



# Lattice Boltzmann large eddy simulation of subcritical flows around a sphere on non-uniform grids

M. Stiebler\*, M. Krafczyk, S. Freudiger, M. Geier

*Institute for Computational Modeling in Civil Engineering, Technische Universität Braunschweig, Germany*

## ARTICLE INFO

### Keywords:

Lattice Boltzmann  
Large eddy simulation  
Transitional flow  
Local grid refinement  
Distributed simulation

## ABSTRACT

In this work, the suitability of the lattice Boltzmann method is evaluated for the simulation of subcritical turbulent flows around a sphere. Special measures are taken to reduce the computational cost without sacrificing the accuracy of the method. A large eddy simulation turbulence model is employed to allow efficient simulation of resolved flow structures on non-uniform computational meshes. In the vicinity of solid walls, where the flow is governed by the presence of a thin boundary layer, local grid-refinement is employed in order to capture the fine structures of the flow. In the test case considered, reference values for the drag force in the Reynolds number range from 2000 to 10 000 and for the surface pressure distribution and the angle of separation at a Reynolds number of 10 000 could be quantitatively reproduced. A parallel efficiency of 80% was obtained on an Opteron cluster.

© 2011 Elsevier Ltd. All rights reserved.

## 1. Introduction

In recent years, the lattice Boltzmann method (LBM) has been established as a method for solving weakly compressible and incompressible flow problems. The LBM was found to be competitive with traditional methods based on the direct discretization of the Navier–Stokes equations in a wide range of applications with complex flow physics, including multiphase flows, suspensions in fluids, and flows in porous media [1–3]. LBM has also been demonstrated to be an efficient simulation tool for laminar flows [4]. However, the efficiency of LB methods for flow problems around bluff bodies in the turbulent subcritical regime has not been investigated in depth, which may be due to the fact that parallel LBM implementations based on turbulence models on non-uniform hierarchical grids require substantial programming effort. Yet, the efficient simulation of turbulent engineering problems requires the use of non-uniform meshes to resolve the large velocity gradients in the boundary layer. As a direct numerical simulation (DNS) of technically relevant turbulent flows is prohibitively expensive, turbulence modeling is required. Two main types of turbulence models have previously been introduced into the LBM context: Reynolds averaged Navier–Stokes (RANS) type models [5] and large eddy simulation models (LES) [6]. Additionally, hybrid approaches have been developed, such as the detached Eddy simulation [7]. In RANS models, a time- or ensemble-averaged velocity is computed and the turbulent features of the flow have to be modeled, including all of the spatial scales. LES models, on the other hand, apply a spatial low-pass filter to the hydrodynamic fields. The optimal filter would eliminate only high frequencies in the flow field while their influence on the large eddies, which are resolved in the simulation, is modeled. However, the optimal filter could only be defined in Fourier space and, hence, would require prohibitively large stencils. This is true for the lattice Boltzmann model and a classical finite difference Navier–Stokes solver alike. However, the simplest possible filter based on shear stress can be implemented very efficiently in the lattice Boltzmann method because of the local availability of the shear stress in the non-equilibrium part of the distribution

\* Corresponding author. Tel.: +49 1637296245; fax: +49 5313917599.

E-mail addresses: [stiebler@irmb.tu-bs.de](mailto:stiebler@irmb.tu-bs.de) (M. Stiebler), [kraft@irmb.tu-bs.de](mailto:kraft@irmb.tu-bs.de) (M. Krafczyk), [freud@irmb.tu-bs.de](mailto:freud@irmb.tu-bs.de) (S. Freudiger), [geier@irmb.tu-bs.de](mailto:geier@irmb.tu-bs.de) (M. Geier).

function. This information can be employed to implement an implicit local low-pass filter also for non-uniform grids at very low computational cost.

In this article, an efficient implementation of a lattice Boltzmann method with LES turbulence modeling on non-uniform grids is introduced, which allows a relatively simple turbulence modeling to be successfully applied to problems relevant in engineering. The first part of this article will briefly recall some basic features of the lattice Boltzmann multiple-relaxation-time method with a subgrid stress model. The second part will deal with extensions of the method for non-uniform grids. In the third part, the method will be applied to the calculation of some distinct flow features of a sphere in a channel in the subcritical regime.

## 2. Lattice Boltzmann method with subgrid stress model (LBM-SGS)

Unlike traditional numerical methods in Computational Fluid Dynamics that apply some discretization technique to the Navier–Stokes equations, the lattice Boltzmann method is a discretization of the Boltzmann equation in a discrete velocity space. Its solutions can be shown to converge asymptotically to the solution of the incompressible Navier–Stokes equation [8]. The starting point for the derivation of the lattice Boltzmann method is a momentum density distribution of identical particles,  $f(\mathbf{x}, \boldsymbol{\xi}, t)$ . The distribution is discrete in space, velocity space, and time. In the lattice Boltzmann method the discretization of space, called the lattice, corresponds exactly to the discretization of velocity space, so that a virtual particle sitting on a node in the lattice moves to a neighboring node in a discrete time step. Hence, interpolation is not required. The Navier–Stokes equation is found to be the equation of motion for the first moment  $\sum_i \mathbf{e}_i f_i$  of the momentum distribution, provided that a sufficiently isotropic velocity set  $\mathbf{e}_i$  is used. A common choice for the lattice Boltzmann velocity set is the D3Q19 (three dimensions, 19 speeds, cf. Appendix B) model [8]. Its microscopic velocity components correspond to the third, fifth and seventh rows of the matrix given in Appendix C. The evolution of the lattice Boltzmann model can be split into two steps: first, free streaming or propagation of the distribution  $f_i$  according to its respective speed and, second, collision on the lattice nodes. In order to obtain the Navier–Stokes equation, the collision operator has to fulfill some conditions concerning conservation laws, Galilean invariance, and dissipation. Still, the collision operator is not uniquely defined. Different choices will have different stability characteristics. In this paper, a version of a multiple-relaxation-time [9] collision operator is employed. That specific version was used in previous works of the authors (e.g. [10]) and the description is recapped here.

