Interactive physically-based modeling for gaseous object using Smoothed particle hydrodynamics

Suriyong Lertkulvanich , Sanan Srakaew, King Mongkut's University of Technology Thonburi

Abstract— **This paper presents an interactive method of modelling the various kind of fluid based on Smoothed Particle Hydrodynamics (SPH) in modelling and simulating the gas flow with free surfaces. The modelling technique uses particle system together with molecular dynamics and also any possible external forces. The fluid equations previously used to model compressible fluid are now used to model incompressible fluid, the stiff gas, which is often used in Eulerian grid-based approach.**

In modelling stage, boundaries and obstacles are determined in order to make a real response of fluid with the boundaries. The leap-frog scheme is used to advance the fluid to the next state. However, the model still needs further modification to improve the realistic and simulation time, which is the goal of every simulator.

***Index Terms*— Computational Fluid Dynamics, gaseous object, particle method, Smoothed Particle Hydrodynamics,**

I. Introduction

Physically-based modelling in this day plays an important role in modelling the fluid like phenomena such as liquid, gas, smoke, dust, and cloud, which are in the form of fluid. Almost all of them are based on two main approaches. The first approach is the Eulerian scheme, which is used to model the fluid field on a specified grid position. The other is Lagrangian approach, which models fluid as a group of particles placed at any locations in space. Although the first one is established well in the research area called Computational Fluid Dynamics (CFD), this field of research is still open because of the fluid complexity, abundant resource requirement, and intensive computational time. One important reason is that the numerical dissipation arises due to the advection process in the Navier-Stokes equation [16]. This make the treatment of the event very complicated. For the particle method, there are two approaches; uncoupled and coupled particles. In the uncoupled particle method, particle has no effect to each other, while the coupled particle method takes into consideration the interaction among particles. Since the first one the particle is uncoupled, it doesn't look realistic. The other one generates more realistic look. One of well-known technique used to simulate the coupled particles is the Smoothed particle hydrodynamics, which allows particles affect with each other when they are close, attract each other at medium distance, and have no force with others at further distance. This technique models the fluid efficiently.

The Smoothed particle hydrodynamic (SPH) was first developed for astrophysical computation by Lucy [4] and Gingold and Monaghan [3]. The method is comprehensive and general enough to model wide range problems. The SPH, Lagrangian approach, is an interpolation method for particle systems. In Eulerian method, the information is placed at the grid or particle-in-cell and the grid system is used to calculate spatial derivatives for the next advanced step by interpolating it back and forth between grids, while the particle information are ignored [16]. For the SPH, the information is placed on particle itself and updated by integral interpolant approximated by summation from its neighbors. Similarly, the derivative expression of variable for each particle is evaluated by summation interpolation based on the derivative of the kernel function which is obtained analytically.

Mr. Suriyong Lertkulvanich is a graduate student in Computer Engineering Department, King Mongkut's University of Technology, Bangkok Thailand; email: suriyong.ler@kmutt.ac.th.

Mr. Sanan Srakaew is a lecturer at Computer Engineering Department, King Mongkut's University of Technology, Bangkok Thailand; email: sanan@cpe.kmutt.ac.th.

II. Related works

Fluid modeling was first introduced in 1822 by Claude Navier. In 1845, George Stokes formed the famous equation called Navier-Stokes equations which described the dynamics of any kind of fluids. Since then those equations have been widely used to help solving the equation with a large amount of computations.

An obvious evolution of fluid modeling first began in 1983. Reeves W.T.[13] introduced particle method as a system to model a fuzzy object. Since then there was a lot of particle modeling literatures splitting in two main approaches, the particle-based or Lagrangian approach and the grid-base or the Eulerian approach. Desbrun and Canie [1] use particles to animate a 2D particle simulation. Since then particle system has been used in various forms in computer graphic modeling such as a hair modeling, cloth modeling etc.

