Hindawi
Advances in Mathematical Physics
Volume 2021, Article ID 5157197, 9 pages
https://doi.org/10.1155/2021/5157197

Research Article

Computer Simulation of Water Flow Animation Based on Two-Dimensional Navier-Stokes Equations

Rushi Cao [1] **and Ruyun Cao** [2]

[1] *Department of Art, Sichuan University, Chengdu 610000, China*
[2] *School of Design and Art, China Academy of Art, Hangzhou 310000, China*

Correspondence should be addressed to Rushi Cao; 2019141070044@stu.scu.edu.cn

Received 8 September 2021; Accepted 27 September 2021; Published 16 October 2021

Academic Editor: Miaochao Chen

Simulation of water flow animation is a significant and challenging subject in computer graphics. With the continuous development of computational fluid dynamics and computer graphics, many more effective simulation methods have been developed, and fluid animation simulation has developed rapidly. In order to obtain realistic flow animation, one of the key aspects is to simulate flow motion. Based on the two-dimensional Navier-Stokes equations, a mathematical model is established to solve the boundary conditions required by the physical flow field of water. The coordinate transformation formula is introduced to transform the irregular physical area into a regular square calculation area, and then, the specific expressions of the liberalized Navier-Stokes equation, continuity equation, pressure Poisson equation, and dimensionless boundary conditions are given. Using animation software to sequence graphics and images of all kinds of control and direct operation of the relevant instructions, through the computer technology to simulate the flow of animation, based on the stability of fluid simulation method and simulation efficiency, so as to make realistic flow animation. The results show that FluidsNet has considerable performance in accelerating large scene animation simulation on the basis of ensuring the rationality of prediction, and the motion of water wave is realistic. The application of computer successfully simulates water flow.

1. Introduction

With the computational fluid mechanics, computer graphics, the continuous development of disciplines, and crossing each other, resulting in a much more effective simulation method, among them, the model based on computer simulation of fluid animation developed rapidly and has not only become a research hotspot in the field but also become the current and future scientific solution and engineering which is an important direction of the development of the computer and visualization. At present, the research on the flow animation simulation has made a lot of achievements. There are two areas of representative research: The first is computational fluid mechanics, which mainly studies the scientific solution of fluid motion. Navier-Stokes equations or their simplified forms are used to accurately solve and describe fluid motion, so as to reveal the memory rules of fluid motion and solve engineering calculation problems in fluid-related fields. A large number of high-precision and efficient numerical methods have been developed and have made abundant achievements [1]. Second, fluid simulation in the field of computer graphics pays attention to the visual effect of fluid phenomena. By using the principles and methods of computer graphics or general graphics, the flow field data is converted into graphics and images and presented in an intuitive form, so as to obtain vivid and lifelike visual effects of fluid movement. It has also been applied to market demands such as movie special effects and video games, and many achievements have been made in this direction [2]. The research of fluid animation simulation based on physical model needs the cross fusion of computational fluid dynamics and computer graphics, which can be divided into two types according to the different research background and content. One is to make use of computational fluid dynamics results and adopt appropriate models and methods to develop flow field representation methods and drawing technologies required by corresponding fluid animation simulation from the perspective of computer graphics. The research work is mainly focused on computer graphics in fluid animation simulation [3]. The other is based on computational fluid dynamics, according to the development level of computer hardware and computer graphics, the development of CFD models, and methods suitable for fluid animation simulation, using the method of computer graphics to complete fluid animation simulation, not only to scientific solution but also to obtain vivid, realistic fluid animation.

The demand for dynamic simulation of fluid, liquid, or gas movement in virtual reality system is gradually increasing, and the movement effect is very useful in simulation and training system, computer art, advertising, and network virtual environment [4]. They can greatly improve the realism of virtual systems. The movement of water flow has the characteristics of unsmoothness, uncertainty, irregularity, and movement variation, and this variation is extremely complicated and chaotic and is affected by the surrounding environment, so it is necessary to simulate its movement correctly. The key is to build the right flow model and use the right rendering algorithm. From the perspective of computer graphics, the scene in the animation should look reasonable in order to make it appealing [5]. To be close to the real scene, this requires that the scene in the animation conforms to the physical laws of the real world. Therefore, it was a natural idea to have the elements of an animated scene move within the constraints of physical laws to make it seem forced.