The lattice Boltzmann model can be written as

$$f_i(t + \Delta t, \mathbf{x} + \mathbf{e}_i \Delta t) = f_i(t, \mathbf{x}) + \Omega_i, \quad i = 0, \dots, b-1, \quad (1)$$

where  $\Delta t$  is the time step and the collision operator is given by

$$\Omega = \mathbf{M}^{-1} \mathbf{S} [(\mathbf{M} \mathbf{f}) - \mathbf{m}^{\text{eq}}]. \quad (2)$$

Matrix  $\mathbf{M}$  is the transformation matrix given in Appendix C, composed of the 19 orthogonal basis vectors  $\{\Phi_i, i = 0, \dots, b-1\}$  given in Appendix A, which are orthogonal with respect to a weighted inner product,  $\langle \Phi_i, \Phi_j \rangle_w = \sum_k \Phi_{ik} \Phi_{jk} w_k = 0$ , if  $i \neq j$  (in contrast to [9], where  $\langle \Phi_i, \Phi_j \rangle = \sum_k \Phi_{ik} \Phi_{jk} \neq 0$ , if  $i \neq j$ ). The vector  $\mathbf{w}$  is composed of the weights  $\{w_i, i = 0, \dots, b-1\}$ :

$$\mathbf{w} = \left( \frac{1}{3}, \frac{1}{18}, \frac{1}{18}, \frac{1}{18}, \frac{1}{18}, \frac{1}{18}, \frac{1}{18}, \frac{1}{36}, \frac{1}{36}, \frac{1}{36}, \frac{1}{36}, \frac{1}{36}, \frac{1}{36}, \frac{1}{36}, \frac{1}{36}, \frac{1}{36}, \frac{1}{36}, \frac{1}{36} \right).$$

The moments  $\mathbf{m} = \mathbf{M} \mathbf{f}$  are labeled as

$$\mathbf{m} = (\rho, e, \epsilon, j_x, q_x, j_y, q_y, j_z, q_z, 3p_{xx}, 3\pi_{xx}, p_{ww}, \pi_{ww}, p_{xy}, p_{yz}, p_{xz}, m_x, m_y, m_z).$$

$\mathbf{m}^{\text{eq}}$  is the vector composed of the equilibrium moments given in Eq. (5) and  $\mathbf{S} = \{s_{ii}, i = 0, \dots, b-1\}$  is the diagonal collision matrix. The non-zero collision parameters  $s_{ii}$  (the eigenvalues of the collision matrix  $\mathbf{M}^{-1} \mathbf{S} \mathbf{M}$ ) are:

$$\begin{aligned} s_{1,1} &= s_a \\ s_{2,2} &= s_b \\ s_{4,4} &= s_{6,6} = s_{8,8} = s_c \\ s_{10,10} &= s_{12,12} = s_d \\ s_{9,9} &= s_{11,11} = s_{13,13} = s_{14,14} = s_{15,15} = -\frac{\Delta t}{\tau} = s_\omega \\ s_{16,16} &= s_{17,17} = s_{18,18} = s_e. \end{aligned} \quad (3)$$

The relaxation time  $\tau$  is chosen as

$$\tau = 3 \frac{\nu}{c^2} + \frac{1}{2} \Delta t, \quad (4)$$

where  $\nu$  is the kinematic viscosity. The parameters  $s_a, s_b, s_c, s_d$  and  $s_e$  can be freely chosen in the range  $[-2, 0]$  and tuned to improve stability [11]. While the optimal values for these parameters depend on both the geometry and the

initial and boundary conditions of the specific system under consideration, reasonable values are given in [9]. We choose  $s_a = s_b = s_c = s_d = s_e = -1.0$ . The non-zero equilibrium distribution functions  $\{m_i^{\text{eq}}, i = 0, \dots, 18\}$  are given by

$$m_0^{\text{eq}} = \rho, \quad m_3^{\text{eq}} = \rho_0 u_x, \quad m_5^{\text{eq}} = \rho_0 u_y, \quad m_7^{\text{eq}} = \rho_0 u_z, \quad (5a)$$

$$m_1^{\text{eq}} = e^{\text{eq}} = \rho_0(u_x^2 + u_y^2 + u_z^2), \quad (5b)$$

$$m_9^{\text{eq}} = 3p_{xx}^{\text{eq}} = \rho_0(2u_x^2 - u_y^2 - u_z^2), \quad (5c)$$

$$m_{11}^{\text{eq}} = p_{zz}^{\text{eq}} = \rho_0(u_y^2 - u_z^2), \quad (5d)$$

$$m_{13}^{\text{eq}} = p_{xy}^{\text{eq}} = \rho_0 u_x u_y, \quad (5e)$$

$$m_{14}^{\text{eq}} = p_{yz}^{\text{eq}} = \rho_0 u_y u_z, \quad (5f)$$

$$m_{15}^{\text{eq}} = p_{xz}^{\text{eq}} = \rho_0 u_x u_z, \quad (5g)$$

where  $\rho_0$  is a constant density,  $\rho$  is a density variation, and the velocities are derived from the moments representing momenta:  $u_\alpha = j_\alpha / \rho_0$ .

The spatially filtered Navier–Stokes equation for the filtered velocity  $\bar{\mathbf{u}}$  is formulated according to the turbulent-viscosity hypothesis:

$$\frac{\partial \bar{u}_\alpha}{\partial t} + \frac{\partial \bar{u}_\alpha \bar{u}_\beta}{\partial x_\beta} = -\frac{\partial P / \rho}{\partial x_\alpha} + \frac{\partial}{\partial x_\beta} \left[ (v_0 + v_T) \left( \frac{\partial \bar{u}_\alpha}{\partial x_\beta} + \frac{\partial \bar{u}_\beta}{\partial x_\alpha} \right) \right], \quad (6)$$

where  $v_0$  is the constant molecular viscosity (here and in the remainder of the text, whenever we use the term “viscosity”, we refer to kinematic viscosity as opposed to dynamic viscosity) and  $v_T$  the time- and space-dependent turbulent viscosity. Analogously, to implement an LB model for the filtered equation, we replace the relaxation rate  $\tau$  in Eq. (3) with a time- and space-dependent  $\tau_{\text{total}} = \tau_0 + \tau_T$ , which is computed as follows:

According to the Smagorinsky model, the eddy, or turbulent, viscosity  $v_T$  depends on the strain rate:

$$v_T = (C_S \Delta x)^2 \|\mathbf{S}\| \quad (7)$$

with the strain rate tensor

$$S_{\alpha\beta} = \frac{1}{2} \left( \frac{\partial \bar{u}_\alpha}{\partial x_\beta} + \frac{\partial \bar{u}_\beta}{\partial x_\alpha} \right). \quad (8)$$

