МАТЕМАТИЧНЕ ТА КОМП'ЮТЕРНЕ МОДЕЛЮВАННЯ

МАТЕМАТИЧЕСКОЕ И КОМПЬЮТЕРНОЕ МОДЕЛИРОВАНИЕ

MATHEMATICAL AND COMPUTER MODELING

UDC 004.93

APPLICATION OF R-FUNCTIONS METHOD AND SMOOTHED PARTICLE HYDRODYNAMICS FOR FLUID SIMULATION

Artiukh A. V. – PhD, Associate Professor of the Department of Applied Mathematics, Kharkiv National University of Radio Electronics, Kharkiv, Ukraine.

Kozyrenko S. I. – PhD, Associate Professor, Associate Professor of the Department of Applied Mathematics, Kharkiv National University of Radio Electronics, Kharkiv, Ukraine.

Shabalin O. O. – Researcher of the Department of Applied Mathematics, Kharkiv National University of Radio Electronics, Kharkiv, Ukraine.

Lytvyn I. R. – Student of the Department of Applied Mathematics, Kharkiv National University of Radio Electronics, Kharkiv, Ukraine.

ABSTRACT

Context. Existing fluid simulation methods have several disadvantages and can be improved with the help of new approaches to the solution of problems of computational fluid dynamics, which confirms the relevance of the work.

Objective. The goal of the work is to improve existing methods of mathematical modeling of fluid based on smoothed particle hydrodynamics and R-functions method.

Method. A new approach of joint use of smoothed particle hydrodynamics, marching cubes and R-functions method is proposed. Smoothed particle hydrodynamics helps to simulate fluid movement in real time. The method considers fluid as a discrete number of sample points (particles), which have mass, velocity, position and physical field quantities (pressure, temperature, mass-density, etc.). The R-functions method allows to solve the inverse problem of analytic geometry: finding an analytical equation of a 2D (3D) object based on its geometrical representation. Using the obtained equation, one can simply detect a particle collision with the object boundary and plot the object surface with the help of marching cubes algorithm. The suggested method allows to achieve good simulation quality and to perform all needed calculations and rendering in real time.

Results. Computational experiments for the problem of fluid simulation were carried out. Various numbers of particles were used. Different kinds of objects were put into the considered region in order to investigate the fluid behavior.

Conclusions. The results of visual simulations allow us to say that the obtained approach works as expected. Therefore, this method can be applied to several problems of fluid simulation where the collision detection with arbitrary objects is considered. Further research may be devoted to the optimization of neighbor-search algorithm, to performing all calculations in graphics processing unit or to taking into account other physical quantities.

KEYWORDS: Navier-Stokes equations, fluid simulation, R-functions method, smoothed particle hydrodynamics, marching cubes algorithm.

ABBREVIATIONS

SPH is the Smoothed Particle Hydrodynamics;
FDM is the Finite Difference Method;
RFM is the R-Functions Method;
CPU is the Central Processing Unit;
GPU is the Graphics Processing Unit.

NOMENCLATURE

x, y, z are coordinates in the Cartesian system;
t is a time;
Ω is a flow domain;
$\partial\Omega$ is a boundary of a body;
χ is a characteristic function of Ω;
Σ is an object represented by inequality;
n is a number of sub-domains describing Ω;
ρ is a density;
m is a particle mass;
l is a fluid threshold;
σ is a surface tension;

DOI 10.15588/1607-3274-2019-3-3

h is a support radius;
k is a gas stiffness;
p is a pressure;
$\mathbf{v}$ is a field of fluid velocities, $\mathbf{v} = (v_x, v_y, v_z)$;
$\mathbf{a}$ is an acceleration vector, $\mathbf{a} = (a_x, a_y, a_z)$;
ν is a coefficient of viscosity;
$\mathbf{f}$ is a field of forces;
$\mathbf{F}$ is a field of internal and external forces;
$\omega = 0$ is a normalized equation of $\partial\Omega$;
ω is a sufficiently smooth function describing the geometry of the domain Ω;
W is a smoothing (kernel) function;
$\mathbf{r}$ is a random point in Ω;
$F(\mathbf{r})$ is an arbitrary function;
V is a volume;
R is a universal gas constant;
T is a temperature;
κ is a Boltzmann constant;
$\mathbf{n}$ is a inward surface normal of fluid;
g is a gravity constant;
c_R is a coefficient of restitution;
c is a smoothed value of color field;
∇ is a gradient operator;
N is a number of particles.

INTRODUCTION

This work introduces a new method of fluid simulation based on the joint use of SPH and RFM. Obviously, there is a huge scope of applications of fluid simulation: poring water, steam, ocean waves, simulations in astrophysics, medicine etc. Therefore, new approaches that can improve simulation are welcomed.

The Navier-Stokes equations are basis of fluid simulation. Commonly, two approaches are used for obtaining its solution: grid-based (Eulerian) and particle-based (Lagrangian) methods. Each of them has its pros and cons.

The Eulerian approach assumes that fluid is composed of molecules and allows obtaining a solution which satisfies almost all fluid physical properties. The Navier-Stokes equations are solved using FDM. As a result, we have a grid-based approach which imposes restrictions on solution existence only inside the grid domain and remains a bunch of computational recourses.

On the other hand, the Lagrangian method allows to simplify the Navier-Stokes equations. The main assumption is that fluid consists of a finite number of particles with fixed mass. Each particle should be considered as a fixed amount of fluid with appropriate physical quantities. This assumption leads to decreasing computational cost and significantly simplifies evaluations on each time step. However, this method has two main disadvantages: it is difficult to treat boundary conditions and computational cost increases with the number of particles. SPH is one of the mesh-free Lagrangian methods.

Finally, RFM allows to solve easily one of the most complicated problems: modeling collisions with 3D objects. It is tedious to describe a complex object which consists of few simpler objects. RFM provides a solution how to get an analytical equation of such objects and a simple way of collision detection.

The object of study is an unsteady flow of viscous incompressible fluid in three-dimensional space, and **the subject of study** – the mathematical apparatus for describing such flows.

The purpose of this paper is to develop a new method based on SPH and RFM of getting an approximate solution of the Navier-Stokes equations. The main advantage of these methods is that they can be easily implemented and the solution can be obtained in real time of a differential equation can be obtained in an analytical form. These benefits allow one to perform fluid simulations without any restrictions on the target domain.