Navier-Stokes equations, of great significance in fluid mechanics, are nonlinear partial differential equations for numerical solutions of Navier-Stokes equation problems. Hoult and Kovtun numerically simulated incompressible Navier-Stokes equations by the SUPG (streamline upwind Petrov-Galerkin) method. Based on the function space of coarse and fine grids, the steady-state and incompressible Navier-Stokes equations are transformed into a linear problem to be solved [6]. Jin et al. present a stable formula for solving compressible linear Navier-Stokes equations by combining pressure gradient projection with the discontinuous finite element method. The Taylor-Galerkin method is used to discreet avadavat equations of LS function. The P1P1 split finite element method was used to solve Navier-Stokes equations, and incompressible problems were simulated based on structural mesh [7]. Based on the hexahedron grid element, Gassner et al. adopt the interpolation method to construct the quadrilateral surface grid element of the object plane as curved surface grid element, which is used to solve three-dimensional compressible Euler equations and Navier-Stokes equations. Numerical solutions are also carried out by the flow around the three-dimensional sphere and the flow around the rotating streamline body. The high-precision characteristics of the DG method are verified [8]. Danchin and Mucha prove that Stokes operators of incompressible Navier-Stokes equations under Navier boundary conditions are related to slip conditions and converge to solutions of Navier-Stokes equations without slip boundary conditions in energy space. In both linear and nonlinear systems, weak and strong solutions are uniformly bounded with respect to slip coefficients [9]. Seregin proposed the pressure-robust mixed discontinuous Galerkin method (pressure-robust zed discontinuous Galerkin method) and proved that velocity error estimation of steady-state Navier-Stokes equations is independent of pressure. In addition, it also shows that the estimation of velocity and pressure is optimal, and numerical examples show that this method is globally optimal and convergent [10].

2. Computer Animation Production Method

2.1. The Production of Water Flow Animation. With programmable computers equipped with cathode-ray tube displays and pen or inkjet plotters, drawing and logic processing are combined, and images can be defined and manipulated on the computer. Computer has been widely used in all kinds of design image production, and a large number of animation creators have begun to turn to this new way of work, greatly promoted the development of computer animation this new industry. Computer animation is different from animation in the traditional sense. In terms of effect, the former can not only achieve all the effects of the latter but also produce visual effects that the latter cannot achieve, or even things that do not exist in virtual reality or cannot be seen by the human eyes [11, 12]. Build physical models to describe the movement of elements in a moving scene. Then, obtain the numerical solution of the corresponding physical equation, get the state of each element in the scene at each time step, and then draw the scene at each time step. This is the basic idea of animation based on physical model. With the physics-based approach to animation, the producer does not have to be like the traditional key framing method. The design specifies the state of each element in the animation scene at each time step directly, but only the initial state of each element. The states of the remaining frames are automatically generated by solving physics equations. A person who does not understand art creation can also use animation tools based on physical models to create an artistic appeal to animation. From the perspective of computational fluid dynamics, numerical solutions of fluid dynamics equations are obtained in the form of a large number of discrete data, such as height field, velocity field, and pressure field. It is difficult to see the distribution law from the data alone. These data fields need to be visualized. In a three-dimensional space, this data is graphed using the methods of computer graphics adding light and shadow to make the figure more pleasing to the eye or more distinctive. It is also a natural need. If you want to consider the changes of the data field in each time step, it is a natural idea to animate the corresponding images. We want to integrate computer graphics and numerical analysis to create computational animation as a tool for both engineering and art. On the one hand, improving the accuracy of solution can make animation used in engineering, and on the other hand, increasing the sense of reality of computer graphics can make animation used in artistic creation.

2.2. Computer Animation Production Common Techniques. From the production point of view, different from the traditional animation produced by using film or video tape to record by frame, computer animation is the animation produced by using animation production software to control the sequence of graphics and images and directly run the relevant instructions. It can be divided into man-machine animation, mechanical animation, and programming animation; different types correspond to different production methods [13]. Man-machine animation technology is to use artificial drawing animation principle or artificial animation model, scanning or shooting into the computer after further processing through the computer or action control, and then by the computer synthesis and broadcast. Mechanical animation technology refers to the creation of original animation and model construction to the final action design, sound, synthesis, playback, and output that are all completed in the computer. Practice has proved that all types of animation can be completed by using computers.

Programming animation production refers to the animation is completely realized by programming, rather than through painting and then computer synthesis playback. A typical example is computer fractal animation. The fractal picture is a combination of applied mathematical knowledge and programming. The shape of the fractal picture generated is realized by automatically changing the variable values in the program. Some programming tools rely on an animation software, that is, animation software with some programming modules to write procedures and directly put in the corresponding animation software to run. Some use specialized graphics development tools such as the willows platform under the two commonly used tools to develop 3D graphics applications: OpenGL and DirectX; the animations generated by these two tools often have stunning, amazing effects. This article makes water flow using one of these two tools, OpenGL.

2.3. OpenGL Working Mechanism and Basic Workflow. OpenGL is an API (application program interface), that is, 3D graphics and model library. It has incomparable advantages in interactive 3D graphics modelling ability and programming side. OpenGL provides object description, rotation, translation, scaling, material, lighting, texture, interaction, and graphics performance commands, which basically cover the development of two-dimensional and three-dimensional graphics procedures that need all aspects. OpenGL is hardware independent and can be implemented on different hardware platforms. For example, using AutoCAD or 3D Studio Max software package can also be convenient to establish three-dimensional geometric model, but it is difficult to transplant and control. If these models are transformed into OpenGL program, you can control these models as you like: make CAD, 3D animation, virtual simulation, and so on. Because it has nothing to do with the hardware system, we can easily transplant the application to another operating system, and it can call the 3D graphics processing function directly to the hardware, so the processing speed is very fast.