$C_S$  is the Smagorinsky model constant, which we set to 0.18 for our simulations. The strain rate tensor can be locally computed from the non-equilibrium distribution functions [12]  $\mathbf{f}^{\text{neq}} = \mathbf{M}^{-1}(\mathbf{M}\mathbf{f} - \mathbf{m}^{\text{eq}})$  as

$$S_{\alpha\beta} = -\frac{3}{2\tau_{\text{total}}c^2} \Pi_{\alpha\beta}^{\text{neq}}, \quad (9)$$

where the momentum flux tensor  $\Pi^{\text{neq}}$  is defined as

$$\Pi_{\alpha\beta}^{\text{neq}} = \sum_i (e_{i\alpha} e_{i\beta} f_i^{\text{neq}}). \quad (10)$$

With Eq. (7), (9) and the equivalent of Eq. (4),

$$\tau_{\text{total}} = \frac{3}{c^2} v_{\text{total}} + \frac{1}{2} \Delta t = \frac{3}{c^2} (v_0 + v_T) + \frac{1}{2} \Delta t, \quad (11)$$

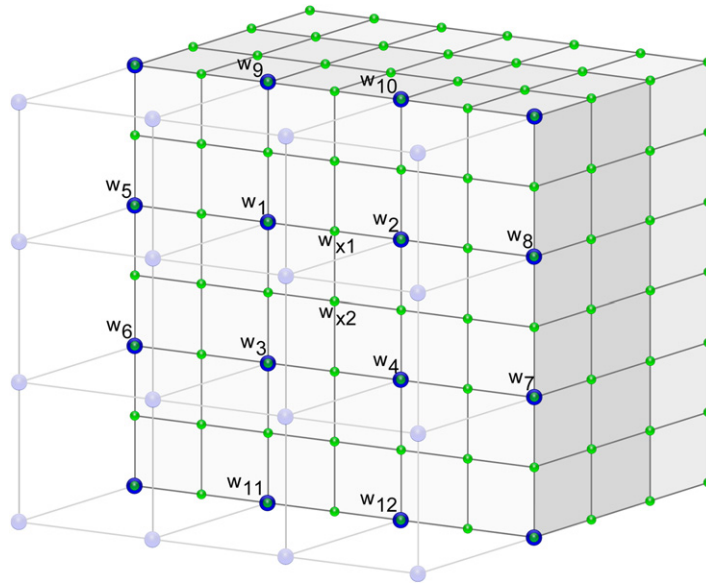
a quadratic equation for  $\tau_{\text{total}}$  can be derived, with the solution

$$\tau_{\text{total}} = \frac{3}{c^2} v_0 + \frac{1}{2} \Delta t + \frac{\sqrt{\tau_0^2 + \frac{18C_S^2 \Delta t^2 Q}{c^2}} - \tau_0}{2} \quad (12)$$

where  $Q = \sqrt{\sum_{\alpha\beta} 2\Pi_{\alpha\beta}^{\text{neq}} \Pi_{\alpha\beta}^{\text{neq}}}$ .

### 3. LBM-SGS on hierarchical block structured grids

Due to the emergence of boundary layers in turbulent flows, high velocity gradients occur near solid walls. To capture these features of the flow, the resolution required near walls is much higher than the resolution required in the rest of the simulation domain. Thus, local refinement of the computational grid is a mandatory part of an efficient method for solving



**Fig. 1.** Schematic view of the interface between two lattice Boltzmann grids with different resolution. Missing distributions on the fine nodes  $w_{x1}$  and  $w_{x2}$  are spatially and temporally interpolated from distributions at the coarse nodes  $w_{1-12}$  [15].

turbulent flow problems. In this work, a block-structured type of grid is used. In each block the resolution is fixed. At the boundary between two blocks the spatial resolution can change by a factor of two.

The hyperbolic scaling  $\Delta t_l c = \Delta x_l^1$  with a level-independent microscopic velocity  $c$  is employed. The Mach and the Reynolds numbers are thus equal on all grids. This naturally leads to a nested time stepping scheme. One time step on a coarse grid corresponds to two time steps on the next finer grid level. An approach where the Mach number is scaled down on finer grid levels to ensure convergence to the *incompressible* Navier–Stokes equations can be found in [13]. That approach requires four time steps on the fine grid during one time step on the coarse mesh.

The local value of the relaxation time  $\tau_{\text{total}}$  is rescaled by introducing the level-dependence of the time step into Eqs. (12) and (4), i.e.

$$\tau_{\text{total}} = \frac{3}{c^2} \nu_0 + \frac{1}{2} \Delta t_l + \frac{\sqrt{\tau_{0,l}^2 + \frac{18c_s^2 \Delta t_l^2 Q}{c^2}} - \tau_{0,l}}{2}, \quad (13)$$

$$\tau_{0,l} = 3 \frac{\nu_0}{c^2} + \frac{1}{2} \Delta t_l. \quad (14)$$

It is important to note that the local relaxation time includes the contributions from the local eddy viscosity, which according to Eq. (7) jumps by a factor of four at the grid interface as the eddy viscosity  $\nu_T$  is proportional to the square of the filter width (i.e. grid spacing). Thus the eddy viscosity contribution will go to zero for sufficiently fine grids, i.e. when the flow is sufficiently resolved.

A typical grid interface is shown in Fig. 1. As the propagation step does not provide all distributions on one grid level at the interface, the missing distributions on each grid level have to be computed from the adjacent grid level by a cubic spatial and linear temporal interpolation [14].

