Flowmap Baking with LBM-SWE

Wei Li, Haozhe Su, Zherong Pan,
Xifeng Gao, Zhenyu Mao, and Kui Wu

20.1 Introduction

The real-time simulation of water dynamics holds paramount importance across various industries, particularly in gaming and virtual reality applications. The integration of realistic water rendering significantly enhances user experiences, fostering immersive and visually captivating virtual landscapes. The ability to achieve dynamic and responsive water surfaces encompassing oceans, rivers, and lakes contributes to the creation of a more authentic and engaging virtual environment. While numerous research directions, including position-based fluids (PBF) [Macklin and Müller 13], smoothed particle hydrodynamics (SPH) [Koschier et al. 22], and shallow water equations (SWE) [Su et al. 23, Chentanez and Müller 10], have advanced the field of real-time water simulation, these methods encounter challenges when addressing large-scale simulation domains with the demand for high-fidelity outcomes.

An alternative approach employed in the gaming industry involves precomputing a 2D velocity field texture, known as a *flowmap*. This flowmap is utilized to animate water movement by displacing the water surface texture in UV space. Further details on flowmaps can be found in [Vlachos 10]. Leveraging a flowmap enables game developers to achieve realistic and dynamic water movement while significantly reducing computational costs during gameplay. Unfortunately, the current flowmap creation pipeline heavily relies on manual craftsmanship by artists, leading to time-consuming processes. Although conventional 2D Navier—Stokes simulation methods can generate flowmaps, these approaches typically neglect underwater terrain considerations, thereby impacting the fidelity of simulation results. The lattice Boltzmann model for shallow water equations (LBM-SWE) [Zhou 02] presents an approach for simulating large bodies of water, including oceans, seas, and large lakes, under the shallow water assumption. This model has been employed [Grenier 18] to compute flowmaps, benefiting from its high computational efficiency and effective handling of underwater terrain. However, a significant drawback emerges in its susceptibility to instability issues when simulating turbulent cases. These challenges frequently lead to failures in generating accurate flowmaps, particularly in complex scenarios.

In the following sections, we present an enhanced LBM-SWE formulation incorporating a new force model to bolster stability and robustness in turbulent flow, particularly at high Reynolds numbers. Our approach yields an accurate and stable LBM solver capable of addressing both laminar and turbulent flow problems. Experimental results demonstrate the efficacy of our model, showcasing its ability to produce accurate simulations. Additionally, our model supports a wide range of Reynolds numbers in complex geometries, further substantiating its versatility and applicability.

20.2 Related Work

The shallow water equations are employed to simulate the behavior of water in environments where the water depth is significantly smaller than the horizontal dimensions, particularly in large-scale scenarios, such as oceans, seas, or large lakes. Derived from the full Navier–Stokes equations for fluid dynamics, SWE makes specific assumptions that enable more computationally efficient simulations. This approach has found widespread use in diverse applications, including ocean engineering [Salmon 99], hydraulic engineering [Valiani et al. 02], and coastal engineering [Tubbs and Tsai 19]. Traditional methods for solving SWE include the finite difference method [Su et al. 23, Chentanez and Müller 10], finite volume method [Brodtkorb et al. 12], and finite element method [Ricchiuto and Bollermann 09]. However, these conventional techniques introduce notable numerical dispersion, resulting in pronounced viscosity in the output results. They often encounter difficulties in accurately simulating bed slope and friction forces due to inherent numerical errors [García-Navarro et al. 19]. Moreover, these methods face challenges in handling complex geometries and are computationally expensive.

The lattice Boltzmann method (LBM) [Chen and Doolen 98] is a mesoscopic method based on statistical theory and has gained prominence as an effective alternative for simulating turbulent, incompressible flow, grounded in the Boltzmann equation and kinetic theory. A key advantage of LBM lies in its utilization of local streaming and collision operators, endowing it with high parallelizability [Li et al. 20, Lyu et al. 21, Li et al. 21, Li et al. 22, Li and Desbrun 23, Li et al. 23]. The Bhatnagar–Gross–Krook lattice Boltzmann method (BGK-LBM) has been extensively applied to solve shallow water equations with appropriate equilibrium distribution functions [Zhou 02]. In the pursuit of enhanced accuracy and stability, multiple-relaxation time (MRT) collision models have been proposed in various works [García-Navarro et al. 19]. Among these MRT methods, the non-orthogonal central-moment (NO-CMR) based lattice Boltzmann scheme has demonstrated superior performance [De Rosis 17].

