INTERNATIONAL JOURNAL FOR NUMERICAL METHODS IN FLUIDS
Int. J. Numer. Meth. Fluids 2006; **51**:439–468
Published online 20 December 2005 in Wiley InterScience (www.interscience.wiley.com). DOI: 10.1002/fld.1140

A generic, mass conservative local grid refinement technique for lattice-Boltzmann schemes

M. Rohde[1,2,*,†], D. Kandhai[1], J. J. Derksen[1] and H. E. A. van den Akker[1]

[1]*Kramers Laboratorium voor Fysische Technologie, Technische Universiteit Delft, Prins Bernhardlaan 6, 2628 BW Delft, The Netherlands*
[2]*Radiation, Radionuclides and Reactors Department, Technische Universiteit Delft, Mekelweg 15, 2629 JB Delft, The Netherlands*

SUMMARY

A generic, mass conservative local grid refinement technique for the lattice-Boltzmann method (LBM) is proposed. As a volumetric description of the lattice-Boltzmann equation is applied, mass conservation can be imposed by allowing the lattice-Boltzmann particles to move from coarse grid cells to fine grid cells and vice versa in the propagation step. In contrast to most existing techniques, no spatial and temporal interpolation of particle densities is applied. Moreover, since the communication between the coarse and the fine grids is independent on the collision step, the method can be used for any LBM scheme.

It was found that the method is second-order accurate in space for 2-D Poiseuille flow and different grid setups. The method was also applied to the case of 2-D lid driven cavity flow at $Re = 1000$, where half of the cavity was locally refined. It was found that the locations of the two lower vortices could be captured accurately. Finally, a direct numerical simulation (DNS) of turbulent channel flow at $Re_\tau = 360$ was performed where the grid was locally refined near the walls of the channel. Good first- and second-order turbulence statistics were obtained, showing the applicability of the local grid refinement technique for complex flows. Copyright © 2005 John Wiley & Sons, Ltd.

KEY WORDS: lattice-Boltzmann method; local grid refinement; turbulence

1. INTRODUCTION

Many systems in nature and engineering involving transport processes can be characterized by a broad spectrum of length and time scales; the most notable example being turbulence. The smallest scales (i.e. Kolmogorov scales) of a turbulent flow are determined by the local flow conditions and may vary significantly in size from position to position. In computational

*Correspondence to: M. Rohde, Radiation, Radionuclides and Reactors Department, Technische Universiteit Delft, Mekelweg 15, 2629 JB, Delft, The Netherlands.
†E-mail: m.rohde@tnw.tudelft.nl

Received 11 April 2005
Revised 30 September 2005
Accepted 18 October 2005

fluid dynamics (CFD), flow inhomogeneity is effectively accounted for by adapting the grid in accordance with the local flow conditions.

In conventional CFD, based on, e.g. finite-volume and finite-difference methods, numerous techniques for adapting the grid to the local flow conditions (i.e. *local grid refinement*) were developed and investigated during the past decades. On the other hand, for LBM, such developments were initiated only some years ago. These techniques can be subdivided into several categories, such as 'finite-difference lattice-Boltzmann method (FDLBM)' [1, 2], 'finite-volume LBM (FVLBM)' [3–5], 'interpolation supplemented LBM (ISLBM)' [6, 7] and so-called locally embedded uniform grid techniques [8–12].

In this paper, a generic, mass conservative technique in the category of locally embedded uniform grids will be described. A number of benchmark cases (ranging from 2-D laminar channel flow to turbulent channel flow) will be used to show that no interpolation and rescaling of particle densities is needed to obtain accurate results. Since no rescaling of particle densities is applied, the technique can be used for any collision operator.

1.1. Locally embedded uniform grids

The locally embedded uniform grid techniques differ from other existing techniques in the sense that the refined uniform grids and the main coarse grid live on different space and time scales. These scales are connected through the level of refinement (i.e. $\Delta x_c = n\Delta x_f$ and $\Delta t_c = n\Delta t_f$, where n denotes the level of refinement, the subscript c refers to the coarse, and f to the fine grid), so that the lattice velocities $\mathbf{c}_i$ are equal for all grids. The consequence is that one needs to perform less time steps on the coarse grid than on the fine grid, because also time is refined *locally*. It can be shown that the reduction in computational effort is $(G_c + nG_f)/(n^4G_c + nG_f)$, where G denotes the number of grid cells on a grid. Another consequence is that the relaxation parameter of the LBGK scheme has to be adjusted, such that the viscosities (in physical units) on both grids are equal; since $v_c = (1/n)v_f$, the relaxation parameters on both grids are related by $\tau_c = (\tau_f - \frac{1}{2})/n + \frac{1}{2}$. The velocities and gravitational acceleration scale according to (in lattice-Boltzmann units!) $u_c = u_f$ and $g_c = ng_f$. A typical grid with its time and space discretization is shown in Figure 1.

