

Article

Lattice-Boltzmann and Eulerian Hybrid for Solid Burning Simulation

Eunchan Jo ¹, Byungmoon Kim ² and Oh-Young Song ^{1,*}¹ Department of Software, Sejong University, Seoul 05006, Korea; graphics.joe.kr@gmail.com² Adobe Research, San Jose, CA 95110, USA; bmkim@adobe.com

* Correspondence: oysong@sejong.edu

Received: 22 October 2019; Accepted: 11 November 2019; Published: 14 November 2019



Abstract: We propose a new hybrid simulation method to model burning solid interactions. Unlike gas fuel, fire and smoke interactions that have been relatively well studied in the past, simulations of solid fuel combustion processes remain largely unaddressed. These include pyrolysis/smoldering, interactions with oxygen and flow inside porous solid. To advance this simulation problem, we designed a new hybrid of the Lattice-Boltzmann method (LBM) and a Eulerian grid based Navier-Stokes equation (NSE). It uses the LBM, which has symmetrical directions of particle velocities in a cell, for inside the solid fuel and the NSE, which has a representative velocity in a cell, for outside the solid. At the interface where the two methods join, we develop a novel method to exchange physical quantities and show a natural transition between the two methods. Since LBM allows us to directly manage the quantity of exchanges from the microscopic perspective, that is, between lattice points, we can easily simulate the burning speed and the shape change of burning an inhomogeneous solid. Also, we derive an LBM version of the previously proposed porous Navier-Stokes equation to simulate gas flow inside the porous solid. In addition, we use the NS solver outside the solid where macroscopic behavior is much more dominant and, hence, LBM is less efficient than NS solver. Our results show us the physical stability and accuracy and visual realism.

Keywords: simulation; fire; heat; solid; combustion; Lattice-Boltzman; porosity

1. Introduction

Fire is common in everyday life in a variety of aspects from candle fires to larger scale forest fires. Fire also plays an important role in content like animations, movies and game applications. Because of this, fire simulation has been broadly studied in the graphics community. Over the last few decades, most of the focus in fire simulation research has been on the temperature, smoke, and gas fuel interactions to generate realistic behavior. To describe realistic fire or flame simulation, the computer graphics community introduces the physics theory with PDE equations. These formulas are discretized using numerical methods. However, there are few studies on the combustion of solid, in particular solid fuel. To simulate burning solid objects, solids and fire simulation should be separately simulated and combined somehow, which have always involved challenges or perhaps weakly principled treatments. Often the physical combustion model for burning solid is not too complicated, namely, some or all of the gas fuel, oxygen, char, ash, or soot, as well as material property, are not fully considered. This makes the solid burning process less accurate. In this paper, we carefully model the combustion phenomenon to enhance realism.

Depending on a solid's internal structure and material property, the shape and propagation speed of fire, smoke and temperature and chemical state change can be very different. Common solid fuels such as woods, papers, plastics and rubbers are not dense and burn in various ways, going through a complex combustion process—thermochemical process where molecules are violently

composed and decomposed. For graphics applications, we do not consider the invisible chemical process and the focus is on modeling the thermal phenomena only. We first develop a governing equation, which is introduced at the fluid simulation in physics area for a combustion and porous medium, which describes the evolution of density, smoke, temperature and heat inside and outside uneven solid material. To solve this equation, it will be discretized using numerical modeling to Navier-Stokes equation (NSE) or Lattice-Boltzmann method (LBM). For this, we converted the heat equation to LBM in Section 4. Solid is modeled with porosity and oxygen is also used to increase realism. Finally, the discretized solid material is rendered using the method of graphics area for realistic visualization.

LBM is developed in the computational fluid dynamics community as a method to compute microscopic fluid flows, whereas NS is from the macroscopic perspective and used in the graphics community to simulate liquid seeping or pigment percolation in painting canvas or to simulate even free liquid surfaces. In contrast to the traditional Eulerian based NSE simulation, which updates values of nodes with a discretized NS solver, LBM considers fluid flow as fictive particle exchanges amongst discrete lattice grids with fixed numbers of connections, usually using 9 in 2D and 19 in 3D with symmetry lattice to a neighboring grid. That is, NSE has a representative value in a cell but LBM has a static number of values with symmetric directions in a cell. Because of this, they are called the macroscopic and microscopic values, respectively. Since LBM is derived from the momentum of microscopic quantity, control of local behaviors, such as local flow being partially blocked in a certain direction by uneven porosity of solid, is much easier than the NS solver.

LBM has many advantages over NSE to simulate fluid flow. All operations are simple and local. It has only two steps—streaming and collision—but has no global operations like diffusion and projection. Because of this, the Poisson equation in NSE is eliminated and the LBM can be easily implemented with parallel hardware. Each cell contains not only velocity and density but also the value of distribution in fixed directions. This makes it possible to handle complex boundaries and incorporate relatively microscopic physics. So, we can efficiently model porous medium as wood. Despite the many advantages, LBM has some drawbacks. As LBM performs an explicit time step, velocities are limited to keep stability. But this will not be an issue in this paper, because combustion on solids is quite slow compared to air free flow. Another disadvantage of LBM is that it requires more memory for fictive particles and LBM can be applied only to discrete regular square grids.

Previously, LBM is used to simulate fluids but we adapt LBM to a burning solid simulation. Because temperature dominates the fire and burning solid object state (Section 3.1), we consider burning solids as a type of propagation of temperature. For this, we derive the temperature governing equation using LBM. Also for more interesting temperature flow and burning simulation we chose wood as the burning material. Wood is a typical porous medium. Because wood has very complicated internal structures, sometimes it has tube-like vessels called a vascular system. These internal structures can influence the combustion propagation direction or the burning pattern. Also, a lot of interesting things occur like incomplete combustion, reactions with oxygen, object bending, distortion, and charcoal.

Our contributions include:

- By separating combustion stages, we design governing equations for each stage.
- We introduce LBM for physical quantity exchanges in solid fuel, for example, microscopic heat exchange.
- We derive the Lattice-Boltzmann equation for physical quantity exchange in a porous material like wood.
- We apply the oxygen to control the combustion time and propagation.
- We propose an LBM and NSE hybrid simulator and a new method to exchange physical quantities.

We present the designed combustion stage and basic LBM concept in Section 3. Next, we derive the LBM version of the previously proposed porous Navier-Stokes equation for porous thermal flow. Section 5 describes the method of coupling between LBM and NSE at the interface.

2. Related Work

Since Reeves [1], procedural models using particles have been widely used to produce fire or flame animation [2–4]. These models can be easily implemented and offer easy control mechanisms to the user; however, they lack physical realism. In the past decade, researchers have introduced physics-based methods for simulating fluid phenomena. In general, fluids simulation methods have been developed based on a Navier-Stokes solver. After Stam [5] introduced an unconditionally stable solver for incompressible fluid, non-Newtonian fluid [6–8] as well as a lot of Newtonian fluid [9–14], including fire or flame simulation [15–20], were introduced. Fedkiw et al. [21] introduced a simple buoyancy force model coupled with change of temperatures to simulate heated gas and adopted a vorticity confinement model to create interesting swirling effects of smoke. Nguyen et al. [16] have augmented this framework to model fire by incorporating a level-set method to capture a thin flame surface. To add the highly turbulent details on a thin flame, Hong et al. [18] adopted detonation shock dynamics to force create a wrinkled flame structure. In these thin flame models, flammable materials are gasified to gas fuels that burn at the infinitely thin surface. Alternatively, several researchers [14,19,22] tailored smoke simulation directly to produce flame without level-set capturing of thin flame. Compared to the thin flame models, this model can generate results much faster while it shows diffusive flame shapes and smoke-like fire movement. Son et al. [20] improved the combustion process for gas fuel. Thanks to these physics-based flame modeling, computer-generated flame scenes become realistic. However, none of these works modeled an actual combustion process for solids. In Melek and Keyser [23], oxygen is modeled simply as the total gas amount minus the amount of gas fuel and smoke per cell.