The lattice Boltzmann model for shallow water equations, as introduced in [Zhou 02], offers an effective approach for simulating large bodies of water due to its computational efficiency, derived from the LBM scheme. However, a limitation arises from its reliance on the BGK collision model, which, due to its first-order nature, is constrained in handling low Reynolds numbers. To address this limitation, the non-orthogonal central moment (NO-CMR) was introduced into LBM-SWE [De Rosis 17]. This method transforms the distribution function into a central-moment space, where each component relaxes toward its equilibrium with an individual rate before being converted back. Despite this improvement, the force term in [De Rosis 17] still employs the low-order expression from [Zhou 02], which may not be accurate enough for turbulent flows. In our new LBM-SWE solver, we propose a high-order forcing model in moment space and a second-order weighted estimation for force calculation to obtain accurate force evaluation. To obtain stable boundary treatment in complex geometries, we also propose a hybrid bounce-back method for turbulent flowmaps.

20.3 Background

In this section, we first review the shallow water equations and then their corresponding lattice Boltzmann model.

20.3.1 Shallow Water Equations

The 2D shallow water equations are derived according to the continuity and momentum conservation, which can be written in a tensor form as

$$\frac{\partial h}{\partial t} + \frac{\partial h\mathbf{u}}{\partial \mathbf{x}} = 0, \tag{20.1}$$

$$\frac{\partial h\mathbf{u}}{\partial t} + \nabla \cdot (h\mathbf{u}\mathbf{u}) = -\frac{g}{2}\nabla^2 h + \nu\nabla \cdot \left(h\left(\nabla\mathbf{u} + (\nabla\mathbf{u})^T\right)\right) + F, \tag{20.2}$$

where $\mathbf{x}$ is the the Cartesian coordinate, h is the water height, $\mathbf{u}$ is the depth-averaged velocity component, $g = 9.81$ m/s^2 is the gravitational acceleration, ν is the kinematic viscosity, and F is the external force term. In particular, the external force term is defined as the sum of the hydrostatic pressure and bed shear stress:

$$\mathbf{F} = -gh\nabla z - \frac{\boldsymbol{\tau}_b}{\rho}, \tag{20.3}$$

where z is terrain height and $\boldsymbol{\tau}_b$ is the bed shear stress.

20.3.2 BGK-LBM for Shallow Water Equations

With the BGK approximation, Zhou [Zhou 02] first formulated SWE into the following lattice Boltzmann equation:

$$f_i(\mathbf{x} + \Delta t\mathbf{c}_i, t + \Delta t) - f_i(\mathbf{x}, t) = \frac{1}{\tau}\left[f_i^{\mathrm{eq}}(\mathbf{x}, t) - f_i(\mathbf{x}, t)\right] + \Delta t\mathcal{F}_i(\mathbf{x}, t), \tag{20.4}$$

where $\mathcal{F}_i(\mathbf{x}, t)$ is the force term, f_i and f_i^{eq} are the current distribution function and local equilibrium distribution function along ith direction on the nine-speed square lattice, as shown in Figure 20.1. The lattice velocity vector $\mathbf{c}_i$ is defined as

$$\mathbf{c}_i = \begin{cases} [0, 0], & i = 0; \\ [1, 0], [0, 1], [-1, 0], [0, -1], & i = 1, 2, 3, 4; \\ [1, 1], [-1, 1], [-1, -1], [1, -1], & i = 5, 6, 7, 8. \end{cases} \tag{20.5}$$

The relaxation factor τ is based on the fluid kinematic viscosity ν as $\tau = 3\nu + 0.5$, and the time step $\Delta t = 1$ is the convention as LBM space is dimensionless space.

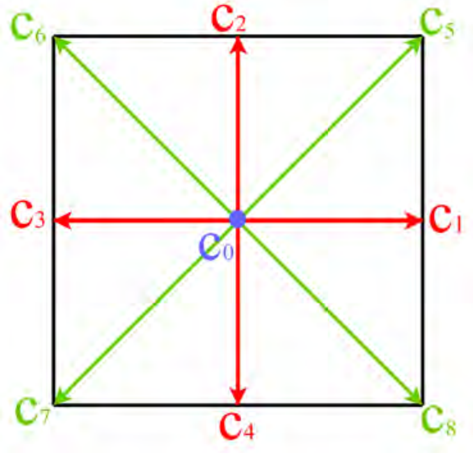

Figure 20.1. Two-dimensional nine-speed (D2Q9) lattice structure.

