Available online at www.sciencedirect.com
ScienceDirect
Computers and Mathematics with Applications 55 (2008) 1461–1468
An International Journal
computers & mathematics
with applications
www.elsevier.com/locate/camwa

Simulation of moving particles in 3D with the Lattice Boltzmann method

Klaus Iglberger*, Nils Thürey, Ulrich Rüde

Chair for System Simulation, Friedrich-Alexander University Erlangen/Germany, Cauerstr. 6, 91058 Erlangen, Germany

Abstract

The following paper presents a way to simulate the behavior of particle agglomerates in a fluid flow by coupling the Lattice Boltzmann Method to a rigid body physics engine. By extending the basic algorithm by a fluid/particle force interaction method, the hydrodynamic forces acting on the particles can be calculated. By the use of this force interaction between the fluid and the particles and by the use of the rigid body physics engine, the movement and collision behavior of particles in a flow can be simulated. Additionally, this coupled simulation system is able to simulate the internal particle forces in the connections between sintered particles, which could break due to the forces and torques of a shear flow. This permits a prediction of possible break-ups or structural displacements.

Keywords: Lattice Boltzmann method; Curved boundaries; Force interaction; Rigid body physics; Particles; Agglomerates

1. Introduction

In the field of particle technology, knowledge about the behavior of complex particles in a flow is essential to predict the properties of the final products containing these particles. However, the behavior of the particles is dependent on their structure. Due to the complex structure of these particles, the analytical examination is limited to only very simple structures. Numerical simulation is one way to efficiently examine the structure and behavior of the particles and therefore create products faster and at lower cost.

This paper proposes to simulate particles and particle agglomerates in a flow by coupling a rigid body physics engine to a Lattice Boltzmann fluid simulation. This new approach is to our current knowledge the first attempt to simulate moving particle agglomerates in a flow that also offers the possibility to simulate arbitrarily complex agglomerates and to calculate contact forces and torques. These can be used to simulate break-ups or structural displacements within the agglomerates. We achieve this by combining several extensions for the fluid simulation like the treatment of moving curved boundaries with the scheme of Yu et al. [1] and a fluid/particle force interaction method with the momentum exchange method of Ladd [2]. An overview of these techniques can be found in [3]. The simulations in Section 5 will show that our approach yields stable and accurate results of arbitrarily complex agglomerates.

* Corresponding author.
E-mail addresses: Klaus.Iglberger@informatik.uni-erlangen.de (K. Iglberger), Ulrich.Ruede@informatik.uni-erlangen.de (U. Rüde).

0898-1221/$ - see front matter © 2007 Elsevier Ltd. All rights reserved.
doi:10.1016/j.camwa.2007.08.022

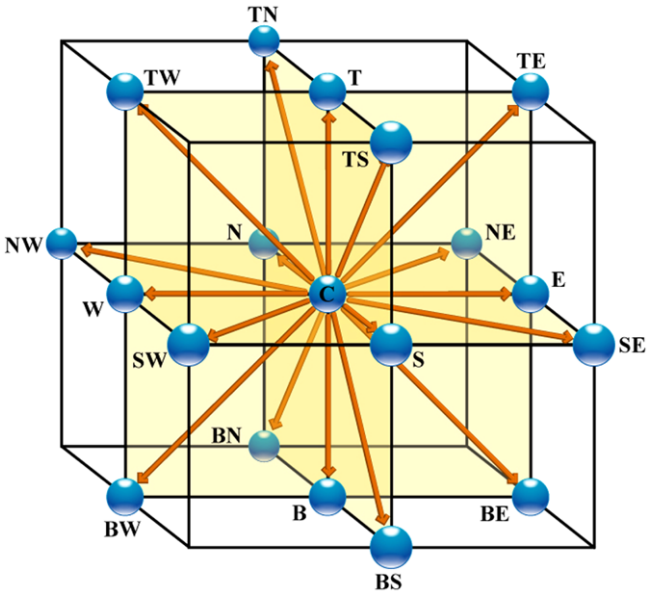

Fig. 1. The D3Q19 model for the 3D LBM.