In these recent years, the other approach, the grid-based plays a major role in modeling fluid such gas, fire, water flow, etc. The first fascinating and outstanding animated fluid was introduced by Foster and Metaxas in 1997. Later, the improved Forster's and Metaxas's method by Stam [17] is certainly an important step for the real-time fluid simulation. In comparison to the particle based method; the grid-based method is not suitable for fluid simulation when it breaks apart into many separate pieces. Other problems are the lack of interaction force and possible need in recalculation if the environment changes. With the particle method, this kind of problems can be solved.

Since our goal is to make a realistic simulation of gas, is the particle based method is used in conjunction with the smoothed particle hydrodynamics. The remaining of the paper discusses the theory of SPH and its implementation to the gas object, the results and conclusion, and finally the future works.

III. PARTICLE MODELLING

A. Smooth Particle Hydrodynamics (SPH)

For modeling the object like fluid, there are two methods mostly used in this day; grid-based and particle-based methods. For the grid based-method, the object properties such as mass, density, and velocity are placed or aligned on fixed grid location. On the contrary, for the smoothed particle hydrodynamics (SPH) approach, particle properties are simulated and placed on the particles themselves. They follow the object and make the mass conserved. This means that the mass conserved equations and convection terms are dispensable [10]. Behind the idea of SPH is the determination of characteristics of fluid by interpolating from the set of unorganized particles. They are discretized and computed from the neighboring particles according to weighting scheme, the smoothing kernel W_h, which gives the spatial mass distribution profile over smoothing length h. For the scalar quantity A, it is interpolated to next state by comparing its location to each neighbor with distance r calculated by weighted sum from other particles:

$$A_i(\mathrm{r}) = \sum_j m_j \frac{A_j}{\rho_j} W_i(\mathrm{r}_i - \mathrm{r}_j, h) \tag{1}$$

where i is the particle order, r_i is its position, j is the iteration order due to the particle j, A_j, r_j, m_j and ρ_j are its field quantity, position, mass and density respectively. W is the scalar function called smoothing kernel with the support distance h from the core. The kernel will smear out the core effect in space. Suppose the density is an interesting property, it needs to smooth with other particles, the equation (1) will be:

$$\begin{aligned}\rho_i &= \sum_j m_j \frac{\rho_j}{\rho_j} W_i(\mathrm{r}_i - \mathrm{r}_j, h) \\ &= \sum_j m_j W_i(\mathrm{r}_i - \mathrm{r}_j, h)\end{aligned} \tag{2}$$

There are several kernel functions. Each function is suitable for each situation. Good kernels must have a Fourier transform that falls rapidly with the wave numbers [1]. In general the cubic B-spline kernel is used:

$$W_i(\mathrm{r}_{ij}, h) = \frac{\sigma}{h^D}\begin{cases} 1 - \frac{3}{2}s^2 + \frac{3}{4}s^3 & ;0 \le s \le 1 \\ \frac{1}{4}(2-s)^3 & ;1 < s \le 2 \\ 0 & ;2 < s \end{cases} \tag{3}$$

where $s = |\mathrm{r}|/\mathrm{h}$, $\mathrm{r}_{ij} = \mathrm{r}_i - \mathrm{r}_j$, σ is the normalized constant and it equals to $1/\pi$ for three dimension model and D is the number of dimension. In most fluid equations, a derivative or gradient of field quantity is also needed for computing the rate of change of the characteristic. Its form is transformed from the derivative to summation of the product of quantity and gradient of smoothing kernel. Thus the gradient cubic B-spline kernel is

$$\nabla_i W_i(\mathrm{r}_{ij}, h) = \frac{\sigma}{h^{D+1}}\hat{\mathrm{r}}\begin{cases} -3s + \frac{9}{4}s^2 & ;0 \le s \le 1 \\ -\frac{3}{4}(2-s)^2 & ;1 < s \le 2 \\ 0 & ;2 < s \end{cases} \tag{4}$$

where $\hat{\mathrm{r}}$ is the normalized vector of r_{ij} and the gradient of the field of any quantity A can be evaluated to

$$\nabla A(\mathrm{r}_{ij}) = \sum_j m_j A_j \nabla W_{ij}(\mathrm{r}_{ij}, h) \tag{5}$$