1 PROBLEM STATEMENT

The unsteady flow of a viscous incompressible fluid in the space region $\Omega \in \mathbb{R}^3$ is described by the Navier-Stokes equations which allow to find the flow speed $\mathbf{v}$ when all other flow characteristics are known [1, 2]:

$$\rho\left(\frac{\partial}{\partial t} + \mathbf{v}\cdot\nabla\right)\mathbf{v} = -\nabla p + \nu\nabla\cdot(\nabla\mathbf{v}) + \mathbf{f}_{\text{external}}, \quad (1)$$

$$\nabla\cdot\mathbf{v} = 0, \quad (2)$$

where $\rho = \text{const}$, $\mathbf{v}: \Omega\times[0,t] \to \mathbb{R}^3$, $p: \Omega\times[0,t] \to \mathbb{R}$.

Taking into account the Lagrangian approach, where fluid is represented by the fixed number of particles, the continuity or mass conservation equation (2) is guaranteed and can be omitted. Furthermore, the particles define the fluid completely and therefore any physical quantity depends on time only. Finally, the Lagrangian formulation of the Navier-Stokes equations (1) can be read as:

$$\rho\frac{d\mathbf{v}}{dt} = -\nabla p + \nu\nabla^2\mathbf{v} + \mathbf{f}_{\text{external}}. \quad (3)$$

In the right hand side of (3) there are two types of forces: internal (pressure, viscosity) and external (gravity). Let us combine them into $\mathbf{F} = \mathbf{f}_{\text{internal}} + \mathbf{f}_{\text{external}}$.

Then, taking into consideration the denotation for the forces, for particle i equation (3) can be rewritten as:

$$\mathbf{a}_i = \frac{d\mathbf{v}_i}{dt} = \frac{\mathbf{F}_i}{\rho_i}. \quad (4)$$

2 REVIEW OF THE LITERATURE

At present, SPH is widely used in solving problems of fluid simulation. Paul Cleary and others [3] use SPH for modeling cast systems. They manage to model cast systems in 3D. The geometric complexity of the cast systems leads to the fact that it is difficult to represent an object as an analytical expression.

Tahakiro Harada, Seiichi Koshizuka and Yoichiro Kawaguchi [4] propose and improve computation model of wall boundary in SPH. The method uses a distance function calculated from a polygon model as a wall

DOI 10.15588/1607-3274-2019-3-3

boundary. Generally, particle methods calculate a wall boundary by converting it to wall particles. Thus, the shape of the boundary gets much complex and the number of wall particles increases.

Zlii Dai and others [5] apply SPH to simulate rapid landslide motion across 3D terrain. They present landscapes as different amount of particles for each landscape.

A. Barreiro and others [6] use SPH for the coastal engineering problems. They represent boundaries as discrete set of boundary particles that exert a repulsive force on the fluid particles when they approach.

Todd B. Silvester and Paul W. Cleary [7] bring into play SPH to simulate wave-structure interaction. For the physical boundaries they use two separate approaches. In the first instance, the boundaries are defined by a single line of stationary particles. The second approach constructs the wall boundaries from two or more layers of stationary fluid particles.

Randles and Libersky [8] consider boundary particles whose properties, including their position, vary each time step. When a real particle is close to a boundary (at a distance shorter than the kernel smoothing length) then a virtual (ghost) particle is generated outside of the system, constituting the image of the incident one.

The R-function method allows us to describe any complex geometric objects in an analytical form in 2D or 3D, which will help us easily detect collisions of particles with the object that will enhance existing methods and reduce total amount of particles used in fluid simulation.

3 MATERIALS AND METHODS

Inverse problem of analytic geometry and RFM.

Let region $\Omega \in \mathbb{R}^3$ be given with the piecewise-smooth boundary. It is necessary to construct the function $\omega(x, y, z)$, which should be positive inside Ω, negative outside Ω, and equal to zero at its boundary $\partial\Omega$. I.e. equation $\omega(x, y, z) = 0$ in implicit form defines the locus for points lying on the boundary $\partial\Omega$ [9]. It is required that $\omega(x, y, z)$ should be an elementary function and have a unique analytic representation. This function can be easily constructed be means of RFM.

RFM was developed by V. L. Rvachev in [10]. The main idea is that among the functions of a continuous argument there are such functions (R-functions), which approximate to the logic algebra functions. These functions form a set that has a non-empty intersection with a set of elementary functions. Thus, each R-function was associated with the corresponding Boolean function, which further made it possible to use the developed apparatus for constructing the solution of the inverse problem of analytic geometry.

Let us denote a characteristic function corresponding to a region Ω_i as $\chi_i(\omega_i(x, y, z) \geq 0)$, $\Omega = \bigcup \Omega_i$, $i = \overline{1, n}$. Then one can construct a predicate $\chi = F(\chi_1, ..., \chi_n) = F((\omega_1 \geq 0), ..., (\omega_n \geq 0))$. Therefore, the region Ω can be constructed by means of the Boolean algebra.

Hereby, an object Σ exists and can be represented as a superposition of objects $\Sigma_1, ..., \Sigma_n$ using logic operations: negation, disjunction and conjunction. Currently, a large number of R-functions systems is known [11], but in present work the simplest one is used – the $\mathcal{R}_0$ system, which has the following form

$$\bar{u} \equiv -u, \quad u \vee_0 v \equiv u + v + \sqrt{u^2 + v^2},$$
$$u \wedge_0 v \equiv u + v - \sqrt{u^2 + v^2}.$$

Let us consider some examples.