2. The Lattice Boltzmann method

In contrast to the classical macroscopic Navier–Stokes (NS) approach to simulate fluids, the Lattice Boltzmann Method (LBM) uses a mesoscopic simulation model [4]. Instead of directly solving the macroscopic fluid quantities, such as velocity and pressure, the movement of fluid particles is modeled. The fluid domain is discretized in uniform Cartesian cells. Each cell holds a fixed number of distribution functions, which represent the number of fluid particles moving in these discrete directions. For this work, the most popular model for the 3D case, the D3Q19 model, which consists of 19 distribution functions, has been used. This model is illustrated in Fig. 1.

The distribution functions are calculated by solving the Lattice Boltzmann equation (LBE), which is a special discretization of the kinetic Boltzmann equation. Based on the Bhatnagar–Gross–Krook (BGK) model, the update of the distribution functions can be formulated as in Eq. (1):

$$f_\alpha(\mathbf{x}_i + \mathbf{e}_\alpha \delta t, t + \delta t) - f_\alpha(\mathbf{x}_i, t) = -\frac{1}{\tau}[f_\alpha(\mathbf{x}_i, t) - f_\alpha^{(\mathrm{eq})}(\mathbf{x}_i, t)], \tag{1}$$

where δt denotes the lattice time step, $\delta \mathbf{x} = \mathbf{e}_\alpha \delta t$ denotes the lattice cell size, τ denotes the lattice relaxation time, $\mathbf{e}_\alpha$ is the discrete lattice velocity in direction α, and $f_\alpha^{(\mathrm{eq})}$ is the equilibrium distribution (see Eq. (4)). Eq. (1) is usually solved in two steps:

$$\tilde{f}_\alpha(\mathbf{x}_i, t + \delta t) = f_\alpha(\mathbf{x}_i, t) - \frac{1}{\tau}[f_\alpha(\mathbf{x}_i, t) - f_\alpha^{(\mathrm{eq})}(\mathbf{x}_i, t)] \tag{2}$$

$$f_\alpha(\mathbf{x}_i + \mathbf{e}_\alpha \delta t, t + \delta t) = \tilde{f}_\alpha(\mathbf{x}_i, t + \delta t). \tag{3}$$

Eq. (2) is called the collide step. This step models various fluid particle interactions like collisions and calculates new distribution functions according to the distribution functions of the last time step. It also models the equilibrium distribution functions, which are calculated with Eq. (4).

$$f_\alpha^{(\mathrm{eq})} = w_\alpha \cdot \rho \cdot \left[1 + \frac{3}{c^2}\mathbf{e}_\alpha \cdot \mathbf{u} + \frac{9}{2c^4}(\mathbf{e}_\alpha \cdot \mathbf{u})^2 - \frac{3}{2c^2}\mathbf{u} \cdot \mathbf{u}\right]. \tag{4}$$

In this equation, w_α is a weighting factor depending on the LBM model used, ρ is the lattice fluid density, and $\mathbf{u}$ is the lattice fluid velocity.

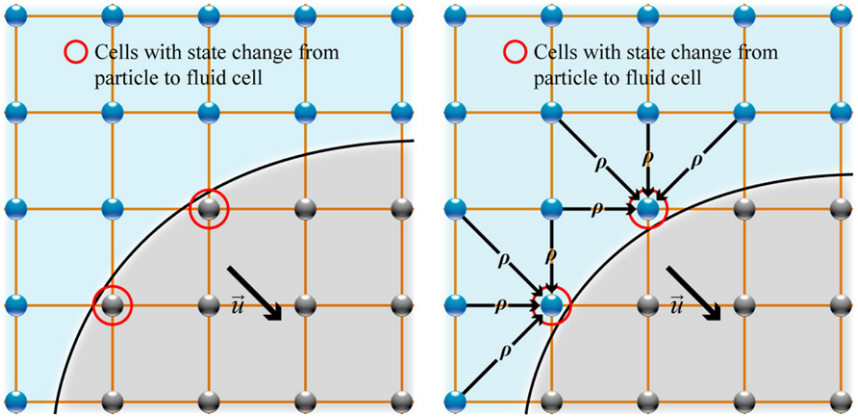

Fig. 2. Flag state change from particle to fluid cell due to particle or agglomerate movement.