B. Modelling with SPH

1) Equation of state

For the grid based method, the evolving of the gaseous is described by using Navier's Stokes equation. The evolving of fluid particle, so as the gas, is described by velocity field u, a density filed ρ and a pressure field p. The evolution of these parameters over time are described by three main equations

$$\nabla \cdot \mathrm{u} = 0 \tag{6}$$

$$\rho\left(\frac{\partial \mathrm{u}}{\partial t} + \mathrm{u}.\nabla \cdot \mathrm{u}\right) = -\nabla p + \rho g + \kappa \nabla^2 \mathrm{u} + F \tag{7}$$

$$\frac{\partial \rho}{\partial t} + \nabla \cdot (\rho \mathrm{u}) = 0 \tag{8}$$

Where g is the gravitational acceleration, κ is the fluid viscosity coefficient, E is the energy and F is the external force which we can model by disturbing the fluid from the mouse motion or else. The equation (6) is the mass conserved equation, (7) is the momentum conserved, (8) is the mass convection equation.
Another equation describes the conservation of energy.

$$\frac{dE}{dt} = -\left(\frac{p}{\rho}\right)\nabla \cdot \mathrm{u} \tag{9}$$

where E is the thermal energy. These equations can describe the dynamics of gas and must be used to fully describe the behavior of fluid which defines the functional relationship between temperature, density and pressure. One important issue in evolving the particle is the particle interactions. This kind of force is implicitly handled by the smoothing kernel.

For the form used for an ideal gas, one equation needed to update pressure for each iteration is:

$$p = (\zeta - 1)\rho E \tag{10}$$

where ζ is the ratio of the specific heat of particular gas being simulated. This equation may be changed depending on the type of fluid. For stiff gas which almost incompressible, Monaghan suggested a different form of equation of state:

$$p_i = p_o\left[\left(\frac{\rho}{\rho_0}\right)^{\gamma} - 1\right] \tag{11}$$

where p_0 and ρ_0 is the reference pressure and the reference density, respectively. In general $\gamma = 7$ means a few percent change in density results a large temperature change [14].
In general, when update, the particle density equation (2) can be used. For the stiff gas, equation (8) is used instead in order to prevent the zero density which can happen with the close-to-boundary particle.

In addition to these equations, as mentioned before modeling fluid using particle method, there are some advantageous: First, since the number of particles is constant and they have constant mass, then the equation (6) is not necessary. Second, the particles move with the fluid, the left hand side of the equation (7) can be simplified to only derivative without convection $(\mathrm{u}.\nabla \mathrm{u})$ term. Thus, for the evolution

of fluid, we model only the right hand side of the equation (7) in the form of acceleration equation

$$\mathrm{a}_i = \frac{d\mathrm{u}_\mathrm{i}}{dt} = \frac{F_i}{\rho_i} \tag{12}$$

where a_i stands for acceleration of particle i and F_i is the right hand side of equation (7)

Since we use the SPH to model the fluid, the equation (7) and (8) can be transformed from the gradient form to the interpolating form by using the smoothing kernel. The pressure gradient used to compute the force on particle is estimated using:

$$\frac{\nabla p_i}{\rho_i} = \sum_j m_j \left(\frac{p_i}{\rho_i^2} + \frac{p_j}{\rho_j^2} \right) \nabla_j W_{ij} \left(\mathrm{r}_{ij}, h \right) \tag{13}$$

All of the Navier's-Stokes equation in the form of SPH equations of motion cab be derived as:

$$\frac{d\mathrm{u}_i}{dt} = -\sum_j m_j \left(\frac{p_i}{\rho_i^2} + \frac{p_j}{\rho_j^2} + \Pi_{ij} \right) \nabla_i W_{ij} + F_i \tag{14}$$

$$\frac{d\rho_i}{dt} = \sum_j m_j \mathrm{u}_{ij} . \nabla_i W_{ij} \tag{15}$$

$$\frac{dE_i}{dt} = \frac{1}{2} \sum_j m_j \left(\frac{p_i}{\rho_i^2} + \frac{p_j}{\rho_j^2} + \Pi_{ij} \right) \mathrm{u}_{ij} . \nabla_i W_{ij} \tag{16}$$