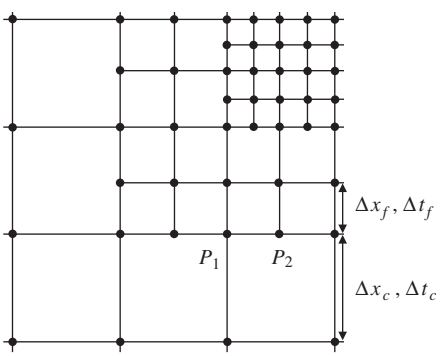

Figure 1. A typical computational grid, consisting of locally embedded uniform grids.

Both the collision and propagation step take place on the fine and coarse grid, following the standard LBM for uniform cubic grids. The question now arises what particle distributions n_i reside on the grid nodes shared by both grids. More specifically, what are the incoming particle distributions travelling from one grid to the other grid? A lot of pioneering work in the field of locally embedded uniform grids has been performed by Filippova and Hänel [8, 9]. They adjusted the collision step on the shared grid nodes by correcting the non-equilibrium part of the particle distribution n_i^{neq} for the different time discretization and relaxation parameter on both grids. Accurate results have been obtained for cases such as two-dimensional flow around an airfoil at $Re = 1.5 \times 10^6$ [13]. Lin and Lai [10] considered the communication step between both grids after the propagation step instead of the collision step. They assumed the particle distributions on the shared grid nodes to be transferable between both grids, without any rescaling of the non-equilibrium part of the particle distribution. Hence, $n_i(\mathbf{x}_c) = n_i(\mathbf{x}_f)$. This assumption, however, is not entirely correct, because the non-equilibrium part n_i^{neq} depends on the relaxation parameter (in the case of the LBGK scheme) and the time step,

$$n_i^{\text{neq}} \sim \Delta t \cdot \tau((\mathbf{c}_i \cdot \nabla)(\mathbf{c}_i \cdot \mathbf{u}) - c_s^2 \nabla \rho \mathbf{u})$$

Without rescaling the non-equilibrium part, an error is introduced in the local components of the stress tensor. Simulations show that without rescaling, indeed a significant error is introduced [11]. Dupuis and Chopard [11] therefore apply such a rescaling, but, in contrast to Filippova and Hänel, after the propagation step. This results in a technique that resembles the method of Filippova and Hänel, but is simpler to implement and contains no singularity for $\tau = 1$.

1.2. Drawbacks of existing techniques

Originally, the LBM scheme is a finite-difference scheme. This implies that conservation of mass cannot be easily imposed when particle distributions need to be interpolated (which might be necessary for applying accurate boundary conditions or local grid refinement techniques). The same issue becomes relevant when locally embedded grids are considered. Filippova and Hänel [8, 9], Lin and Lai [10] and Dupuis and Chopard [11] apply first- and second-order interpolation in space and time for properly transferring particle densities from one grid to the other and for keeping the overall accuracy of the simulation second order.

Another issue relates to the fact that most existing techniques use the rescaling of the non-equilibrium distribution. A consequence of such a rescaling step is that these techniques are restricted to a specific collision operator. Since a range of flow problems require more sophisticated lattice-Boltzmann schemes (such as the multiple relaxation time lattice Boltzmann model [14] and the scheme as proposed in References [15, 16]), a scheme-independent grid refinement technique would be preferable.

2. AN ALTERNATIVE SCHEME FOR LOCALLY EMBEDDED GRIDS

2.1. Methodology

One way to impose conservation of mass is by using a volumetric description of the transport and behaviour of the lattice-Boltzmann particles (i.e. the lattice-Boltzmann equation). An example of such a grid, with each embedded grid consisting of grid *cells* rather than grid

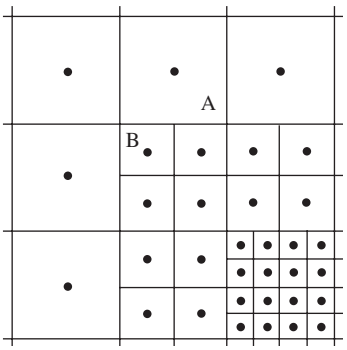

Figure 2. A set of locally embedded grids, consisting of grid cells. A represents a coarse cell, B a fine cell. The level of refinement reads $n = 2$.

nodes, is shown in Figure 2. The volumetric information of the different grids can now be used to ensure mass conservation. In fact, we can use the rescaling of the non-equilibrium distribution as described by Dupuis and Chopard [11], use second-order interpolation in time and a mass-conservative second-order interpolation technique in space to obtain second-order accuracy and mass conservation. In this section, however, a different approach is proposed. The idea of using the volumetric description of the grid for mass conservation is still applied, but no interpolation technique or rescaling of the non-equilibrium distribution is applied.