The continuity of the stress tensor requires rescaling of the non-equilibrium part  $\mathbf{m}^{\text{neq}}$  [16,14]. Rescaling is applied after the streaming step by

$$m_{i,l-1}^{\text{neq}} = \frac{s_{ii,l}}{s_{ii,l-1}} \frac{\Delta t_{l-1}}{\Delta t_l} m_{i,l}^{\text{neq}}. \quad (15)$$

With Eqs. (3) and (4) we obtain

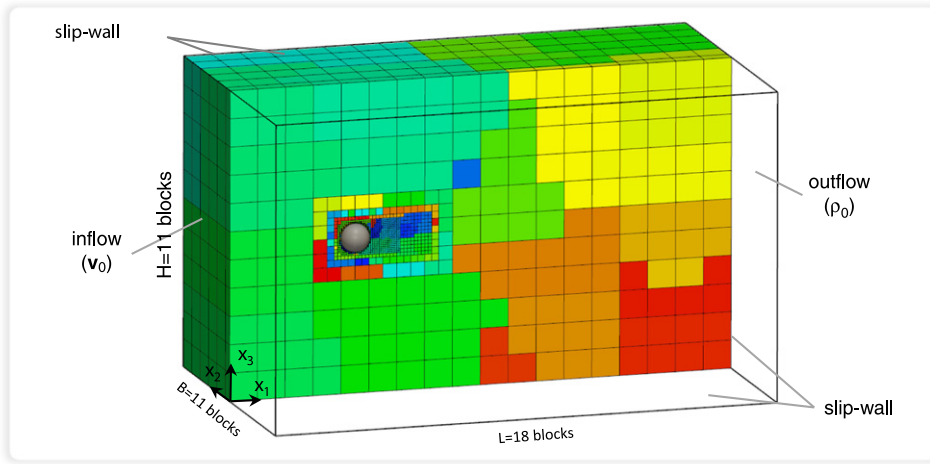
$$m_{i,l-1}^{\text{neq}} = 2\beta_{i,l} m_{i,l}^{\text{neq}} \quad (16)$$

with  $\beta_{i,l} = 1$  for  $i \in \{1, 2, 4, 6, 8, 10, 12, 16, 17, 18\}$  and

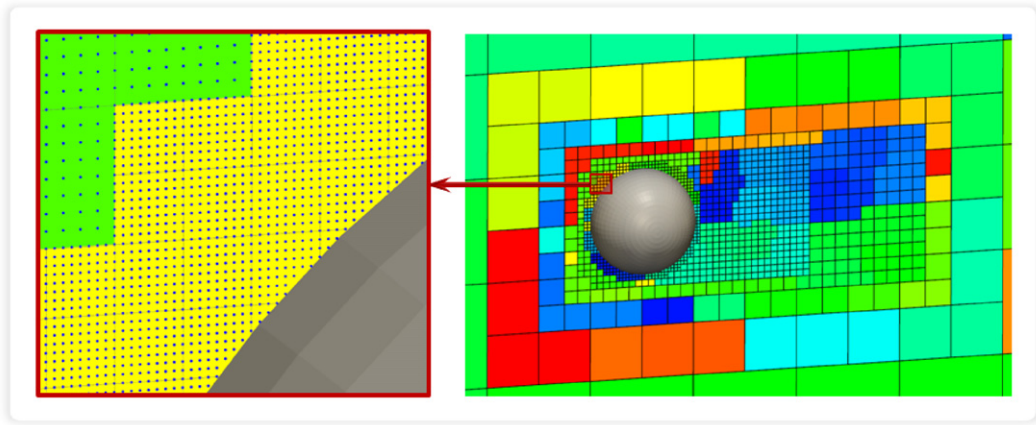
$$\beta_{i,l} = \frac{\Delta t_l (6\nu_0 + \Delta t_{l-1} c^2)}{\Delta t_{l-1} (6\nu_0 + \Delta t_l c^2)} \quad (17)$$

for  $i \in \{9, 11, 13, 14, 15\}$ , where  $\nu_0$  is the molecular viscosity.

<sup>1</sup> Here and in the following,  $l$  denotes the level of refinement of a block, higher numbers representing a higher resolution, i.e.  $2^l \cdot \Delta x_l = \text{const.}$  As a subscript,  $l$  or functions of  $l$  denote a level-specific value of an entity.



**Fig. 2.** Setup of the test case with six block levels: each square in the lower right picture represents a  $9 \times 9 \times 9$  block of grid points. The total number of grid points is  $\approx 2E08$ . The block color represents the corresponding subdomain index.



**Fig. 3.** Block grid with six levels next to sphere (right) and individual grid nodes (left).

Temporal and spatial interpolations are required because of the different mesh spacings  $\Delta x_i$  and time steps  $\Delta t_i$ . A cubic interpolation in space is used for the those grid nodes which do not have corresponding neighboring nodes of the same grid level in at least one of the  $b - 1$  lattice directions (Fig. 1). Details of the algorithm can be found in [14,17,18].

#### 4. Test case and numerical results

The presented test case for the method is the turbulent flow around a sphere in a square channel (Figs. 2 and 3). The blockage ratio, i.e. the ratio of channel width  $h$  and sphere diameter  $d$ , is  $\lambda = h/d = 11$ . In the inlet of the channel, a constant velocity  $u_{in}$  is prescribed. It is well known, that the turbulent intensity of the upstream flow might influence the turbulence statistics, yet this is much more relevant for supercritical bluff body flows with Reynolds numbers exceeding  $10^5$ . Thus we do not consider turbulent inflows in this study. On the surface of the sphere, no-slip (i.e. second-order bounce-back [19]) boundary conditions are used. On the wall of the channel, a slip velocity boundary condition is applied.

The test case is evaluated for the Reynolds number range  $Re = u_{in}d/\nu \in [2000, 10\,000]$ . The drag force on the sphere  $F_d$  is computed with the momentum-exchange method [20]. After temporal averaging, the computed drags are compared with a curve taken from experimental and numerical studies [21]:

$$\log_{10}(c_d) = -2.4571 + 2.5558 \log_{10}(Re) - 0.9295 \log_{10}(Re)^2 + 0.1049 \log_{10}(Re)^3 \quad (18)$$

where the drag force coefficient  $c_d$  is defined by

$$c_d = 2 \frac{F_d}{\pi \rho u_{in}^2 r^2}, \quad (19)$$

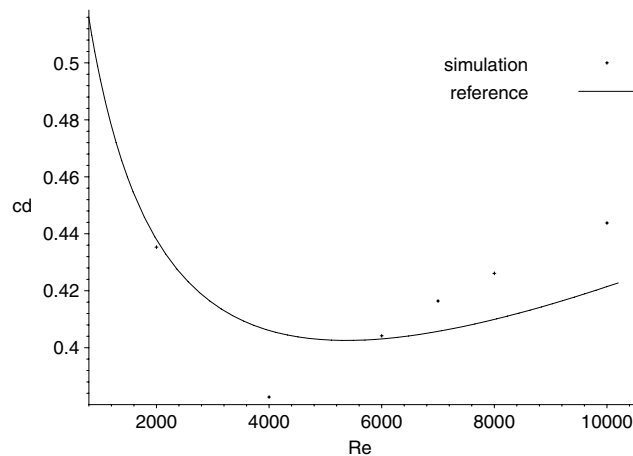


Fig. 4. Comparison of computed drag with the empirical results given by Eqs. (18) and (20).