Example 1. A cube defined by its vertices $O(0,0,0)$, $A(0,0,3)$, $B(0,3,0)$, $C(0,3,3)$, $D(3,0,0)$, $E(3,0,3)$, $F(3,3,0)$, $G(3,3,3)$. The cube can be represented as

$$\Omega = \Sigma_1 \wedge \Sigma_2 \wedge \Sigma_3,$$

where $\Sigma_1 = (z(3 - z) \geq 0)$ – the region bounded by planes $z = 0$ and $z = 3$, $\Sigma_2 = (x(3 - x) \geq 0)$ – by $x = 0$ and $x = 3$, $\Sigma_3 = (y(3 - y) \geq 0)$ – by $y = 0$ and $y = 3$. Then the boundary equation has the form

$$\omega(x, y, z) \equiv [z(3 - z) \wedge_0 x(3 - x)] \wedge_0 y(3 - y) = 0.$$

Example 2. A pawn (figure 1a) can be described as

$$\Omega = (\Sigma_1 \wedge \Sigma_2 \wedge \Sigma_3) \vee (\Sigma_4 \vee \Sigma_5).$$

Here $\Sigma_1 = (\omega_1 \geq 0)$, $\omega_1 = 0.25 - (x - 1.5)^2 - (y - 1.5)^2$ – a right circular cylinder, $\Sigma_2 = (\omega_2 \geq 0)$, $\omega_2 = z(1 - z)$ – a region between planes $z = 0$ and $z = 1$, $\Sigma_3 = (\omega_3 \geq 0)$, $\omega_3 = -20((x - 1.5)^2 + (y - 1.5)^2) + 1 + 10(z - 0.75)^2$ – a hyperboloid, $\Sigma_4 = (\omega_4 \geq 0)$, $\omega_4 = 0.125 - (x - 1.5)^2 - (y - 1.5)^2 - 20(1 - z)^2$ – an ellipsoid, $\Sigma_5 = (\omega_5 \geq 0)$, $\omega_5 = 0.05 - (x - 1.5)^2 - (y - 1.5)^2 - (1.25 - z)^2$ – a sphere. The boundary equation $\partial\Omega$ can be represented as:

$$\omega(x, y, z) \equiv [\omega_1 \wedge_0 \omega_2 \wedge_0 \omega_3] \vee_0 [\omega_4 \vee_0 \omega_5] = 0. \quad (5)$$

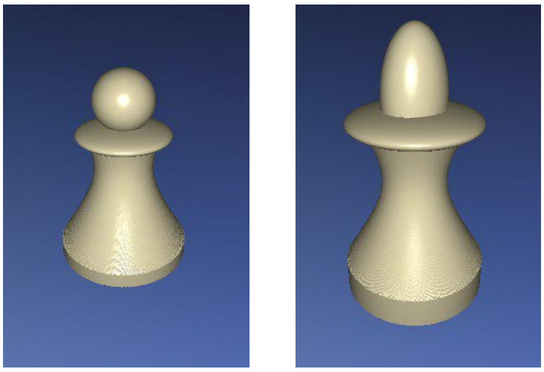
Example 3. A bishop (figure 1b) can be described as

$$\Omega = (\Sigma_1 \wedge \Sigma_2 \wedge \Sigma_3) \vee (\Sigma_4 \vee \Sigma_5),$$

where $\Sigma_1 = (\omega_1 \geq 0)$, $\omega_1 = 0.25 - (x - 1.5)^2 - (y - 1.5)^2$ – a right circular cylinder, $\Sigma_2 = (\omega_2 \geq 0)$, $\omega_2 = z(1.25 - z)$ – a region between $z = 0$ and $z = 1.25$, $\Sigma_3 = (\omega_3 \geq 0)$, $\omega_3 = -20((x - 1.5)^2 + (y - 1.5)^2) + 1 + 10(z - 0.85)^2$ – a hyperboloid, $\Sigma_4 = (\omega_4 \geq 0)$, $\omega_4 = 0.2 - (x - 1.5)^2 - (y - 1.5)^2 - 20(1.25 - z)^2$ – an ellipsoid, $\Sigma_5 = (\omega_5 \geq 0)$, $\omega_5 = 0.2 - 5(x - 1.5)^2 - 4(y - 1.5)^2 - (1.4 - z)^2$ – an ellipsoid. The boundary equation $\partial\Omega$ has the form:

$$\omega(x, y, z) \equiv [\omega_1 \wedge_0 \omega_2 \wedge_0 \omega_3] \vee_0 [\omega_4 \vee_0 \omega_5] = 0. \quad (6)$$

DOI 10.15588/1607-3274-2019-3-3



a b

Figure 1 – Drawn regions: a – pawn, b – bishop

Marching cubes algorithm.

Marching cubes is a well-known algorithm that was originally developed by Lorensen and Cline [12] and it is used for plotting complex shapes in 3D. It allows to extract a polygonal mesh of a shape surface. Then one can pass this mesh to OpenGL and plot the shape surface.

Suppose we have a shape and it can be described by the boundary equation $F(x, y, z) = 0$. Firstly, we split the domain into a uniform grid of cubes. Then for each vertex of a cube we check if it is outside or inside the shape checking the sign of the function at this point. After that we look into the look up table and depending on how many vertices are marked as inside the shape we choose an appropriate line to be plotted [13].

Therefore, the application of marching cubes algorithm in our work is to plot 3D objects by equation obtained with the help of RFM.

Smoothed particle hydrodynamics.

SPH has various applications [14–17]. It is based on the kernel interpolation, which is used to approximate mathematical calculations with acceptable threshold of approximation. The special case of applying is to define density – the initial value of expressing further equations. The smoothing function, which is placed for specifying connection between particles, must confirm conditions:

$$\int_{\Omega} W(\mathbf{r}, h)d\mathbf{r} = 1, \quad W(\mathbf{r}, h) \geq 0,$$

$$W(\mathbf{r}, h) = W(-\mathbf{r}, h), \quad W(\mathbf{r}, h) = 0 \Leftrightarrow \|\mathbf{r}\| > h .$$

For any function $F(\mathbf{r})$ smoothing interpolation $F_s(\mathbf{r})$ can be found by using kernel convolution W :

$$F_s(\mathbf{r}) = \int F(\mathbf{r}')W(\mathbf{r} - \mathbf{r}', h)d\mathbf{r}' .$$

Because of lightweight computational complexity kernel function is replaced by spline, which is annulled beyond the pale of two smoothing radii. It means that any physical magnitude is composed by values of neighbour particles, which are within the ambit of smoothing area. Spline usage requires symmetry and sufficient smoothness that guarantees twice continuously differentiable kernel. Corresponding spline takes the following form:

$$W(\mathbf{r}, h) = \frac{315}{64\pi h^9} \begin{cases} \left(h^2 - \|\mathbf{r}\|^2\right)^3, & 0 \leq \|\mathbf{r}\| \leq h, \\ 0, & \|\mathbf{r}\| > h. \end{cases}$$

In case when the variety of points is known and its mass and density are predefined, the overall volume of final element can be approximated by changing integral with the following sum:

$$F_s(\mathbf{r}) = \sum_j \frac{\mathrm{m}_j}{\rho_j} F(\mathbf{r}_j) W(\mathbf{r} - \mathbf{r}_j, h) . \tag{7}$$

An important thing is that approximation $F_s(\mathbf{r})$ is defined everywhere and differentiable because of kernel differentiability. In case $F(\mathbf{r}) = \rho(\mathbf{r})$ the approach of density can be expressed with

$$\rho_s(\mathbf{r}) \approx \sum_j \mathrm{m}_j W(\mathbf{r} - \mathbf{r}_j, h),$$

which depends only on mass and coordinate of a particle.