The alternative method consists of several steps, which are described below and are schematically shown in Figure 3. Only cells A and B in Figure 2 and one velocity direction and its corresponding opposite direction (denoted by $\downarrow$ and $\uparrow$) are considered for clarity.

The steps of the method are as follows:

Step 1: *Collision step on coarse*(A) *and fine*(B) *grid cells.* Perform the collision step and apply a body force, if any, on the coarse and fine grid cells. Note that the collision operator Ω_i and the body force are grid dependent. Formally,

$$n_i(\mathbf{x}, t_*) = n_i(\mathbf{x}, t) + \Omega_i(\mathbf{N}) + t_{\mathrm{f},\ i}\rho(\mathbf{c}_i \cdot \mathbf{G})$$

where t_* denotes the moment after the collision and the last term represents the addition of momentum to the fluid due to the presence of a body force $\mathbf{G}$.

Step 2: *Homogeneous redistribution of particle densities from coarse to fine grid cells.* Redistribute the *mass* distribution $N_i(\mathbf{x}, t_*)$, which is homogeneously distributed in a coarse cell, into $n^{\mathscr{D}}$ smaller cubic cells ($\mathscr{D}$ denotes the spatial dimension), following

$$(N_i(\mathbf{x}_p, t_*))_{\mathrm{f}} = \frac{1}{n^{\mathscr{D}}}(N_i(\mathbf{x}_{\mathrm{c}}, t_*))_{\mathrm{c}} \quad \text{with} \quad \begin{cases} p = 1 \ldots n^{\mathscr{D}} \\ \mathbf{x}_p = \mathbf{x}_{\mathrm{c}} \pm \left(\dfrac{1+2k}{2n}\right)\Delta\mathbf{x}_{\mathrm{c}} \\ \qquad \text{with } k = 0 \cdots \dfrac{n}{2} - 1 \end{cases} \tag{1}$$

where $N_i = \Delta V n_i$ and ΔV is the volume of a grid cell. Note that, according to Figure 3, only one layer of coarse grid cells needs to be redistributed. Equation (1) implies that the particle *densities* in the smaller cubic cells are equal to the original particle density in the coarse

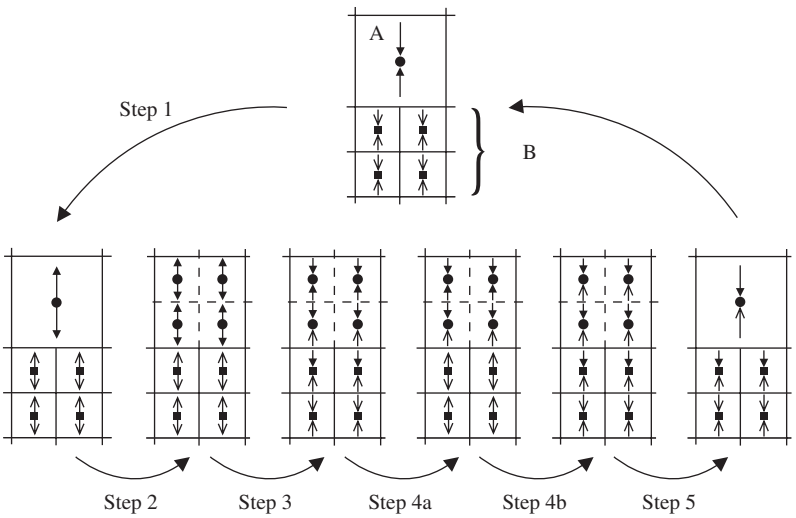

Figure 3. Schematic representation of our locally embedded grid refinement technique. The cell marked with 'A' represents a coarse grid cell, the cells marked with 'B' are fine grid cells. The circles (•) denote the centre of the cells in the coarse grid, the squares (■) the centre of the cells in the fine grid. The open arrow heads (∧ and ∨) correspond to particle densities originating from the fine grid, the closed arrow heads (▲ and ▼) to particle densities that originate from coarse grid cells. Particle distributions after a propagation step are represented by arrows pointing *towards* the cell centre, particle distributions after a collision step are denoted by arrows pointing *from* the cell centre.

cell, hence $(n_i)_f = (n_i)_c$. It is important to note that no spatial interpolation and no rescaling of particle densities take place. This contrasts to the methods of Filippova and Hänel [8, 9] and Dupuis and Chopard [11], where second-order interpolations in space are applied and where the non-equilibrium distributions are rescaled when particle densities are interchanged between grids of different levels of refinement.

Step 3: *Propagation step on the coarse and fine grid.* Perform the propagation step on the coarse and the fine grid.[‡] This step actually represents a communication step between both grids. It should be stressed that particles originating from the coarse grid remain unchanged and simply propagate to the fine grid, though a collision operator corresponding to the coarse grid was applied on these particles before. The numerical implications and the effect on the simulation results will be discussed in more detail in the Section 3.

