A Framework for Real-Time Animation of Liquid-Rigid Body Interaction

Melih KANDEMİR Tolga ÇAPIN Bülent ÖZGÜÇ
Computer Engineering Department
Bilkent University

Abstract— Both high-end training simulators and computer games lack convincingly realistic and interactive animation of liquids. However, almost none of the game engines support liquid models in the games scenes. This is because of the huge computational time and memory requirements of the existing liquid simulation models. In this paper, an old height-field based model is extended and generalized for use in interactive graphics applications. Although its limitations and lack of accuracy, the model has such little computational requirements that it can be freely used in a game scene containing an intense geometry. The model is extended by a novel technique for the representation of viscosity, which is the most discriminative dynamic property of the fluids. Thus, it is able to simulate different types of liquids with different physical properties in a real-time environment.

Keywords—Liquid Animation, Natural Phenomena, Viscosity

1 Introduction

Recent advances in computational power have made realistic virtual environments possible. Experience has shown that the accuracy of simulated physics in an interactive graphics application is the primary factor that gives the user the sense of realism [3].That's why the research on physics engine development has become one of the hot topics in game industry.

Although recently proposed solutions can detect and handle interactions of rigid bodies in a reasonable computational cost [8], they ignore almost all kinds of interactions between rigid bodies and liquids. This is due to the high computational cost of complex physical equations involved in the physics of liquids. Actually, very successful models exist in the literature for realistic liquid simulation and rendering. However, these models have one very crucial drawback in common: it is impossible to use them in real-time virtual environments.

Most of the proposed models in the literature are based on computational fluid dynamics (CFD). These models depend on iterative calculation of vector fields in a discretized scene, moving some particles inside the volume according to these vector fields and generation of the liquid surface by combining the particles close to the surface. The vector fields in the scene are generally calculated using the Navier-Stokes non-linear equation system [6].

In this paper, a less accurate but a significantly more cost-effective method is used for modeling liquid behavior and surface rendering. The proposed solution is based on the work of O'Brien & Hodgins [1]. The method was not proposed as a real-time solution, because its computational cost was huge for the hardware technology of that time. Since the method can not compete with CFD based methods in terms of accuracy, it has not been widely adopted. However, in today's technology, an extended and generalized version of this method can be used as a cheap and practical liquid model in real-time virtual reality environments with very low computational cost. This is the main idea behind taking this method as the basis.

O'Brien & Hodgins' method offers the representation of the liquid body by a 2D grid of columns, also known as height-fields. The liquid flows among the columns through virtual pipes according to basic laws of hydrostatics. Although the method does not depend on particle-based modeling of fluid dynamics, it still provides a considerable amount of realistic animation of the surface.

Our work proposes a novel method for the representation and controllability of viscosity of the liquid as an extension to O'Brien et al.'s work. Since viscosity is the most discriminative dynamic property of the fluids, such an extension makes the liquid model a generic and configurable one which can be customized for use in an arbitrary real-time virtual environment. An experimental implementation of the model is done and it is verified that the method provides a convincingly realistic liquid behavior with very little computational cost in real-time.

The remaining sections of the paper are organized as follows: in Section 2, other studies on the subject are

This work is supported by European Commission Sixth Framework Program with Grant No: 511568 (3DTV Network of Excellence Project) and Meteksan Sistem Corporation.

presented; in Section 3 the motivation of the presented work is stated; in Section 4, our contribution is briefly described; in Section 5, the model used to represent the liquid behavior is explained; in Section 6, the novel technique for viscosity modeling is proposed; in Section 7, the controllable parameters of the proposed model are defined; in Section 8 the results are evaluated; and in Section 9, a conclusion is given and suggestions for future work are stated.

2 Related Work

The studies about liquid simulation can be classified into two major subfields: i) simulation of large bodies of water, and ii) accurate simulation of complex behavior of small bodies of water.