The length of a smooth radius may depend on $h = h(\mathbf{r})$, which is a precondition of variety calculations in density sample. There are two approaches: «scattering» and «gathering». The first one has a great advantage: a volume integral in the smoothed area returns total mass:

$$\int \rho(\mathbf{r})d\mathbf{r} = \sum_i \mathrm{m}_i .$$

On the other hand, where kernel is described by $W(\mathbf{r}_i - \mathbf{r}_j, h(\mathbf{r}_i))$, the only necessary value for density definition of i particle is h_i .

Thereby, because of the only requirement is particle density value, but total mass is a constant, an unambiguous equality between the volume integral in general form and total mass is unimportant. In other words, the «gathering» approach simplifies (7) a lot, so it becomes:

$$\rho_i = \sum_{j=1}^{N} \mathrm{m}_j W(\mathbf{r}_i - \mathbf{r}_j, h_i) . \tag{8}$$

The described above approach of the kernel interpolation allows to declare the rest of the hydrodynamic values. The first ones are internal forces: pressure and viscosity. To evaluate pressure the ideal gas law should be reviewed

$$pV = NRT .$$

In case of isothermal fluid with permanent mass the right side can be replaced with Boltzmann constant:

$$pV = \kappa, \quad p/\rho = \kappa, \quad p = \kappa\rho . \tag{9}$$

In terms of satisfying Newton's third law, pressure force must apply the symmetry property as follows:

$$\mathbf{f}_i^{\text{pressure}} = -\sum_{j \neq i} \frac{p_i + p_j}{2} \frac{\mathrm{m}_j}{\rho_j} \nabla W(\mathbf{r}_i - \mathbf{r}_j, h). \tag{10}$$

DOI 10.15588/1607-3274-2019-3-3

Combining the origin formula of interpolation kernel and pressure force equation, the corresponding pressure kernel and gradient take the following form:

$$W_{\text{pressure}}(\mathbf{r},h)=\frac{15}{\pi h^6}\begin{cases}\left(h-\|\mathbf{r}\|\right)^3, & 0\le\|\mathbf{r}\|\le h,\\ 0, & \|\mathbf{r}\|>h,\end{cases}$$

$$\nabla W_{\text{pressure}}(\mathbf{r},h)=-\frac{45}{\pi h^6}\frac{\mathbf{r}}{\|\mathbf{r}\|}\left(h-\|\mathbf{r}\|\right)^2.$$

The viscosity force has to apply the symmetry property either:

$$\mathbf{f}_i^{\text{viscosity}}=\nu\sum_{j\ne i}(\mathbf{u}_j-\mathbf{u}_i)\frac{m_j}{\rho_i}\nabla^2 W(\mathbf{r}_i-\mathbf{r}_j,h). \quad (11)$$

The viscosity kernel is received similar:

$$W_{\text{viscosity}}(\mathbf{r},h)=\frac{15}{2\pi h^3}\begin{cases}-\frac{\|\mathbf{r}\|^3}{2h^3}+\frac{\|\mathbf{r}\|^2}{h^2}+\frac{h}{2\|\mathbf{r}\|}-1, & 0\le\|\mathbf{r}\|\le h,\\ 0, & \|\mathbf{r}\|>h,\end{cases}$$

$$\nabla^2 W_{\text{viscosity}}(\mathbf{r},h)=\frac{45}{\pi h^6}(h-\|\mathbf{r}\|).$$

The only external force is gravity and it is trivially formulated using Newton's second law

$$\mathbf{f}_i^{\text{gravity}}=\rho_i g. \quad (12)$$

But there is an additional force that can be applied to the free surface of a liquid fluid – the surface tension force. It is normally not a part of the Navier-Stokes equations as it is considered as a boundary condition.

For a Lagrangian fluid the boundaries can be identified by the particles. Its behaviour can be described by

$$\mathbf{f}_i^{\text{surface}}=-\sigma\nabla^2 c_i\frac{\mathbf{n}_i}{\|\mathbf{n}_i\|}. \quad (13)$$

The main attribute of this value is identity of gradient and the inward surface normal of the fluid:

$$\mathbf{n}_i=\nabla c_i=\sum_j\frac{m_j}{\rho_i}\nabla W(\mathbf{r}_i-\mathbf{r}_j,h). \quad (14)$$

The overall algorithm has the following form [18]:

1. Initialization of hydrodynamic system:

– creation of a liquid substance;

– creation of N particles and the initial values of their characteristics;

– preparation of kernel function and calculation of smoothing radius;

– construction of equations of collision objects (obstacles) with the help of RFM and getting a polygonal mesh of an obstacle;

– initialization of the Verlet method for calculating particle acceleration.

2. Calculation of density and pressure:

– search for neighbors for each particle i;

– calculation of particle density ρ_i using (8);

– calculation of particle pressure p_i using (9).

3. Calculation of internal forces:

– calculation of pressure force $\mathbf{f}_i^{\text{pressure}}$ on the particle using of kernel interpolation (10);

– calculation of viscosity force $\mathbf{f}_i^{\text{viscosity}}$ on the particle using kernel interpolation (11);

– summing of pressure and viscosity forces to get internal forces $\mathbf{f}_i^{\text{internal}}=\mathbf{f}_i^{\text{pressure}}+\mathbf{f}_i^{\text{viscosity}}$.

4. Calculation of external forces:

– calculation of gravity force $\mathbf{f}_i^{\text{gravity}}$ (12);

– calculation of the surface normal $\mathbf{n}_i$ (14);

– calculation of the surface tension force $\mathbf{f}_i^{\text{surface}}$ for a plurality of particles located on or near the surface of the liquid (13);

– summing of forces $\mathbf{f}_i^{\text{external}}=\mathbf{f}_i^{\text{gravity}}+\mathbf{f}_i^{\text{surface}}$.