Where F_i is the external force which can be any kind of force and we use it as an interactive perturbation force to the velocity field. For the equation (14) and(15), they are used for updating the pressure and density at the next time step. The term Π_{ij} is an artificial viscosity added to handle shocks. Several variations of this expression but one is preferred suggested by [3]:

$$\Pi_{ij} = \begin{cases} \dfrac{-\alpha \bar{c}_{ij} \mu_{ij} + \beta \mu_{ij}^2}{\bar{\rho}_{ij}} & \text{if } \mathrm{u}_{ij} . \mathrm{r}_{ij} < 0 \\ 0 & \text{if } \mathrm{u}_{ij} . \mathrm{r}_{ij} \geq 0 \end{cases} \tag{17}$$

where $\mu_{\mathrm{ij}} = \dfrac{h \mathrm{u}_{ij} . \mathrm{r}_{ij}}{\mathrm{r}_{ij}^2 + \eta^2}$, $\bar{\rho}_{ij} = \dfrac{\rho_i + \rho_j}{2}$ the mean density, $\mathrm{r}_{ij} = \mathrm{r}_i - \mathrm{r}_j$ distance between particle, $\mathrm{u}_{ij} = \mathrm{u}_i - \mathrm{u}_j$ velocity difference between particle i and j, α and β are viscous constants (usually set to 1 and 2 respectively) , $\eta = 0.1h$ the denominator zero preventing term, and $\bar{c}_{ij}$ the particle mean sound speed [8], [1]. The speed of sound for each particle i is:

$$c_i = \sqrt{\frac{\gamma P}{\rho}} \tag{18}$$

The first term in (17) is the shear and bulk viscosity and the second is the Von Neumann-Richtmyer artificial viscosity, which prevents particle interpenetration at high speed [1].

C. Forces

Clearly, the term of forces of equation (14) are the external forces of various kinds that act on the fluid. Some of them are used in our simulation process and they need different kinds of simulating kernel. Muller [10] suggested the following:

1) Viscosity Force

This force dependeds on the different of velocity. Thus

$$F_i^{viscosity} = \mu \nabla^2 \mathrm{u} = \mu \sum_j m_j \frac{\mathrm{u}_j - \mathrm{u}_i}{\rho_j} \nabla_{ij}^2 W \left(\mathrm{r}_{ij}, h \right) \tag{19}$$

2) Bouyant force

This kind of force is belonging to the gaseous and it moves the gas in the direction upward. The gas will rise and move against the gravity. Foster and Metaxas [2] model this force by:

$$F_{bv} = \beta g_v \left(T - T_0 \right) \tag{20}$$

where g_v is the gravity in the vertical direction, β is the coefficient of thermal expansion, T is modified to the particle element temperature, and T_0 is the reference temperature of the gas. This equation is applied directly to the moving state equation.

IV. Gas simulation

A. Data structure

Since the smoothing process is concerned with smoothing kernel, the value of kernel which is outside the $2h$ length has no meaning for calculating and it takes a lot of computational time to visit all of the other particles. Thus, former literatures suggested that it should be designed the grid to support for the particles called voxels with the width in each direction of $2h$ [9], [10], [14]. Thus, this means that the particle is randomly assigned into grid voxel at initial state and smoothed with its neighboring voxels and no need to smooth with particles which are further than one cell. After each iteration, the old grid position is deleted and the new grid structure is assigned.

For the structure of particles in cell, a double linked list is used for particles and their boundary particles which is one neighboring of cell length and its location and the list is reassigned when the calculation is moving to the next cell.