In the simulation of large bodies of water, there are two main challenges. The first is the realistic simulation and rendering of waves. Tessendorf [9] has a seminal study about integrating the wave models developed by oceanographers into computer graphics. In this study, the illumination properties of ocean water are also described. The other challenge about simulation of large bodies of water is the massive geometry necessary for representing the water surface. As an alternative to classical approaches to resolution management (i.e. level-of-detail based methods), Hinsinger et al [10] have recently introduced an adaptive scheme that sets the geometric resolution according to the viewing distance. Bristol et al [12] offer a method for modeling the effects of wind on liquid surface at real-time. The real-time performance is achieved by reducing the accuracy of liquid model, in a very similar way it is done in this paper. However, these methods represent the liquid surface in terms of wave models. Although these models are successful in representing large liquid bodies and their interaction with natural phenomena, like wind, they are inherently not able to represent the surface structures due to collisions; since they do not represent the liquid volume. In this paper, our concern is the simulation of the interaction of smaller liquid bodies with rigid scene objects.

All methods in the literature about the simulation of complex behavior of small bodies of water depend on discretizing the liquid volume. However, there are several approaches in the way the grid structure is manipulated in order to produce the surface geometry. If physical accuracy is critical, the liquid volume is discretized by a 3D grid and the liquid flow in the voxels are evolved by the Navier-Stokes equations. This provides a particle-based simulation of the liquid behavior, since the Navier-Stokes equations model the dynamics of a single fluid particle. Kass & Miller's study is a very famous early work that uses this approach [5]. Losasso et al. have recently developed a very accurate model which supports very complex phenomena like interaction of combustive gases and liquids [4]. Although these methods provide very accurate and very realistic results, and a complete physical model of the liquids, they require enormous amounts of computational time and memory. Thus, they are inapplicable in real-time applications.

The height-field based methods discretize the liquid volume by a 2D grid of height fields. Each cell in the grid is a column full of water. The liquid behavior is modeled by the basic laws of hydrostatics [1]. In this paper, we used this approach for liquid representation.

Surface rendering is another major problem in liquid simulation, especially if the liquid is represented in a particle-based manner. Although the liquid volume is represented by a grid, a particle system should be used in order to determine the surface polyhedron. There are several approaches to that problem. The most popular approach is to spawn particles only in the boundary voxels and remove the particles when they go out of the boundary voxels. Foster & Metaxas propose such a technique in [6]. In [2], Foster & Fedkiw use dynamic level sets as an extension to isocontour technique [7].

In the recent work of Irving et al. [11], the particle-based approach and height-fields approach are successfully combined. In such a way, large bodies of water could be very realistically simulated. Although the proposed model is one of the most powerful models, it is not practical for real-time applications, because it is prohibitively expensive in terms of computational cost.

3 Motivation

None of the methods described above can directly be used in interactive virtual environments, because their computational costs are huge for a real-time application. Since most of the models in a virtual environment scene are rigid bodies, one of the most critical features of a liquid model is its ability to respond to rigid body interactions with considerable realism. A computationally efficient liquid model with such a feature would be useful in virtual reality environments.

4 Our Contribution

We have extended the O'Brien's method [1] by defining a novel technique for simulation of viscosity. This stands for an additional controllable feature for the model. In fact, the controllability of viscosity is crucial, because viscosity is the most discriminative dynamic property of the fluids. Since the liquid model is generalized in such a way, it can be used in a real-time graphics application.

5 Liquid Representation

In order to model the liquid, the method described in [1] is taken as the basis. The liquid is taken as a two-part system: the main volume and the free surface of the liquid. Spray is not included in the model for performance reasons. It can simply be modeled as a trivial particle system and managing such a system involves a great overhead.