Eq. (3) is called the streaming step. In this step, fluid particles are streamed from one cell to a neighboring cell according to the velocity of the fluid particles in the original cell. This streaming operation can either be performed as a pushing operation from one cell to the surrounding cells, or as a pulling operation into one cell from the surrounding cells.

3. Particles and agglomerates

This work focuses on the simulation of particles and agglomerates of a size between 1 and 100 microns. Particles are an unbreakable compound of atoms and molecules and are of arbitrary shape and mass. The connection to other particles results in agglomerates, which are of arbitrary and often of very complex structure, but breakable at certain contact points. In this work, the structure of a single particle is approximated by a sphere. Although this is an abstraction from the real world structure, experiments show that the particle behavior in a flow resembles the behavior of a sphere. Therefore, agglomerates consist of several spheres, which are connected by point contacts or contact areas for overlapping particles. For contact points of interest, the contact forces and torques are calculated in order to predict break-ups and structural displacements.

4. Moving particles with the Lattice Boltzmann simulation

In order to simulate moving agglomerates in a flow, the LBM, which simulates the fluid behavior, has to be coupled to a rigid body physics engine, which simulates the correct rigid body movements of agglomerates. Due to this two-system simulation, the basic LBM algorithm has to be extended.

The first extension handles the dynamic character of the flag values due to the movement of the particle agglomerates. These flag values have to be recalculated in each time step due to the movement of the particles and agglomerates. Cells may change from fluid cell to particle cell if they are inside one of the spherical particles, but they may also change from particle cell to fluid cell. In this case, new distribution functions have to be created for the new fluid cell. This situation is illustrated in Fig. 2, where two new fluid cells result due to the agglomerate movement to the lower right.

In the case of a flag change from particle to fluid cell, the missing distribution functions are calculated as equilibrium distribution functions using the local density, which is derived by a local density interpolation, and the surface velocity of the particle surface point nearest to the new fluid cell.

The second extension deals with the moving curved particle surfaces. The standard link, bounce-back, no-slip boundary condition always assumes that the wall is placed exactly between the fluid and the particle node. Due to the arbitrary position of the particles and the curved particle surface, the particle surface can intersect the link between two nodes at arbitrary distances. These distance values are called delta values:

$$\delta = \frac{\text{Distance between fluid node and particle surface}}{\text{Distance between fluid node and particle node}} \in (0, 1]. \tag{5}$$

```
1: RaySphereIntersect(o,d,c,r)
2: returns(delta)
3: l = c - o
4: s = l · d
5: l² = l · l
6: m² = l² - s²
7: q = √ r² - m²
8: t = s-q
9: return( t / |d| )
```

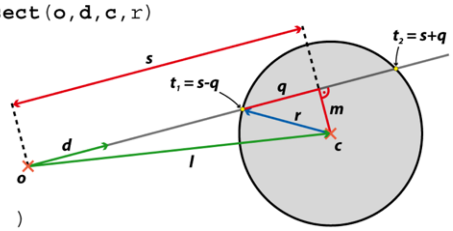

Fig. 3. Delta calculation for the curved particle surfaces.

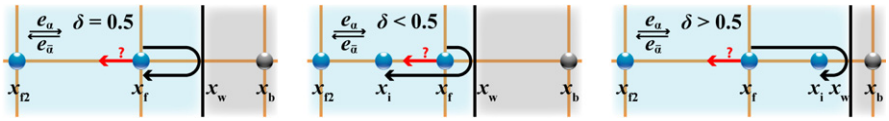

Fig. 4. Bounce back boundaries for different values of δ.