Compared to flame simulation techniques, studies on solid burning are rare in the computer graphics field. In work by Yu et al. [24], a simple fire propagation model on solids was introduced. Burnt parts are simply removed along the fire front of an object. Losasso et al. [25] introduced a solid erosion model to simulate burning objects by tracing the changing surface of the remaining object region. At first glance, their results seemed to be plausible. The deformation of burnt regions, however, is not incorporated into the model and also no charred residue remained. Later, Melek [26] introduced a simple free-form deformation model to simulate burning woods and the crumpling of burning papers. Recently, Pirk [27] simulated a dynamic tree model for wood combustion. Because tree branches are made by particles bonding, this can be broken when the temperature rises over the threshold. Although the model provides an intuitive control mechanism for large-scale deformation, it is difficult to generate complex and detailed deformation including topological changes of solids. All of these approaches utilized oversimplified combustion models on solid objects without considering the complex interactions between internal structures of objects and oxygen.

In order to simulate fluid flow through porous media that have complex internal structures, researchers prefer to use the Lattice-Boltzmann method (LBM) [24,28,29]. Compared to the traditional Navier-Stokes solvers, the LBM scheme can easily incorporate internal phenomena inside objects that are hard to describe microscopically. Chu and Tai [30] devised an LBM-based 2D fluid solver for simulating percolation in disordered media, like paper. Later, Echevarria et al. [29] presented a framework to hybridize a thin two-layered Navier-Stokes solver and LBM to simulate liquid flows to produce analog photographic manipulations. LBM with porous media also has been researched in other fields. Aaltosalmi [31] introduces fluid flow with porous media and porous media with rock geometry which was simulated by Boek and Venturoli [32] in mathematics. Grucelski and Pozorski [33] use thermal LBM to simulate the transformation of geometry by heat. They make porous media using a lot of small obstacles and deal with porous media like a boundary. The heat can flow into an obstacle. This porous media is of relatively large size, namely an obstacle that can occupy many cells, but we design many porous sections in a cell.

The unique characteristics of LBM make it well-suited to parallel computing. Therefore, the LBM framework has been proposed as a direct alternative to Navier-Stokes. Wei et al. [34] presented an LBM-based method to produce gaseous phenomena in real-time while maintaining plausible details.

Later, in order to simulate liquids, Thuerey et al. [35,36] augmented an LBM scheme for representing a dynamically evolving liquid surface, which is based on combining it with a VOF (volume of fluid) model [37]. In work by Zhu et al. [38,39], liquid-liquid mixture phenomena are simulated by introducing a two-fluid LBM model. Although these challenging works demonstrated the potential of LBM, the NS solver and SPH still offer higher quality results. On the other hand, LBM enjoys unique characteristics and behaviors that NS solver and SPH do not, which is well demonstrated by Chu and MoXi [30]. Moving on, in this paper, we study LBM formulation of gaseous fluid inside a solid and hybridize the solver with NS solver.

3. Burning Solid and Lattice-Boltzmann Method

3.1. Burning Process

We first briefly review the solid burning process and propagation of a physical quantity from a combustion engineering perspective. The process of combustion consists of three relatively distinct steps with overlapping phases—preheating, gaseous and charcoal phases. Each phase depends on temperature and multiple transition states. The physical quantity change rate at each phase is represented by time differentials.

In the preheating phase, the solid is heated up to the ignition point. Due to the heat, the chemical process decomposes solid into flammable gas and charcoal, but there is no flame in this phase. Solid and gas are mixed up and heat exchange between them occurs at the microscopic scale inside the solid. This phase can be described with solid fuel s , gas ρ and charcoal C quantities as

$$\dot{s} = -k_{\text{pre}} T^{\frac{1}{2}} e^{-\frac{1}{RT}}, \quad \dot{\rho} = -k_{\text{sp}} \dot{s}, \quad \dot{C} = -k_c \dot{s}. \quad (1)$$

T is temperature, R is gas constant. k_{pre} , k_{sp} and k_c are coefficients for the reduction of solid fuel and the generation of gas fuel and charcoal, respectively. These equations are derived from Arrhenius equation. When solid is heated up, solid fuel s will be decreased at relatively low speed and gas and charcoal will be slightly evolved proportionally to the fuel consumed, as in Equation (1). If an external heat source keeps the temperature high enough to continue pyrolysis, but low enough not to ignite flammable gas generated, pyrolysis can continue without flame. Eventually, the temperature reaches the ignition point of the flammable gas. When gas starts to burn, a much greater amount of heat is released and the solid can burn without an external heat source. Then the second stage starts.

The second stage is the gaseous phase, where flammable gases are mixed with oxygen and ignited at much a higher temperature than the pyrolysis point. In this phase, flames are often visible. The flame, also the energy, is produced in the form of heat and light. Heat transfer from the combustion to the solid fuel continues to evolve the flammable gases. Simultaneously, this flammable gas is burned releasing large amounts of heat that increase the temperature and maintains the flames. The internal structures of the solid will be changed chemically and physically in a drastic way. Combustion also intensively consumes the oxygen around it. If oxygen is insufficient, incomplete combustion occurs producing residual products like soot or carbon monoxide. Because of this drastic reaction, the reaction constant is much higher than in the preheating phase. Oxygen O , Temperature T and smoke S_m is evolved by

$$\begin{aligned} \dot{s} &= -k_{\text{ign}} T^{\frac{1}{2}} e^{-\frac{1}{RT}}, \quad \dot{\rho} = -k_{\text{sc}} \dot{s}, \quad \dot{O} = k_{\text{oxy}} \dot{s}, \\ \dot{T} &= -k_{T,w} \dot{s}, \quad \dot{S}_m = -k_{S_m} \dot{s}. \end{aligned} \quad (2)$$

k_{ign} , k_{oxy} , k_{sc} , $k_{T,w}$ and k_{S_m} are coefficients for decreasing solid fuel and oxygen and increasing gas fuel, heat and smoke, respectively.

As flammable gas runs out and the temperature decreases, the final charcoal phase starts. When the temperature is higher than the ignition point of the charcoal, the charcoal burns to ashes. The reaction constant is much lower than combustion in order to burn slowly, and also a small

amount of heat will be generated. Heat loss may drop temperatures faster than the generated heat. Then, the burned charcoal will be cooled down and the fire disappears.

$$\dot{C} = -k_{\text{ign},c} T^{\frac{1}{2}} e^{-\frac{1}{RT}}, \quad \dot{T} = -k_{T,c} \dot{C} \quad (3)$$

C is the reduction of charcoal. $k_{\text{ign},c}$ and $k_{T,c}$ are coefficients for charcoal and heat generation in the charcoal phase.

3.2. Lattice Boltzmann Method

The Lattice Boltzmann Method is a grid based simulation method for fluids flow used in the area of computational fluid dynamics. Compared with the typical Navier-Stokes equation, LBM can be used to calculate physical propagation using streams and collision operations at each lattice grid. The LBM approach assumes fictive particles are randomly moving in a divided simulation area with a regular lattice. These particles propagate grid cells in a constrained direction using distribution functions (DFs) f_i as statistical physics. There are many ways to constrain direction. We use the LBM model D2Q9 for 9 lattice velocity directions in 2D and D3Q19 for 19 velocity directions in 3D at each cell. The lattice velocity is presented by $\vec{e}_i (i = 1 \cdots 19)$ in 3D and that is microscopic velocity. During time stepping, particles exchange physical quantities on these lattice velocities through stream and collision step.