5. Time integration and collision management:

– summing of forces $\mathbf{F}_i=\mathbf{f}_i^{\text{external}}+\mathbf{f}_i^{\text{internal}}$;

– calculation of particle acceleration using (4);

– calculation of particle velocity and location using the Verlet method;

– performing a collision search;

– if a collision occurred, then correct the particle arrangement and update its velocity vector $\mathbf{v}_i=\mathbf{v}_i-(1-c_R)(\mathbf{v}_i\cdot\mathbf{n})\mathbf{n}$, where $0\le c_R\le 1$;

– approximation of a new velocity vector.

6. Plotting the obstacle and the particles located in the positions obtained at the previous step.

4 EXPERIMENTS

The algorithm was implemented in C++ object-oriented programming language. The source code could be easily compiled on Windows, Linux and Mac OS X with the help of CMake tool as project generator and an appropriate compiler. Overall, the project skeleton consists of four main parts:

– algorithms library, where Marching Cubes and Neighbor Search methods, R-operations and 3D Point are introduced;

– demo executable, which is responsible for graphical representation of the achieved results;

– sph library, where SPH is carried out;

– thirdparty, that contains Google Test and FreeGLUT source code.

The interactions between these main parts are performed as follows: sph uses algorithms library, demo binary links both of them and FreeGLUT library as well for the graphical representation of the achieved results.

Google Test framework takes care of unit testing. Code coverage of the project is about 95%. Project quality is measured by Codacy code analysis and equal to A. Every pull request is checked by Travis continuous inte-

DOI 10.15588/1607-3274-2019-3-3

gration tool, which performs Linux build on random Linux-build OS and runs all unit tests.

The source code is introduced as an open source project in GitHub https://github.com/aartiukh/sph-sdk. Any help or bug report are appreciated.

The simulation based on the proposed algorithm was carried out in cube, i.e. $\Omega = [0,3]\times[0,3]\times[0,3]$, for two obstacles: the pawn and bishop described by their equations (5) and (6) respectively. The number of particles N was equal to 10000.

The physical parameter values that were used for fluid simulation are presented in the table 1 [18].

A few features were implemented:

– it is possible to rotate the cube with particles and inserted object as well;

Table 1 – The physical parameter values of simulation

Name	Symbol	Value	Unit
Density	ρ	998.29	kg / m^3
Particle mass	m	0.02	kg
Viscosity	ν	3.5	$Pa \cdot s$
Surface tension	σ	0.0728	N / m
Threshold	l	7.065	n / a
Gas stiffness	k	3.0	J
Support radius	h	0.1	m

– particle color is changing depending on its speed: red corresponds to a very fast particle, yellow – medium speed and blue – slow motion;

– gravity vector gets updated during cube rotation;

– one can restore the original cube position.

In addition, the performance of the algorithm was measured depending on the different number of particles. See figure 4 for the details. The data was taken on Ubuntu 18.04 x64, which has Intel Core i5-5200U 2.2 GHz and 12Gb RAM. Only one processor was used.

5 RESULTS

Figures 2 and 3 depict the results of simulation for 2 cases described above in the different periods of time. The first experiment demonstrates the behavior of particles with pawn (5) as an obstacle. The second experiment reveals how flow behaves when bishop (6) is inserted into cube. Time is denoted in seconds.

Thus, both experiments show us how the real-time simulation is moving. In the beginning (Figures 2a, 3a) one can see a drop constructed from 10000 particles that is falling at the obstacle (pawn or bishop) under the power of gravity. Then, calculating the value of the function on the boundary, the particles are striking against an obstacle and flying apart by changing their direction (Figures 2b, 3b). On figures 2c and 3c the particles are pushing off the cube boundary by changing its speed to 0. Then, after some time, they are calming down (Figures 2d, 3d). Some of the particles are getting stuck on the obstacle surface that depicts the wetting phenomenon.

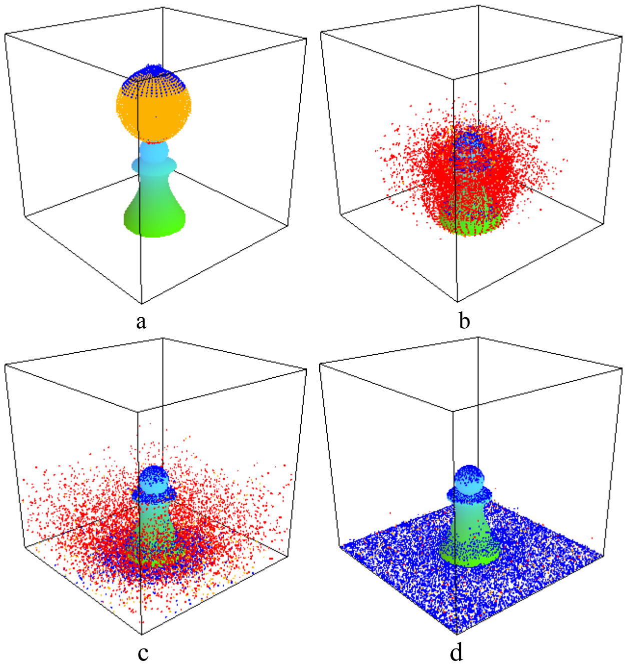


Figure 2 – Experiment 1,
$N = 10000$, obstacle is the pawn:
a – $t = 2$, b – $t = 15$, c – $t = 30$, d – $t = 45$

Figure 4 shows a dependency of the average running time of the implemented SPH algorithm from the number of particles involved into simulation.