Step 4: *Repeat steps* 4*a and* 4*b* $n-1$ *times.*

Step 4*a*: *Collision step on the fine grid.* Apply the collision step *on the fine grid only*. Two issues should be mentioned here: (i) Particles that propagate from the fine into the coarse grid (i.e. cross the interface that separates the fine and the coarse grid[§]) will not take part in any collision step again. (ii) In most methods that relate to locally embedded uniform grids, the particle densities on $\tilde{t} = \tilde{t}_0 + \Delta\tilde{t}_f$, located on shared grid nodes such as nodes P_1 and P_2

[‡]i.e. on the fine grid cells located on the fine and coarse grid.
[§]From now on, the interface between the fine grid and the coarse grid is denoted as the *grid transition interface*.

in Figure 1, are obtained by interpolating the rescaled coarse distributions on $\tilde{t}_0$ and $\tilde{t}_0 + \Delta\tilde{t}_c$. No temporal interpolation of particle densities is applied in the method presented here.

Step 4*b*: *Propagation step on the coarse and fine grid.* Perform the propagation step on both grids. This step is methodologically similar to step 3.

Step 5: *Homogeneous redistribution of particle densities from fine to coarse grid cells.* Sum the masses of the $n^{\mathscr{D}}$ small cubic cells for all velocity directions $\mathbf{c}_i$ pointing to wards the coarse grid, resulting in the new incoming particle distributions for the coarse grid. Formally,

$$(N_i(\mathbf{x},t))_c = \sum_{p=1}^{n^{\mathscr{D}}} (N_i(\mathbf{x}_p,t))_f$$

or in terms of particle densities

$$(n_i(\mathbf{x},t))_c = \frac{1}{n^{\mathscr{D}}} \sum_{p=1}^{n^{\mathscr{D}}} (n_i(\mathbf{x}_p,t))_f \tag{2}$$

Similar to step 2, no rescaling of particle densities is applied here.

3. VALIDATIONS

With the help of several well-known benchmark cases, the method as described in the previous section will be tested on numerical and physical consistency. These benchmark cases are 2-D channel flow, lid driven cavity flow at $Re = 1000$ and turbulent channel flow at $Re_\tau = 360$.

3.1. 2-D laminar channel flow

We start with one of the simplest benchmark cases, which is 2-D laminar channel flow driven by a constant body force. The direction of the body force (and thus the flow) with respect to the orientation of the grid transition interface will be varied as follows:

- The body force is *parallel* to the grid transition interface.
- The body force is *normal* to the grid transition interface.

Both cases will be studied in the subsequent sections.

3.1.1. Body force parallel to the grid transition interface. First we consider a channel with a coarse grid in the top half of the channel and a fine grid in the bottom part (see Figure 4).

The settings for the density ρ, the height of the channel H, the viscosity ν, the body force g, the maximum velocity at the centreline of the channel $U_{\max}$ and the location of the grid transition interface z_S can be found in Table I. In this case, the position of the grid transition interface is at the centreline of the channel, hence the height (in physical units) of the both grids is equal, i.e. $\tilde{H} = 8\Delta\tilde{x}_f = 4\Delta\tilde{x}_c$. The lattice-Boltzmann scheme used in the simulations is the so-called D3Q19 scheme, which is a single relaxation time scheme [17] with the collision operator:

$$\Omega_i(\mathbf{x},t) = -\frac{1}{\tau}(n_i(\mathbf{x},t) - n_i^{\mathrm{eq}}(\mathbf{x},t))$$

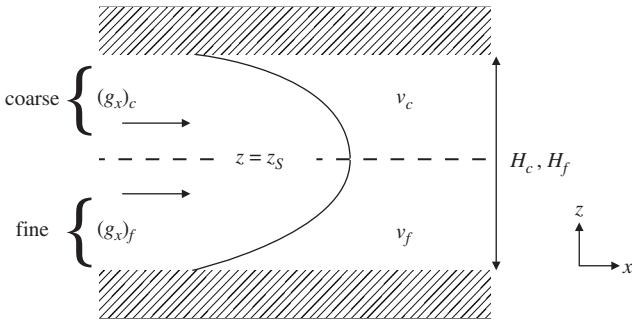

Figure 4. Grid setup for 2-D laminar flow. The direction of the body force is parallel along the grid transition.

Table I. Settings for a 2-D channel flow simulation. The direction of the body force is parallel along the grid transition.

	In coarse lattice units	In fine lattice units
ρ (ls^{-3})	1	1
H (ls)	8	16
v (ls^2.lt^{-1})	$\frac{1}{6}$	$\frac{1}{3}$
g_x (ls.lt^{-2})	$\frac{1}{480}$	$\frac{1}{960}$
$U_{\max}$ (ls.lt^{-1})	0.1	0.1
z_S (ls)	4	8

'ls' and 'lt' refer to the unit for length and time in lattice-Boltzmann schemes, respectively.