For each pair of neighboring fluid and particle nodes, one delta value has to be calculated. Delta values of zero are not possible due to the fact that a node on the surface is counted as a particle node. The algorithm for this calculation is taken from computer graphics [5]. It tests for an intersection between a ray in direction d and a sphere, and calculates the distance between the fluid cell and the particle surface. Fig. 3 presents the algorithm and an illustration for the calculation of a delta value for the case that a ray is shot from a fluid node towards the neighboring particle node.

Fig. 4 shows the three possible situations for delta values in the interval of (0, 1]. Since the fluid particles in the LBM are always considered to move one cell length per time step ($\delta\mathbf{x}/\delta t$), for delta values smaller than 0.5 and larger than 0.5, the fluid particles would come to rest at an intermediate node ($\mathbf{x}_i$). In order to calculate the reflected distribution function in node $\mathbf{x}_f$, an interpolation scheme has to be applied.

In [6] several interpolation schemes ranging from linear interpolations, which only take two fluid nodes into account, to quadratic interpolations, which use three fluid nodes, are compared. Based on this work, the linear interpolation scheme of Yu et al. was chosen [1,3]. This scheme uses two fluid nodes and the interpolation is represented by a single equation independent of whether the value of δ is less or larger than 0.5. With this scheme, the reflected distribution function is calculated as

$$f_{\tilde{\alpha}}(\mathbf{x}_f, t+\delta t) = \frac{1}{1+\delta}\cdot\Big[(1-\delta)\cdot f_{\alpha}(\mathbf{x}_f, t+\delta t) + \delta\cdot f_{\alpha}(\mathbf{x}_b, t+\delta t) + \delta\cdot f_{\tilde{\alpha}}(\mathbf{x}_{f2}, t+\delta t) - 2w_a\rho_w\frac{3}{c^2}\mathbf{e}_a\cdot\mathbf{u}_w\Big], \tag{6}$$

where w_α is the same weighting factor as in Eq. (4), ρ_w is the fluid density in node $\mathbf{x}_f$, and $\mathbf{u}_w$ is the velocity of the bounce-back wall. For the coupling between the fluid and the particle agglomerates, the LBM also has to be extended by a force interaction scheme. In order to calculate the fluid force acting on the particle surface, the momentum exchange method was used [2,3]. With this method, the total physical force acting on a particle agglomerate is calculated as the sum over all fluid/particle node pairs, resulting in

$$F = \sum_{\mathbf{x}_b}\sum_{\alpha=1}^{19}\mathbf{e}_\alpha[f_\alpha(\mathbf{x}_b, t) + f_{\tilde{\alpha}}(\mathbf{x}_f, t)]\delta\mathbf{x}/\delta t. \tag{7}$$

An example for the accuracy of this approach can be found in [7], where the drag force on several agglomerates was determined.

After the force calculations, one step in the coupled rigid body physics engine can be simulated in order to move the particle agglomerates according to the applied forces. Since no open source physics engine provides the capabilities to accurately calculate the resulting contact forces and torques, a special rigid body physics engine was implemented, which focuses on the simulation of complex particle agglomerates. This engine uses a second order Störmer–Verlet time integration for solving the equations of motion. For the contact force and torque calculation, the free body

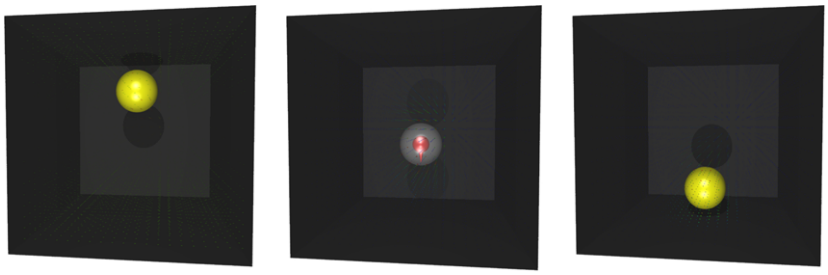

Fig. 5. Sinking particle for the comparison between analytical and simulation results.