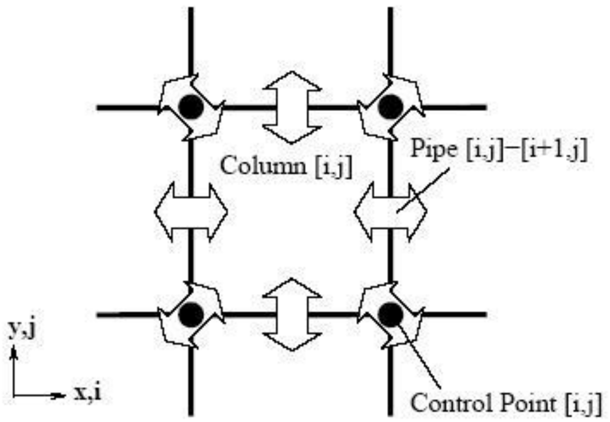

Fig. 1 Each column is connected to other columns in its 8-neighborhood by virtual pipes [1]

5.1 Volume Model

The volume is modeled using a formulation that divides the body into a rectilinear grid of connected columns, as described in [1]. Each column is connected to other columns in its 8-neighborhood by virtual pipes (Figure 1). The liquid flows among the columns through these pipes in order to equalize the total pressure, which is the sum of static and external pressures, on every column. The interaction of a solid object with the liquid surface is animated by changing the external pressures of the columns with which the object interacts.

The equations to determine the flow in the pipes are derived from laws of hydrostatics. These laws have some analogies with Newtonian rigid-body dynamics. In this paper, the term "analogy" is used in order to illustrate the phenomenal resemblances between different concepts. Their physical correctness is not taken into consideration. The novel viscosity model explained in the following section is constructed on these analogies.

Pressure is the fundamental phenomenon of hydrostatics. It is also at the heart of the liquid model adopted in this paper. The definitions of all of the other concepts used in the liquid model directly or undirectly depend on the definition of pressure. In the model, three types of pressure is used; the definitions of which are given below :

$$P_{total} = P_{static} + P_{external} \quad (5.1)$$

$$P_{static} = h\rho g + P_{atmosp\ here} \quad (5.2)$$

In (5.2), ρ is the density of the liquid, g is the gravitational constant, $P_{atmosp\ here}$ is the atmospheric pressure, and $P_{external}$ is the total pressure applied by an external factor (e.g. another object, wind etc.). These three properties are configurable constants. h is the column height. The value of this parameter changes as the liquid surface changes shape along the animation time. When the liquid is at rest, each column has equal total pressure. When a rigid body interacts with the liquid surface, it exerts external pressure to the columns with which it interacts. This breaks the equilibrium and the liquid starts to flow from the columns with high total pressure to the ones with low total pressure. In such a way, the liquid responds to the changes in external pressure by rearranging the static pressures of the columns by reorganizing their heights. The flow stops and equilibrium is regained when all columns have equal total pressure.

The equations given below define the physical laws of the liquid flow.

Pressure Difference→External Force : Pressure difference triggers the motion of the liquid through the columns. Its direction and magnitude determines the direction and magnitude of the flow. As an object starts to move when the magnitude of the total external force exerted on that object is greater than 0, the liquid starts to flow when the magnitude of the pressure difference between two neighbor columns gets greater than 0 after the influence of an external factor. Hence, it is analogous to external force in Newtonian dynamics. Here is the formal definition of external force used in the model:

$$F_{external} = \delta P = P_{total_{this}} - P_{total_{neig\ hbor}} \quad (5.3)$$

Flow→Velocity : The flow difference between consecutive timesteps equals to the timestep times the total force on the column times a constant, k. This is the same as velocity-force dependency in Newtonian dynamics ($\delta v = \frac{F}{m} \cdot \delta t$) where the velocity difference equals to the timestep times the total force times the reciprocal of the mass. Hence, flow is analogous to velocity. The following is the flow update equation used in the model:

$$Q_t = D \cdot Q_{t-1} + \delta t \cdot k \cdot (F_{external} - F_{viscosity}) \quad (5.4)$$

Here, D is the "vanishing quotient" which determines the stabilization time of the surface after an interaction. If D=1, the surface will never stabilize and it will finally diverge after some time, since the flow in the current step will always be augmented to the flow in the next step. In order to let the surface stabilize, D should be set to a value between 0 and 1. In our implementation, it is set to 0.93. This value is chosen as the one that gives the most realistic results after experimentation of a set of different values. This quotient can be regarded as a configurable parameter and it can be set to different values for various purposes. $F_{viscosity}$ is the force due to viscosity. This is analogous to frictional force in Newtonian dynamics. The details about the formula of viscosity force and its derivation are given in Section 6.