The discretized particle velocities $\mathbf{c}_i$ are defined as follows:

$$\mathbf{c}_i = \begin{cases} (0,0,0) & \text{for } i = 0 \\ (\pm 1,0,0),(0,\pm 1,0),(0,0,\pm 1) & \text{for } i = 1 \ldots 6 \\ (0,\pm 1,\pm 1) & \\ (\pm 1,0,\pm 1) & \text{for } i = 7 \ldots 18 \\ (\pm 1,\pm 1,0) & \end{cases}$$

In the LBGK scheme used here, the equilibrium distribution n_i^{eq} is equal to

$$n_i^{\text{eq}}(\mathbf{x},t) = t_{p,i}\rho(\mathbf{x},t)(1 + 3(\mathbf{c}_i \cdot \mathbf{u}) + \tfrac{9}{2}(\mathbf{c}_i \cdot \mathbf{u})^2 - \tfrac{3}{2}\mathbf{u}^2)$$

where $t_{p,i} = \frac{1}{3}$ for $i = 0$, $t_{p,i} = \frac{1}{18}$ for $i = 1 \ldots 6$ and $t_{p,i} = \frac{1}{36}$ for $i = 7 \ldots 18$. The speed of sound for this scheme is equal to $c_s = \sqrt{\frac{1}{3}}$ (ls.lt^{-1}) and the pressure satisfies the equation of state $p = \rho c_s^2$ (ls^{-1}.lt^{-2}). At the top and bottom walls, the no-slip boundary condition is realized by using the halfway bounce-back method.

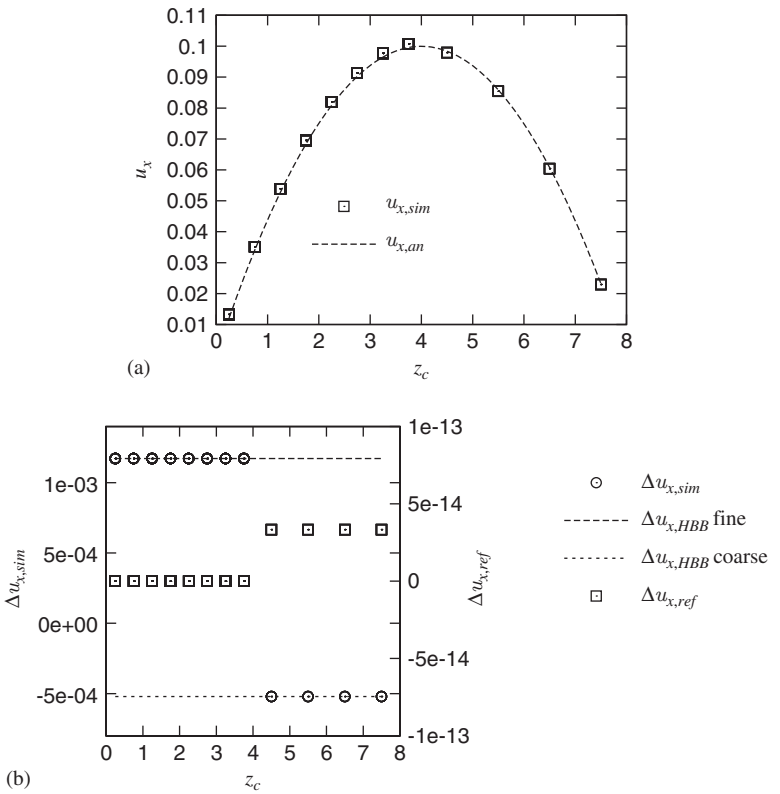

Figure 5. (a) Axial velocity profile; and (b) error profiles in a 2-D channel, according to the numerical setup as shown in Figure 4 and Table I. 'HBB' refers to 'halfway bounce-back'. The definitions, used in the figure, are: $\Delta u_{x,\mathrm{sim}} = u_{x,\mathrm{sim}} - u_{x,\mathrm{an}}$, $\Delta u_{x,\mathrm{ref}} = \Delta u_{x,\mathrm{sim}} - \Delta u_{x,\mathrm{HBB}}$ and $u_{x,\mathrm{an}} = 4U_{\max}H^{-2}z(L-z)$, the latter being the analytical solution of 2-D channel flow. The $\Delta u_{x,\mathrm{HBB}}$ refers to the error as found for simulations performed on a uniform grid (fine and coarse). Note the different scales in panel (b).