diagram method is used: for each contact point, the rigid body is split into two sections, which are considered to be rigid bodies on their own. For both sections, the connected other section is replaced by a contact force and contact torque [8]. The contact forces $\mathbf{F}_c$, which might be used to simulate ruptures in the particle agglomerate structure, are calculated as

$$\mathbf{F}_c(t) = M_s \cdot \mathbf{a}_s(t) - \mathbf{F}_s(t), \tag{8}$$

where M_s is the total mass of one of the two sections of the connection, $\mathbf{a}_s$ is the acceleration of this section, and $\mathbf{F}_s$ is the total force acting on this section [8]. The contact torque $\mathbf{T}_c$, which might cause structural displacements within the particle agglomerate, or which also might cause ruptures, can be calculated as

$$\mathbf{T}_c(t) = \mathbf{I}_s(t) * \dot{\omega}(t) + \omega(t) \times \mathbf{I}_s(t)\omega(t) - \mathbf{d}_c \times \mathbf{F}_c(t) - \mathbf{T}_s(t), \tag{9}$$

where $\mathbf{I}_s$ is the moment of inertia, $\dot{\omega}$ is the angular acceleration, ω is the angular velocity, and $\mathbf{T}_s$ is the torque of one of the two sections. $\mathbf{d}_c$ is the distance between the center of this section and the center of mass of the particle agglomerate [8].

5. Results

One of the possible test scenarios for a comparison between analytical results and the moving particle simulation is to compare the sinking velocities of a heavy particle [9]. A particle with a density of $2.0\frac{\text{kg}}{\text{dm}^3}$ (lattice density of 2) is put in the upper half of a box completely filled with fluid of density $1.0\frac{\text{kg}}{\text{dm}^3}$ (lattice density of 1) and having no external stimulations (Fig. 5). The analytical sinking velocity for spherical bodies and Reynolds numbers smaller than 0.25 ($\text{Re} < 0.25$) can be calculated as

$$v = \frac{2gr^2 \cdot (\rho_P - \rho_F)}{9\eta}, \tag{10}$$

where g is the acceleration due to gravity which is $9.81\frac{\text{m}}{\text{s}^2}$, r is the particle radius, ρ_P is the particle density, ρ_F the fluid density and η is the dynamic fluid viscosity. The analytical solution calculated with Eq. (10) is $v = 5.45 \times 10^{-3}\frac{\text{m}}{\text{s}}$ for a particle with a physical radius $r = 5 \times 10^{-5}$ m and a lattice radius of $r = 5$. Fig. 6 shows the simulation results for the sinking velocity and the total force acting on the sinking particle for a simulation in a 60^3 domain with free-slip boundary conditions. Although slightly larger than the analytical result, the sinking velocity essentially agrees with the exact value. The difference can be explained by the periodic force fluctuations caused by the change in the number of fluid cells surrounding the particle due to the flag changes.

Based on the verification of test problems, which can be compared to analytical results, several test cases of complex particle agglomerates were investigated. One example is the simulation of a symmetric dual star particle agglomerate with 13 particles and 12 connections, which is rotating in a symmetric shear flow (see Fig. 7).

Due to the completely symmetric setup, the agglomerate is supposed to start a pure rotational movement around its center of mass in the center of the central sphere. Each particle has a radius of 4×10^{-4} m, which results in a maximum agglomerate length of 4×10^{-3} m. The shear flow results from a flow to the right in the upper half of the fluid domain and a flow to the left in the lower half. The value of interest in this simulation is the force in the two connections of the central particle. Due to the complexity of the simulation, no analytical solution for this force exists, so no comparisons