In (5.4), k is a configurable constant as given below:

$$k = \frac{A_{pipe} \cdot g}{L_{pipe}} \quad (5.5)$$

where $\boldsymbol{A_{pipe}}$ is the area and $\boldsymbol{L_{pipe}}$ is the length of the pipe. The effect of this constant on liquid characteristics is explained in Section 7.

Volume→Distance : The volume difference in a column between consecutive timesteps is equal to the flow difference times the timestep. It is analogous to velocity-distance relationship in Newtonian dynamics ($\delta x = \delta v \cdot \delta t$) where the displacement equals to the timestep times the velocity difference. Hence, volume is analogous to distance. Here is the volume update equation used in the model :

$$V_t = V_{t-1} + \delta t \cdot (Q_t - Q_{t-1}) \quad (5.6)$$

After the volume at the next timestep is calculated, the new height of the column can be determined using the following formula:

$$h_t = \frac{V_t}{dx \cdot dy} \quad (5.7)$$

where dx and dy are the lengths of a column in the x and y axes respectively. They are both configurable constants.

5.2 Surface Model

The surface model also consists of a rectilinear grid of control points as described in [1]. All control points of all cells define the surface geometry. The vertical position of a particular control point is determined by averaging the heights of the four columns surrounding that grid point:

$$z_{ij} = \frac{h_{i,j} + h_{i,j+1} + h_{i+1,j} + h_{i+1,j+1}}{4} \quad (5.8)$$

5.3 External Objects

External pressure is the only parameter in the liquid model that can be changed by the external factors, such as collisions with rigid bodies. When an object collides with the liquid surface, all the columns that collide with the object are applied an external pressure of magnitude:

$$P_{E_{ij}} = -\frac{f_e}{4 d_x d_y} \quad (5.9)$$

where f_e equals to the sum of the weight of the external object and other external forces that act on the object; and d_x and d_y are lengths of the column edges.

6 Viscosity Representation

Viscosity is the most important dynamic characteristic of liquids in terms of computer animation, because it determines how the liquid responds to shape changes. However, the original version of this model does not provide a representation of viscosity. Here, a method for representing viscosity in this model is proposed.

In computational fluid dynamics, viscosity is a quotient that describes a fluid's internal resistance to flow. In the model used here, this corresponds to an internal force which acts against liquid flow among columns. Intuitively, viscosity force could be defined as linearly dependent to δP. This would be a direct analogy to frictional force in Newtonian dynamics. However, in the model used here, this would be the same as narrowing the virtual pipes between the columns. The experiments have shown that applying such a viscosity force does not even generate a visible effect. It just slows down the stabilization of the liquid surface after interaction.

For that reason, higher-order dependencies are experimented. The best visual result is obtained when the viscosity force is cubically dependent to the pressure difference.

The viscosity force is defined as follows:

$$F_{viscosity} = \delta P^3 \cdot V; \quad \delta P \leq \sqrt{\frac{1}{V}} \quad (6.1)$$

$$F_{viscosity} = \delta P \cdot V; \quad \delta P > \sqrt{\frac{1}{V}} \quad (6.2)$$

Here, V is the viscosity quotient. It has a value between 0 and 1. Since viscosity force is cubically dependent to δP, its magnitude exceeds the external force if δP is greater than $\sqrt{\frac{1}{V}}$. Since an internal force-like frictional force- can not be greater than an external force, the viscosity force remains equal to the external force in that case (Figure 2,3). Figure 2 and Figure 3 also show how viscosity can be controlled by changing V.