OpenGL works as a client server. The client sends OpenGL commands to the server, and the server is responsible for interpreting these commands. In most cases, the client and server run on the same machine [14]. Of course, it can also be used in the network environment, so OpenGL has network transparency, and in Win32 graphics device interface to encapsulate the graphics function library in the dynamic connection library GD132.DLL, OpenGL's graphics library is also encapsulated in a dynamic connection library called openg132.dll. Calls to OpenGL functions published from the client application are first called, after processing by Openg132. DLL sent to the server, and then further processing by WinSRV. DLL sent to the Win32 device driver interface, and finally to the video display driver processing graphics command. The entire process is shown in Figure 1.

In the basic process of OpenGL work, OpenGL generates frames to be displayed after processing geometric vertex data and image pixel data through a series of operations [15]. The basic workflow is shown in Figure 2.

All data, including geometric vertex data and image pixel data, are first stored in a display list and then processed at once. Then, it deals with the evaluation stage of the pipeline and provides an efficient method to approximate the drawing of geometric curves and surfaces through the evaluation polynomial of the input data. In the next stage, the operation of each vertex is combined with primitives, the vertex is changed and illuminated, and the primitives are trimmed to fit the viewport size. The next step is pasteurization, which includes both geometric and physical mapping. Geometric operation is to transform the pixel into a two-dimensional image. The physical operation is to calculate information such as the colour and depth of each point in the image. Thus, pasteurization of a pixel consists of a two-part operation one that determines which squares of an integer grid in window coordinates are occupied by the image. The second part calculates the colour and depth values for each of these squares. In this process, a series of frame cache addresses and associated values are generated through two-dimensional descriptions of geometric and physical operations. All fragments are processed slice by slice, and the processing results are stored as pixels in the frame cache. The operations include conditionally updating the frame cache based on input and previously stored Z -values, mixing input pixel colours with stored colours, masking, and other logical operations on pixel values. When the input data is pixel data, such data including pixels, drawings, and images will skip the above processing "solver" and be processed as pixel values during the pixel manipulation phase. The results of this phase of processing are either stored in texture memory for raster manipulation or pasteurized in the same form as geometric data, and the results are integrated into the frame cache.

2.4. Data-Driven Fluid Animation. The amount of computation for fluid animation depends on the number of meshes or particles. With the increase of this number, the amount of computation will gradually increase, and the demand for computing resources will also increase accordingly. When the simulation object is expanded to large-scale water bodies such as lakes and oceans, the calculation time of each frame update will increase rapidly to be measured in minutes, which is far beyond the requirement of real-time simulation for single-frame calculation time. Therefore, the traditional physics-based method is not suitable for real-time

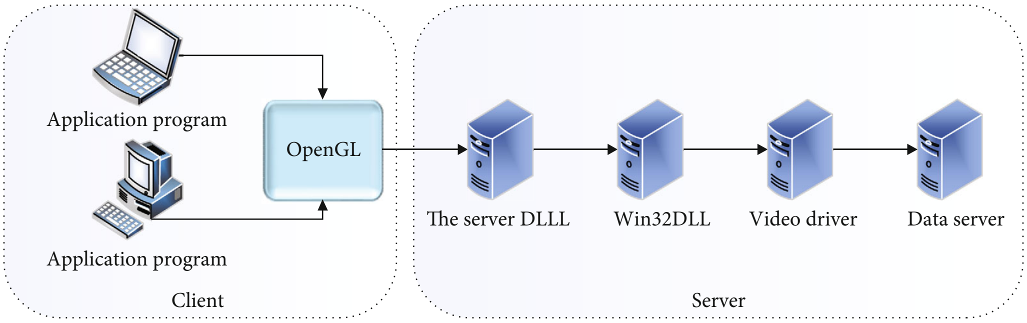

Figure 1: System process mechanism process diagram.

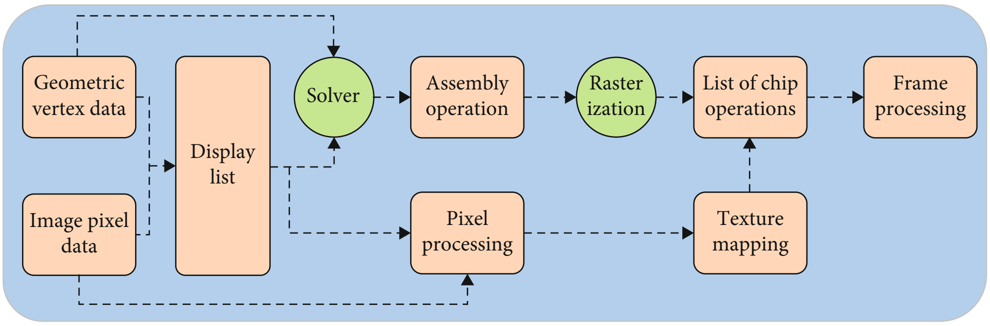

Figure 2: OpenGL basic workflow flow chart.

simulation of large-scale water bodies. Theoretically, the physical system can be reconstructed and estimated by collecting physical information data, but considering that the reconstruction and analysis of video in the field of computer vision is still very difficult, we choose to obtain cheap, flexible, and reliable simulation data to drive. Navier-Stokes equations enable learning and prediction of physical models directly from complex or unknown experiments with underlying causal models. In fluid simulation, there are usually more than 5,000 fluid particles in the scene, and each particle is calculated independently and related to each other. Therefore, although data-driven fluid simulation also requires the learning of fluid dynamics through physical learning, with the development of artificial intelligence technology, machine learning algorithms shine in the field of computer vision, and many data-driven methods based on deep learning also appear in the field of computer animation simulation.