B. Algorithm

In order to do the time simulation for our particle objects, the particle needs initialized with a slightly difference in parameters, position, velocity, pressure, density and thermal temperature. If there is no difference in these parameters the particle does not move. The system is evolved in a manner similar to the following pseudo code:

For all particles
  Search for neighboring particle advance in 1 cell
  Calculate smoothing kernel and its gradient, using neighboring cell
  Calculate the pressure gradient and artificial viscosity
  Calculate acceleration using SPH form of momentum conserved equation (equation(13)) and corresponding forces
  Calculate rate of change of thermal energy (16)
  Calculate rate of change of density (15)
  Handle collision and particle interaction force
For all particles
  Update positions
  Update velocity
  Update thermal energy
  Update particle density
  Calculate new particle pressure due to equation (18)

C. Time integration

For the time integration to the next iteration, the leap-frog method is used. The advantage of this scheme is that it needs low requirement of

resources such as memory, storage and its computational efficiency which is accurate to second-order [1]. The leap-frog, the particle position and velocity are offset by a half time step. The rate of change of density and velocity are used to advance the density and velocity for a half time step while position is advanced in full time step [11] thus all of the evolution equations are shown as follow:

$$
\begin{aligned}
t &= t + \Delta t \\
\rho_i\left(t+\frac{\Delta t}{2}\right) &= \rho_i(t) + \frac{\Delta t}{2} D\rho_i(t) \\
\mathrm{u}_i\left(t+\frac{\Delta t}{2}\right) &= \mathrm{u}_i(t) + \frac{\Delta t}{2} D\mathrm{u}_i(t) \\
x_i(t+\Delta t) &= x_i(t) + \Delta t \mathrm{u}_i(t+\Delta t/2)
\end{aligned}
\tag{21}
$$

In order to keep the calculations consistent at each subsequent time step, at the start of each subsequent time step, the density and velocity of each particle need to be predicted at half a time step to coincide the position.

$$
\begin{aligned}
\rho_i(t) &= \rho_i\left(t-\frac{\Delta t}{2}\right) + \frac{\Delta t}{2} D\rho_i(t-\Delta t) \\
\mathrm{u}_i(t) &= \mathrm{u}_i\left(t-\frac{\Delta t}{2}\right) + \frac{\Delta t}{2} D\mathrm{u}_i(t-\Delta t)
\end{aligned}
\tag{22}
$$

At the end of the subsequent time step, the particle density, velocity and position are advanced in the standard leap-frog scheme

D. Boundary conditions and collision handling

There is no need to handle the particle collision among each other since the smoothing process handles this instead. Thus the particle handling is needed when the particle hits a boundary or an obstacle. Unlike Eulerian scheme, which the boundaries or obstacles are discretized to small polygon and aligned to each grid and boundary, the obstacle of particle method are aligned and formed as big polygons in space. For the particle which is near the boundary or obstacle, the point/plane method [19] is used with two most success responses, the particle bouncing and repelling. If X is a point on plane with normal n which pointing inside and P is particle location, the dot product of $(X\text{-}P).n$ is used for this event. If it is greater than zero, it means the particle is inside, otherwise is not allowed. For the response on collision plane, the normal component (V_N) of particle velocity is for bouncing response and tangentenial component (V_T) is for repelling as shown in Figure 1.

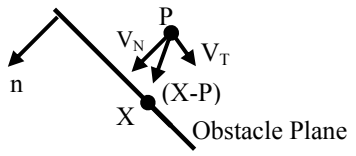


Figure 1 Boundary condition handling

E. Implementation

It is obviously seen that the time requirement for each iteration of particle is $O(n^2)$ if it smoothes with its all particles even though the kernel effect only $2h$ length. To improve the computational time, iterating only $2h$ means the spatial volume is divided into voxel with approximated size of $2h$ length. Each voxel containing the linked list of pointers to particle can have many particles which has linked list of pointers to particles data structure but each particle is associated with only one voxel and interacts only with particle in neighboring voxel. Thus the i^{th} particle only smoothes with its neighbors one cell apart. With this technique the computation time complexity is reduced from $O(n^2)$ to $O(nm)$ where m is the number of its neighboring particles which are in the same grid cell [10].