7 Controllability

As the paper title mentions, the method described in the paper is proposed as a framework. Generality and controllability are crucial properties of framework models in computer animation. A more controllable model is also a more general one. Thus, it is important to criticize at what extent the model proposed here is controllable. Here are the controllable parameters that the model provides:

Density : The density of the liquid can be configured.

Pipe length & Pipe Area : The length and the area of the pipes determine the time the liquid returns to steady state. They have effect on the stability of the surface. However, they don't have any effect on the visual properties of the surface.

Timestep : The model successfully works for both large and for small timesteps.

Viscosity : The viscosity of the liquid can be controlled by a single parameter. As the value of this parameter increases, the liquid gets more viscous.

Resolution : The model works both at coarse and at fine resolution without any side effects.

Height : The model successfully works for both deep and shallow liquids.

External Force : By applying external forces to particular columns, many kinds of interactions with rigid bodies can be simulated.

Atmospheric Pressure : The atmospheric pressure can be set to represent different atmospheric conditions.

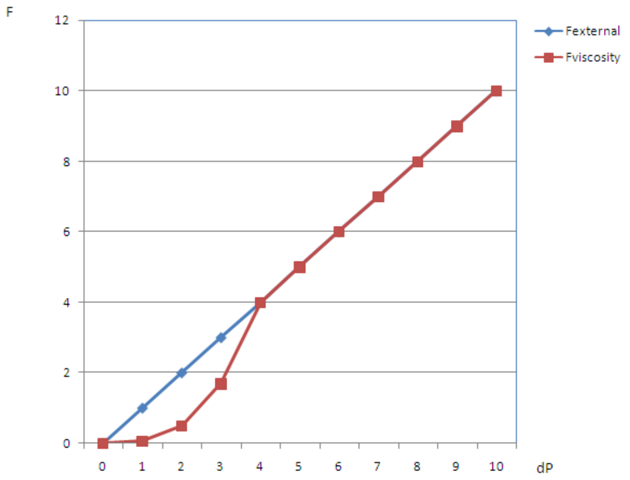

Fig. 2 External force (F) against Viscosity Force (Fvis) when V=1/16. Two forces become equal at dP=4. Fvis remain equal to F if dP is greater than 4.

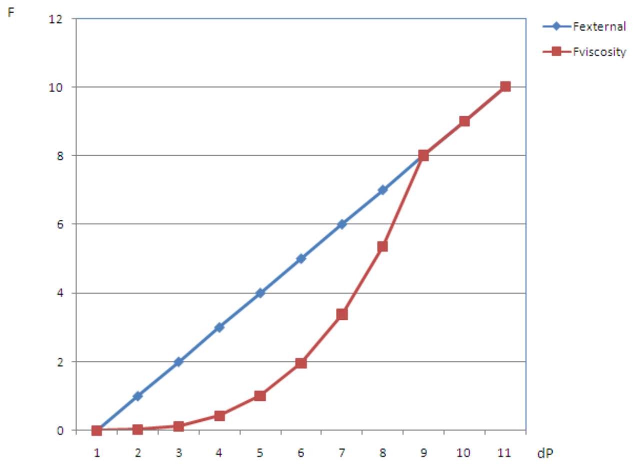

Fig. 3 External force (F) against Viscosity Force (Fvis) when V=1/64. Two forces become equal at dP=8. Fvis remain equal to F if dP is greater than 8.

Column Sizes : Column sizes can be configured to determine the size of the entire liquid body regardless of the resolution.

8 Results

The implementation is done in C++ and OpenGL is used as the graphics library. The system architecture is given in Figure 4. The system is organized in the scene graph structure. The “scene” is the root model. It manages the general operations related to the entire scene. The “liquid tank” model includes the “tank geometry” and the “liquid body” geometry. The liquid body consists of columns. There may be additional objects in the scene. As in a typical scene graph structure, every node in the graph solves and renders its own geometry, and manages the child nodes.

All the capabilities of the model stated in this paper are tested on several desktop computers with different hardware configurations and they are validated.

