cellular Automata and the liquid by Navier-Stokes equations and the Smoothed Particle Hydrodynamics (SPH) method. We combine the advantage of the low computational cost of cellular automata, the realistic fluid dynamics inherent in the Navier-Stokes and the efficiency of SPH for problems with dynamic boundaries to develop a new animating framework for computer graphics applications. In this work, we discuss the theoretical elements of our proposal and show the mentioned advantages for 2D flows in the experimental results.

1 Introduction

Physically-based techniques for the animation of natural elements like fluids (gas or liquids), elastic, plastic and melting objects, among others, have taken the attention of the computer graphics community. The motivation for such interest rely in the potential applications of these methods and in the complexity and beauty of the natural phenomena that are involved [11].

In this paper, we focus on physically-based fluid animation for computer graphics applications (see [11] and references therein). Basically, the works in this area fall in two categories: Realistic fluid and Interactive, or Real-Time, fluid animation. The former is more suitable for the special effects industry while the later is appropriate for interactive applica-

tions like computer games and virtual surgery. The work [4] is a remarkable one in this area which includes fluid equations and numerical technique [8], shortly Computational Fluid Dynamics (CFD), and scientific visualization methods. The literature of this field reports gas [4] and water simulations [12], interaction between liquids and deformable solids and others [11].

A majority of fluid animation methods in computer graphics use 2D/3D mesh based approaches that are mathematically motivated by the Eulerian methods of Finite Element (FE) and Finite Difference (FD), in conjunction with Navier-Stokes equations of fluids [8]. Recently, the mesh-free methods like *Smoothed Particle Hydrodynamics (SPH)* [12], *Moving-Particle Semi-Implicit (MPS)* and the *Method of Characteristics* have been applied (see [7] and references therein).

On the other hand, cellular automata based modeling tries to emulate the system through a set of *agents* that encapsulate the behaviors of the *particles* that make up the system [1]. For instance, that is the philosophy behind Lattice Gas Cellular Automata (LGCA) models for fluids [3, 5]. These are discrete models based on point particles that move on a lattice, according to suitable and simple rules in order to mimic a fully molecular dynamics. Such bottom-up framework needs low computational resources for both the memory allocation and the computation itself.

In this paper we focus on gas-liquid modeling for computer graphics applications. We model and simulate the gas through a Lattice Gas Cellular Automata, the FHP model, and the liquid by Navier-Stokes equations and the Smoothed Particle Hydrodynamics (SPH) method. The FHP model, introduced by Frisch, Hasslacher and Pomeau [3], is simpler than the known Lattice Boltzmann models, but the obtained results for lower density flows have been suitable for our computer graphics applications. Also, by Chapman-Enskog expansion, a known multiscale technique in this area, it can be demonstrated that the Navier-Stokes model can

be reproduced by FHP technique. However, **there is no need** to solve Partial Differential Equations (PDEs) to obtain a high level of description. Therefore, when combining Navier-Stokes, LGCA and SPH, we get the advantage of the low computational cost of cellular automata, the realistic fluid dynamics inherent in the Navier-Stokes and the efficiency of SPH for problems with dynamic boundaries to develop a new animating framework for computer graphics applications.

We could try to simulate both the phases through a Lattice Gas Model [3]. 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 paper is organized as follows. Section 2 describes the FHP model its multiscale analysis to get Navier-Stokes behavior. Section 3 describes the SPH method. The experimental results are presented on section 4. Conclusions and future works are given on Section 5.

2 FHP and Navier-Stokes

The FHP was introduced by Frisch, Hasslacher and Pomeau [6] 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 site at the same time with the same velocity. This restriction is the *exclusion principle*; it ensures that six Boolean variables at each lattice site 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 site.

When exactly two particles enter the same site 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. For symmetry reasons, the two possibilities are chosen randomly, with equal probability.

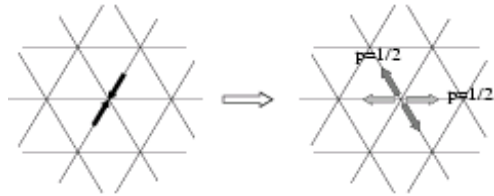


Figure 1: The two-body collision in the FHP.