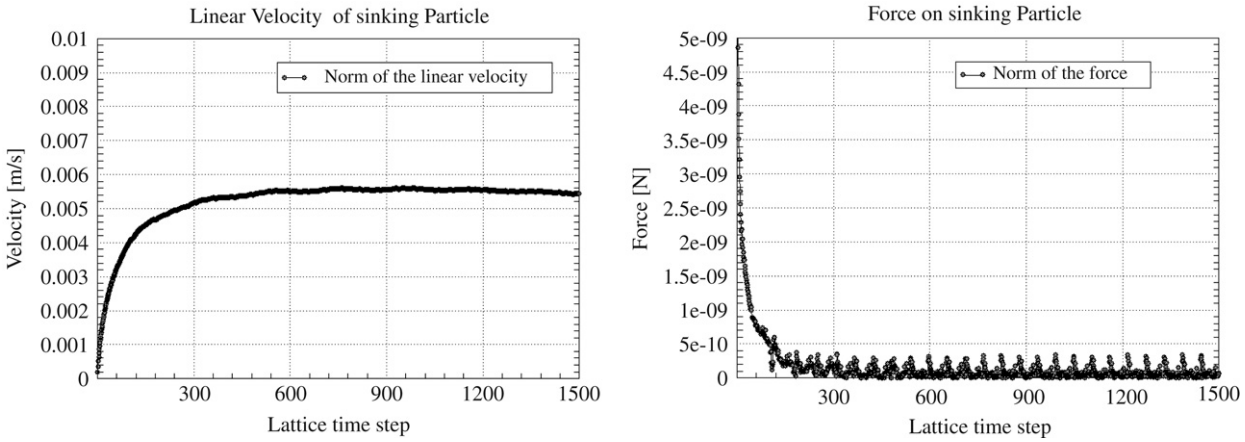

Fig. 6. Simulation results for the sinking particle simulation. The used size of the LBM simulation domain was 60^3 cells and the radius of the sphere was 5 cells.

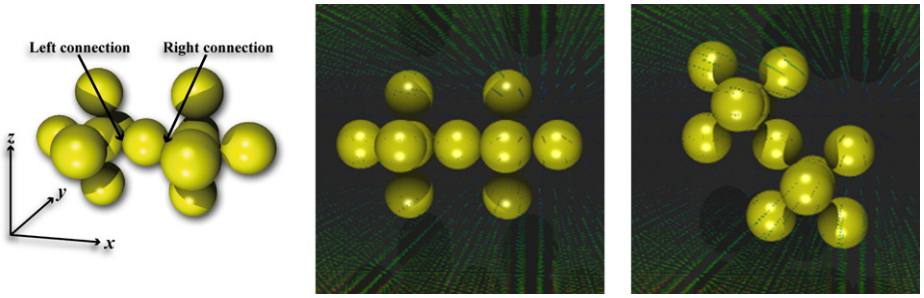

Fig. 7. Rotating dual star particle agglomerate.

between analytical results and the simulation can be made. However, we expect to see a sinusoidal shape of the forces in x- and z-directions and no force in y-direction, since the agglomerate is rotating around the y-axis. The forces in the connection to the right of the central sphere should be the inverse of the forces in the connection to the left of the central sphere. The norm of the force in both connections should result in the same value due to the symmetric setup of the simulation. For the first simulation run, a lattice cell length of $dx = 5 \times 10^{-5}$ m is used, which results in a lattice domain discretized to 120^3 lattice cells and a particle radius of 8 lattice cells. Fig. 8 shows the forces on both center connections.

The basic sinus oscillation of the x- and z-forces can be monitored, which results from the pure rotational movement of the dual star agglomerate due to its symmetric structure. It also becomes obvious that the force on the right connection is exactly the inverse of the force on the left connection due to the symmetric setup of this simulation. In both cases, a maximum norm of 7.00557×10^{-7} N acts on the connections. It is obvious that the gradient of the curve is not completely smooth and that several leaps in the force calculation occur. This again results from the changing number of fluid cells around the particle agglomerate.

Fig. 9 shows three examples for complex, asymmetric particle agglomerates. The left illustration shows an agglomerate with 16 spheres, the second illustration an agglomerate with 32 spheres, and the third illustration an agglomerate with 64 spheres.