The performance evaluations clearly show that the model is able to run at real-time without any difficulty. The experimental software uses approximately 10MB RAM. When the grid resolution for the liquid body is 100x100 and the environment is illuminated by Gouraud shading, the frame rate is between 20-25 frames/second. There is no GPU usage in the software. Some example outputs are given below. In these outputs, the liquid surface is rendered as wireframe in order to highlight the surface geometry.

In Figure 5., a light object is dropped into a liquid tank. The columns at the place the object falls get lower because the external pressures of these columns increase due to object contact. The external force is exerted until the object hits the ground. However, before the object hits the ground, the columns start to stabilize. This is because the flow from the previous frame is partially transferred to the next frame. The percentage of this transmission is determined by the vanishing quotient (D). This quotient can be considered as the frictional force inside the liquid. When the object reaches at the bottom, it suddenly stops to exert the external force. This suddenly reduces the total pressure of the contacting columns and a splashing effect occurs. After a while, the liquid surface returns to the steady state.

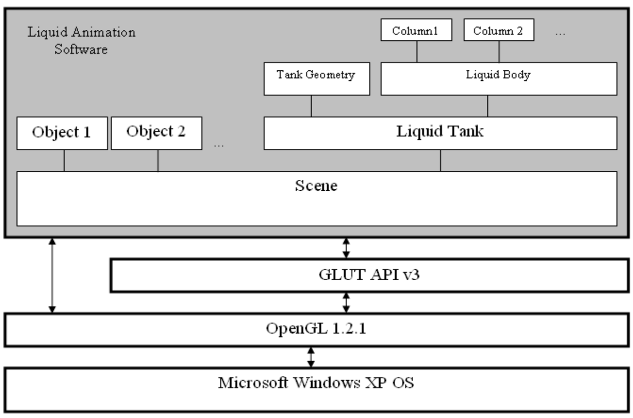

Fig. 4 The system architecture

In Figure 6, a heavy object is dropped into the tank. This time, a stronger splashing effect occurs. The contacting columns oscillate with higher amplitude and they return to the steady state a bit later.

In Figure 7, an object with horizontal velocity is dropped into a liquid with low viscosity. The surface geometry correctly reflects the horizontal velocity of the contacting object by producing an asymmetric geometry at the place of contact. The liquid with low viscosity gives strong response to the contact.

In Figure 8, an object with horizontal velocity is dropped into a liquid with high viscosity. The liquid with high viscosity gives weak response to the contact.

9 Conclusion and Future Work

In this paper, a practical liquid model is extended with the representation of viscosity. Since the original model does not support the representation of such a crucial property, it could not be utilized as a generic model that could be used for representing different types of liquids. Controllability of viscosity makes the model configurable for different real-time virtual environments. According to different viscosity quotients, the liquid body responds to rigid body interactions in different ways.

One feature of the model is that it provides easy integration to other models. Since the liquid surface is represented as height-fields, sophisticated height-field based methods developed for ocean animation [10] can be integrated to this model. Such a hybrid model can be used for both animating the ocean waves and handling rigid body interactions with the ocean water.

As another future work, the surface rendering mechanism can be improved. In the current surface model, all the vertices are connected by straight lines. Better surface geometry may be obtained if the vertices are connected by curves. Bézier or B-Spline interpolation techniques could be used for calculation of these curves. The liquid surface can be rendered further realistic by mapping dynamically generated textures whose structure vary with the heights of the patch vertices on which a particular texture is being mapped.

References

[1] O'Brien J., Hodgins J. "Dynamic Simulation of Splashing Fluids",Computer Animation 1995, 198–205.

[2] Foster N. , Fedkiw R. "Practical Animation of Liquids", SIGGGRAPH 2001, p.15-22.

[3] Chris Hecker "Physics, The Next Frontier", Game Developer Magazine, Oct/Nov 1996

[4] Losasso F., Shinar T., Selle A., Fedkiw R., "Multiple InteractingFluids", SIGGGRAPH 2006, p.812-819.