The velocity profile $u_{x,\mathrm{sim}}$ of the simulations is shown in Figure 5(a), together with the analytical solution $u_{x,\mathrm{an}}$. In Figure 5(b), the numerical error $\Delta u_{x,\mathrm{sim}}$ is shown. This figure clearly shows that this numerical error fully originates from the halfway bounce-back method [18], so that the error due to the grid refinement technique, $\Delta u_{x,\mathrm{ref}}$, is of the order of machine precision. This result is also found when the grid transition interface is located at positions different from the centreline (not shown in the figure). We therefore conclude that, in this case, the grid refinement technique does not introduce any significant numerical error. A detailed explanation for the absence of an error can be found in Reference [19]. Finally, it is remarkable that the error profiles contain a discontinuity at the grid transition interface, since the transfer of numerical information among grid cells usually induce a more continuous profile.

3.1.2. Body force normal to the grid transition. The second case is similar to the previous one, except that the grid transition interface is normal to the direction of the body force

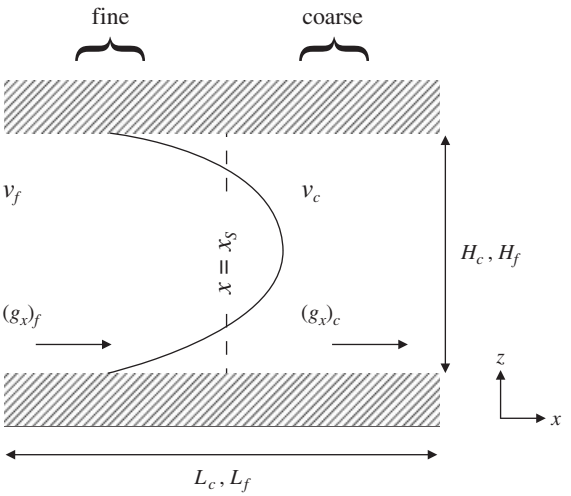

Figure 6. Grid setup for 2-D laminar flow. The direction of the body force is normal to the grid transition interface.

Table II. Settings for a 2-D channel flow simulation. The direction of the body force is normal to the grid transition interface.

	In coarse lattice units	In fine lattice units
ρ (ls^{-3})	1	1
H (ls)	8	16
ν (ls^2.lt^{-1})	$\frac{1}{6}$	$\frac{1}{3}$
L (ls)	100	200
g_x (ls.lt^{-2})	$\frac{1}{480}$	$\frac{1}{960}$
$U_{\max}$ (ls.lt^{-1})	0.1	0.1
x_S (ls)	50	100

(see Figure 6). Consequently, the only gradient present in 2-D channel flow, $\partial u_x/\partial z$, is oriented differently with respect to the grid setup. On top of that, since the grid is not homogeneous in the x direction, an artificial gradient $\partial u_x/\partial x$ may be introduced. Especially close to both grid transition interfaces at $x = 0$ and x_S (note that a periodic boundary is applied at $x = 0$ and L), this artificial gradient may be present in the flow. We therefore choose the length of the channel L large enough, so that this artificial gradient becomes practically zero far away from these grid transition interfaces. We then can compare the velocity profiles both close to and far away from the grid transition interface. The numerical settings of the simulation can be found in Table II.

The results of the simulation are shown in Figures 7(a) and 8(a). We can see in Figure 7(a) that, in contrast to the previous test case, the velocities in the stream- and spanwise directions show a staggered profile with small staggering amplitude. This clearly indicates a numerical artifact of the refinement technique, which can also be seen in the error profiles in

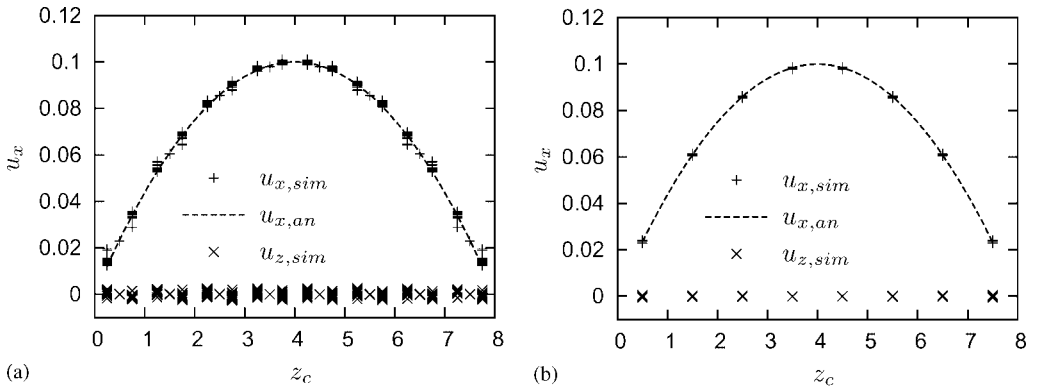

Figure 7. Axial velocity profiles in a 2-D channel, according to the numerical setup as shown in Figure 6 and Table II. The velocity profiles for *all* x-positions are shown here: (a) without post-processing; and (b) after post-processing.