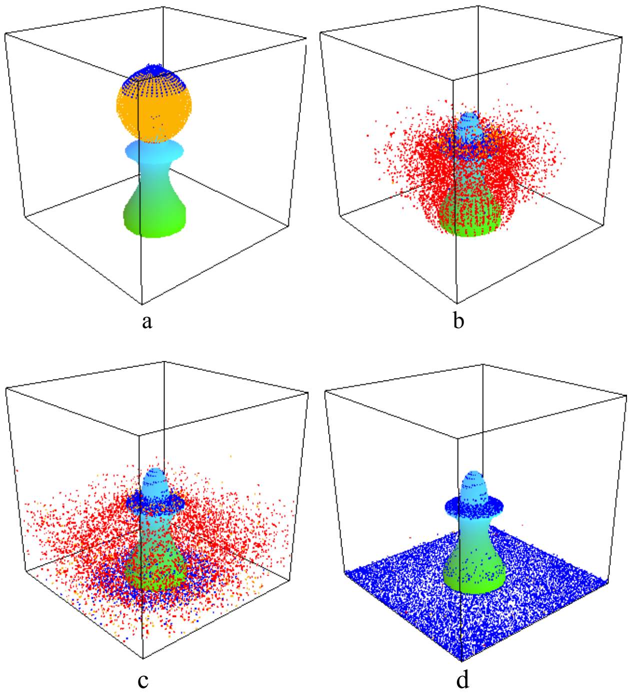


Figure 3 – Experiment 2,
$N = 10000$, obstacle is the bishop:
a – $t = 2$, b – $t = 15$, c – $t = 30$, d – $t = 45$

DOI 10.15588/1607-3274-2019-3-3

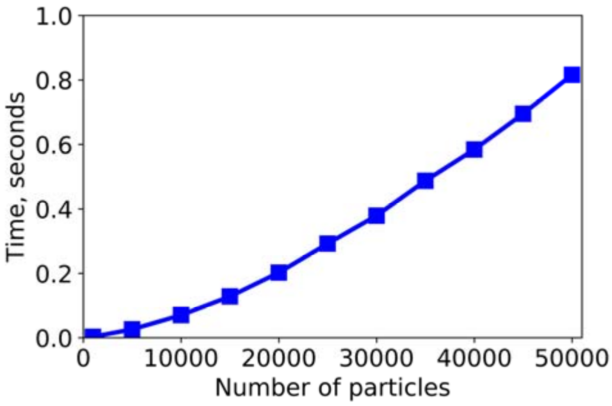


Figure 4 – The average running time in seconds of one SPH iteration depending on number of particles

6 DISCUSSION

Obviously, 10000 particles are not enough to simulate water-like behavior. The graphical representation shows that it is necessary to add more particles in order to achieve better simulation results. Other authors use one million particles to get a naturally looking simulation.

However, one cay say that Zlii Dai's work [5], where the terrain is represented as different number of particles, is improved. This improvement is achieved in our research by using RFM, which replaces the common approach of boundary representation as particles with an analytical boundary equation.

As one can see on figure 4, the average running time of one SPH iteration strongly depends on the number of particles. The largest part of computation time is consumed by the nearest neighbor search algorithm. Implemented neighbor search has $O(n\log(n))$ complexity. In further research it is reasonable to move all computations to GPU. GPU has more arithmetic logic units comparing with CPU that increases the ability to process simple operations in parallel. This will help to achieve the desired performance and allow to get better simulation results in real-time.

Thus, it is clear that new investigations are required. They will allow to develop new instruments and methods in order to help researchers in different areas of science.

CONCLUSIONS

The solution method for 3D simulation of unsteady flows is introduced. The solution algorithm is based on smooth particle hydrodynamics, marching cubes and the R-functions methods, combined usage of which allows to reduce particles number that are involved in a simulation. The various numerical experiments are carried out. Moreover, our C++ implementation is published in GitHub that grants access to it for anyone and allows to comment it or to report an issue.

The scientific novelty of the obtained algorithm is that it allows to change the considered obstacle easily and render its graphical representation. This advantage gives scientists an opportunity to perform a simulation faster and easier. The only difficulty here could be how to build a boundary equation of an obstacle.

The practical significance of the developed algorithm is that it can be easily implemented and has various practical applications. In addition, the implemented software follows SOLID principle that makes it more understandable, extendable and maintainable. This is crucial for further development and code reuse.

The prospects for further research can be dedicated to Neighbor Search algorithm speed up; performing all computations on GPU in order to reduce computation time and to avoid copying the data from CPU to GPU; implementation of Dual Contouring which is faster than Marching Cubes algorithm; performing more experiments with other shapes constructed by means of RFM.

ACKNOWLEDGEMENTS

The work was carried out at the department of Applied Mathematics at Kharkiv National University of Radio Electronics within the framework of collaborative scientific research conducted by the department. The authors express their gratitude to Inna Savchecko for her help.

REFERENCES

1. Landau L. D., Lifshits E. M. Teoreticheskaya fizika. V 10 t. T. VI. Gidrodinamika. Moscow, Fizmatlit, 2003, 736 p.
2. Ladyizhenskaya O. A. Matematicheskie voprosyi dinamiki vyazkoy neszhimaemoy zhidkosti. Moscow, Nauka, 1970, 288 p.
3. Rudman M., Cleary P. Using Smoothed particle hydrodynamics to study wave impact on floating offshore platforms, *Seventh International Conference on CFD in the Minerals and Process Industries,* 2009, pp. 1–7.
4. Takahiro H. Seiichi K., Yoichiro K. Smoothed particle hydrodynamics in complex shapes, *Proceedings of 23rd Spring Conference on Computer Graphics*, 2007, pp. 10–15.
5. Dai Z., Huang Y., Cheng H., Xu Q. 3D numerical modeling using smoothed particle hydrodynamics of flow-like landslide propagation, *Engineering Geology*, 2014, Vol. 180, pp. 21–33. DOI: 10.1016/ j.enggeo.2014.03.018.
6. Barreiro A., Crespo A., Domínguez J. M., Gómez-Gesteira M. Smoothed Particle Hydrodynamics for coastal engineering problems, *Computers & Structures,* 2013, V. 120, pp. 96–106. DOI: 10.1016/j.compstruc.2013.02.010.
7. Cleary P. W., Ha J. Three-dimensional smoothed particle hydrodynamics simulation of high pressure die casting of light metal components, *Journal of Light Metals*, 2002,V. 2, No. 3, pp. 169–183. DOI: 10.1016/S1471-5317(02)00043-3.
8. Randles P., Libersky L. Smoothed Particle Hydrodynamics: Some recent improvements and applications, *Computer Methods in Applied Mechanics and Engineering,* 1996, V. 139, No. 1–4, pp. 16–25. DOI: 10.1016/S0045-7825(96)01090-0.
9. Rvachev V. L. Ob analiticheskom opisanii nekotoryih geometricheskih ob'ektov, *Dokl. AN SSSR,* 1963, Vol. 153, № 4, pp. 765–768.
10. Rvachev V. L. Teoriya R-funktsiy i nekotoryie ee prilozheniya. Kyiv, Naukova dumka, 1982, 552 p.
11. Kravchenko V. F., Rvachev V. L. Algebra logiki, atomarnyie funktsii i veyvletyi v fizicheskih prilozheniyah. Moscow, Fizmatlit, 2006, 416 p.
12. Lorensen W., Cline H. Marching cubes: A high resolution 3D surface construction algorithm, *Proceedings of the 14th*