[5] Kass M., Miller G. "Rapid, Stable Fluid Dynamics For Computer Graphics", SIGGRAPH 1990, p.49-57.

[6] Foster N., Metaxas D., "Realistic Animation of Liquids", Graphical Models and Image Processing 1996, p.471-483.

[7] Bloomenthal J., Bajaj C., Blinn J., Cani M.-P., Rockwood, A., Wyvill, B., and Wyvill, G., "Introduction to Implicit Surfaces", 1997, Morgan Kaufmann publishers Inc.

[8] Geiger B., "Real-Time Collision Detection and Response for Complex Environments", CGI 2000, p.105

[9] Tessendorf, J. "Simulating Ocean Water", SIGGRAPH 1999, Course Notes, Addison-Wesley.

[10] Hinsinger D., Neyret F., Cani M.-P., "Interactive Animation of Ocean Waves", SIGGRAPH 2002, p.161-166.

[11] Irving G., Guendelman E., Losasso F., Fedkiw R., "Efficient Simulation of Large Bodies of Water by Coupling Two and Three Dimensional Techniques", SIGGRAPH 2006, p.805-811

[12] Bristol A., Varsamidis T., "Real-Time Modelling of the Action of Wind on Liquid Surfaces", EUROGRAPHICS UK 2006 Theory and Practice of Computer Graphics, p.111-114

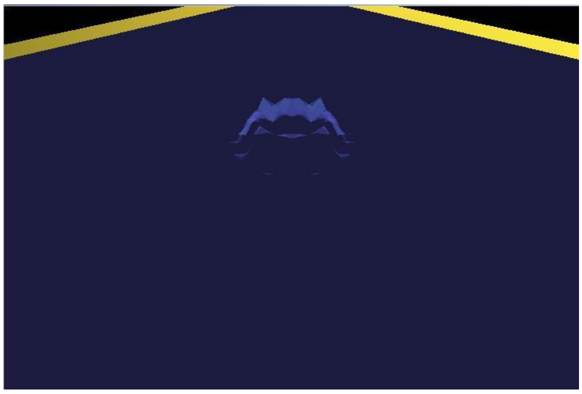

Fig .5a A light rigid ball is dropped into a tank filled with a liquid.

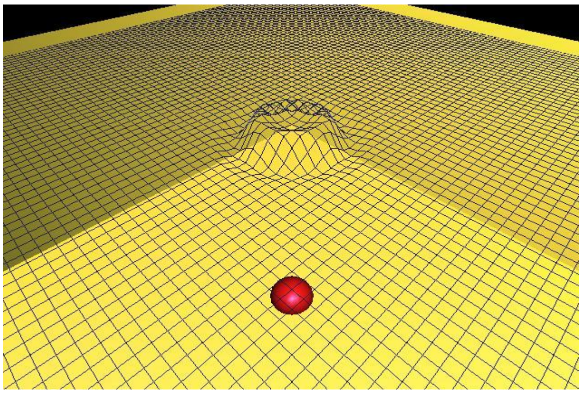

Fig .5b A light rigid ball is dropped into a tank filled with a liquid. The surface geometry is rendered as wireframe

Fig. 6a A heavy rigid ball is dropped into a tank filled with a liquid. The wireframe view of the surface geometry is above.

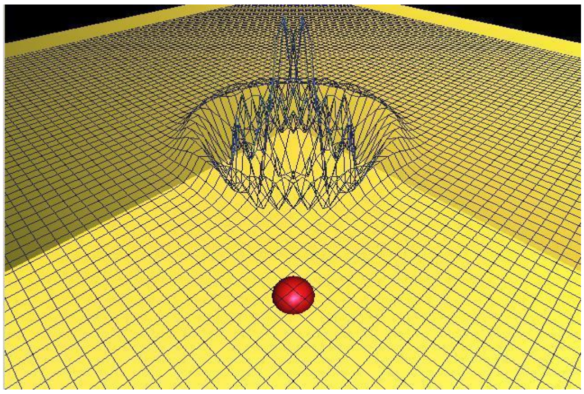

Fig. 6b A heavy rigid ball is dropped into a tank filled with a liquid. The surface geometry is rendered as wireframe.