3. Mathematical Models of Two-Dimensional Navier-Stokes Equations

3.1. Mathematical Model. According to Newton's second law and assumptions, the following mathematical models of linearized incompressible viscosity Navier-Stokes equations were established [16]:

$$\frac{\partial u}{\partial t} + \mu' u - 2\theta_1 v = -\frac{1}{\rho}\frac{\partial p}{\partial x} + y\theta_2 - x{\theta_1}^2 + \left(\frac{\partial^2 u}{\partial^2 x} + \frac{\partial^2 v}{\partial^2 y}\right), \tag{1}$$

$$\frac{\partial u}{\partial t} + \mu' u + 2\theta_1 v = -\frac{1}{\rho}\frac{\partial p}{\partial y} + x\theta_2 - y{\theta_1}^2 + \left(\frac{\partial^2 u}{\partial^2 x} + \frac{\partial^2 v}{\partial^2 y}\right), \tag{2}$$

where u and v are the velocity components of the fluid in the x and y directions, respectively, t represents the time, p and ρ represent the pressure and density of the fluid, respectively, and θ, θ_1, and θ_2 represent the rotation angle, angular velocity, and angular acceleration in tilting excitation motion, respectively. The continuity equation for incompressible fluid is as follows:

$$\frac{\partial u}{\partial x} + \frac{\partial u}{\partial y} = 0. \tag{3}$$

The velocity field solved by using the liberalized Navier-Stokes equations must satisfy the continuity equation, but if the velocity field solved does not satisfy the continuity equation, the original pressure field needs to be modified. Therefore, Poisson equation, which can correct the pressure field, should be introduced here.

$$\nabla^2 p = 2\rho\theta_1\left(\frac{\partial u}{\partial x} - \frac{\partial u}{\partial y}\right), \tag{4}$$

where ∇^2 is the Laplace operator. The equation is obtained by taking the partial derivatives of Equations (1) and (2) with respect to x and y, respectively, then adding and substituting into the continuity equation as constraints.

3.2. Boundary Conditions. When solving the physical flow field of water, the boundary conditions are composed of free liquid surface boundary conditions and solid-liquid surface boundary conditions. The free surface boundary is a moving boundary that changes with time. On the free liquid surface, the fluid must satisfy the kinematic and dynamic boundary conditions. Since the free surface always meets the condition of continuous surface; that is, fluid particles on the liquid surface always exist on the liquid surface without breaking waves, the kinematic boundary conditions of the free surface can be expressed as follows [17]:

$$h_t = v - uh_x. \tag{5}$$

If surface tension is not taken into account and pressure changes on the air side of the free surface caused by changes in the liquid level are also ignored, then dynamic boundary conditions on the free surface can be expressed as follows [18]:

$$\begin{aligned} u_y &= v_x + \frac{4v_y h_x}{(h_x)^2 - 1}, \\ \frac{p_0}{\rho} &= g(h - h_0) + 2v\frac{1 - (h_x)^2}{1 + (h_x)^2}v_y, \end{aligned} \tag{6}$$

where the column represents the pressure on the free liquid surface.

Since the fluid is impenetrable to the wall, there is no slip boundary condition on the two sides and bottom of the tank, and the pressure gradient boundary condition on the solid-liquid surface can be expressed as follows:

$$\frac{\partial p}{\partial x} = \rho\left(-g\cos\theta + y\sin\theta_2 + x{\theta_1}^2\right). \tag{7}$$

3.3. Coordinate Transformation and Equation Dimensionless. The coordinate transformation formula is introduced to transform the irregular physical area into a regular square calculation area, and then, the specific expressions of the linearized Navier-Stokes equation, continuity equation, pressure Poisson equation, and dimensionless boundary conditions are given. During the fluid sloshing, the shape of the whole region is constantly changing, and the function of the free surface wave changes instantaneously. Coordinate transformation is used to map irregular fluid region to square calculation region, and the coordinate transformation function is as follows [19]:

$$\begin{aligned} X &= \frac{x - b_1}{b_2 - b_1}, \\ Y &= \frac{y - d_0}{h}. \end{aligned} \tag{8}$$

The left boundary can be mapped to $x = 0$ and the right boundary to $y = 1$ by the coordinate transformation formula given above. We linearize the Navier-Stokes equation into a dimensionless format.

$$\begin{aligned} U &= \frac{u}{u_0}, \\ V &= \frac{v}{u_0}, \\ P &= \frac{p}{\rho g d_0}, \\ H &= \frac{h}{d_0}, \\ \mu &= \frac{\mu' d_0}{u_0}, \end{aligned} \tag{9}$$

where inhibition is the characteristic velocity of fluid. Using the above coordinate transformation and dimensionless format, the dimensionless form of the linearized Navier-Stokes equations can be written as follows:

$$\begin{aligned} U_T + C_5U_X + C_6U_Y + \mu U - 4\pi V\theta_T &= -\frac{1}{F^2}\sin(2\pi\theta) \\ &\quad - \frac{1}{F^2}(C_1P_X + C_2P_Y) + C_8\theta_T + C_9{\theta_T}^2, \\ V_T + C_5U_X + C_6U_Y + \mu U - 4\pi V\theta_T &= -\frac{1}{F^2}\cos(2\pi\theta) \\ &\quad - \frac{1}{F^2}(C_3P_X + C_4P_Y) - C_7\theta_T + C_{10}{\theta_T}^2. \end{aligned} \tag{10}$$