Table 1

Comparison of setup 1 used for simulations at Reynolds number 2000–10 000 and setup 2 used additionally for Reynolds number 10 000.

	Number of fluid nodes	Number of fluid nodes weighted by $2^l$	CPU time per coarse time step (750 000 LUPS) (s)
Setup 1 (6 levels)	1.9E + 07	4.0E + 08	533
Setup 2 (7 levels)	4.6E + 07	3.2E + 09	4267

where  $r = d/2$  is the radius of the sphere. While Eq. (18) applies to channels with very large blockage ratios, i.e.  $h \gg d$ , [21] gives an empirical correction factor for  $c_d$  for the case of moderate blockage ratios:

$$c_{d,\text{corr}} = k_c c_d \quad (20)$$

$$k_c = \frac{1}{1 - 1.6\lambda^{1.6}}.$$

The instantaneously measured drag force is subject to strong turbulent fluctuations that require long temporal averaging. The high computational cost of the simulation restricts our measurement periods to intervals that only allow for preliminary results. The measurement was started after  $10 \frac{L}{u}$ , with  $L$  being the length of the simulation domain (Fig. 2), to allow the initial aperiodic flow to leave the domain. Averaging was applied over 25 turnover times  $\frac{d}{u}$ . In simulation time this corresponds to the interval  $[36\,250\Delta t_c, 41\,250\Delta t_c]$ . The results of the comparison can be seen in Fig. 4.

It can be stated that the simulation can reproduce the correct trend of the empirical curve and that the results for each data point are off by  $\leq 6\%$ , which is within the range of the uncertainty of the reference results.

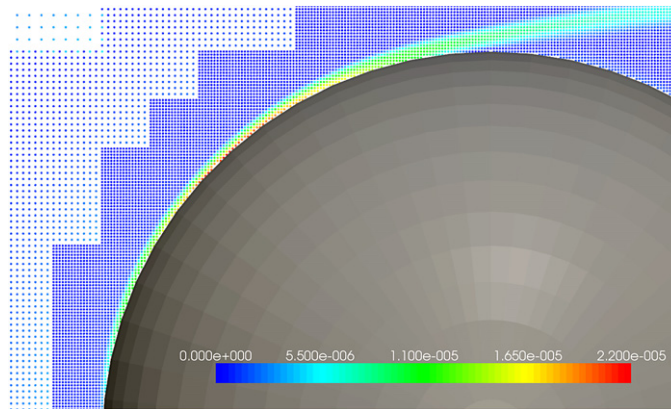
The thickness of the boundary layer can be estimated as  $1.3d/\sqrt{\text{Re}}$  in the region of the stagnation point upstream of the sphere and as  $1.0d/\sqrt{\text{Re}}$  near the center plane of the sphere perpendicular to the main flow direction [21]. We choose the resolution around the sphere such that the sphere diameter is resolved by 256 grid points and, therefore, the boundary layer is resolved with  $\approx 3$  points at  $\text{Re} = 10\,000$  and  $\approx 6$  points at  $\text{Re} = 2000$ .

With the given resolution, one time step on the coarsest grid level requires  $4 \cdot 10^8$  lattice node updates (Table 1). The LB research prototype software package VIRTUALFLUIDS, described in detail in [15], was used for the distributed simulations. It has been executed on a 64-bit-AMD-Opteron cluster with 47 dual processor boards ( $2 \times 1.4$  GHz), each connected via a full duplex Myrinet 2000 network interface. On a single core processor, the number of lattice node updates per second (LUPS) is  $\approx 9 \times 10^5$ . Seventy processors were used for the simulations in this work and a parallel efficiency of 80% was obtained. The limiting factors for the efficiency were the specific properties of the given network topology and the imperfect load balancing of the domain decomposition done by METIS [22] for non-uniform grids.

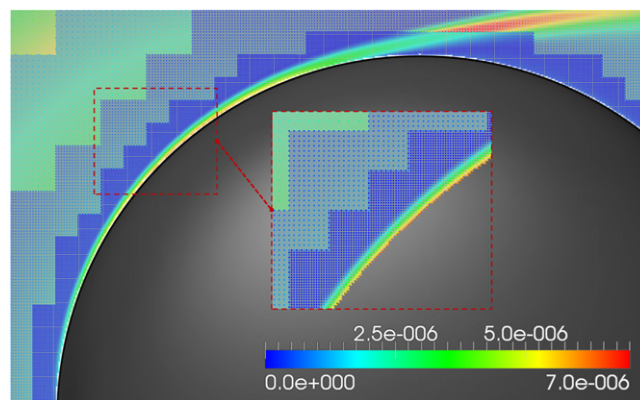
#### 4.1. A detailed look at the case of $\text{Re} = 10\,000$

For the case of  $\text{Re} = 10\,000$  and  $\text{Ma} = 0.04c/c_s \approx 0.069$  (with  $c_s = c/\sqrt{3}$  being the speed of sound), the flow is still slightly under-resolved at the finest grid level six. The eddy viscosity, which would vanish in the limit of infinite resolution, is found to be  $\nu_T \approx 0.7\nu_0$  (see also Fig. 5), i.e. given the large velocity gradient in the boundary layer the contribution of the eddy viscosity to the total viscosity is much smaller than in the bulk flow, where the eddy viscosity may surpass the molecular viscosity by two orders of magnitude. Motivated by this, an additional simulation was set up where the vicinity





**Fig. 5.** The color in the plot indicates the amount of eddy viscosity produced by the Smagorinsky model. For comparison, the numerical value of the molecular viscosity is  $3.2 \cdot 10^{-5}$  (measured in the units of the coarsest grid used in the simulation, i.e.  $[\nu] = c\Delta x_0$ ). Thus, in the boundary layer the contribution of eddy viscosity to the total viscosity is significant.