In the streaming step, all DFs are advected with their respected normalized velocities. Use of the normalized vectors makes it possible to handle this advection by simply copying as

$$f_i^*(\vec{x} + \vec{e}_i \Delta t, t + \Delta t) = f_i(\vec{x}, t), \quad (4)$$

where Δt is the time step. Obviously, the streaming step alone is not enough to simulate the incompressible movement of fluids. So, the colliding step is required. In this step, the collision of fluid particles redistributes DFs toward their equilibrium DFs f_i^{eq} . For example, in the D3Q19 model,

$$f_i^{\text{eq}} = w_i \rho \left[1 + 3\vec{e}_i \cdot \vec{u} - \frac{3}{2} u^2 + \frac{9}{2} (\vec{e}_i \cdot \vec{u})^2 \right], \quad (5)$$

where ρ is the fluid density and the weights are given by $w_1 = 1/3$, $w_{2 \dots 7} = 1/18$ and $w_{8 \dots 19} = 1/36$. The corresponding lattice velocities are symmetrical and are given by $e_1 = (0, 0, 0)$, $e_{2,3} = (\pm 1, 0, 0)$, $e_{4,5} = (0, \pm 1, 0)$, $e_{6,7} = (0, 0, \pm 1)$, $e_{8 \dots 11} = (\pm 1, \pm 1, 0)$, $e_{12 \dots 15} = (0, \pm 1, \pm 1)$ and $e_{16 \dots 19} = (\pm 1, 0, \pm 1)$. In this equation, macroscopic fluid quantities like velocity and density used in the traditional grid method can be computed by summation of all the DFs for one cell

$$\rho = \sum f_i, \quad \vec{u} = 1/\rho \sum \vec{e}_i f_i. \quad (6)$$

The equilibrium DFs represent a stationary state of the fluid—a very viscous state. To make an incompressible state, the streamed DFs f_i^* need to mix equilibrium DFs up.

$$f_i(\vec{x} + \vec{e}_i \Delta t, t + \Delta t) = (1 - \omega) f_i^*(\vec{x} + \vec{e}_i \Delta t, t + \Delta t) + \omega f_i^{\text{eq}}. \quad (7)$$

This is performed by linearly weighting each of the streamed DF with the equilibrium DF. With these two simple steps, fluid motion can be simulated. It was proven that LBM actually solves incompressible Navier-Stokes equations that are validated on small Mach numbers ($\ll 1$) $|\vec{u}|/C_s$, where C_s is the sound speed, with error $O(Ma^2)$ [40]. Since burning speed is extremely slow from the microscopic perspective in wood burning simulations, our method satisfies the low-Mach number limit. Readers who are interested in more detail may refer to Reference [41].

4. LBM Model for Porous Thermal Flow

Simulation steps of a burning object are determined by temperature that is a variable of time and space. Fluid simulation for computer graphics usually use the temperature field for heat transfer.

$$\frac{\partial T}{\partial t} + \vec{u} \cdot \nabla T = \nu_T \nabla^2 T - c_T T^4. \quad (8)$$

This heat equation represents heat transfer and is also known as a simple form of energy equation. Like NSE, it includes advection, diffusion and heat loss due to radiation. To solve the governing equations for wood burning, we begin with a basic LBM model. We extend the LBM to handle the temperature field. We just add an energy distribution function g_i for temperature. The temperature can be flowed with streaming (9) and colliding (10) processes in a time step the same as in the basic LBM. But the equilibrium DF for temperature is simpler, approximated up to the first order of velocity because the calculation of temperature is less critical than density or velocity calculations. The first order approximation is sufficient to our experiments.

$$g_i^*(\vec{x} + \vec{e}_i \Delta t, t + \Delta t) = g_i(\vec{x}, t), \quad (9)$$

$$g_i^{eq} = w_i \rho [1 + 3\vec{e}_i \cdot \vec{u}]. \quad (10)$$

To satisfy incompressible state, the energy distribution function Equations (9) and (10) need to be mixed up.

$$g_i(\vec{x} + \vec{e}_i \Delta t, t + \Delta t) = (1 - \omega) g_i^*(\vec{x} + \vec{e}_i \Delta t, t + \Delta t) + \omega g_i^{eq}. \quad (11)$$

The macroscopic temperature quantity is a summation of microscopic quantity g_i . It is used in boundary conditions in Section 5.

$$T = \sum g_i, \quad (12)$$

In order to model the porous propagation of flow and heat, the existing governing equation must be modified. Because wood is a typical porous medium, the governing equations should be able to represent porous flow with heat transfer inside wood. Conventional NS equations are not sufficient. The model for porous medium with a variable porosity was proposed by Nithiarasu et al. [42]. The model can be expressed by the following generalized Navier-Stokes equation:

$$\nabla \cdot \vec{u} = 0, \quad (13)$$

$$\frac{\partial \vec{u}}{\partial t} + \vec{u} \cdot \frac{\nabla \vec{u}}{\epsilon} = -\frac{1}{\rho} \nabla(\epsilon p) + \nu_u \nabla^2 \vec{u} + \vec{F}, \quad (14)$$

where \vec{u} and p are the velocity and pressure in the medium with the porosity ϵ , respectively, and ν_u is an effective viscosity. \vec{F} represents the total body force due to the porous medium and buoyancy force and is given by

$$\vec{F} = -\frac{\epsilon \nu_u}{\kappa} \vec{u} - \frac{1.75}{\sqrt{150\epsilon\kappa}} |\vec{u}| \vec{u} + \epsilon \beta (T - T_m) \vec{e}_y, \quad (15)$$

where β , T_m and \vec{e}_y are a thermal expansion coefficient, an ambient temperature and the upward direction, respectively. The permeability κ of the porous medium is related to the porosity ϵ based on Mavko and Nur's work [43] and can be expressed as

$$\kappa = R_\kappa \frac{(\epsilon - \epsilon_p)^3}{(1 - \epsilon + \epsilon_p)^2}, \quad (16)$$

where R_κ is a proportional permeability coefficient and ϵ_p is a percolation porosity of 0.025 [43]. In the RHS of Equation (15), the first two terms are for porous mediums. These terms play a role in resisting flow in the opposite direction. The last term presents buoyancy force. Actually, when porosity becomes

1, permeability will become nearly infinite, these equations reduce to typical Navier-Stokes equations with heat transfer coupled by only a buoyancy force since porous medium terms vanish. All of these equations are coupled by velocity, temperature and force. We can design the porous LBM model corresponding to the macroscopic Equations (13) and (14) through the Chapman-Enskog expansion (see Appendix A for details). The derived density and heat streaming step,

$$f_i^*(\vec{x} + \vec{e}_i \Delta t, t + \Delta t) = \epsilon f_i(\vec{x}, t) + (1 - \epsilon) f_i(\vec{x} + \vec{e}_i \Delta t, t) \quad (17)$$

$$g_i^*(\vec{x} + \vec{e}_i \Delta t, t + \Delta t) = \epsilon g_i(\vec{x}, t) + (1 - \epsilon) g_i(\vec{x} + \vec{e}_i \Delta t, t), \quad (18)$$

are controlled by the amount of porosity in a cell and interpolate between the previous value and the streaming value. Low porosity means slow streaming. These modified versions keep simply still. Next, the derived colliding steps are

$$f_i^{eq} = w_i \rho \left[1 + 3\vec{e}_i \cdot \vec{u} - \frac{3}{2\epsilon} \vec{u}^2 + \frac{9}{2\epsilon} (\vec{e}_i \cdot \vec{u})^2 \right], \quad (19)$$

$$\vec{u} = \frac{\vec{v}}{c_0 + \sqrt{c_0^2 + c_1 |\vec{v}|}}, \quad (20)$$

$$\vec{v} = \sum \frac{\vec{e}_i f_i}{\rho} + \frac{\epsilon \beta (T - T_m) \vec{e}_y}{2}, \quad (21)$$

$$c_0 = \frac{1 + \frac{\epsilon v_u}{2\kappa}}{2}, \quad (22)$$

$$c_1 = \frac{1.75\epsilon}{2} \sqrt{150\epsilon\kappa}. \quad (23)$$

Though slightly complicated, when porosity and permeability become higher and higher while burning, flow becomes faster and faster with this equation. The reason is that the denominator becomes smaller and smaller.