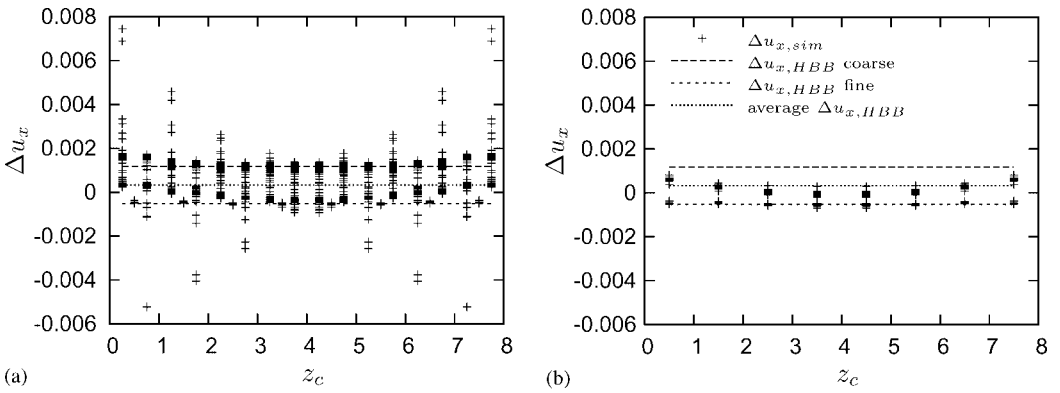

Figure 8. Error profiles with respect to the axial velocity in a 2-D channel, according to the numerical setup as shown in Figure 6 and Table II. 'HBB' refers to 'halfway bounce-back'. The error profiles for *all* x-positions are shown here: (a) without post-processing; and (b) after post-processing.

Figure 8(a). The staggered profiles are also shown in Figure 9, where the velocities in the streamwise direction are plotted as a function of x. It is found that the location of the maxima and minima of the fluctuations is always the same on the coarse time scale (not shown in the figure). The latter figure also indicates that the staggering only occurs on the fine grid, for the particle distribution on the coarse grid is always a time and space average of all incoming particle distributions originating from the fine grid (see Equation (2)).

Staggering is a common phenomenon found in lattice-Boltzmann simulations (see e.g. References [20, 21]) and may be cured by applying an averaging step in time and/or space. Figures 7(b) and 8(b) show that the staggered profiles collapse when a simple post-processing step is applied. This post-processing step, which is schematically shown in Figure 10, only consists of averaging the fine grid particle distributions in space to coarse grid particle

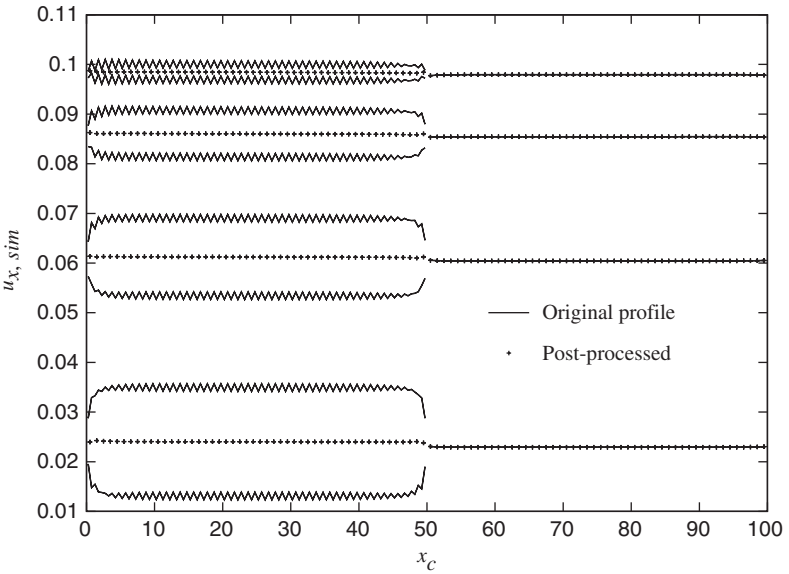

Figure 9. Axial velocities in the streamwise direction along the length of the channel, according to the numerical setup as shown in Figure 6 and Table II. The lines represent the original velocities on the coarse grid ($x_c = 50 \cdots 100$) and the fine grid ($x_c = 0 \cdots 50$), the symbols the velocities after a post-processing step. The original velocities on the fine grid are located on $z_c = 0.25, 0.75, 1.25, \ldots, 7.75$, the velocities on the coarse grid are located on $z_c = 0.5, 1.5, \ldots, 7.5$. Units relate to the *coarse* grid.