DOI 10.15588/1607-3274-2019-3-3

annual conference on Computer graphics and interactive techniques (SIGGRAPH ‘87), 1987, pp. 163–169. DOI: 10.1145/37401.37422.
13. Newman T. S. A survey of the marching cubes algorithm, *Computers & Graphics,* 2006, V. 30, No. 5, pp. 854–879. DOI: 10.1016/ j.cag.2006.07.021.
14. Liu M. B., Liu G. R. Smoothed Particle Hydrodynamics (SPH): an Overview and Recent Developments, *Archives of Computational Methods in Engineering*, 2010, Vol. 17, No. 1, pp. 25–76. DOI: 10.1007/s11831-010-9040-7.
15. Gomez-Gesteira M., Rogers B., Dalrymple R. A., Crespo A. State-of-the-art of classical SPH for free-surface flows, *Journal of Hydraulic Research,* 2010, Vol. 48, pp. 6–27. DOI: 10.1080/00221686.2010.9641242.
16. Price D., Wurster J., Tricco T., Nixon C., Toupin S., Pettitt A. Phantom: A Smoothed Particle Hydrodynamics and Magnetohydrodynamics Code for Astrophysics, *Publications of the Astronomical Society of Australia,* 2018, Vol. 35, pp. 10–35. DOI: 10.1017/pasa.2018.25.
17. Crespo A., Domínguez J. M. DualSPHysics: Open-source parallel CFD solver based on Smoothed Particle Hydrodynamics, *Computer Physics Communications,* 2015, V. 187, pp. 204–216. DOI: 10.1016/j.cpc.2014.10.004.
18. Kelager M. Lagrangian fluid dynamics using smoothed particle hydrodynamics, University of Copenhagen, 2006, 88 p.

Received 19.04.2019.
Accepted 08.07.2019.

УДК 004.93

ЗАСТОСУВАННЯ МЕТОДУ R-ФУНКЦІЙ ТА ГІДРОДИНАМІКИ ЗГЛАДЖЕНИХ ЧАСТИНОК ДЛЯ МОДЕЛЮВАННЯ РІДИНИ

Артюх А. В. – канд. фіз.-мат. наук, доцент кафедри прикладної математики Харківського національного університету радіоелектроніки, Харків, Україна.

Козиренко С. І. – канд. тех. наук, доцент, доцент кафедри прикладної математики Харківського національного університету радіоелектроніки, Харків, Україна.

Шабалін О. О. – здобувач кафедри прикладної математики Харківського національного університету радіоелектроніки, Харків, Україна.

Литвин І. Р. – студент кафедри прикладної математики Харківського національного університету радіоелектроніки, Харків, Україна

АНОТАЦІЯ

Актуальність. Існуючі методи моделювання рідини мають ряд недоліків та можуть бути вдосконалені за допомогою нових підходів до розв’язання задач обчислювальної гідродинаміки, що свідчить про актуальність роботи.

Мета роботи. Метою роботи є вдосконалення існуючих методів математичного моделювання рідини на основі гідродинаміки згладжених частинок та методу R-функцій.

Метод. Запропоновано новий підхід спільного використання гідродинаміки згладжених частинок та методу R-функцій. Гідродинаміка згладжених частинок дозволяє моделювати рух рідини в реальному часі. Метод розглядає рідину як дискретне число точок вибірки (частинок), які мають масу, швидкість, положення і величини фізичного поля (тиск, температура, щільність тощо). Метод R-функцій дозволяє розв’язати обернену задачу аналітичної геометрії: знайти аналітичне рівняння 2D (3D) об’єкту на основі його геометричного представлення. Використовуючи отримане рівняння об’єкту, можна просто виявити зіткнення частинок з межею цього об’єкту та побудувати поверхню об’єкту за допомогою алгоритму крокуючих кубиків. Запропонований спосіб дозволяє досягти гарної якості моделювання та виконати всі необхідні розрахунки та відображення в реальному часі.

Результати. Обчислювальні експерименти були проведені для задачі моделювання рідини. Для моделювання використовувалась різна кількість частинок. Для вивчення поведінки рідини в розглянуту область були додані різні види об’єктів.

Висновки. Результати візуального моделювання дозволяють стверджувати, що отриманий метод працює як і очікувалось. Цей метод може бути застосовано до різних задач моделювання рідини, де береться до уваги виявлення зіткнень з довільними об’єктами. Подальші дослідження можуть бути присвячені оптимізації алгоритму пошуку сусідів, виконанні всіх обчислень у графічному процесорі або врахуванню інших фізичних величин.

КЛЮЧОВІ СЛОВА: рівняння Нав’є-Стокса, моделювання рідини, метод R-функцій, гідродинаміка згладжених частинок, алгоритм крокуючих кубиків.

УДК 004.93

ПРИМЕНЕНИЕ МЕТОДА R-ФУНКЦИЙ И ГИДРОДИНАМИКИ СГЛАЖЕННЫХ ЧАСТИЦ ДЛЯ МОДЕЛИРОВАНИЯ ЖИДКОСТИ

Артюх А. В. – канд. физ.-мат. наук, доцент кафедры прикладной математики Харьковского национального университета радиоэлектроники, Харьков, Украина.

Козыренко С. И. – канд. тех. наук, доцент, доцент кафедры прикладной математики Харьковского национального университета радиоэлектроники, Харьков, Украина.

Шабалин А. О. – соискатель кафедры прикладной математики Харьковского национального университета радиоэлектроники, Харьков, Украина.

Литвин И. Р. – студент кафедры прикладной математики Харьковского национального университета радиоэлектроники, Харьков, Украина.

DOI 10.15588/1607-3274-2019-3-3

АННОТАЦИЯ

Актуальность. Существующие методы моделирования жидкости имеют ряд недостатков и могут быть усовершенствованы с помощью новых подходов к решению задач вычислительной гидродинамики, что подтверждает актуальность работы.