The dimensionless form of the continuity equation can be written as follows:

$$(C_1U_X + C_2U_Y) + (C_3V_X + C_4V_Y) = 0. \tag{11}$$

The dimensionless form of the pressure equation can be written as follows:

$$\begin{aligned} S &= \left({C_1}^2 + {C_3}^2\right)P_X + 2(C_1C_2 + C_3C_4)P_{XY} + \left({C_2}^2 + {C_4}^2\right)P_Y, \\ \bar{S} &= F^2[\mu(C_1U_X + C_2U_Y + C_3V_X + C_4V_Y) - 4\pi\theta(C_1V_X \\ &\quad - C_2V_Y - C_3U_X - C_4U_Y) - 8\pi^2{\theta_T}^2]. \end{aligned} \tag{12}$$

Coefficients $C_1 - C_{10}$ are the coefficients generated by regional coordinate transformation and are expressed as follows:

$$
\begin{aligned}
C_1 &= d_0 \frac{\partial X}{\partial x}, \\
C_2 &= d_0 \frac{\partial Y}{\partial x}, \\
C_3 &= d_0 \frac{\partial X}{\partial y}, \\
C_4 &= d_0 \frac{\partial Y}{\partial y}, \\
C_5 &= \frac{d_0}{u_0} \frac{\partial X}{\partial t}, \\
C_6 &= \frac{d_0}{u_0} \frac{\partial Y}{\partial t}, \\
C_7 &= \frac{2\pi x}{d_0}, \\
C_8 &= \frac{2\pi y}{d_0}, \\
C_9 &= \frac{4\pi^2 x}{d_0}, \\
C_{10} &= \frac{4\pi^2 y}{d_0}.
\end{aligned} \tag{13}
$$

4. Data-Driven Large Scene Simulation Acceleration

Fluid simulation methods based on data drive mainly use fluid simulation data as dataset. The simulation method is to divide the simulation domain into uniform two-dimensional subspace and then perform physical calculation on each node [20]. High quality fluid simulations with challenging details are achieved by learning from different levels of features using different network structures. FluidsNet, moreover, can use small scene training dataset which is a good model to predict fluid animation of the same kind of big scene; this can not only solve the traditional simulation method for fluid animation of big scene simulation speed of the problem of large memory consumption but also can solve the data-driven algorithm that often requires a lot of big scene samples to model learning problems.

4.1. FluidsNet Network Structure. The overall structure of FluidsNet is shown in Figure 3. FluidsNet designs networks with different structures for different levels of features. The network takes SPH fluid animation simulation data as input and calculates local features for fluid states [21]. Then, the simulation scene is divided into uniform 3D vowels through the background grid. The vowel learning layer is used to generate 4-dimensional vowel feature tensor, which is used for learning global features. Finally, the velocity field is predicted and output for all fluid states after the basic, local, and global characteristics are combined.

4.2. Training Process. All datasets were trained in the same way. The tensor flow framework and ADAM optimizer were used to implement FluidsNet. The initial learning rate was 0.003 and decreased to 1 in the middle of the training iteration. The complete training details are shown in Table 1.

Dataset 1 contains 25 animation simulations, with each non-Newtonian fluid animation simulation containing approximately 2000 frames of data. The training results are shown in Figure 4. The general viscosity of non-Newtonian fluid is greater than that of Newtonian fluid, and there are few splash and other phenomena in animation simulation. FluidsNet mainly learns the viscosity change in the flow process and meets surface wrinkles and other phenomena generated by solids, so all simulation animation frames are directly used as the original data, without intercept.

Datasets 2 and 3 contain 38 animation simulations, each of which contains approximately 1500 frames of data. Since the animation simulation of Newtonian fluid usually moves smoothly and has weak flow characteristics before 300 frames and after 1500 frames, and the training of FluidsNet for Newtonian fluid aims at scenes with high complexity and rich fluid details, 300 to 1500 frames are taken for each animation for training. Because the value variation between successive frames is small, the original dataset is sampled on average, with approximately 1 frame in every 10 frames taken for training and testing. 80% of the dataset is used for training and the remaining 20% for testing. The number of training iterations is about 2.4 K, and 8 samples can be batch processed simultaneously. When the dataset was fully utilized, it was randomly shuffled and continued for the next round of training.

Figure 5 illustrates the training iteration process for FluidsNet. It can be observed that the error of FluidsNet decreases during the training process and finally converges to a stable solution for each dataset training. In dataset 2, the final error is slightly higher than that in other datasets, which is due to the larger area of fluid-structure coupling collision and stronger water waves and splints in this scene, which leads to higher complexity of animation simulation. At the same time, dataset 2 also needs more training iterations.