Equation (20.4) can be solved using the splitting method with two steps:

Step 1: Streaming First, note that

$$f_i^*(\mathbf{x}+\Delta t\mathbf{c}_i, t+\Delta t) = f_i(\mathbf{x}, t). \tag{20.6}$$

Then, the macroscopic variables h and $\mathbf{u}$ can be updated as

$$h = \sum_{i=0}^{8} f_i, \tag{20.7}$$

$$\mathbf{u}^* = \sum_{i=0}^{8} \frac{f_i^* \mathbf{c}_i}{h}. \tag{20.8}$$

Step 2: Collision For the second step,

$$f_i(\mathbf{x}, t) = f_i^*(\mathbf{x}, t) + \frac{1}{\tau}\left[f_i^{eq}(\mathbf{x}, t) - f_i^*(\mathbf{x}, t)\right] + \Delta t \mathcal{F}_i(\mathbf{x}, t), \tag{20.9}$$

where the local equilibrium distribution function f_i^{eq} is defined as

$$f_i^{\mathrm{eq}} = \begin{cases} h - \frac{5gh^2}{6} - \frac{2h}{3}\mathbf{u}\cdot\mathbf{u}, & i = 0; \\ \frac{gh^2}{6} + \frac{h}{3}\mathbf{c}_i\cdot\mathbf{u} + \frac{h^2}{2}(\mathbf{c}_i\cdot\mathbf{u})^2 - \frac{h}{6}\mathbf{c}_i\cdot\mathbf{u}, & i = 1,2,3,4; \\ \frac{gh^2}{24} + \frac{h}{12}\mathbf{c}_i\cdot\mathbf{u} + \frac{h^2}{8}(\mathbf{c}_i\cdot\mathbf{u})^2 - \frac{h}{24}\mathbf{c}_i\cdot\mathbf{u}, & i = 5,6,7,8. \end{cases} \tag{20.10}$$

The force term $\mathcal{F}_i(\mathbf{x}, t)$ can be computed by accounting for the external force $\mathbf{F} = (F_x, F_y)$:

$$\mathcal{F}_i(\mathbf{x}, t) = \frac{1}{6}\mathbf{c}_i \cdot \mathbf{F}(\mathbf{x}, t), \tag{20.11}$$

where the external force $\mathbf{F} = [F_x, F_y]$ is defined in Equation (20.3).

20.4 Our Method

In this section, we first review the central-moment MRT model for LBM, then introduce our high-order forcing formula, followed by our hybrid bounce-back method for stable boundary treatment.

20.4.1 NO-CMR LBM-SWE

The central-moment MRT model was first introduced to significantly improve LBM accuracy [De Rosis 17]. In 2D, this model uses non-orthogonal central moments $\mathbf{m}$ derived from distribution functions $\mathbf{f} = [f_0, ..., f_8]$ via a linear transform $\mathbf{m} = \mathbf{M}\mathbf{f}$, where $\mathbf{M}$ is a matrix known in closed form as a function of the macroscopic velocity $\mathbf{u}$. In particular, $\mathbf{M} = [m_0, ..., m_8]^T$ is a 9×9 moment-space projection matrix in which each row is defined as

$$m_0 = [1, 1, 1, 1, 1, 1, 1, 1, 1], \tag{20.12}$$

$$m_1 = [\bar{\mathbf{c}}_{0,x}, ..., \bar{\mathbf{c}}_{8,x}], \tag{20.13}$$

$$m_2 = [\bar{\mathbf{c}}_{0,y}, ..., \bar{\mathbf{c}}_{8,y}], \tag{20.14}$$

$$m_3 = [\bar{\mathbf{c}}^2_{0,x} + \bar{\mathbf{c}}^2_{0,y}, ..., \bar{\mathbf{c}}^2_{8,x} + \bar{\mathbf{c}}^2_{8,y}], \tag{20.15}$$

$$m_4 = [\bar{\mathbf{c}}^2_{0,x} - \bar{\mathbf{c}}^2_{0,y}, ..., \bar{\mathbf{c}}^2_{8,x} - \bar{\mathbf{c}}^2_{8,y}], \tag{20.16}$$

$$m_5 = [\bar{\mathbf{c}}_{0,x}\bar{\mathbf{c}}_{0,y}, ..., \bar{\mathbf{c}}_{8,x}\bar{\mathbf{c}}_{8,y}], \tag{20.17}$$

$$m_6 = [\bar{\mathbf{c}}^2_{0,x}\bar{\mathbf{c}}_{0,y}, ..., \bar{\mathbf{c}}^2_{8,x}\bar{\mathbf{c}}_{8,y}], \tag{20.18}$$