Цель работы. Целью работы является усовершенствование существующих методов математического моделирования жидкости на основе гидродинамики сглаженных частиц и метода R-функций.

Метод. Предложено новый подход совместного использования метода гидродинамики сглаженных частиц и R-функций. Гидродинамика сглаженных частиц позволяет моделировать движение жидкости в реальном времени. Метод рассматривает жидкость как дискретное число точек выборки (частиц), которые имеют массу, скорость, положение и величины физического поля (давление, температура, плотность и т.д.). Метод R-функций позволяет решить обратную задачу аналитической геометрии: найти аналитическое уравнение 2D (3D) объекта на основе его геометрического представления. Используя полученное уравнение объекта, можно просто определить столкновения частиц с границей объекта и построить поверхность объекта с помощью алгоритма шагающих кубиков. Предложенный способ позволяет достичь хорошего качества моделирования и выполнить все необходимые расчеты и прорисовку в реальном времени.

Результаты. Вычислительные эксперименты были проведены для задачи моделирования жидкости. Для моделирования использовалось разное количество частиц. Для изучения поведения жидкости в рассматриваемую область были добавлены различные виды объектов.

Выводы. Результаты визуального моделирования позволяют утверждать, что полученный метод работает как и ожидалось. Этот метод может быть применен к различным задачам моделирования жидкости, где берется во внимание обнаружение столкновений с произвольными объектами. Дальнейшие исследования могут быть посвящены оптимизации алгоритма поиска соседей, выполнении всех вычислений в графическом процессоре или учету других физических величин.

КЛЮЧЕВЫЕ СЛОВА: уравнения Навье-Стокса, моделирование жидкости, метод R-функций, гидродинамика сглаженных частиц, алгоритм шагающих кубиков.

ЛІТЕРАТУРА / ЛИТЕРАТУРА

1. Ландау Л. Д. Теоретическая физика. В 10 т. Т. VI. Гидродинамика / Л. Д. Ландау, Е. М. Лифшиц. – М. : Физматлит, 2003. – 736 с.
2. Ладыженская О. А. Математические вопросы динамики вязкой несжимаемой жидкости / О. А. Ладыженская. – М. : Наука. – 1970. – 288 с.
3. Rudman M. Using Smoothed particle hydrodynamics to study wave impact on floating offshore platforms / M. Rundman, P. Cleary // Seventh International Conference on CFD in the Minerals and Process Industries. – 2009. – P. 1–7.
4. Takahiro H. Smoothed particle hydrodynamics in complex shapes / H. Takahiro, K. Seiichi, K. Yoichiro. // Proceedings of 23rd Spring Conference on Computer Graphics. – 2007. – P. 10–15.
5. 3D numerical modeling using smoothed particle hydrodynamics of flow-like landslide propagation / [Z. Dai, Y. Huang, H. Cheng, Q. Xu] // Engineering Geology. – 2014. – V. 180. – P. 21–33. DOI: 10.1016/ j.enggeo.2014.03.018.
6. Smoothed Particle Hydrodynamics for coastal engineering problems / [A. Barreiro, A. Crespo, J. M. Domínguez, M. Gómez-Gesteira] // Computers & Structures. – 2013. – V. 120. – P. 96–106. DOI: 10.1016/j.compstruc.2013.02.010.
7. Cleary P. W. Three-dimensional smoothed particle hydrodynamics simulation of high pressure die casting of light metal components / P. W. Cleary, J. Ha // Journal of Light Metals. – 2002. – V. 2, № 3. – P. 169–183. DOI: 10.1016/S1471-5317(02)00043-3.
8. Randles P. Smoothed Particle Hydrodynamics: Some recent improvements and applications / P. Randles, L. Libersky // Computer Methods in Applied Mechanics and Engineering. – 1996. – V. 139, № 1–4. – P. 16–25. DOI: 10.1016/S0045-7825(96)01090-0.
9. Рвачев В. Л. Об аналитическом описании некоторых геометрических объектов / В. Л. Рвачев // Докл. АН СССР. – 1963. – Т. 153, №4. – С. 765–768.
10. Рвачев В. Л. Теория R-функций и некоторые ее приложения / В. Л. Рвачев. – К. : Наукова думка, 1982. – 552 с.
11. Кравченко В. Ф. Алгебра логики, атомарные функции и вейвлеты в физических приложениях / В. Ф. Кравченко, В. Л. Рвачев. – М. : Физматлит, 2006. – 416 с.
12. Lorensen W. Marching cubes: A high resolution 3D surface construction algorithm / W. Lorensen, H. Cline // Proceedings of the 14th annual conference on Computer graphics and interactive techniques (SIGGRAPH ’87). – 1987. – P. 163–169. DOI: 10.1145/37401.37422.
13. Newman T. S. A survey of the marching cubes algorithm / T. S. Newman // Computers & Graphics. – 2006. – V. 30, № 5. – P. 854–879. DOI: 10.1016/ j.cag.2006.07.021.
14. Liu M. B. Smoothed Particle Hydrodynamics (SPH): an Overview and Recent Developments / M. B. Liu, G. R. Liu // Archives of Computational Methods in Engineering. – 2010. – V. 17, № 1. – P. 25–76. DOI: 10.1007/s11831-010-9040-7.
15. State-of-the-art of classical SPH for free-surface flows / [M. Gomez-Gesteira, B. Rogers, R. A. Dalrymple, A. Crespo] // Journal of Hydraulic Research. – 2010. – V. 48. – P. 6–27. DOI: 10.1080/00221686.2010.9641242.
16. Phantom: A Smoothed Particle Hydrodynamics and Magnetohydrodynamics Code for Astrophysics / [D. Price, J. Wurster, T. Tricco et al] // Publications of the Astronomical Society of Australia. – 2018. – V. 35. – P. 10–35. DOI: 10.1017/pasa.2018.25.
17. Crespo A. DualSPHysics: Open-source parallel CFD solver based on Smoothed Particle Hydrodynamics / A. Crespo, J. M. Domínguez // Computer Physics Communications. – 2015. – V. 187. – P. 204–216. DOI: 10.1016/j.cpc.2014.10.004.
18. Kelager M. Lagrangian fluid dynamics using smoothed particle hydrodynamics / M. Kelager. – University of Copenhagen, 2006. – 88 p.

DOI 10.15588/1607-3274-2019-3-3