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, $n_i(\vec{r}, t)$, of particle entering site \vec{r} at time t with a velocity pointing along direction \vec{c}_i , where $i = 1, 2, \dots, 6$ 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

$$\vec{v}_i = \frac{\Delta_r}{\Delta_t} \vec{c}_i. \quad (1)$$

The microdynamics of a LGCA is written as

$$n_i(\vec{r} + \Delta_r \vec{c}_i, t + \Delta_t) = n_i(\vec{r}, t) + \Omega_i(n(\vec{r}, t)) \quad (2)$$

where Ω_i is called the collision term [2].

We now define the ensemble average $N_i(\vec{r}, t) = \langle n_i(\vec{r}, t) \rangle$ of the microscopic occupation variables. Note that, $N_i(\vec{r}, t)$ is also the probability of having a particle entering the site \vec{r} , at time t , with velocity given by equation (1).

Following the usual definition of statistical mechanics, the local density of particles is the sum of the average number of particles traveling along, each direction \vec{c}_i

$$\rho(\vec{r}, t) = \sum_{i=0}^z N_i(\vec{r}, t). \quad (3)$$

Similarly, the particle current, which is the density ρ times the velocity field \vec{u} , is expressed by.

$$\rho(\vec{r}, t) \vec{u}(\vec{r}, t) = \sum_{i=0}^z \vec{v}_i N_i(\vec{r}, t). \quad (4)$$

Another quantity which will play an important role in the up coming derivation is the momentum tensor

Π defined as

$$\Pi_{\alpha\beta} = \sum_{i=0}^z \vec{v}_{i\alpha} \vec{v}_{i\beta} N_i(\vec{r}, t) \quad (5)$$

where the greek indices α and β label the d spatial components of the vectors. The quantity Π represents the flux of the α -component of momentum transported along the β -axis. This term will contain the pressure contribution and the effects of viscosity.

The starting point to obtain the macroscopic behavior of the CA fluid is to derive an equation for the N_i 's. Averaging the microdynamics (2) yields

$$N_i(\vec{r} + \Delta_r \vec{c}_i, t + \Delta_t) - N_i(\vec{r}, t) = \langle \Omega_i(n(\vec{r}, t)) \rangle \quad (6)$$

It is important to notice that $\Omega_i(n)$ has some generic properties, namely

$$\sum_{i=1}^z \Omega_i = 0 \quad \text{e} \quad \sum_{i=1}^z \vec{v}_i \Omega_i = 0 \quad (7)$$

expressing the fact that particle number and momentum are conserved during the collision process (the incoming sum of mass or momentum equals the outgoing sum).

The N_i 's vary between 0 and 1 and, at a scale $L \gg \Delta_r$ e $T \gg \Delta_t$, one can expect them to be smooth functions of the space and time coordinates. Therefore, equation (6) can be Taylor expanded up to second order and gives

$$\begin{aligned} & \Delta_r (\vec{c}_i \cdot \nabla) N_i(\vec{r}, t) + \Delta_t \partial_t N_i(\vec{r}, t) \\ & + \frac{1}{2} (\Delta_r)^2 (\vec{c}_i \cdot \nabla)^2 N_i(\vec{r}, t) + \Delta_r \Delta_t (\vec{c}_i \cdot \nabla) \partial_t N_i(\vec{r}, t) \\ & + \frac{1}{2} (\Delta_t)^2 (\partial_t)^2 N_i(\vec{r}, t) = \langle \Omega_i(n(\vec{r}, t)) \rangle. \end{aligned} \quad (8)$$

where $(\partial_t)^2$ is the second derivative in respect to the time parameter t .

Following [6], we introduce a new space variable \vec{r}_1 such that

$$\vec{r}_1 = \epsilon \vec{r} \quad \text{e} \quad \partial_r = \epsilon \partial_{\vec{r}_1} \quad (9)$$

with $\epsilon \ll 1$. We also introduce the extra time variables $t_1 = \epsilon t$ and $t_2 = \epsilon^2 t$, as well as new functions N_i^ϵ depending on \vec{r}_1 , t_1 and t_2 , $N_i^\epsilon = N_i^\epsilon(t_1, t_2, \vec{r}_1)$ and substitute into equation (8)

$$N_i \rightarrow N_i^\epsilon \quad \partial_t \rightarrow \epsilon \partial_{t_1} + \epsilon^2 \partial_{t_2} \quad \partial_r \rightarrow \epsilon \partial_{\vec{r}_1} \quad (10)$$

together with the corresponding expressions for the second order derivatives. Then obtain new equations for the new function N_i^ϵ . Thus, following the Chapman-Enskog method, a standard procedure used in statistical mechanics to solve an equation like (8) with a perturbation parameter ϵ , we assume that [13],

$$N_i^\epsilon = N_i^{(0)} + \epsilon N_i^{(1)} + \epsilon^2 N_i^{(2)} + \dots \quad (11)$$