V. Result

A. Programming language and Interactive simulation

The program is written in C++ using OpenGL library with interactive components. At first the interactive part is designed using keyboard and mouse for shaping the boundary, setting the particle elements, etc. Further, in order to making our simulation program look more versatile and interesting, we design the interface as Graphical User Interface (GUI). We add one more library, the Graphic Library User Interface (GLUI) [12] to our program. Its control component (spinner, translation, arcball etc.) are designed for 3D modeling. Beside the particle and simulation parameter, we add the viewing, translation and rotation control in order to set a suitable view for user. For this part the designed panel is show in figure xxx

B. Testing the simulation

To test the particle system for gaseous object, we set the system to incompressible fluid or stiff gas which often use as an experiment like the one that use in the CFD system testing. Thus the particle is set to have the density about 1000 kg/m^2 with a small density deviation. For the initializing state, particle properties such as its position, velocities, pressure and thermal energy are randomly set. In order to let the particle have some movement, particle should have a slightly difference in pressure and thermal energy. While the particles are moving, their velocity and movement should not change too much. The Courant-Friedrichs-Lewy criterion called Courant condition is used to check for the simulation stability. This condition can be written as: $u\Delta t / \Delta x \le 1$ [1], where Δt is the time step and Δx is the grid size and u is the maximum particle velocity. If particle moves further than this distance or the time step is too large, the result of integration will give rise to instability during integration.
We set the experiment into three types. First we test the simulation process for the various numbers of particles in order to make sure that the simulating is stable. The system is set without external force. The results are shown in Figure 2, Figure 3 and Figure 5. Second, boundary test is shown in Figure 4, which the container built with a form of funnel. Last, the drop test; we apply the gravity force which is an external force to the system. The result is shown in Figure 4.

All of the tests are performed and run by using PC with Pentium IV, 2.5 GHz system with 512 Mbytes of memory.

VI. Conclusion and Future work

A. Conclusion

This research has presented a particle-based method called Smoothed Particle Hydrodynamic for fluid-like object use gaseous object as a case study. Instead of using the particle a discrete mass, it properties come from its neighbours. The smoothing kernel is an important part that smoothes particles as samples of mass smeared out in space. In modelling the gaseous object, the Navier-Stokes equation is used and it is modelled as an incompressible gas similar to Eulerian system. Although the results are not photo realistic, it can be further proved. Our preliminary results are promising prototypes for the next stage.

B. Future work

We intend to do for the future in 2 ways: First we aim to render our system by creating surface or reconstruction surface such as marching

cubes [6] which the particle can be seen as cloud point. For our experiment at this time, the surface creation and rendering consumes an enormously time. It can make the system to lost the real-time simulation feature. The other is adding more parameters such as angular velocity for the particle spinning phenomena. There are also potential ways to increase the simulation performance. This technique can also be applied to other fluid models such as fire, water wave and so on.

VII. REFERENCES

[1] Desbrun, M. and Cani, M. P., Smoothed particles: A new paradigm for animating highly deformable bodies. In Computer Animation and Simulation '96 (Proceedings of EG Workshop on Animation and Simulation), pages 61–76. Springer-Verlag, Aug 1996.

[2] Foster, N. and Metaxas, D., Modeling the Motion of a Hot, Turbulent Gas. In Computer Graphics Proceedings, Annual Conference Series, 1997, pages 181–188, August 1997.

[3] Gingold, R.A. and Monaghan. Smoothed Particle Hydrodynamics: Theory and Application to Non-spherical Stars, MNRAS 181 (375), 1977.

[4] Lucy, L.B. A Numerical Approach to the Testing of the Fission Hypothesis. Astrophysical Journal 82 (1013), 1977.

[5] Liu, M.B., Xie, W.P. and Liu, G.R., Modeling incompressible flows using a finite particle method, Applied Mathematical Modelling 29 (2005) 1252-1270

[6] Lorensen W.E. and Cline H.E., Marching cubes: A high resolution 3D surface construction algorithm. SIGGRAPH, pages 163–169, 1987.