5. NSE and LBM Interface Handling

In this section, we handle 3 types of boundary conditions. First, the simulation domains have a free surface boundary that has equal values of the nearest simulation grid and can spread physical quantities out like smoke and temperature. Second, because air and object space simulate different numerical methods for each NSE and LBM, physical quantity exchange needs to be considered between these regions. Lastly, the boundary surface between oxygen and combustion of the object was considered. Because the first boundary condition is common in computer graphics, we focus on the other two.

5.1. Exchanging Quantity between Solid (LBM) and Air (NSE) Domains

We simulate the inside of the burning solid using LBM to describe the microscopic flow of physical quantities and the outside of the solid using the NS solver to propagate macroscopic flows of physical quantities such as heat, smoke, gas fuel and oxygen. At the boundary of the solids, the LBM grid meets the NSE grid and the two methods need to communicate with each other.

First, when solving an NS grid cell, having a neighbor of an LBM cell, at the boundary, the macroscopic quantities are needed like density, velocity and temperature from the LBM cell. These are easily computed by the summation of distribution functions (DF) at each of the LBM grid locations. These quantities can be set as boundary conditions of the Navier-Stokes domain. Figure 1 illustrates this using velocity as an example. The top row is the LBM cell and the bottom row is the NSE cell. When updating the cell *B* that is a neighbor of the LBM cell, the *β* cell has to convert

the microscopic (blue dot arrows) value to a macroscopic (green arrow) value by the summation of DF values.

Second, with the other way, solving the LBM grid at the boundary, the microscopic quantity needed to transfer air-to-solid is more complex. Macroscopic values from the NS solver should be decomposed into microscopic values for each DF as illustrated in Figure 1 (A, B and C cells). For this, we derive the following equation.

$$f_i = w_i \left[1 + 3\vec{e}_i \cdot \vec{u} - \frac{3}{2}u^2 + \frac{9}{2}(\vec{e}_i \cdot \vec{u})^2 \right] \quad (24)$$

This equation, similar to the equilibrium equation, converts a macroscopic quantity into distribution functions that is a set of directional vectors of particle velocities f_i . When updating the cell β , the macroscopic (red arrows) value converts to a microscopic (gray dot arrows) value and these quantities currently belong to an NS solver region. In the LBM streaming steps, these quantities (blue dot arrow in A, B, C, α and γ cells) will be streamed to LBM cells β .

According to solid density, matter has a different penetration ratio. Because of this we designed the weight to quantity. At the streaming stage—the air to solid boundary—the physical quantity must be scaled as a function of porosity. If the porosity of an LBM cell is close to 1, which means the cell has little solid but is almost full of air and thermal conductivity is as high as the neighbor air cell. At this time, DFs will be weakly scaled down. In contrast, if porosity is close to 0, DFs are scaled down to very small numbers, which means that air hardly flows into the solid. Since air and solids have different characteristics, the thermal capacity changes as porosity varies. As the porosity of a solid increases, the thermal capacity is rapidly decreased. Therefore, in the case of temperature, we apply a more drastically changing graph related to heat capacity as shown in Figure 2. This results in a natural evolution of burning shapes as shown in Figure 3 and video in the supplemental materials.

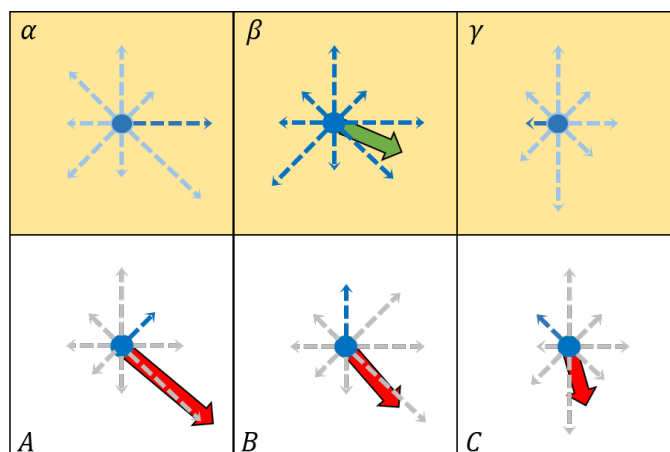


Figure 1. Illustration of physical quantity exchanges across LBM and NSE interface. Upper row is the LBM domain and bottom row is the NSE domain. The macroscopic value (green row) from LBM is required for evolution of the NSE grid cell B. The value of the NSE grid cell has to decompose for the streaming of the LBM grid cell β .

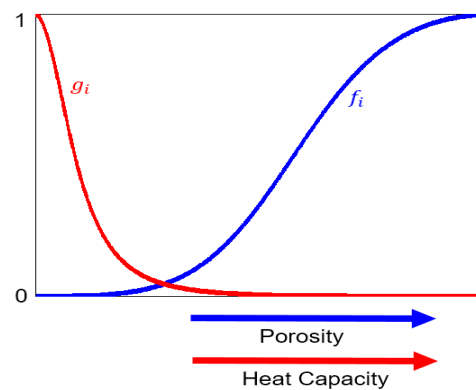


Figure 2. Weighting function for temperature streaming.



Figure 3. Wood burning simulation in 3D. The wood is eroded by the collaboration of velocity fields computed in the NSE field, weighted streaming into solid domain (NSE to LBM) and solid-internal flows simulated by LBM.

5.2. Oxygen Considering

In the real world, it is well known that combustion requires three elements—fuel, temperature and oxygen. In much of the research about fire in computer graphics, oxygen is rarely used in terms of the shape, speed and the direction of fire being deeply affected. To increase the realism of simulation, we add an oxygen field. Oxygen can flow into the solid cell with scaling down using the function of the porosity and is infinitely supplied from outside the simulation domain. It is also only consumed when the solid and gas fuel is in the gaseous phase of the burning process. If oxygen is short in the gaseous phase, the burning solid is delayed and oxygen will be filled enough in the next step so that the burning solid is restarted. Because of this, there are various burning solid shapes and we can control the burning speed using the oxygen threshold. For example, some sphere object is supposed to be set on fire. Without oxygen, the object will be burned bottom to top gradually. It means the top of the object furthestmost from the ignition point is burned last. But in the real world with oxygen, the combustion goes through an object from the outside to the inside gradually and the center of the sphere would be burned last. We design oxygen fields like the other physical quantities including an advection and a diffusion step and simulate consumption by burning at the gaseous step.

$$\frac{dO}{dt} = -k_{oxy} T^{\frac{1}{2}} e^{-\frac{1}{RT}} + \nu_O \nabla^2 O - \vec{u} \cdot \nabla O, \quad (25)$$

where ν_O is the oxygen diffusion coefficient. This equation is similar to solid fuel consumption and we can control the decreasing quantity. If oxygen is below a threshold at a cell, a burning object will stop or defer.

6. Experiments

All the experiments were performed on an Intel Core i7 3770 3.4 GHz processor with 24 GB of memory. We used an octree grid based Eulerian simulation for discretization. For the LBM domain of a solid, we refined the octree grid uniformly. We used a root array of $16 \times 32 \times 16$ and refined up to four levels. Hence, the maximum simulation grid size would be $256 \times 512 \times 256$. Perlin noise was

used to generate the porosity of solid objects. Our solver was parallelized using the OpenMP library. Heat sources with 1000 degrees for each experiment are provided for the first 0.1 s, in order to provide enough initial heat to make the objects start the combustion process. An exception is the pyrolysis test, which cannot self-sustain without external heat. Therefore, we proved a low temperature heat source constantly. In the experiments, we used the following constants: $k_{pre} = 0.005$, $k_{sp} = 0.05$, $k_c = 0.9$, $k_{ign} = 0.15$, $k_{sc} = 0.05$, $k_{T,w} = 40$, $k_{sm} = 15$, $k_{ign,c} = 0.1$, $k_{T,c} = 3.3$, $c_T = 0.3$, $\nu_u = 0.1$ and $\beta = 0.01$, $\nu_O = 0.0001$, $k_{oxy} = 0.3$, and for stability simulation, we set ω to 0.2 in Equation (11). This value is well known in the LBM community for stability [44].