Besides, it is supposed that $\langle \Omega_i(n) \rangle$ can be factorized into $\Omega_i(N)$. Thus, we write the contributions of each order in ϵ for right-hand side of (8):

$$\Omega_i(N) = \Omega_i(N^{(0)}) + \epsilon \sum_{j=1}^z \left(\frac{\partial \Omega_i(N^{(0)})}{\partial N_j} \right) N_j^{(1)} + \mathcal{O}(\epsilon^2) \quad (12)$$

Using expressions (9)-(11) in the left-hand side of (8) and comparing the terms of the same order in ϵ in the equation (12), yields

$$O(\epsilon^0) : \Omega_i(N^{(0)}) = 0 \quad (13)$$

and

$$\begin{aligned} O(\epsilon^1) : & \partial_{1\alpha} v_{i\alpha} N_i^{(0)} + \partial_{t_1} N_i^{(0)} \\ & = \frac{1}{\Delta_t} \sum_{j=1}^z \left(\frac{\partial \Omega_i(N^{(0)})}{\partial N_j} \right) N_j^{(1)} \end{aligned} \quad (14)$$

where the subscript 1 in spatial derivatives (e.g. $\partial_{1\alpha}$) indicates a differential operator expressed in the variable \vec{r}_1 and $\frac{\Delta_r}{\Delta_t} (\vec{c}_i \cdot \nabla_{r_1}) = \partial_{1\alpha} v_{i\alpha}$, from equation (1).

We also impose the extra conditions that the macroscopic quantities ρ and $\rho \vec{u}$ are entirely given by the zero order of expansion (11)

$$\rho = \sum_{i=1}^z N_i^{(0)} \quad \text{and} \quad \rho \vec{u} = \sum_{i=1}^z \vec{v}_i N_i^{(0)} \quad (15)$$

and therefore

$$\sum_{i=1}^z N_i^{(l)} = 0 \quad \text{and} \quad \sum_{i=1}^z \vec{v}_i N_i^{(l)} = 0, \quad \text{for } l \geq 1 \quad (16)$$

Thus, following the development present in [5], we can obtain (see [7] for some details):

$$\partial_t \rho + \text{div } \rho \vec{u} = 0 \quad (17)$$

$$\partial_t \vec{u} + 2C_4 G(\rho) (\vec{u} \cdot \nabla) \vec{u} = -\frac{1}{\rho} \nabla p + \nu \nabla^2 \vec{u} \quad (18)$$

where

$$\nu = \Delta_t v^2 b C_4 \left(\frac{1}{\Lambda} - \frac{1}{2} \right) = \frac{\Delta_t v^2}{d+2} \left(\frac{1}{\Lambda} - \frac{1}{2} \right) \quad (19)$$

is the kinematic viscosity of our discrete fluid and the pressure term is

$$p = a C_2 v^2 \rho - \left[\frac{C_2}{d} - C_4 \right] \rho G(\rho) u^2 \quad (20)$$

Equation (17) is the continuity equation. Expression (18) is equivalent to the standard Navier-Stokes equation for incompressible flows [??] if $C_4 G(\rho) = 1$. The above development shows that Navier-Stokes model can be approximately reproduced by the FHP technique **without solving** any differential equation.

3 Smooth Particle Hydrodynamics

In this work, we aim to simulate the liquid phase by the Navier-Stokes equation (expression (18) with $C_4G(\rho)$ set to 1) plus a state equation for the pressure field, like $P = c^2\rho$, where c is the speed of sound in the fluid. We can also add the gravity field (\vec{g}) to the model, to obtain the following Navier-Stokes expression:

$$\rho \left(\frac{\partial \vec{u}}{\partial t} + \vec{u} \cdot \nabla \vec{u} \right) = -\nabla p + \rho \vec{g} + \nu \nabla^2 \vec{u}, \quad (21)$$