All three simulations offer insight into the behavior of the particle agglomerates in a flow. However, when using several particles in the simulation, the memory requirements soon reach critical levels due to the growing number of particles and a fixed minimal particle radius that is necessary because of accuracy constraints. The last illustration in Fig. 9 shows a simulation of three agglomerates, which requires 640 MByte of memory. Additionally, the boundary influences should be minimized by a huge distance between the agglomerates and the boundaries, which only increases the memory problem. Therefore, for the simulation of more agglomerates, memory reduction techniques like grid refinement [10,3], grid compression [11] or the parallelization [12] of the simulation have to be applied.

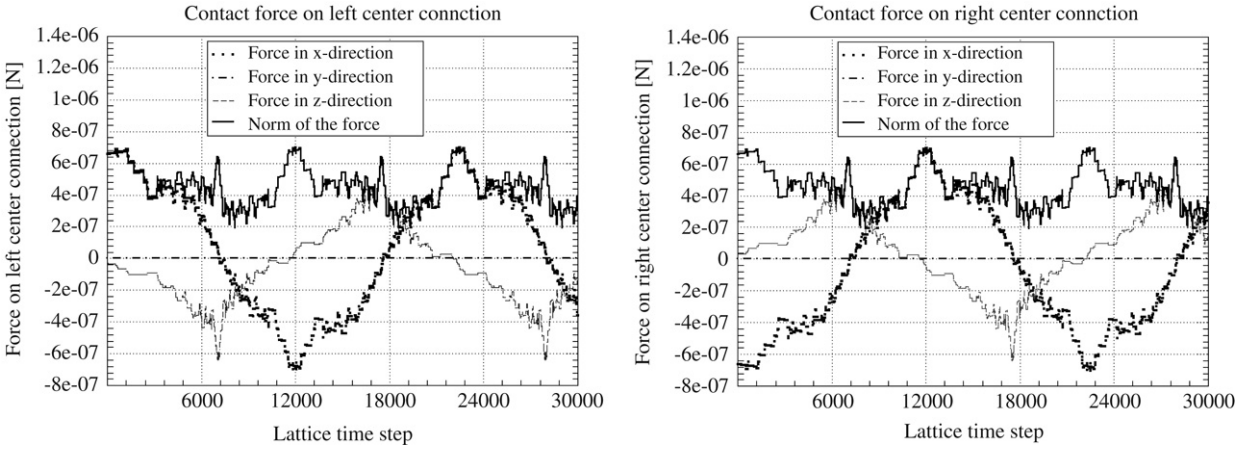

Fig. 8. Contact forces for the two center contacts of the dual star agglomerate. The used size of the LBM simulation domain was 120^3 cells and the radius of the sphere was 8 cells.

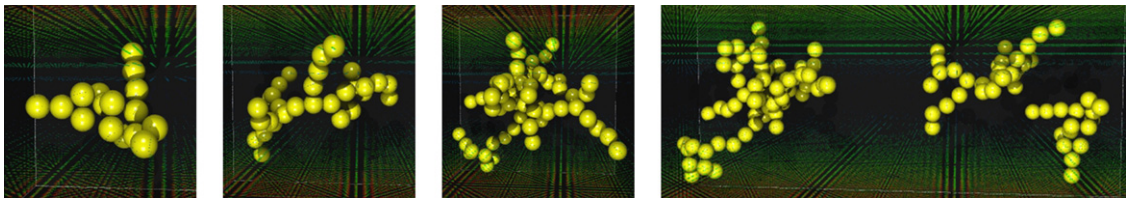

Fig. 9. Examples for complex particle agglomerates with 16, 32 and 64 spheres. The right illustration shows a simulation of three complex agglomerates in one simulation domain.