**Fig. 6.** Eddy viscosity around a sphere with higher resolution (setup 2). For comparison, the molecular viscosity is  $3.2 \cdot 10^{-5} c\Delta x_{l=0}$ . Thus, eddy viscosity is about an order of magnitude less.

of the sphere containing the boundary layer was resolved by one more grid level (seven). As expected, the contribution of the eddy viscosity could thereby be reduced by a factor of four (Fig. 6).

Yet, even with the coarser resolution, we observe that the pressure distribution over the surface and the angle of separation are almost identical and thus well captured using six grid levels. The parallel efficiency does not decrease for seven grid levels. Outside of the boundary layer, the eddy viscosity quickly vanishes to a negligible amount, acting only as a filter which eliminates undesired high frequencies in the flow field. In order to satisfy the boundary layer theory the eddy viscosity near the wall should also vanish. To avoid excessive grid refinement near boundaries, wall models such as Van Driest damping [23] or more advanced wall models could be incorporated in the code in future work.

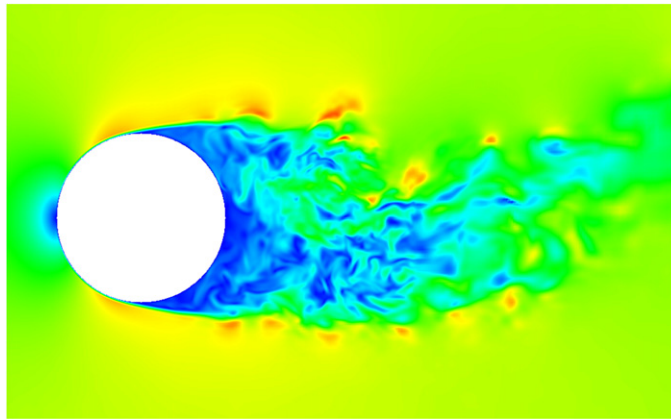
Fig. 7 shows a snapshot of an instantaneous velocity field close to the sphere and Fig. 8 shows the mean pressure distribution.

The distribution of the mean pressure coefficient  $c_p = \frac{p-p_\infty}{0.5\rho u_\infty^2}$  (with  $p_\infty$  the free stream pressure and  $u_\infty$  the free stream velocity) around the sphere is shown in Fig. 9. Its distribution is almost independent of the Reynolds number in the subcritical regime. The computed values agree very well with results from the literature [24]. The angle of separation was measured at 83.5 degrees and resides in the range of values from Achenbach (83.5° at  $Re = 10\,000$ ) [25] and Bakic (80°–83° at  $Re = 50\,000$ ) [26].

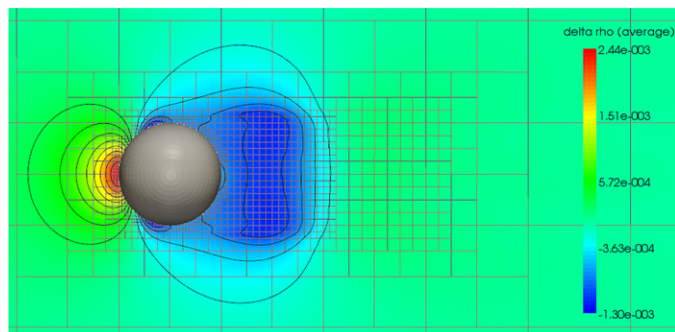
Fig. 10 shows the contours of one component of the Reynolds stress tensor  $\overline{v'_x v'_y}$ , where the prime denotes temporal fluctuations of the velocities and the overline denotes temporal average,  $x$  is the streamwise direction and  $y$  is the other coordinate direction which is parallel to the cutting plane shown.

## 5. Conclusion

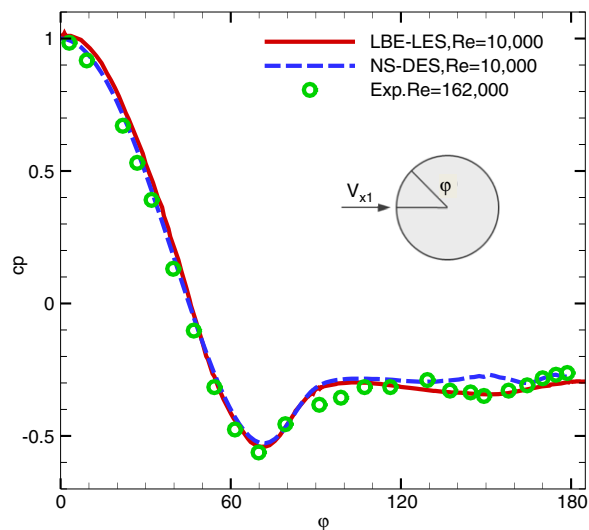
A lattice Boltzmann model and parallel implementation for the computation of subcritical flow around a sphere has been presented. It features LES/Smagorinsky SGS turbulence modeling and a non-uniform computational grid, which are



**Fig. 7.** Snapshot of the instantaneous velocity magnitude on a plane cutting through the center of the sphere in the main direction of the flow and perpendicular to the walls.



**Fig. 8.** Pressure time average on a plane cutting through the center of the sphere in the main direction of the flow and perpendicular to the walls.



**Fig. 9.** Comparison of the simulated mean pressure coefficient distribution on the surface of the sphere with literature values.

necessary for a sufficient resolution of the boundary layer while keeping the overall computational effort low. The method lends itself to high parallel efficiency.

For the test case of flow around a sphere in a channel, for a range of Reynolds numbers between 2000 and 10000, our simulations capture the essential features of these flows; compared to values reported in the literature, the drag coefficient



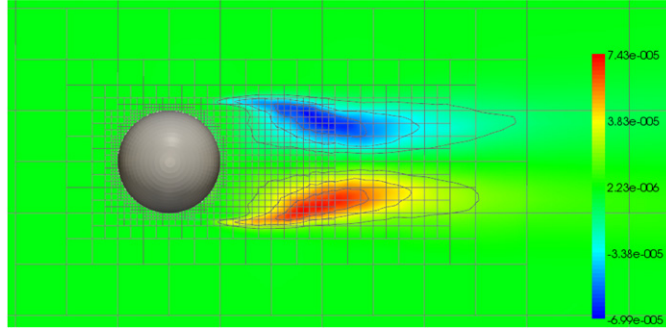


Fig. 10. Reynolds stress  $\overline{v'_x v'_y}$ .

is reproduced with a relative error of 6% at worst. For the case of  $Re = 10\,000$ , the detachment angle and surface pressure distribution are captured very well. We conclude that the lattice Boltzmann method is suitable for subcritical flows around bluff bodies even with the simple turbulence model used in this work. For a further reduction of the computational time, the development of a wall model compatible with the LB framework is desirable. In combination with a near-wall compatible SGS model as described in [27] we expect an improved performance. These investigations are presently being conducted and will be the subject of future publication.