4.3. Velocity Field Prediction. The necessity of velocity field prediction based on global feature is evaluated. Figure 6 shows the iterative changes of the network model after removing global features. The numerical value of the output velocity field predicted by the network model is reflected by colour. When the velocity is 0, the fluid particles are shown as blue. The greater the value of the velocity field, the whiter the fluid particles appear. The initial prediction results of random frames in the dataset by the untrained network model show that the output values have strong randomness. As the training iteration goes on, the network model without global features tends to predict the velocity fields of all fluid particles to be the same value approximately 0, while the model can correctly predict the corresponding velocity fields of different fluid particles.

Although the velocity field predicted by FluidsNet is slightly lower than the simulation reference value as the

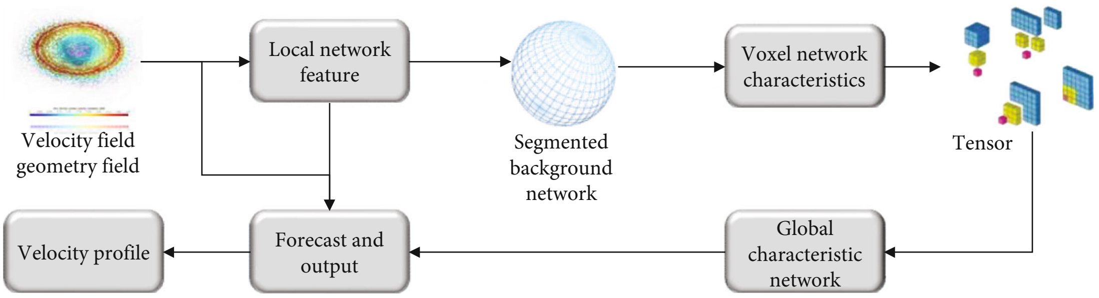

FIGURE 3: FluidsNet overall structure diagram.

TABLE 1: The complete training details.

Scene	Non-Newtonian tumours	Newton tumours	Water movements
Original data	25 animation, 5000 frames	38 animation, 45600 frames	38 animation, 45600 frames
Training dataset	4000	3680	3680
Testing dataset	1000	965	965
Model parameter	$K = 10$; $n = 0.5$	$K = 0$; $n = 0$	$K = 0$; $n = 0$
Number of batches	8	8	8
Training iteration	2600	2380	2380
Training time	268	221	215

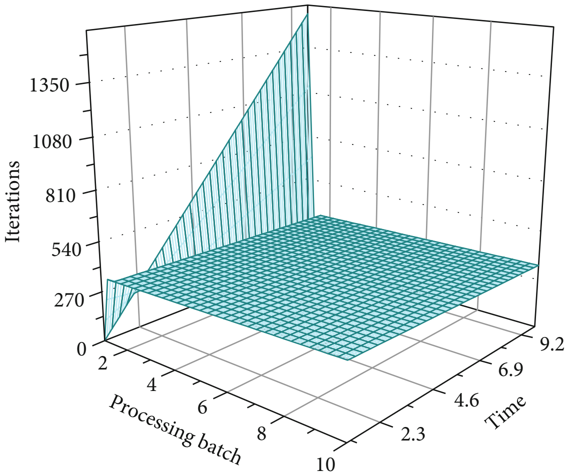

FIGURE 4: Number set 1 training result diagram.

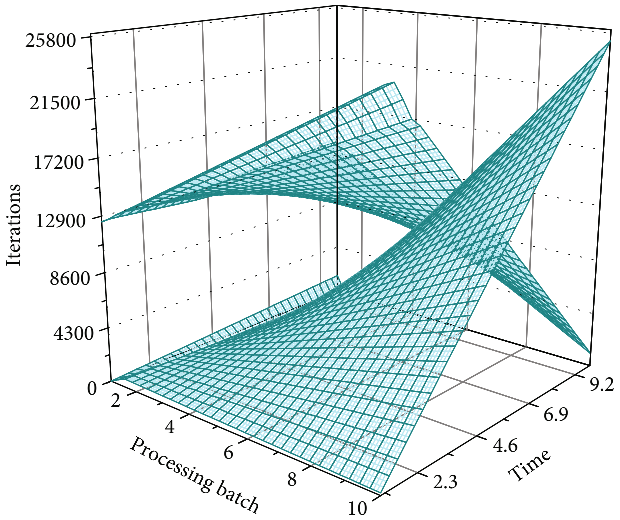

FIGURE 5: FluidsNet training iteration process.

error accumulates, the error is within the acceptable and reasonable range, and FluidsNet can reasonably predict the non-Newtonian fluid. Although FluidsNet has good accuracy and stability with error accumulation, it can reasonably predict the velocity field of various fluid simulation scenarios with random initial state. In addition, the dataset of FluidsNet can use small scene training good model prediction velocity field of the same kind of big scene. The model solves the problem of slow speed and large memory consumption in large scene simulation by traditional fluid animation simulation method. The model can also solve the problem that other data-driven algorithms need a large number of large scene datasets for model learning. FluidsNet has considerable performance in accelerating large scene animation simulation on the basis of ensuring the rationality of prediction.