Such model can be simulated by the SPH method described next, following the references [9]. 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:

$$\langle A(\mathbf{r}) \rangle = \int_{Space} A(\mathbf{r}') W(\mathbf{r} - \mathbf{r}', h) d\mathbf{r}', \quad (22)$$

where the function $W(\mathbf{r} - \mathbf{r}', h)$ is an (interpolation) kernel which must satisfies the following properties [9]:

1) Volume conservation:

$$\int_{Space} W(\mathbf{r} - \mathbf{r}', h) d\mathbf{r}' = 1, \quad (23)$$

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

$$\lim_{h \rightarrow 0} W(\mathbf{r} - \mathbf{r}', h) = \delta(\mathbf{r} - \mathbf{r}'). \quad (24)$$

If we take a sampling of A then the $A(\mathbf{r}')$ in equation (22) will be known only at a discrete set of N points $\mathbf{r}_1, \mathbf{r}_2, \dots, \mathbf{r}_N$. Hence, thorough properties (22)-(23) it is possible to show that [9]:

$$\langle A(\mathbf{r}) \rangle = \sum_{j=1}^N \frac{m_j}{\rho(\mathbf{r}_j)} A(\mathbf{r}_j) W(\mathbf{r} - \mathbf{r}_j, h). \quad (25)$$

$$\langle \nabla_{\mathbf{r}} A(\mathbf{r}) \rangle = \sum_{j=1}^N \frac{m_j}{\rho(\mathbf{r}_j)} A(\mathbf{r}_j) \nabla_{\mathbf{r}} W(\mathbf{r} - \mathbf{r}_j, h). \quad (26)$$

An analogous expression can be obtained by the Laplacian. From equation (26) we can observe that there is no need for a mesh to compute spatial derivatives. With equations (25) and (26), we are ready to write the discrete version of the fluid equations (21).

The smoothing length 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 (21), using this approach, as

$$\begin{aligned} \vec{f}_i^{press} &= - \sum_j m_j \frac{p_i + p_j}{2\rho_j} \vec{\nabla} W(\vec{r}_i - \vec{r}_j, h) \\ \vec{f}_i^{visc} &= \mu \sum_j m_j \frac{\vec{v}_j - \vec{v}_i}{\rho_j} \vec{\nabla}^2 W(\vec{r}_i - \vec{r}_j, h) \\ \vec{f}_j^{grav} &= \rho_j \vec{g}_j \end{aligned}$$

where the f_i^{press} and f_i^{visc} are the pressure and viscosity forces. Only the gravity force f_i^{grav} is considered as external force.

The density at each particle can be found from the following equation

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

However, we are considering the calculate density less the initial density. This is necessary to avoid extras forces in the surface of interaction between the two fluids.

In order to have stability we adopt the following kernel

$$\begin{aligned} W_{grav}(r, h) &= \frac{315}{64\pi h^9} \begin{cases} (h^2 - r^2)^3, & 0 \leq r \leq h, \\ 0, & \text{cc.} \end{cases} \\ W_{press}(r, h) &= \frac{15}{\pi h^6} \begin{cases} (h - r)^3, & 0 \leq r \leq h, \\ 0, & \text{cc.} \end{cases} \\ W_{visc}(r, h) &= \frac{15}{2\pi h^3} \begin{cases} -\frac{r^3}{2h^3} + \frac{r^2}{h^2} + \frac{h-2r}{2r}, & 0 \leq r \leq h, \\ 0, & \text{cc.} \end{cases} \end{aligned}$$

4 Experimental Results

In this work we combine the FHP and Navier-Stokes equations for computer graphics applications. We simulate a gas system through FHP and a liquid by Navier-Stokes plus SPH. From the point of view of our work, the fact that the SPH is a meshfree is attractive due the constant change in the surface of interaction in the two phase flow.

The first point is how to model the interactions between the two phases in the interfacial area. If we take the development presented in [3, 7], we observe that the divergence of the momentum tensor (equation (5)) generates the pressure contribution and effect of viscosity. Other possibilities can be found in the two-phase flows literature [5]. However, for simplicity, in this paper we follows another way. 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 . Then, we set the interaction force as:

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

where τ is force scale parameter. This approach is more intuitive than the other ones [5] and allows the generation of interesting visual effects. The results presented are limited for bidimensional fluid systems. Three dimensional extensions are discussed in the next section.