$$m_7 = [\bar{\mathbf{c}}_{0,x}\bar{\mathbf{c}}^2_{0,y}, ..., \bar{\mathbf{c}}_{8,x}\bar{\mathbf{c}}^2_{8,y}], \tag{20.19}$$

$$m_8 = [\bar{\mathbf{c}}^2_{0,x}\bar{\mathbf{c}}^2_{0,y}, ..., \bar{\mathbf{c}}^2_{8,x}\bar{\mathbf{c}}^2_{8,y}], \tag{20.20}$$

where $\bar{\mathbf{c}}_i = [\bar{\mathbf{c}}_{i,x}, \bar{\mathbf{c}}_{i,y}]$ is the shifted lattice velocities by the local fluid velocity $\mathbf{u}^*$, with $\bar{\mathbf{c}}_{i,x} = \mathbf{c}_{i,x} - \mathbf{u}^*_x$ and $\bar{\mathbf{c}}_{i,y} = \mathbf{c}_{i,y} - \mathbf{u}^*_y$. Row m_0 is simply the zeroth-order moment (equal to density), m_1 and m_2 are the components of the vector representing the first-order moment, $m_3, \ldots, m_5$ are the components of the symmetric second-order moment, and the others are high-order moments.

Then, Equation (20.4) can be rewritten in the moment format as

$$\mathbf{f}^{t+1} - \mathbf{f}^t = \mathbf{M}^{-1}\big(\mathbf{R}(\mathbf{m}^{\mathrm{eq}} - \mathbf{m}) - \mathbf{M}\mathcal{F}\big), \tag{20.21}$$

where $\mathbf{R} = \mathrm{diag}\{r_0, ..., r_1\}$ is the diagonal matrix containing the relaxation rates $r_i = \frac{1}{\tau}$ for $i = 4, 5$ and otherwise set to 1 [De Rosis 17].

20.4.2 High-Order Forcing for NO-CMR

We can divide the force terms into two parts according to the trapezoidal rule or Crank–Nicolson discretization. First, add one-half force into macro velocity after the streaming step:

$$h\mathbf{u}^* = \sum_i f^*_i c_i + \frac{\mathbf{F}}{2} = h\mathbf{u}^t + \frac{\mathbf{F}}{2}. \tag{20.22}$$

Then, the NO-CMR collision model from Equation (20.21) becomes

$$\mathbf{f}^{t+1} = \mathbf{f}^t - \mathbf{M}^{-1}\big(\mathbf{R}(\mathbf{m} - \mathbf{m}^{\mathrm{eq}}) + (\mathbf{I} - \tfrac{1}{2}\mathbf{R})\mathbf{K}\big), \tag{20.23}$$

where $\mathbf{K}$ represents the force terms projected into central-moment space,

$$\mathbf{K} = \left[0, F_x, F_y, 0, 0, 0, \frac{1}{3}F_y, \frac{1}{3}F_x, 0\right]^T. \tag{20.24}$$

It is easy to verify the forcing model's accuracy by taking the first-order moment of Equation (20.23) as

$$\mathbf{u}^{t+1} = \frac{\sum_i f_i^* c_i}{h} + \frac{\mathbf{F}}{2h} = \mathbf{u}^* + \frac{\mathbf{F}}{2h} = \mathbf{u}^t + \frac{\mathbf{F}}{h}. \tag{20.25}$$

For $\mathbf{R}$, we introduce the high-order relaxation times $\nu_6 = \nu_7 = 0.1$ and $\nu_8 = 0.1$ for corresponding to relaxation rate $r_i = 1/(3 * \nu_i + 0.5)$ for a lower dissipation error [Li et al. 20].

Instead of approximating the first term ∇z in Equation (20.3) by the central difference method [Zhou 02], We use the second-order weighted estimation used in [Li et al. 21]:

$$\nabla z(\mathbf{x}) = \sum_i w_i \mathbf{c}_i z(\mathbf{x} + \mathbf{c}_i). \tag{20.26}$$

It is a rotationally symmetric gradient approximation, which is more accurate and stable than the traditional central difference method.