### Acknowledgments

Financial support from the Federal Ministry of Education and Research (BMBF) under the SKALB project grant 01IH08003E is gratefully acknowledged. We also thank the DEISA Consortium ([www.deisa.eu](http://www.deisa.eu)), funded through the EU FP7 project RI-222919, for support within the DEISA Extreme Computing Initiative, and Mr. John Gounley for editorial assistance.

### Appendix A. Orthogonal basis vectors $\{\Phi_i\}$

The 19 orthogonal basis vectors  $\{\Phi_i, i = 0, \dots, b-1\}$  required for the transformation of the D3Q19 momentum distribution function to its corresponding moments are listed below:

$$\Phi_{0,\alpha} = 1, \quad \Phi_{1,\alpha} = e_{\alpha}^2 - c^2, \quad \Phi_{2,\alpha} = 3(e_{\alpha}^2)^2 - 6e_{\alpha}^2 c^2 + c^4, \quad (A.1)$$

$$\Phi_{3,\alpha} = e_{\alpha x}, \quad \Phi_{5,\alpha} = e_{\alpha y}, \quad \Phi_{7,\alpha} = e_{\alpha z}, \quad (A.2)$$

$$\Phi_{4,\alpha} = (3e_{\alpha}^2 - 5c^2)e_{\alpha x}, \quad \Phi_{6,\alpha} = (3e_{\alpha}^2 - 5c^2)e_{\alpha y}, \quad (A.3)$$

$$\Phi_{8,\alpha} = (3e_{\alpha}^2 - 5c^2)e_{\alpha z}, \quad (A.4)$$

$$\Phi_{9,\alpha} = 3e_{\alpha x}^2 - e_{\alpha}^2, \quad \Phi_{11,\alpha} = e_{\alpha y}^2 - e_{\alpha}^2, \quad (A.5)$$

$$\Phi_{13,\alpha} = e_{\alpha x}e_{\alpha y}, \quad \Phi_{14,\alpha} = e_{\alpha y}e_{\alpha z}, \quad \Phi_{15,\alpha} = e_{\alpha x}e_{\alpha z}, \quad (A.6)$$

$$\Phi_{10,\alpha} = (2e_{\alpha}^2 - 3c^2)(3e_{\alpha x}^2 - e_{\alpha}^2), \quad \Phi_{12,\alpha} = (2e_{\alpha}^2 - 3c^2)(e_{\alpha y}^2 - e_{\alpha}^2), \quad (A.7)$$

$$\Phi_{16,\alpha} = (e_{\alpha y}^2 - e_{\alpha}^2)e_{\alpha x}, \quad \Phi_{17,\alpha} = (e_{\alpha z}^2 - e_{\alpha}^2)e_{\alpha y}, \quad (A.8)$$

$$\Phi_{18,\alpha} = (e_{\alpha x}^2 - e_{\alpha}^2)e_{\alpha z}. \quad (A.9)$$

### Appendix B. D3Q19 set of microscopic velocities $\mathbf{e}_i$

$$\{\mathbf{e}_i, i = 0, \dots, 18\}$$

$$= \left\{ \begin{pmatrix} 0 & c & -c & 0 & 0 & 0 & 0 & c & -c & c & -c & c & -c & c & -c & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & c & -c & 0 & 0 & 0 & c & -c & -c & c & 0 & 0 & 0 & 0 & c & -c & c & -c \\ 0 & 0 & 0 & 0 & 0 & c & -c & 0 & 0 & 0 & 0 & 0 & c & -c & -c & c & c & -c & -c & c \end{pmatrix} \right\}.$$

### Appendix C. Transformation matrix M

For convenience, the basis vectors are arranged in matrix M which reads:

$$\begin{bmatrix}
 1 \cdot & (1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1) \\
 c^2 \cdot & (-1 & 0 & 0 & 0 & 0 & 0 & 0 & 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1) \\
 c^4 \cdot & (1 & -2 & -2 & -2 & -2 & -2 & -2 & 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1) \\
 c \cdot & (0 & 1 & -1 & 0 & 0 & 0 & 0 & 1 & -1 & 1 & -1 & 1 & -1 & 1 & -1 & 0 & 0) \\
 c^3 \cdot & (0 & -2 & 2 & 0 & 0 & 0 & 0 & 1 & -1 & 1 & -1 & 1 & -1 & 1 & -1 & 0 & 0) \\
 c \cdot & (0 & 0 & 0 & 1 & -1 & 0 & 0 & 1 & -1 & -1 & 1 & 0 & 0 & 0 & 0 & 1 & -1) \\
 c^3 \cdot & (0 & 0 & 0 & -2 & 2 & 0 & 0 & 1 & -1 & -1 & 1 & 0 & 0 & 0 & 0 & 1 & -1) \\
 c \cdot & (0 & 0 & 0 & 0 & 0 & 1 & -1 & 0 & 0 & 0 & 0 & 1 & -1 & -1 & 1 & 1 & -1) \\
 c^3 \cdot & (0 & 0 & 0 & 0 & 0 & -2 & 2 & 0 & 0 & 0 & 0 & 1 & -1 & -1 & 1 & 1 & -1) \\
 c^2 \cdot & (0 & 2 & 2 & -1 & -1 & -1 & -1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 & -2 & -2 & -2) \\
 c^4 \cdot & (0 & -2 & -2 & 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 & -2 & -2 & -2) \\
 c^2 \cdot & (0 & 0 & 0 & 1 & 1 & -1 & -1 & 1 & 1 & 1 & 1 & -1 & -1 & -1 & -1 & 0 & 0) \\
 c^4 \cdot & (0 & 0 & 0 & -1 & -1 & 1 & 1 & 1 & 1 & 1 & 1 & -1 & -1 & -1 & -1 & 0 & 0) \\
 c^2 \cdot & (0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 & 1 & -1 & -1 & 0 & 0 & 0 & 0 & 0 & 0) \\
 c^2 \cdot & (0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 & -1) \\
 c^2 \cdot & (0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 & 1 & -1 & -1 & 0 & 0 & 0) \\
 c^3 \cdot & (0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 & -1 & 1 & -1 & -1 & 1 & -1 & 0 & 0 & 0) \\
 c^3 \cdot & (0 & 0 & 0 & 0 & 0 & 0 & 0 & -1 & 1 & 1 & -1 & 0 & 0 & 0 & 1 & -1 & 1) \\
 c^3 \cdot & (0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 & -1 & -1 & 1 & -1 & 1)
 \end{bmatrix}$$

The transformation of the nodal momentum distribution function to its corresponding moments is obtained by  $\mathbf{m} = \mathbf{M}\mathbf{f}$ . The multiple-relaxation-time collision operator is applied in moment space according to Eq. (2).