Firstly, we highlight the simplicity and low computational cost FHP. Figure 2 shows an initial configuration with zero density in the middle of the domain. There are 80.000 particles with position and velocity directions randomly distributed. It is not required any extra mathematical machinery to deal with such density discontinuity because system rules do not undergo modifications. Figure 3-5 picture this system after 10, 50, 300 time steps. The lattice resolution for the FHP is 200 by 200 points. The clock time was 22 seconds in a 2.4GHz Pentium IV with 1GB of RAM memory.

The examples below were generated with $\nu = 0.01$ in equation (21). When the SPH particles collide with the box boundary, we simply push them in and reflect the velocity component that is perpendicular to the boundary. The number of particles of SPH was 78 for all results (Figures 14 and 15). The particle systems that represent the liquid can be rendered as a point-based model through the splatting method [10]. The splatting method associates a kernel to each point. The integration of these kernels reconstruct the continuous fields. The liquid phase is visualized by a 2D version of the splatting method. The gas phase is visualized by a simple gray scale map for the gas (particle) density.

Figures 6 shows examples of gas-liquid systems with the forces in the interfacial area given by expression (4). This figure shows a stream of 1000 particles with vertical macroscopic velocity and the liquid (blue) just before interaction. Figure 7, 8 and 9 show three snapshots of the system evolution with the expected deformation of the interfacial surface after 2 seconds. In Figure 10 we show the initial configuration with 2000 particles randomly distributed in space and vertical macroscopic velocity. The obtained visualization (Figures 11, 12 and 13) show also interesting and realistic effects. In this case, the clock time was 6 seconds to complete the 166 iterations.

5 Conclusions and Future Works

In this paper we propose to combine FHP and Navier-Stokes plus SPH for gas-liquid modeling in computer graphics applications. We discuss the theoretical elements of our proposal and present some experimental results. Further works are the incorporation of external forces; for example, due to user interaction. Also, we will implement the Lattice Boltzmann models for 3D gas animation. In

this case, the visualization can be done by volume rendering and/or texture mapping [11]

References

- [1] *Agent-Based Modeling vs. Equation-Based Modeling: A Case Study and Users' Guide*, London, UK, 1998. Springer-Verlag.
- [2] B. Chopard and M. Droz. *Cellular Automata Modeling of Physical Systems*. Cambridge University Press, 1998.
- [3] 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*, 5(2):1-144, 2002. special issue on: Applications of Cellular Automata in Complex Systems.
- [4] N. Foster and D. Metaxas. Modeling the motion of a hot, turbulent gas. In *ACM SIGGRAPH*, pages 181-188. ACM Press, 1997.
- [5] 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*, pages 649-707, 1987.
- [6] U. Frisch, B. Hasslacher, and Y. Pomeau. Lattice-gas automata for the navier-stokes equation. *Phys. Rev.*, page 1505, 1986.
- [7] G. A. Giraldo, A. V. Xavier, A. L. A. Jr, and P. S. Rodrigues. Lattice gas cellular automata for computational fluid animation. Technical report, National Laboratory of Scientific Computing, <http://www.arxiv.org/abs/cs.GR/0507012>, 2005.
- [8] C. Hirsch. *Numerical Computation of Internal and External Flows: Fundamentals of Numerical Discretization*. John Wiley Sons, 1988.
- [9] M. Liu, G. Liu, and K. Lamb. Constructing smoothing functions in smoothed particle hydrodynamics with applications. *Journal of Computational and Applied Mathematics*, 155, 2003.
- [10] J. v. B. M. Zwicker, H. Pfister and M. Gross. Surface splatting. In *Proceedings of the 28th annual conference on Computer graphics and interactive techniques*, pages 371-378. ACM Press, 2001.
- [11] 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*, pages 141-151, New York, NY, USA, 2004. ACM Press.
- [12] M. Müller, D. Charypar, and M. Gross. Particle-based fluid simulation for interactive applications. In *Proceedings of ACM SIGGRAPH symposium on Computer animation*, 2003.
- [13] J. Piasecki. *Echelles de temps multiples en théories cinétique*. Cahiers de physique. Press polytechniques et universitaire romandes, 1997.