20.4.3 Hybrid Bounce-Back

Zhou proposed the bounced-back method for the boundary treatment [Zhou 02], as illustrated in Figure 20.2 (left), which can be written as

$$\mathbf{f}_i(\mathbf{x}, t+1) = \mathbf{f}_{i'}(\mathbf{x}, t). \tag{20.27}$$

It means that the function $\mathbf{f}_{i'}$ at node $\mathbf{x}$ along the i' direction that collides with the solid boundary will bounce back to the ith direction to update

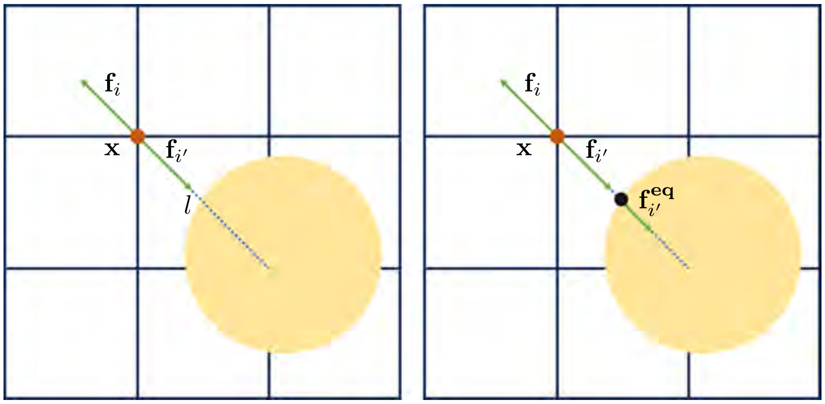

Figure 20.2. Boundary treatment: bounce-back method used in [Zhou 02] (left), and our hybrid bounce-back method (right).

$\mathbf{f}_i(\mathbf{x}, t+1)$. Unfortunately, it is too unstable to handle complex terrain geometries as it brings high-frequency ghost modes into low-order velocity moments, which ruins the simulation.

Motivated by the single-node boundary treatment [Filippova and Hänel 98, Li et al. 22], we proposed the stable hybrid bounce-back method in the LBM-SWE framework. In particular, the incoming distribution function coming from the boundary toward a fluid node along a boundary-crossing link l (dotted blue line in Figure 20.2, left) can be partially approximated by the equilibrium distribution function of a point at the intersection of the link and the boundary. In the context of shallow water fluid simulation with static obstacles, we impose the zero-velocity boundary condition (see Figure 20.2, right) on the obstacle to evaluate the local values of $\mathbf{f}^{\mathrm{eq}}(\mathbf{x})$ and blend the bounce-back distribution $\mathbf{f}_{i'}(\mathbf{x})$ and the equilibrium $\mathbf{f}_{i'}^{\mathrm{eq}}(\mathbf{x})$ on the fluid node $\mathbf{x}$ near the solid:

$$\mathbf{f}_i(\mathbf{x}, t+1) = (1-\alpha)\mathbf{f}_{i'}(\mathbf{x}, t) + \alpha\mathbf{f}_{i'}^{\mathrm{eq}}(\mathbf{x}, t), \tag{20.28}$$

where we set $\alpha = 0.1$ in practice. By adding the equilibrium function, we can filter out non-physical oscillations in the distribution function $\mathbf{f}_i$ after boundary treatment, which shows better stability in turbulent flow simulations.

20.4.4 Inlet/Outlet Boundary Conditions

We use the Dirichlet boundary condition for macroscopic variables, the water height h and the velocity $\mathbf{u}$. Based on the assumption that the distribution function $\mathbf{f}$ deviates only slightly from the equilibrium state $\mathbf{f}^{\mathrm{eq}}$, we use $\mathbf{f}^{\mathrm{eq}}$ to approximate $\mathbf{f}$ based on Equation (20.10).

20.4.5 Implementation Details

We implemented our approach in CUDA, using a structure-of-arrays (SOA) data structure [Chen et al. 22], storing 24 variables per grid node, in which nine variables store the distribution functions $\mathbf{f}$, with two copies to facilitate the time update, two variables for velocity, two variables for force terms, one variable for z, and one variable for h. Listing 20.1 provides the pseudocode of one iteration of our LBM-SWE update scheme. Please see the attached example application for specific implementations.

20.5 Results

In this section, we test our LBM-SWE solver on one benchmark case for verification and two examples with complex terrains to showcase the effectiveness of our method.

```
for each node at x
{
    for each direction $i$
    {
        if (Node $\mathbf{x} - \mathbf{c}_i$ is not water or $h(\mathbf{x} - \mathbf{c}_i) < z(\mathbf{x} - \mathbf{c}_i)$)
        {
            Hybrid bounce-back           // Section 20.4.3
        }
        else
        {
            Streaming                    // Equation (20.6)
        }
    }
    Evaluate $h$ and $\mathbf{u}^*$                // Equation (20.7)
    Calculate force terms              // Section 20.4.2
    Update $\mathbf{u}$                         // Equation (20.22)
    Collision                          // Equation (20.23)
    Boundary treatment                 // Section 20.4.4
}
```

Listing 20.1. Pseudocode of our LBM-SWE solver.