[7] Monaghan, J.J., Huppert, H.E. and Worster, M.G., Solidification using smoothed particle hydrodynamics, Journal of computational Physics 206 (2005) – 684-705

[8] Morris, J.P., Fox, P.J., Zhu, Y., Modeling low Reynolds number incompressible flows using SPH, J. Comput. Phys. 136 (1997) 214–226.

[9] Muller M., Schirm S. and Teschner M., Interactive Blood Simulation for Virtual Surgery Based on Smoothed Particle Hydrodynamics

[10] Muller M., Charypar D. and Gross M.,Particle-Based Fluid Simulation for Interactive Applications, Eurographics/SIGGRAPH Symposium on Computer Animation (2003)

[11] Oxley, S., Modelling the Capture Theory for the Origin of Planetary Systems., Thesis for the Degree of Doctor of Philosophy, University of York Dept. of Physics, 1999

[12] Rademacher P, GLUI A GLUI-Based User Interface Library, Version 2.0, June 10, 1999, http://glui.sourceforge.net/

[13] Reeves, W.T., Particle systems - a technique for modeling a class of fuzzy objects. ACM Transactions on Graphics 2(2), pages 91-108, 1983.

[14] Roy, T.M., Physically-Based Fluid Modeling Using Smoothed Particle Hydrodynamics, Masters Thesis, University of Illinois at Chicago, 1995.

[15] Shi ,L., et.al., Controllable motion synthesis in a gaseous medium, Visual Comput (2005) 21: 474–487, Springer-Verlag 2005

[16] Shao, S and Lo, E.Y.M, Incompressible SPH method for simulating Newtonian and non-Newtonian flows with a free surface, Advances in Water Resources 26(2003) 787-800

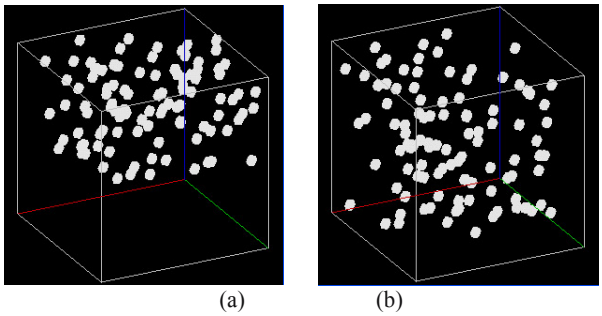
[17] Stam, J.. Stable fluids. In Proceedings of the 26th annual conference on Computer graphics and interactive techniques, pages 121-128. ACM Press/Addison-Wesley Publishing Co., 1999.

[18] Stora, D., Agliati , P.O., Cani, M.P., Neyret, F., & Gascuel, J.D., Animating Lava Flows. Graphics Interface'99, Kingston, Canada, jun 1999.

[19] Witkin, A.: Particle System Dynamics. In: SIGGRAPH '93, Course Notes #60: An Introduction to Physically Based Modeling, pp. C1- C12, August 1993.

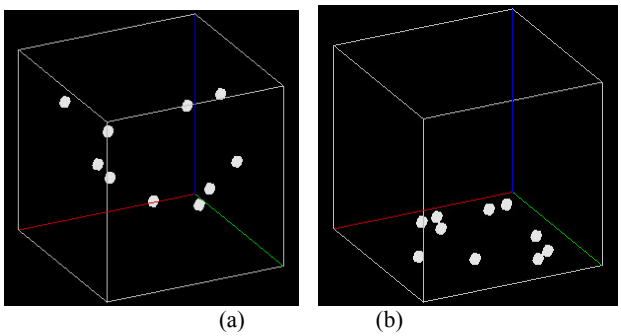
[20] Witkin, A. and Baraff, D., Physically Based Modeling: Principles and Practice Differential Equation Basic., Course Note A SIGGRAPH'97