6.1. LBM Solver vs. NS Solver

In order to compare the results between the LBM solver and the NS solver, we performed a 2D smoke-rising test. In an NS simulation, the advection was computed using the BFECC method. In LBM simulations, the porosity was set to be 0.9 and permeability coefficient changes were from 1000 to 0.001. Figure 4 shows the results. The NS solver produces strong vorticity formation due to the incompressibility of fluid and BFECC (Back and Forth Error Compensation and Correction) advection. In contrast, the LBM solver shows weak vorticity and a rather slow advection of fluid because the LBM solver actually solves a weakly compressible version of the NS solver and the fluid motions are hindered by the porous medium. It is also seen that, as permeability decreases, the fluid motions became slower and more viscous. All the simulations were carried out using a uniform 512×1024 grid. The LBM simulation took about 0.35 s to advance a single time-step while the NS simulation took about 0.6 s.

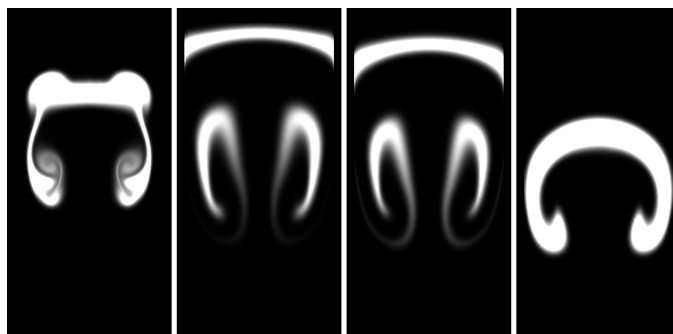


Figure 4. The leftmost image is flow obtained by the NS solver that shows strong vorticity formation. The rest of the three images was done by porous LBM with a uniform porosity of 0.9 and a permeability coefficient of 1000, 1 and 0.001, respectively.

6.2. Porosity Comparison between LBM and NS Solver

In order to prove the effectiveness of the LBM method in a porosity medium over NS method, we simulate burning wood with each method. Figure 5 shows the results. With the same parameter, the propagation of the NS solver is faster, like burning a sponge piece. Because the LBM is a microscopic scale viewpoint using fictive particles, the wood is gradually decomposed and heat release is maintained constantly. In the NS method, a piece of wood is burnt in 10 s, and the process takes place in 10 s. On the other hand, the charcoal process proceeds slowly for 80 s after burning for 40 s in the LBM method. Especially, it is possible to express realistic combustion by keeping the heat longer in the charcoal process with LBM.

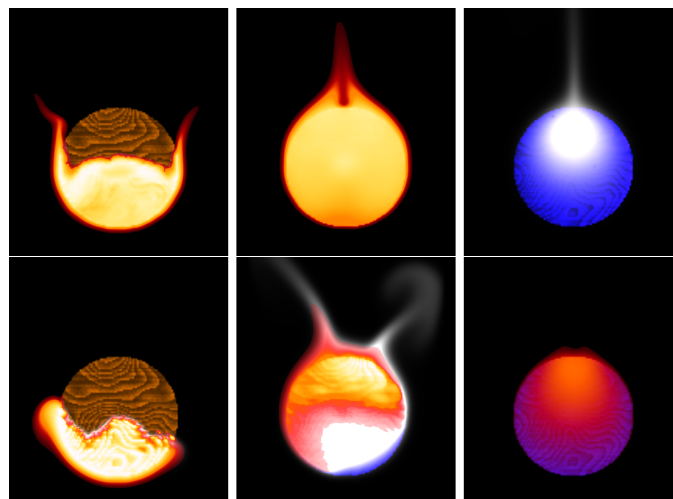


Figure 5. NS and LBM comparison experiment. Upper images are 3 s, 12 s and 30 s using NS solver. Lower images are 15 s, 40 s and 80 s using LBM solver.

6.3. Pyrolysis: Wood into Charcoal

We test pyrolysis simulation by supplying a low temperature heat source continuously. Then the wood is decomposed into charcoal and flammable gas. If the temperature is low enough, the flammable gas is not ignited, which is the same as for low oxygen. This process converts wood into charcoal. Since gas fuel (shown in green color in Figure 6), which is composed of methane and hydrogen, is lighter than air and also continuously heated, the hot gas fuel rises through the air or seeps into the wood. Meanwhile, as the temperature propagates inside the solid, wood is turned into charcoal gradually. Interestingly, charring on the front can be observed though we did not adopt any boundary capturing method such as a level-set. Since the temperature is kept low enough in all frames, gas fuel does not start the combustion process and charcoal smoldering is minimal. As shown in Figure 6 and video in the supplemental materials, pyrolysis without combustion is a very slow process and contains a small-scale thermochemical reaction occurring inside the solid, compared to fast and macroscopic air movements. Our hybrid solver enables us to simulate these phenomena that have different time-and-space scales in a single framework.

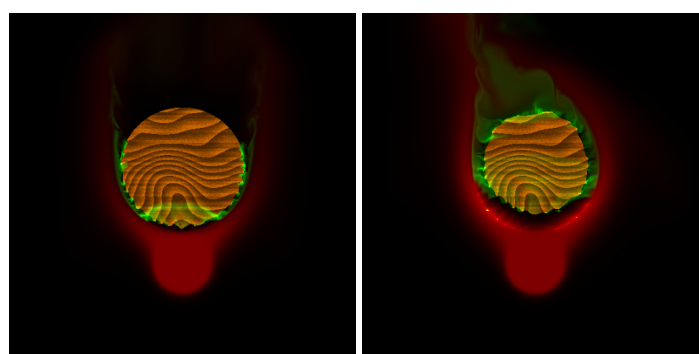


Figure 6. Pyrolysis experiment. Solid fuel is burned by heat underneath and produces charcoal shown in black and gas fuel shown in green. On the right image, burnt wood, on the left, charcoal. However, the temperature is low enough so that gas fuel does not start the combustion process and charcoal smoldering is minimal. Note that the LBM domain includes charcoal in black color including heated in red near the heat source.

6.4. Burning with Oxygen

Oxygen is an essential element that influences the overall combustion process including burning rate, pattern and shapes of flame. To point out the importance of oxygen in the simulation of the

combustion process, we designed an experiment of burning a solid ball with and without the oxygen model in Figure 7 and videos in the supplemental materials. Due to the proposed oxygen model, combustion of solids occurred only where oxygen is percolated (upper row in Figure 7). It is also seen that flames are created in regions where fuels meet oxygen with sufficient heat. In contrast, without considering oxygen, the wood just burns from down to up gradually (lower row in Figure 7).

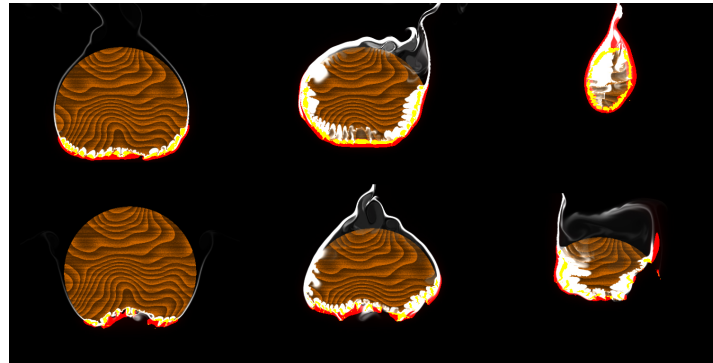


Figure 7. Upper row is burning with oxygen, where we can observe that the oxygen rich region burns first, while in the lower row where oxygen is not considered, burning is less natural and interesting.