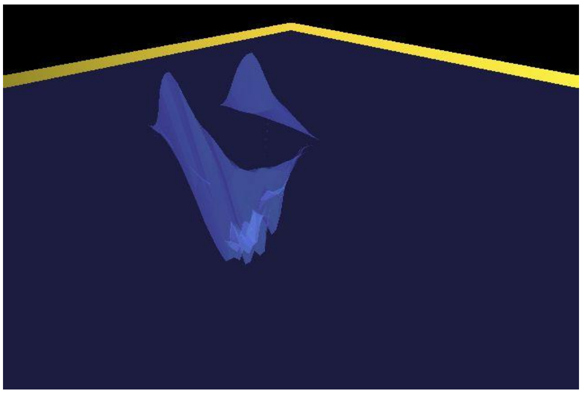

Fig. 7a A rigid ball with horizontal velocity interacts with a liquid with low viscosity.

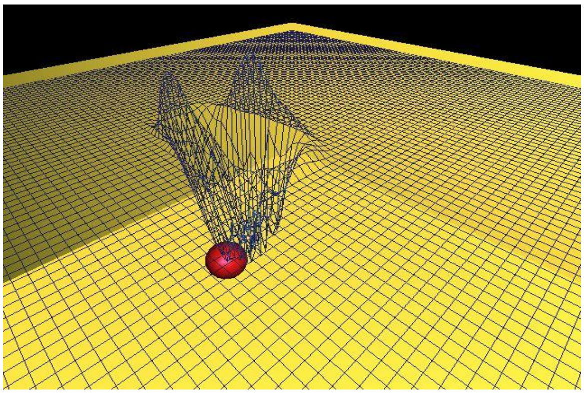

Fig. 7b A rigid ball with horizontal velocity interacts with a liquid with low viscosity. The surface geometry is rendered as wireframe.

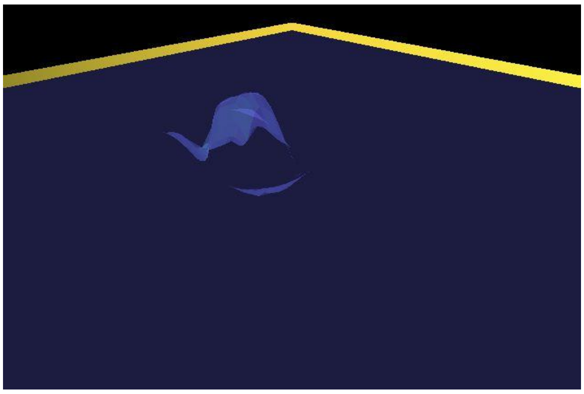

Fig. 8a A rigid ball with horizontal velocity interacts with a liquid with low viscosity.

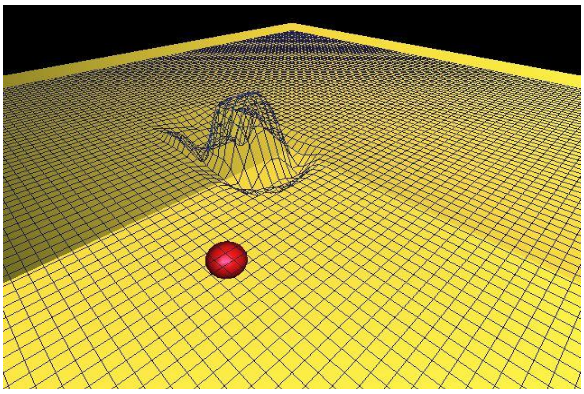

Fig. 8b A rigid ball with horizontal velocity interacts with a liquid with high viscosity. The surface geometry is rendered as wireframe.