4.4. Time Consumption. One of the main advantages of the model is that its prediction speed is faster than the solution speed of fluid animation simulation method. Figure 7 tests the computational efficiency of fluid animation simulation under the solver. The specific numerical value corresponding to the experimental results shown is the total value of fluid particles and solid sample particles, and the time consumption is the average data of all frames of the experimental animation. Each set of experiments ran 50 two-way iterations, including all communications, and then calculated the average time for each iteration. As can be seen, the model shows better performance with an extremely high acceleration ratio. The model can process multiple sets of data separately at the same time. Therefore, the model can

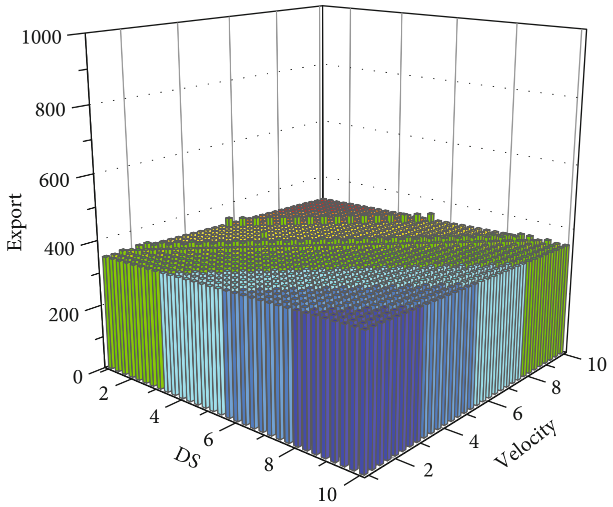

Figure 6: Iterative variation diagram of the network model after removing global features.

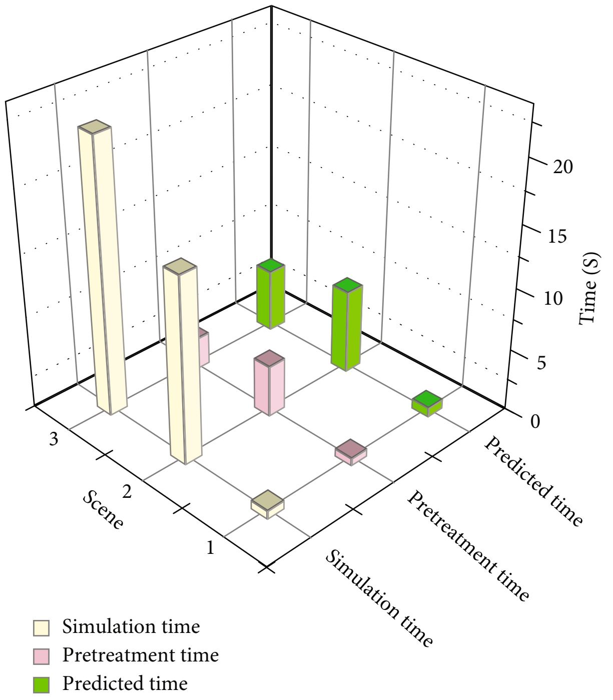

Figure 7: Computational efficiency test diagram of fluid animation simulation under solver.

be effectively applied to output multiple velocity fields at the same time.

FluidsNet greatly reduces the computing time required for output speed field. When the number of fluid particles in the simulation scene is large and the amount of water is large, the acceleration performance of FluidsNet will be better reflected. In addition, when the mechanical model of non-Newtonian fluid is more complex, the prediction time of FluidsNet is only related to the number of fluid particles in the scene and the scene size and has nothing to do with the complexity of the model, so it can achieve a very high speed ratio. Although the predictive part of the network takes some time, FluidsNet still has sufficient performance advantages considering the ability to run multiple samples in batches.

5. Conclusion

Fluid animation in graphics has a wide range of applications in 3D animation, games, computer-aided design, and other fields. Due to the complexity of the real world and the computational complexity of fluid simulation itself, how to simulate complex fluid phenomena more accurately while maintaining the stability of simulation and improving simulation efficiency has always been the most important problem in the field of fluid animation. In this paper, a mathematical model based on two-dimensional Navier-Stokes equations is established to solve the boundary conditions required by the physical flow field of water. Based on the stability and simulation efficiency of data-driven fluid simulation method under large time step, the data-driven method is designed according to the simulation data to achieve large scene simulation acceleration. This method has good accuracy and stability and can reasonably predict the velocity field of various fluid simulation scenarios with random initial states. On the basis of ensuring the rationality of prediction, it has a considerable performance in accelerating large scene animation simulation. In the future, we will explore whether the application of fluid data collected in the real world can be extended, such as debris flow, video, and image of industrial paste, where material parameters cannot be accurately evaluated.

Data Availability

The data used to support the findings of this study are available from the corresponding author upon request.

Conflicts of Interest

The authors declare that they have no known competing financial interests or personal relationships that could have appeared to influence the work reported in this paper.

References

[1] X. Qin, M. Motley, R. LeVeque, F. Gonzalez, and K. Mueller, "A comparison of a two-dimensional depth-averaged flow model and a three-dimensional RANS model for predicting tsunami inundation and fluid forces," *Natural Hazards and Earth System Sciences*, vol. 18, no. 9, pp. 2489–2506, 2018.