6.5. Wood Burning with Permeability

It is well known that there is a strong correlation between burning rate and permeability. Hardwoods with low permeability such as oak or maple create long lasting fires but are hard to ignite. In contrast, softwoods with high permeability such as evergreen tree or conifers catch on fire easily and burn fast. To simulate this, we performed a permeability test (Figures 8 and 9 and videos in the supplemental materials). Compared to hardwood with a low permeability of 0.001, softwood with a high permeability of 1000 has a high tendency to catch on fire since oxygen is easily percolated into the wood. In both experiments, we can observe that the proposed hybrid boundary handling enables flow exchanging flows towards inside and outside the wood, and the proposed porous LBM method generates natural flow inside the wood.

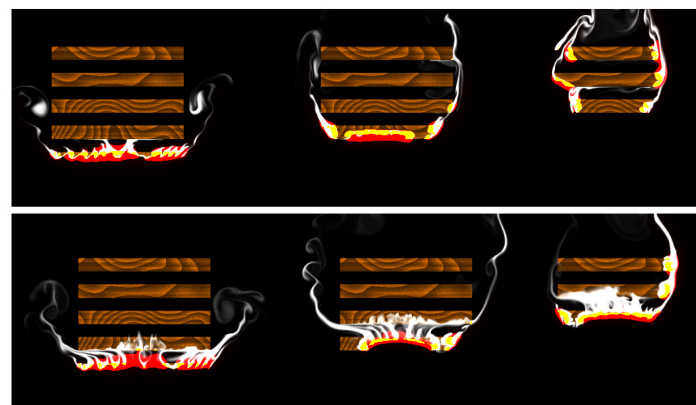


Figure 8. Upper row has low permeability, where flow inside wood is minimal. In the lower row where permeability high, flow occurs naturally across the two simulation domains. We can observe that the proposed hybrid boundary handling enables smoke exchanging flows towards inside and outside woods, and the proposed porous LBM method generates natural flow inside the wood.

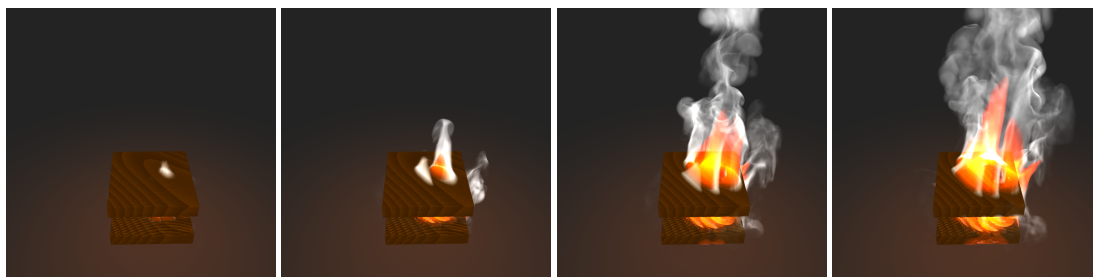


Figure 9. 3D permeability test. Smoke passes through wood grain.

7. Conclusions and Future Work

We proposed a new method to simulate the solid combustion process using a Lattice-Boltzmann and Eulerian hybrid. We show that the NS solver can serve for air regions where the macroscopic behavior is dominant, and LBM can serve for solid regions where microscopic flow along the uneven porosity is dominant. We designed a transitioning physical quantity between NSE and LBM like smoke, temperature and oxygen and demonstrated that natural flow occurs between the two domains. For combustion simulation, we incorporated flammable gas and charcoal generation so that the pyrolysis process can be simulated. By simulation with oxygen, we demonstrate the improvement of combustion realism. We derive the Lattice-Boltzmann formulation of the porous Navier-Stokes equation and translate the solid porosity into permeability. We demonstrated flows of heat, smoke and gas fuel inside porous solid fuel.

Combustion contains complex phenomena and a large portion is still left unaddressed at various accuracy levels. When heat decomposes solid fuel into flammable gas and charcoal, the solid is deformed, broken and collapsed. We ignore these rigid body simulations. More studies on this are needed in the future. Our simulation system can perform smoldering; however, the formation of the unique wood charcoal details, sometimes called allegation, where the realistic ash formation and its fracture are left for future work. Also, modeling the complex geometry involved will be considered. This paper only discusses burning solid simulation without rendering techniques. Research on apposite rendering for charcoal, combustion and solid and gas fuel will help to improve the look of realistic burning solids.

Supplementary Materials: The following are available online at <http://www.mdpi.com/2073-8994/11/11/1405/s1>.

Author Contributions: Conceptualization, E.J. and B.K.; Data curation, E.J.; Formal analysis, E.J.; Funding acquisition, B.K. and O.-Y.S.; Investigation, E.J., B.K. and O.-Y.S.; Methodology, E.J., B.K. and O.-Y.S.; Project administration, O.-Y.S.; Software, E.J.; Supervision, O.-Y.S.; Visualization, E.J.; Writing-original draft, E.J., B.K. and O.-Y.S.; Writing-review and editing, E.J., B.K. and O.-Y.S.

Funding: This research was supported by the MSIT (Ministry of Science and ICT), Korea, under the ITRC (Information Technology Research Center) support program (IITP-2019-2016-0-00312) supervised by the IITP (Institute for Information & communications Technology Planning & Evaluation), the faculty research fund of Sejong University in 2019, and the MSIP (Ministry of Science, ICT & Future Planning), Korea, under the National Program for Excellence in SW (2015-0-00938) supervised by the IITP (Institute for Information & communications Technology Planning & Evaluation).

Conflicts of Interest: The authors declare no conflict of interest.

Appendix A

In this appendix, we derive the macroscopic equations from the porous LBM model. The D2Q9 model is taken as an example. We start with the discrete Lattice Boltzmann Equation (LBE) which combines streaming step (Equation (4)) and collision step (Equation (7)).

$$f_i(\vec{x} + \vec{e}_i \Delta t, t + \Delta t) = f_i(\vec{x}, t) + (f_i^{eq} - f_i(\vec{x}, t)) / \tau. \quad (\text{A1})$$

Note that $\tau = 1/\omega$. Next, we do a 2nd-order Taylor series expansion about the LHS of LBE. The zero derivative term and the first term on the RHS will cancel, leaving only the first and second derivative terms of the Taylor expansion and the last term on RHS [45].

$$\begin{aligned} \frac{\partial f_i}{\partial t} + \vec{e}_i \cdot \nabla f_i + \left(\frac{1}{2} \vec{e}_i \vec{e}_i : \nabla \nabla f_i + \vec{e}_i \cdot \nabla \frac{\partial f_i}{\partial t} + \frac{1}{2} \frac{\partial^2 f_i}{\partial t^2} \right) \\ = (f_i^{eq} - f_i) / \tau, \end{aligned} \quad (\text{A2})$$

where $f_i = f_i(\vec{x}, t)$ for simplicity and “:” is the double dot product between dyads. The macro-dynamical behavior arising from LBE can be found from the Chapman-Enskog multiscale analysis using an expansion parameter K . To do so, the following expansions are introduced:

$$f_i = f_i^{(0)} + K f_i^{(1)} + K^2 f_i^{(2)} + \dots, \quad (\text{A3})$$

$$\frac{\partial}{\partial t} = K \frac{\partial}{\partial t_1} + K^2 \frac{\partial}{\partial t_2}, \quad \nabla = K \nabla_1, \quad (\text{A4})$$

where t_1 is a convective time-scale much bigger than t_2 (diffusive time-scale).