20.5.1 Circular Dam Break

We test our method in a circular dam break scene (proposed in [Peng et al. 16]) with a square periodic domain with length $L = 40$ m. In the center of this domain is a cylindrical water column with the following initial settings:

$$\mathbf{u}(\mathbf{x}, 0) = 0; \tag{20.29}$$

$$h(\mathbf{x}, 0) = \begin{cases} 2.5, & (\mathbf{x} - \mathbf{x}_c)^2 \leq R^2; \\ 0.5, & (\mathbf{x} - \mathbf{x}_c)^2 > R^2; \end{cases} \tag{20.30}$$

where $\mathbf{x}_c = [\frac{L}{2}, \frac{L}{2}]$ and the radius $R = 2.5$ m.

We use 100×100 grid nodes to discretize the domain with world-space $dt = 0.05$ s for each time step. Figure 20.3 visualizes the height profile at $t = 3.5$ s at the cross-section of $\mathbf{x} = x_c$, where circles indicate our result and dots the reference data from [Peng et al. 16]. The graph shows that our method demonstrates the same accuracy and robustness as the raw moment-based collision model does, but our central-moment model also has the ability to handle turbulent flow with complex terrain situations.

20.5.2 Flow through a Canyon

To highlight our handling of complex geometries, we run our LBM-SWE in a canyon in which the terrain height h is normalized in a range of $[0.02, 8]$.

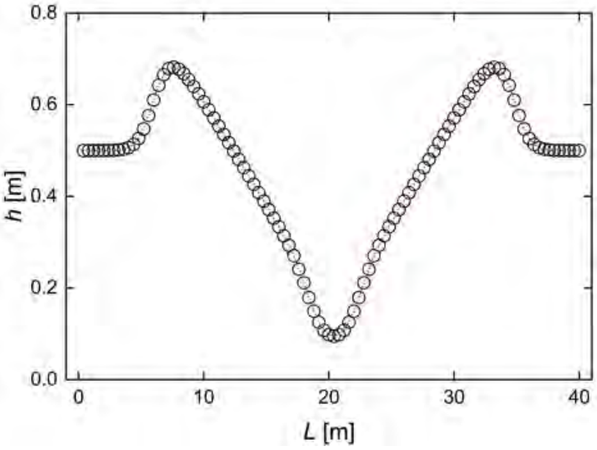

Figure 20.3. The height profile of a circular dam break generated by our solver (circles) and reference data from [Peng et al. 16] (dots).

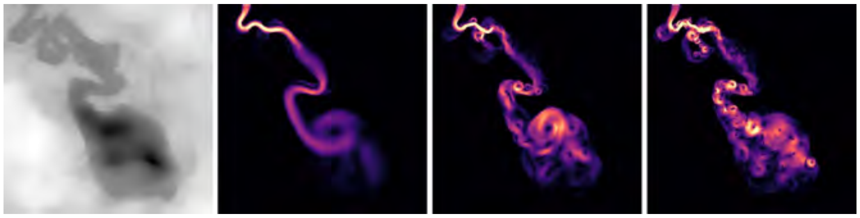

Figure 20.4. Given the height map of the terrain (left), the flowmaps are generated by our LBM-SWE with different Reynolds numbers: Re = 4000, Re = 80000, and Re = 800,000. The brighter the color, the larger the velocity magnitude.

The simulation resolution is 512×512. Figure 20.4 demonstrates the result of the velocity magnitude with different Reynolds numbers settings: Re = 4000, Re = 80000, and Re = 800,000. With a lower Reynolds number, the resulting flowmap demonstrates the smooth behavior of laminar flow with high viscous forces. In contrast, with a high Reynolds number, we can get a turbulent flow with eddies, vortices, and turbulent structures throughout the flow field, which highlights the advantage of our method over previous techniques.

We further evaluate the performance over three different resolutions, 256^2, 512^2, and 1024^2, on NVIDIA GeForce RTX 2070 Super (laptop GPU). We only need one frame of the velocity field as the flowmap to drive the water animation in real time. We run our LBM-SWE simulator for 18,000 steps to get a relatively stable velocity, the total times of which are 9 s, 27 s, and 126 s, respectively.

20.5.3 Flow in a River

We further test our solver by simulating the flow through a river. The simulation domain is discretized into a 502 × 502 grid, and the terrain height z is normalized to the range of $[0.01, 8]$. In this case, we set an inlet to the left of the river terrain and set a circle outlet at the end (Figure 20.5). In practice, we add the force term after 4000 steps to dampen the pressure wave from distribution functions in the initialization stage. Figure 20.6 demonstrates an example of foam and water displacement driven by the flowmap generated by our LBM-SWE solver.