VIII. APPENDIX



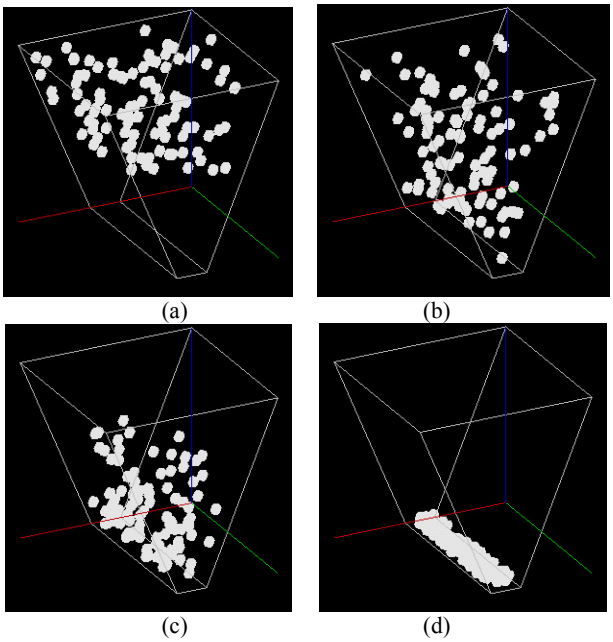
(a) (b)

Figure 2 Gaseous object 100 particles simulation with no external force (a) initial state (b) steady state, particle move with theirs velocity and energy



(a) (b)

Figure 3 10 particle simulation apply with gravity external force (a) initial state (b) final state



(a) (b)

(c) (d)

Figure 4 Boundary test with funnel container (a) Initial state (b) Steady state, (c) and (d) are simulate with external force (drop test): (c) Middle state and (d) Its final state

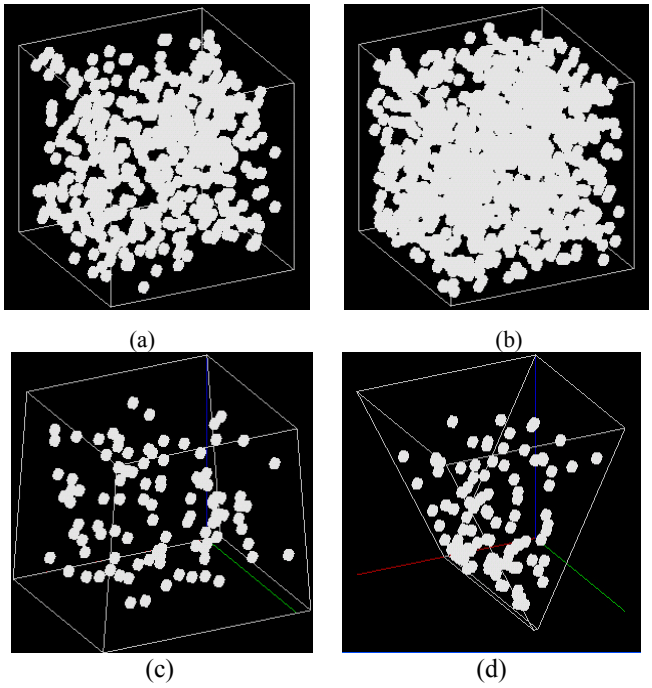


Figure 5 Various type Simulation test (a) 500 and (b) 1000 particles (c) Extend the shape to upward funnel (d) with pivot at the bottom

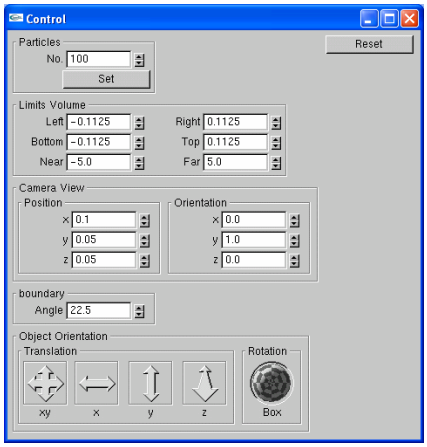


Figure 6 Graphic User Interface for setting the simulation parameters and viewing position