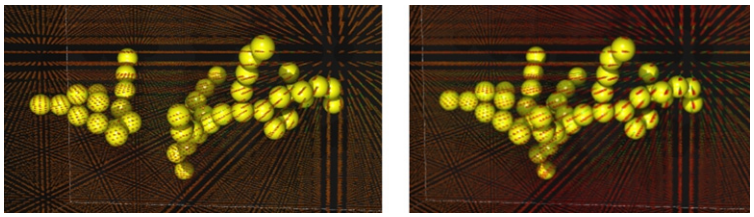

Fig. 10. Collision between two particle agglomerates in a channel flow.

Fig. 10 demonstrates a collision between two particle agglomerates in a channel flow. Due to the kinetic energy during the collision, these two particle agglomerates can concatenate and build a new particle agglomerate.

6. Conclusion

The investigation of the flow behavior of particles can be supported by the numerical simulation approach presented in this paper. By extending the LBM with a treatment for curved boundaries and a force interaction method between the fluid and the particles, and by coupling the LBM fluid simulation to a rigid body physics engine, an accurate simulation of moving agglomerates can be implemented. This simulation is also able to calculate the contact forces and torques between several sintered particles and, therefore, predict connection break-ups. However, in some simulations, the accuracy of the simulation results still need to be improved. For example, the momentum exchange approach could be extended by a finite volume calculation in order to reduce the force fluctuations. Additionally, the problem of the excessive memory usage has to be addressed by memory reduction techniques or parallelization.

References

[1] D. Yu, R. Mei, W. Shyy, A unified boundary treatment in lattice Boltzmann, in: AIAA 41st Aerospace Sciences Meeting & Exhibit, 2003.

[2] A.J.C. Ladd, Numerical simulation of particular suspensions via a discretized Boltzmann equation. Part 2. Numerical results, J. Fluid Mech. 271 (1994) 311–339.
[3] Dazhi Yu, Renwei Mei, Li-Shi Luo, Wei Shyy, Viscous flow computations with the method of lattice Boltzmann equation, Prog. Aerospace Sci. 39 (2003) 329–367. Published by Elsevier Science Ltd. http://www.sciencedirect.com.
[4] Sauro Succi, The Lattice Boltzmann Equation for Fluid Dynamics and Beyond, Oxford University Press, ISBN: 0198503989, 2001.
[5] Tomas Akenine-Möller, Eric Haines, Real Time Rendering, 2nd edition, AK Peters, Ltd, ISBN: 1568811829, 2002, 1.9.
[6] Christian Feichtinger, Drag Force Simulations of Particle Agglomerates with the Lattice-Boltzmann Method. Chair for Computer Science 10 (System Simulation), University of Erlangen-Nuremberg, Germany, June 2005. http://www10.informatik.uni-erlangen.de.
[7] C. Binder, C. Feichtinger, H. Schmid, N. Thürey, W. Peukert, U. Rüde, Simulation of the hydrodynamic drag of aggregated particles, J. Colloid Interf. Sci. 301 (1) (2006) 155–167.
[8] Dominik Geuß, Momentum calculation for particle agglomerate connections, Chair for Computer Science 10 (System Simulation), University of Erlangen-Nuremberg, Germany, April 2006. http://www10.informatik.uni-erlangen.de.
[9] C.K. Aidun, Y. Lu, E.-J. Ding, Direct analysis of particulate suspensions with inertia using the discrete Boltzmann equation, J. Fluid Mech. 373 (1998) 287–311.
[10] Dazhi Yu, Renwei Mei, Wei Shyy, A multi-block lattice Boltzmann method for viscous fluid flows, Internat. J. Numer. Methods Fluids 39 (2002) 99–120.
[11] Klaus Iglberger, Lattice Boltzmann simulation of flow around moving particles, Chair for Computer Science 10 (System Simulation), University of Erlangen-Nuremberg, Germany, June 2005. http://www10.informatik.uni-erlangen.de.
[12] T. Pohl, M. Kowarschik, J. Wilke, K. Iglberger, U. Rüde, Optimization and profiling of the cache performance of parallel Lattice Boltzmann codes, Parallel Process. Lett. 13 (4) (2003) 549–560.