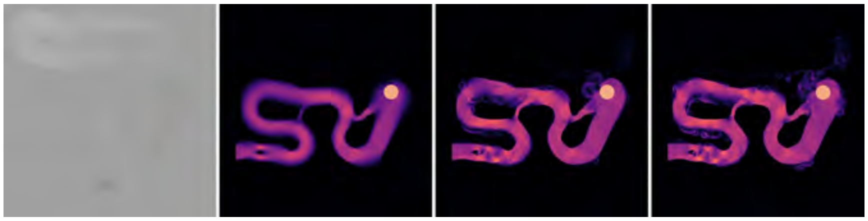

Figure 20.5. Given the height map (left), flowmaps are generated by our LBM-SWE solver with a fixed inlet and a constant outlet boundary (the pale yellow circle). With different Reynolds numbers (Re = 400, Re = 6000, and Re = 60,000), our LBM-SWE solver can capture different vortex details as shown in flowmaps. The brighter the color, the larger the velocity magnitude.

20.6 Conclusion

In this article, we proposed a new LBM-SWE solver with high-order force evaluation in moment space and symmetric gradient approximation for bed gradient, based on the central-moment collision model [De Rosis 17]. A new hybrid bounce-back was formulated to handle complex terrain for turbulent flow.

Leveraging the highly parallel nature of our solver, we achieved the efficient generation of all flowmaps, even at 2K × 2K resolution, within a matter of minutes. This practical and efficient computational performance renders our method well-suited for gaming applications.

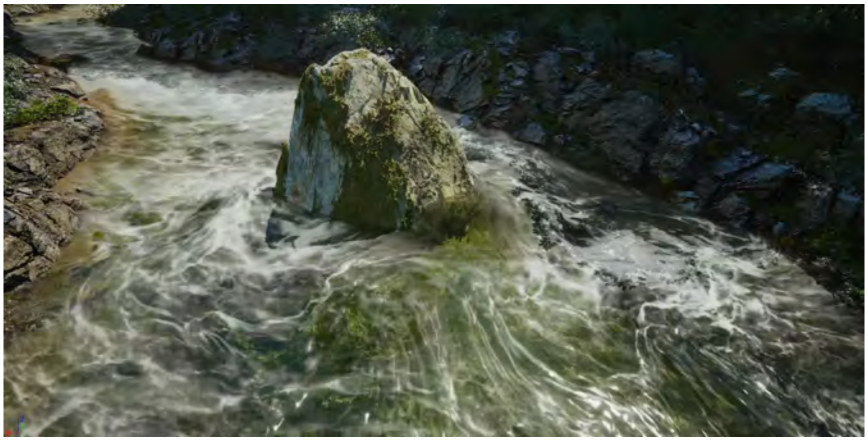

Figure 20.6. The foam and water displacement driven by the flowmap generated by our LBM-SWE.

Bibliography

[Brodtkorb et al. 12] André R. Brodtkorb, Martin L. Sætra, and Mustafa Altinakar. “Efficient Shallow Water Simulations on GPUs: Implementation, Visualization, Verification, and Validation.” *Computers & Fluids* 55 (2012), 1–12.

[Chen and Doolen 98] Shiyi Chen and Gary D. Doolen. “Lattice Boltzmann Method for Fluid Flows.” *Annual Review of Fluid Mechanics* 30:1 (1998), 329–364.

[Chen et al. 22] Yixin Chen, Wei Li, Rui Fan, and Xiaopei Liu. “GPU Optimization for High-Quality Kinetic Fluid Simulation.” *IEEE Transactions on Visualization and Computer Graphics* 28:9 (2022), 3235–3251.

[Chentanez and Müller 10] Nuttapong Chentanez and Matthias Müller. “Real-Time Simulation of Large Bodies of Water with Small Scale Details.” In *Proceedings of the 2010 ACM SIGGRAPH/Eurographics Symposium on Computer Animation*, p. 197–206. Eurographics Association, 2010.

[De Rosis 17] Alessandro De Rosis. “A Central Moments-Based Lattice Boltzmann Scheme for Shallow Water Equations.” *Computer Methods in Applied Mechanics and Engineering* 319 (2017), 379–392.

[Filippova and Hänel 98] Olga Filippova and Dieter Hänel. “Grid Refinement for Lattice-BGK Models.” *Journal of Computational Physics* 147:1 (1998), 219–228.