[2] G. Sadaka, A. Rakotondrandisa, P.-H. Tournier, F. Luddens, C. Lothodé, and I. Danaila, "Parallel finite-element codes for the simulation of two-dimensional and three-dimensional solid-liquid phase-change systems with natural convection," *Computer Physics Communications*, vol. 257, article 107492, 2020.

[3] M.-S. Mohamed, A.-N. Hirani, and R. Samtaney, "Discrete exterior calculus discretization of incompressible Navier-Stokes equations over surface simplicial meshes," *Journal of Computational Physics*, vol. 312, pp. 175–191, 2016.

[4] Y. Zhou and L. Peng, "On the time-fractional Navier-Stokes equations," *Computers & Mathematics with Applications*, vol. 73, no. 6, pp. 874–891, 2017.

[5] C. Fang, Z. Zhao, P. Zhou, and Z. Lin, "Feature learning via partial differential equation with applications to face recognition," *Pattern Recognition*, vol. 69, pp. 14–25, 2017.

[6] R.-E. Hoult and P. Kovtun, "Stable and causal relativistic Navier-Stokes equations," *Journal of High Energy Physics*, vol. 2020, no. 6, pp. 1–15, 2020.

[7] X. Jin, S. Cai, H. Li, and G. E. Karniadakis, "NSFnets (Navier-Stokes flow nets): physics-informed neural networks for the incompressible Navier-Stokes equations," *Journal of Computational Physics*, vol. 426, article 109951, 2021.

[8] G.-J. Gassner, A.-R. Winters, F.-J. Hindenlang, and D. A. Kopriva, "The BR1 scheme is stable for the compressible Navier–Stokes equations," *Journal of Scientific Computing*, vol. 77, no. 1, pp. 154–200, 2018.

[9] R. Danchin and P.-B. Mucha, "The incompressible Navier-Stokes equations in vacuum," *Communications on Pure and Applied Mathematics*, vol. 72, no. 7, pp. 1351–1385, 2019.

[10] G. Seregin, "Liouville type theorem for stationary Navier–Stokes equations," *Nonlinearity*, vol. 29, no. 8, pp. 2191–2195, 2016.

[11] C. Wang, Y. Wang, and Z. Zhang, "Zero-viscosity limit of the Navier–Stokes equations in the analytic setting," *Archive for Rational Mechanics and Analysis*, vol. 224, no. 2, pp. 555–595, 2017.

[12] C. Foias, C.-F. Mondaini, and E.-S. Titi, "A discrete data assimilation scheme for the solutions of the two-dimensional Navier-Stokes equations and their statistics," *SIAM Journal on Applied Dynamical Systems*, vol. 15, no. 4, pp. 2109–2142, 2016.

[13] M. Fujisawa, T. Nakada, and M. Mikawa, "Particle-based shallow water simulation with splashes and breaking waves," *Journal of Information Processing*, vol. 25, pp. 486–493, 2017.

[14] T.-C. Lavoratti, S. Heitkam, U. Hampel, and G. Lecrivain, "A computational method to simulate mono- and poly-disperse two-dimensional foams flowing in obstructed channel," *Rheologica Acta*, vol. 60, no. 10, pp. 587–601, 2021.

[15] J. Kim, J. Jung, and Y. Park, "An energy-stable and second-order accurate method for solving the incompressible Navier-Stokes equations," *Journal of the Korean Society for Industrial and Applied Mathematics*, vol. 23, no. 2, pp. 93–114, 2019.

[16] F.-S. Pereira, F.-F. Grinstein, and D.-M. Israel, "Modeling and simulation of transitional Rayleigh-Taylor flow with partially-averaged Navier-Stokes equations," 2021, https://arxiv.org/abs/2108.06423.

[17] B. Crespin, C.-T. Tran, M. Cerbelaud, A. Videcoq, and E. Darles, "Two-dimensional stochastic rotation dynamics for fluid simulation," *Journal of Computer and Communications*, vol. 8, no. 2, pp. 27–38, 2020.

[18] J.-H. Kim and J. Lee, "Synthesizing large-scale fluid simulations with surface and wave foams via sharp wave pattern and cloudy foam," *Computer Animation and Virtual Worlds*, vol. 32, no. 2, article e1984, 2021.

[19] S.-J. Terrington, K. Hourigan, and M.-C. Thompson, "The generation and conservation of vorticity: deforming interfaces and boundaries in two-dimensional flows," *Journal of Fluid Mechanics*, vol. 890, p. 5, 2020.

[20] S. Bhaumik and T. K. Sengupta, "Impulse response and spatio-temporal wave-packets: the common feature of rogue waves, tsunami, and transition to turbulence," *Physics of Fluids*, vol. 29, no. 12, article 124103, 2017.

[21] K. Qu, H.-S. Tang, and A. Agrawal, "Integration of fully 3D fluid dynamics and geophysical fluid dynamics models for multiphysics coastal ocean flows: simulation of local complex free-surface phenomena," *Ocean Modelling*, vol. 135, pp. 14–30, 2019.