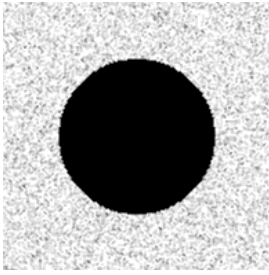


Figure 2: Initial configuration with 80000

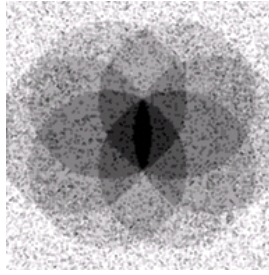


Figure 3: Evolution after 25 steps

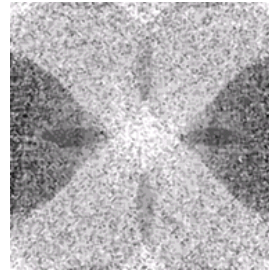


Figure 4: Evolution after 50 steps

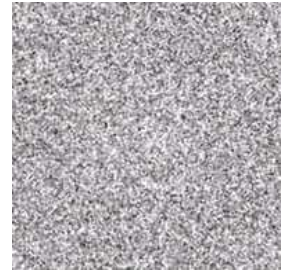


Figure 5: Evolution after 300 steps

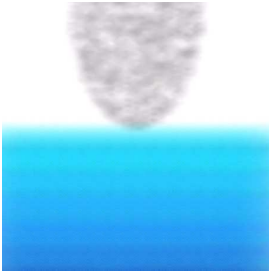


Figure 6: Initial configuration of liquid-gas

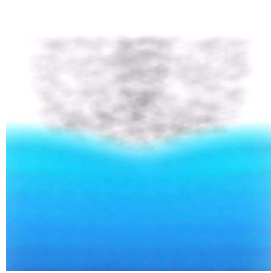


Figure 7: Evolution after 15 steps

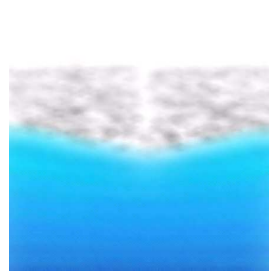


Figure 8: Evolution after 25 steps

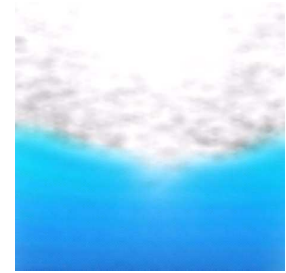


Figure 9: Evolution after 50 steps

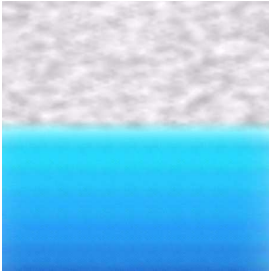


Figure 10: Initial configuration of liquid-gas

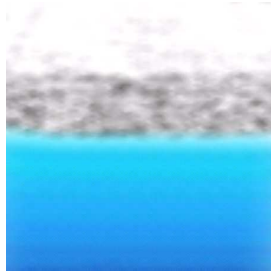


Figure 11: Evolution after 28 steps

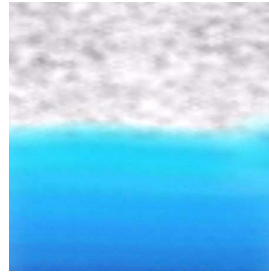


Figure 12: Evolution after 116 steps

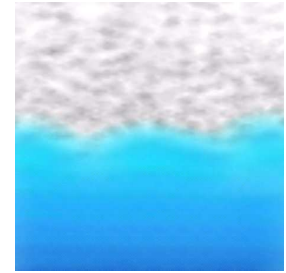


Figure 13: Evolution after 166 steps

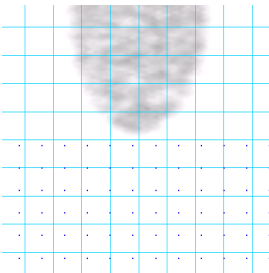


Figure 14: Initial configuration of gas-liquid system without blending technique.

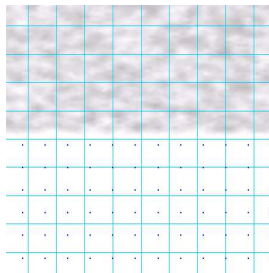


Figure 15: Initial configuration of gas-liquid system without blending technique.

6 Acknowledgments

We would like to acknowledge the Brazilian organization for scientific development, the CNPq and FAPERJ for the financial support for this work.