## References

- [1] A.J.C. Ladd, Numerical simulations of particulate suspensions via a discretized Boltzmann equation, *Journal of Fluid Mechanics* 271 (1994) 285–339.
- [2] X. Shan, H. Chen, Lattice Boltzmann model for simulating flows with multiple phases and components, *Physical Review E* 47 (1993) 1815–1819.
- [3] B. Ahrenholz, J. Tölke, M. Krafczyk, Second-order accurate Lattice Boltzmann flow simulations in reconstructed porous media, *International Journal of Computational Fluid Dynamics* 20 (6) (2006) 369–377.
- [4] S. Geller, M. Krafczyk, J. Tölke, S. Turek, J. Hron, Benchmark computations based on Lattice-Boltzmann, finite element and finite volume methods for laminar flows, *Computers & Fluids* 35 (2006) 888–897.
- [5] C. Teixeira, Incorporating Turbulence Models into the Lattice-Boltzmann method, *International Journal of Modern Physics C* 9(8) (1998) 1159–1175.
- [6] M. Krafczyk, J. Tölke, L.-S. Luo, Large-Eddy simulations with a multiple-relaxation LBE model, *International Journal of Modern Physics B* 17 (2003) 33–39.
- [7] S. Radhakrishnan, U. Piomelli, A. Keating, A. Lopes, Reynolds-averaged and large-eddy simulations of turbulent non-equilibrium flows, *Journal of Turbulence* 7 (2006) N63.
- [8] Y.H. Qian, D. d'Humières, P. Lallemand, Lattice BGK models for Navier–Stokes equation, *Europhysics Letters* 17 (6) (1992) 479–484.
- [9] D. d'Humières, I. Ginzburg, M. Krafczyk, P. Lallemand, L.-S. Luo, Multiple-relaxation-time lattice Boltzmann models in three dimensions, *Royal Society of London Philosophical Transactions Series A* 360 (2002) 437–451.
- [10] J. Tölke, S. Freudiger, M. Krafczyk, An adaptive scheme using hierarchical grids for Lattice Boltzmann fluid flow simulations, *Computers & Fluids* 35 (2006) 820–830.
- [11] P. Lallemand, L.-S. Luo, Theory of the lattice Boltzmann method: dispersion, dissipation, isotropy, Galilean invariance, and stability, *Physical Review E* 61 (6) (2000) 6546–6562.
- [12] S. Hou, J. Sterling, S. Chen, G.D. Doolen, A Lattice Boltzmann subgrid model for high Reynolds number flows, *Contributions to Mineralogy and Petrology* (1994) 1004.
- [13] M. Rheinländer, A consistent grid coupling method for Lattice-Boltzmann schemes, *Journal of Statistical Physics* 121 (1–2) (2005) 49–74.
- [14] B. Crouse, E. Rank, M. Krafczyk, J. Tölke, A LB-based approach for adaptive flow simulations, *International Journal of Modern Physics B* 17 (2003) 109–112.
- [15] S. Freudiger, Entwicklung eines parallelen, adaptiven, komponentenbasierten Strömungskerns für hierarchische Gitter auf basis des Lattice-Boltzmann-Verfahrens, Ph.D. Thesis, Institut für Rechnergestützte Modellierung im Bauingenieurwesen, Technische Universität Braunschweig, 2009.
- [16] O. Filippova, D. Hänel, Boundary-fitting and local grid refinement for Lattice-BGK models, *International Journal of Modern Physics C* 9 (1998) 1271–1279.
- [17] D. Yu, R. Mei, W. Shyy, A multi-block Lattice Boltzmann method for viscous fluid flows, *International Journal for Numerical Methods in Fluids* 39 (2) (2002) 99–120.
- [18] M. Weickert, G. Teike, O. Schmidt, M. Sommerfeld, Investigation of the LES WALE turbulence model within the Lattice Boltzmann framework, *Computers & Mathematics*, doi:10.1016/j.camwa.2009.08.060.
- [19] M. Bouzidi, M. Firdaouss, P. Lallemand, Momentum transfer of a Lattice-Boltzmann fluid with boundaries, *Physics of Fluids* 13 (2001) 3452–3459.
- [20] R. Mei, D. Yu, W. Shyy, L.-S. Luo, Force evaluation in the Lattice Boltzmann method involving curved geometry, *Physical Review E* 65 (4) (2002) 041203.
- [21] R. Clift, J. Grace, M. Weber, *Bubbles, Drops and Particles*, Academic Press, San Diego, 1978.
- [22] G. Karypis, V. Kumar, METIS—a software package for partitioning unstructured graphs, partitioning meshes, and computing fill-reducing orderings of sparse matrices-version 4.0, 1998. Website: <http://glaros.dtc.umn.edu/gkhome/views/metis>.
- [23] E. Van Driest, On turbulent flow near a wall, *Journal of Aerospace Science and Technologies* 23 (6) (1956) 1007–1011.
- [24] G. Constantinescu, K. Squires, LES and DES investigations of turbulent flow over a sphere at Re 10 000, *Flow Turbulence and Combustion* 70 (1–4) (2003) 267–298.
- [25] E. Achenbach, Experiments on the flow past spheres at very high Reynolds numbers, *Journal of Fluid Mechanics* 54 (3) (1972) 565–575.
- [26] V. Bakic, Experimental investigation of turbulent flows around a sphere, Ph.D. Thesis, TUHH Hamburg, 2002.
- [27] F. Nicoud, F. Ducros, Subgrid-scale stress modelling based on the square of the velocity gradient tensor, *Flow, Turbulence and Combustion* 62 (2) (1999) 183–200.