Applying the above expansion to the Equation (A2) and separating into the consecutive order of K , the following equations are derived:

$$O(K^0) : f_i^{(0)} = f_i^{eq}, \quad (\text{A5})$$

$$O(K^1) : D_{1i} f_i^{(0)} = -\frac{f_i^{(1)}}{\tau}, \quad (\text{A6})$$

$$O(K^2) : \frac{\partial f_i^{(0)}}{\partial t_2} + (1 - 1/2\tau) D_{1i} f_i^{(1)} = -\frac{f_i^{(2)}}{\tau}, \quad (\text{A7})$$

where $D_{1i} = \frac{\partial}{\partial t} + \vec{e}_i \cdot \nabla_1$. Note that from Equation (6) and Equations ($\rho = \sum f_i^{eq}$, $\vec{u} = 1/\rho \sum \vec{e}_i f_i^{eq}$) obtained with definition of the equilibrium DF in Equation (19), we can obtain that

$$\sum f_i^{(k)} = 0, \quad \sum \vec{e}_i f_i^{(k)} = 0; k \geq 1. \quad (\text{A8})$$

Applying the relations between DFs and the macroscopic properties from above, the mass and momentum equations are achieved:

$$\frac{\partial \rho}{\partial t} + \nabla \cdot \rho \vec{u} = 0, \quad (\text{A9})$$

$$\frac{\partial \rho \vec{u}}{\partial t} + \nabla \cdot \Pi = \vec{0}, \quad (\text{A10})$$

where momentum flux tensor Π has the following form:

$$\Pi = \sum \vec{e}_{ix} \vec{e}_{iy} [f_i^{eq} + (1 - 1/2\tau) f_i^{(1)}]. \quad (\text{A11})$$

Inserting the equilibrium distribution (Equation (19)) back into the flux tensor leads to:

$$\Pi_{xy}^{(0)} = \sum \vec{e}_{ix} \vec{e}_{iy} f_i^{eq} = \rho / 3 \delta_{xy} + \rho \vec{u}_x \vec{u}_y / \epsilon, \quad (\text{A12})$$

$$\begin{aligned} \Pi_{xy}^{(1)} &= (1 - 1/2\tau) \sum \vec{e}_{ix} \vec{e}_{iy} f_i^{(1)} \\ &= -\rho v_u (\nabla_{1x} \vec{u}_y + \nabla_{1y} \vec{u}_x), \end{aligned} \quad (\text{A13})$$

where $\nu_u = \frac{(\tau-1/2)}{3}\Delta t$. Finally, the macroscopic equation is recovered under the assumption that density variation is small:

$$\frac{\partial \rho \vec{u}}{\partial t} + \nabla \cdot \frac{\rho \vec{u} \vec{u}}{\epsilon} = -\nabla(\epsilon p) + \rho \nu_u (\nabla \vec{u} + \vec{u} \nabla), \quad (\text{A14})$$

where $p = \rho/3\epsilon$. If ρ is constant, Equations (A9) and (A14) reduces to Equations (13) and (14) without the external force.

In order to account for the external force, the LBE should be augmented by adding a microscopic forcing term F_i .

$$f_i(\vec{x} + \vec{e}_i \Delta t, t + \Delta t) = f_i(\vec{x}, t) + (f_i^{eq} - f_i(\vec{x}, t))/\tau + F_i. \quad (\text{A15})$$

The forcing term can be expanded up to the first order in the lattice velocity [46,47] as

$$F_i = 3\Delta t w_i \rho (1 - 1/2\tau) \vec{e}_i \cdot \vec{F}, \quad (\text{A16})$$

Accordingly, the velocity is redefined as

$$\vec{u} = 1/\rho \sum \vec{e}_i f_i^{eq} + 1/2\vec{F}. \quad (\text{A17})$$

Inserting \vec{F} in Equation (15) into Equation (A17), the velocity \vec{u} can be given explicitly by Equation (20) with Equations (21)–(23).

Note that drag effects due to porous medium can be reduced because we ignored the second- and the higher- order term in Equation (A16) and in Equation (10). To compensate the effects, we adopt the bounce-back scheme during the streaming step [28] (Equations (17) and (18)).

References

- Reeves, W.T. Particle systems—A technique for modeling a class of fuzzy objects. *ACM Trans. Graph. TOG* **1983**, *2*, 91–108. [\[CrossRef\]](#)
- Stam, J.; Fiume, E. Depicting fire and other gaseous phenomena using diffusion processes. In Proceedings of the 22nd Annual Conference on Computer Graphics and Interactive Techniques, Los Angeles, CA, USA, 6–11 August 1995; ACM: New York, NY, USA, 1995; pp. 129–136.
- Nishita, T.; Dobashi, Y. Modeling and rendering of various natural phenomena consisting of particles. In Proceedings of the Computer Graphics International 2001, Hong Kong, China, 6 July 2001; pp. 149–156.
- Lamorlette, A.; Foster, N. Structural modeling of flames for a production environment. In Proceedings of the ACM Transactions on Graphics (TOG), San Antonio, TX, USA, 23–26 July 2002; ACM: New York, NY, USA, 2002; Volume 21, pp. 729–735.
- Stam, J. Stable fluids. In Proceedings of the 26th Annual Conference on Computer Graphics and Interactive Techniques, Los Angeles, CA, USA, 8–13 August 1999; ACM Press/Addison-Wesley Publishing Co.: New York, NY, USA, 1999; pp. 121–128.
- Zhu, B.; Lee, M.; Quigley, E.; Fedkiw, R. Codimensional non-Newtonian fluids. *ACM Trans. Graph. TOG* **2015**, *34*, 115. [\[CrossRef\]](#)
- Wang, X.; Ban, X.; He, R.; Wu, D.; Liu, X.; Xu, Y. Fluid-Solid Boundary Handling Using Pairwise Interaction Model for Non-Newtonian Fluid. *Symmetry* **2018**, *10*, 94. [\[CrossRef\]](#)
- Huang, L.; Hädrich, T.; Michels, D.L. On the Accurate Large-scale Simulation of Ferrofluids. *ACM Trans. Graph. TOG* **2019**, *38*, 93. [\[CrossRef\]](#)
- Foster, N.; Fedkiw, R. Practical animation of liquids. In Proceedings of the 28th Annual Conference on Computer Graphics and Interactive Techniques, Los Angeles, CA, USA, 12–17 August 2001; ACM: New York, NY, USA, 2001; pp. 23–30.
- Müller, M.; Charypar, D.; Gross, M. Particle-based fluid simulation for interactive applications. In Proceedings of the 2003 ACM SIGGRAPH/Eurographics Symposium on Computer Animation, San Diego, CA, USA, 26–27 July 2003; Eurographics Association: Aire-la-Ville, Switzerland, 2003; pp. 154–159.
- Kim, D.; Song, O.y.; Ko, H.S. Stretching and wiggling liquids. *ACM Trans. Graph. TOG* **2009**, *28*, 120.