[García-Navarro et al. 19] Pilar García-Navarro, J Murillo, J Fernández-Pato, Isabel Echeverribar, and Mario Morales-Hernández. “The Shallow Water Equations and Their Application to Realistic Cases.” *Environmental Fluid Mechanics* 19 (2019), 1235–1252.

[Grenier 18] Jean-Philippe Grenier. “River Editor: Water Simulation in Real-Time.” Interview by Kirill Tokarev, 80 Level, October 24, 2018. https://80.lv/articles/river-editor-water-simulation-in-real-time/.

[Koschier et al. 22] Dan Koschier, Jan Bender, Barbara Solenthaler, and Matthias Teschner. “A Survey on SPH Methods in Computer Graphics.” *Computer Graphics Forum* 41:2 (2022), 737–760.

[Li and Desbrun 23] Wei Li and Mathieu Desbrun. “Fluid-Solid Coupling in Kinetic Two-Phase Flow Simulation.” *ACM Transactions on Graphics* 42:4 (2023), 123:1–123:14.

[Li et al. 20] Wei Li, Yixin Chen, Mathieu Desbrun, Changxi Zheng, and Xiaopei Liu. “Fast and Scalable Turbulent Flow Simulation with Two-Way Coupling.” *ACM Transactions on Graphics* 39:4 (2020), 47:1–47:20.

[Li et al. 21] Wei Li, Daoming Liu, Mathieu Desbrun, Jin Huang, and Xiaopei Liu. “Kinetic-Based Multiphase Flow Simulation.” *IEEE Transactions on Visualization and Computer Graphics* 27:7 (2021), 3318–3334.

[Li et al. 22] Wei Li, Yihui Ma, Xiaopei Liu, and Mathieu Desbrun. “Efficient Kinetic Simulation of Two-Phase Flows.” *ACM Transactions on Graphics* 41:4 (2022), 114:1–114:17.

[Li et al. 23] Wei Li, Tongtong Wang, Zherong Pan, Xifeng Gao, Kui Wu, and Mathieu Desbrun. “High-Order Moment-Encoded Kinetic Simulation of Turbulent Flows.” *ACM Transactions on Graphics* 42:6 (2023), 1–13.

[Lyu et al. 21] Chaoyang Lyu, Wei Li, Mathieu Desbrun, and Xiaopei Liu. “Fast and Versatile Fluid-Solid Coupling for Turbulent Flow Simulation.” *ACM Transactions on Graphics* 40:6 (2021), 201:1–201:18.

[Macklin and Müller 13] Miles Macklin and Matthias Müller. “Position Based Fluids.” *ACM Transactions on Graphics* 32:4 (2013), 104:1–104:12.

[Peng et al. 16] Y. Peng, J. M. Zhang, and J. G. Zhou. “Lattice Boltzmann Model Using Two Relaxation Times for Shallow-Water Equations.” *Journal of Hydraulic Engineering* 142:2 (2016), 06015017.

[Ricchiuto and Bollermann 09] Mario Ricchiuto and Andreas Bollermann. “Stabilized Residual Distribution for Shallow Water Simulations.” *Journal of Computational Physics* 228:4 (2009), 1071–1115.

[Salmon 99] Rick Salmon. “The Lattice Boltzmann Method as a Basis for Ocean Circulation Modeling.” *Journal of Marine Research* 57:3 (1999), 503–535.

[Su et al. 23] Haozhe Su, Siyu Zhang, Zherong Pan, Mridul Aanjaneya, Xifeng Gao, and Kui Wu. “Real-Time Height-Field Simulation of Sand and Water Mixtures.” In *SIGGRAPH Asia 2023 Conference Papers, SA ’23*, pp. 65:1–65:10. Association for Computing Machinery, 2023.

[Tubbs and Tsai 19] Kevin R. Tubbs and Frank T.-C. Tsai. “MRT-Lattice Boltzmann Model for Multilayer Shallow Water Flow.” *Water* 11:8 (2019), 1623.

[Valiani et al. 02] Alessandro Valiani, Valerio Caleffi, and Andrea Zanni. “Case Study: Malpasset Dam-Break Simulation Using a Two-Dimensional Finite Volume Method.” *Journal of Hydraulic Engineering* 128:5 (2002), 460–472.

[Vlachos 10] Alex Vlachos. “Water Flow in Portal 2.”, 2010. Advances in Real-Time Rendering in 3D Graphics and Games, SIGGRAPH Course.

[Zhou 02] Jian Guo Zhou. “A Lattice Boltzmann Model for the Shallow Water Equations.” *Computer Methods in Applied Mechanics and Engineering* 191:32 (2002), 3527–3539.