12. Lenaerts, T.; Adams, B.; Dutré, P. Porous flow in particle-based fluid simulations. *ACM Trans. Graph. TOG* **2008**, *27*, 49.
13. Pfaff, T.; Thuerey, N.; Gross, M. Lagrangian vortex sheets for animating fluids. *ACM Trans. Graph. TOG* **2012**, *31*, 112. [[CrossRef](#)]
14. Kim, D.; Lee, S.W.; Song, O.Y.; Ko, H.S. Baroclinic turbulence with varying density and temperature. *IEEE Trans. Vis. Comput. Graph.* **2011**, *18*, 1488–1495.
15. Yngve, G.D.; O'Brien, J.F.; Hodgins, J.K. Animating explosions. In Proceedings of the 27th Annual Conference on Computer Graphics and Interactive Techniques, New Orleans, LA, USA, 23–28 July 2000; ACM Press/Addison-Wesley Publishing Co.: New York, NY, USA, 2000; pp. 29–36.
16. Nguyen, D.Q.; Fedkiw, R.; Jensen, H.W. Physically based modeling and animation of fire. *ACM Trans. Graph. TOG* **2002**, *21*, 721–728.
17. Selle, A.; Rasmussen, N.; Fedkiw, R. A vortex particle method for smoke, water and explosions. In Proceedings of the ACM Transactions on Graphics (TOG), Los Angeles, CA, USA, 31 July–4 August 2005; ACM: New York, NY, USA, 2005; Volume 24, pp. 910–914.
18. Hong, J.M.; Shinar, T.; Fedkiw, R. Wrinkled flames and cellular patterns. In Proceedings of the ACM Transactions on Graphics (TOG), San Diego, CA, USA, 5–9 August 2007; ACM: New York, NY, USA, 2007; Volume 26, p. 47.
19. Horvath, C.; Geiger, W. Directable, high-resolution simulation of fire on the GPU. In Proceedings of the ACM Transactions on Graphics (TOG), New Orleans, LA, USA, 3–9 August 2009; ACM: New York, NY, USA, 2009; Volume 28, p. 41.
20. Son, M.; Kim, B.; Wilensky, G.; Lee, S. Still-Frame Simulation for Fire Effects of Images. In *Computer Graphics Forum*; Wiley Online Library: Hoboken, NJ, USA, 2013; Volume 32, pp. 295–304.
21. Fedkiw, R.; Stam, J.; Jensen, H.W. Visual simulation of smoke. In Proceedings of the 28th Annual Conference on Computer Graphics and Interactive Techniques, Los Angeles, CA, USA, 12–17 August 2001; ACM: New York, NY, USA, 2001; pp. 15–22.
22. Melek, Z.; Keyser, J. Interactive simulation of fire. In Proceedings of the 10th Pacific Conference on Computer Graphics and Applications, Beijing, China, 9–11 October 2002; pp. 431–432.
23. Melek, Z.; Keyser, J. *Interactive Simulation of Fire*; Technical Report 2002-7-1; Texas A&M Department of Computer Science: College Station, TX, USA, 2002.
24. Yu, D.; Mei, R.; Luo, L.S.; Shyy, W. Viscous flow computations with the method of lattice Boltzmann equation. *Prog. Aerosp. Sci.* **2003**, *39*, 329–367. [[CrossRef](#)]
25. Losasso, F.; Irving, G.; Guendelman, E.; Fedkiw, R. Melting and burning solids into liquids and gases. *IEEE Trans. Vis. Comput. Graph.* **2006**, *12*, 343–352. [[CrossRef](#)] [[PubMed](#)]
26. Melek, Z.; Keyser, J. Driving object deformations from internal physical processes. In Proceedings of the 2007 ACM Symposium on Solid and Physical Modeling, Beijing, China, 4–6 June 2007; ACM: New York, NY, USA, 2007; pp. 51–59.
27. Pirk, S.; Jarzabek, M.; Hädrich, T.; Michels, D.L.; Palubicki, W. Interactive wood combustion for botanical tree models. *ACM Trans. Graph. TOG* **2017**, *36*, 197. [[CrossRef](#)]
28. Succi, S. *The Lattice Boltzmann Equation: For Fluid Dynamics and Beyond*; Oxford University Press: Oxford, UK, 2001.
29. Echevarria, J.I.; Wilensky, G.; Krishnaswamy, A.; Kim, B.; Gutierrez, D. Computational simulation of alternative photographic processes. In *Computer Graphics Forum*; Wiley Online Library: Hoboken, NJ, USA, 2013; Volume 32, pp. 7–16.
30. Chu, N.S.H.; Tai, C.L. MoXi: Real-time ink dispersion in absorbent paper. *ACM Trans. Graph. TOG* **2005**, *24*, 504–511. [[CrossRef](#)]
31. Aaltosalmi, U. *Fluid Flow in Porous Media with the Lattice-Boltzmann Method*; University of Jyväskylä: Jyväskylän yliopisto, Finland, 2005.
32. Boek, E.S.; Venturoli, M. Lattice-Boltzmann studies of fluid flow in porous media with realistic rock geometries. *Comput. Math. Appl.* **2010**, *59*, 2305–2314. [[CrossRef](#)]
33. Grucelski, A.; Pozorski, J. Lattice Boltzmann simulation of fluid flow in porous media of temperature-affected geometry. *J. Theor. Appl. Mech.* **2012**, *50*, 193–214.
34. Wei, X.; Li, W.; Mueller, K.; Kaufman, A.E. The Lattice-Boltzmann method for simulating gaseous phenomena. *IEEE Trans. Vis. Comput. Graph.* **2004**, *10*, 164–176.

35. Thürey, N.; Iglberger, K.; Rüde, U. Free surface flows with moving and deforming objects for LBM. In Proceedings of the Vision, Modeling and Visualization, Aachen, Germany, 22–24 November 2006; Volume 2006, pp. 193–200.
36. Thürey, N.; Rüde, U. Stable free surface flows with the lattice Boltzmann method on adaptively coarsened grids. *Comput. Vis. Sci.* **2009**, *12*, 247–263. [[CrossRef](#)]
37. Hirt, C.W.; Nichols, B.D. Volume of fluid (VOF) method for the dynamics of free boundaries. *J. Comput. Phys.* **1981**, *39*, 201–225. [[CrossRef](#)]
38. Zhu, H.; Liu, X.; Liu, Y.; Wu, E. Simulation of miscible binary mixtures based on lattice Boltzmann method. *Comput. Animat. Virtual Worlds* **2006**, *17*, 403–410. [[CrossRef](#)]
39. Zhu, H.; Bao, K.; Wu, E.; Liu, X. Stable and efficient miscible liquid-liquid interactions. In Proceedings of the 2007 ACM Symposium on Virtual Reality Software and Technology, Newport Beach, CA, USA, 5–7 November 2007; ACM: New York, NY, USA, 2007; pp. 55–64.
40. Dellar, P.J. Incompressible limits of lattice Boltzmann equations using multiple relaxation times. *J. Comput. Phys.* **2003**, *190*, 351–370. [[CrossRef](#)]
41. Chen, S.; Doolen, G.D. Lattice Boltzmann method for fluid flows. *Annu. Rev. Fluid Mech.* **1998**, *30*, 329–364. [[CrossRef](#)]
42. Nithiarasu, P.; Seetharamu, K.; Sundararajan, T. Natural convective heat transfer in a fluid saturated variable porosity medium. *Int. J. Heat Mass Transf.* **1997**, *40*, 3955–3967. [[CrossRef](#)]
43. Mavko, G.; Nur, A. The effect of a percolation threshold in the Kozeny-Carman relation. *Geophysics* **1997**, *62*, 1480–1482. [[CrossRef](#)]
44. Thürey, N. *Physically Based Animation of Free Surface Flows with the Lattice Boltzmann Method: Physikalische Animation Von Strömungen Mit Freien Oberflächen Mit Der Lattice-Boltzmann-Methode*; Verlag Dr. Hut: München, Germany, 2007.
45. Lattice Boltzmann Methods. Available online: https://en.wikipedia.org/wiki/Lattice_Boltzmann_methods (accessed on 13 November 2019).
46. Ladd, A.; Verberg, R. Lattice-Boltzmann simulations of particle-fluid suspensions. *J. Stat. Phys.* **2001**, *104*, 1191–1251. [[CrossRef](#)]
47. Guo, Z.; Zheng, C.; Shi, B. Discrete lattice effects on the forcing term in the lattice Boltzmann method. *Phys. Rev. E* **2002**, *65*, 046308. [[CrossRef](#)] [[PubMed](#)]



© 2019 by the authors. Licensee MDPI, Basel, Switzerland. This article is an open access article distributed under the terms and conditions of the Creative Commons Attribution (CC BY) license (<http://creativecommons.org/licenses/by/4.0/>).