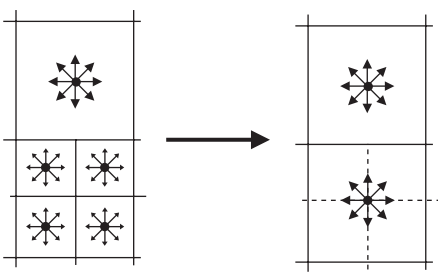

Figure 10. Schematic representation of the post-processing step.

distributions. It should be noted here that the post-processing step is not performed as part of the simulation, but is only applied to the final simulation data.¶

Figure 8(b) shows that the error in the velocity is of the same order as the error caused by the halfway bounce-back method applied here. This suggests that the grid refinement technique is second-order accurate in space for this flow problem. Detailed error analysis reveals that this is indeed the case (data not shown here), in spite of the absence of rescaling, interpolation of particle densities and the presence of a staggered solution on the fine grid.

¶One can perform such a process as part of the simulation, however, it is found that the accuracy is reduced drastically.

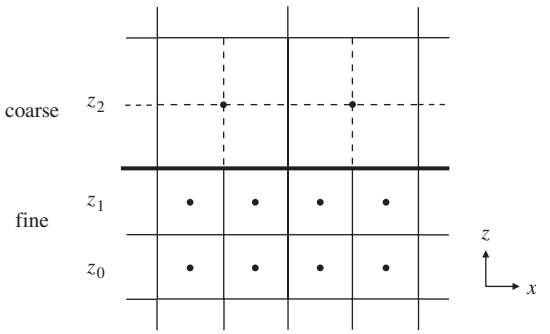

Figure 11. Grid setup for the accelerated, uniform flow field test cases.

3.2. *Uniform flow field accelerated by a body force*

The previous section indicates that the grid refinement technique may add non-physical effects to the solution. In this section, we therefore analyse the grid refinement technique in more detail, with the aim of understanding these unphysical effects better. Two simple cases are considered for this analysis, viz:

- a uniform velocity field U_0 on $t = t_0$, uniformly accelerated with a body force g_x. The body force (and thus the flow) is *parallel* along the grid transition interface;
- a uniform velocity field W_0 on $t = t_0$, accelerated with a body force g_z. The body force is *normal* to the grid transition interface.

Both cases are analysed on a grid as shown in Figure 11.

3.2.1. Body force parallel to the grid transition interface. The velocities on $z = z_0$, z_1 and z_2 can be derived analytically in the case of a D2Q9 scheme after one coarse time step (i.e. on $t_0 + \Delta t_c$), thereby following the steps of the method described in Section 2.1. We may start by applying the equilibrium distribution on $t = t_0$, since no velocity gradients exist in the flow ($n_i^{\text{neq}} = 0$). In the case of a flow parallel along the grid transition interface, the velocities and densities are (in lattice units):

$$\begin{aligned} u(z_0, t + \Delta t_c) &= U_0 + \tfrac{13}{6}(g_x)_f \Delta t_f - \tfrac{5}{108}(g_x)_f / \nu_f \cdot (\Delta x_f)^2 \\ u(z_1, t + \Delta t_c) &= U_0 + 2(g_x)_f \Delta t_f + \tfrac{1}{27}(g_x)_f / \nu_f \cdot (\Delta x_f)^2 \\ u(z_2, t + \Delta t_c) &= U_0 + \tfrac{23}{12}(g_x)_f \Delta t_f + \tfrac{1}{216}(g_x)_f / \nu_f \cdot (\Delta x_f)^2 \\ w(z_{0\ldots2}, t + \Delta t_c) &= 0 \\ \rho(z_{0\ldots2}, t + \Delta t_c) &= \rho_0 \end{aligned} \tag{3}$$

The velocities deviate from the expected velocity with an error $\Delta u(z, t_0 + \Delta t_c) = u(z, t_0 + \Delta t_c) - (U_0 + 2\Delta t_f (g_x)_f)$. Equations (3) apparently show that this error is dependent on the viscosity and the body force. Such a dependency is also found for the velocities in the subsequent time steps, which are obtained by simulations (see Figure 12).

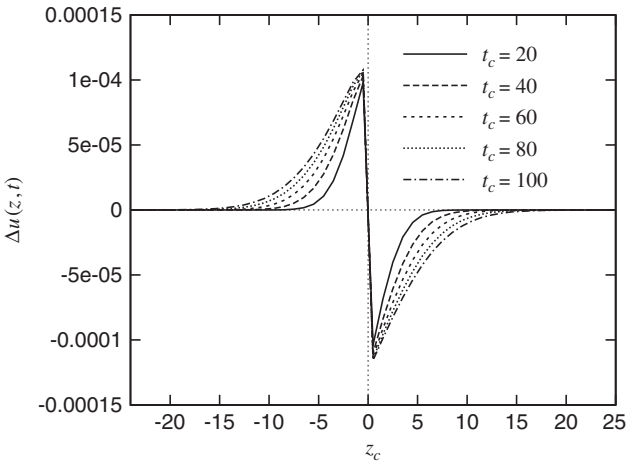

Figure 12. Profiles of the error in the velocity around the grid transition in the case of an accelerated uniform flow field. The negative z values represent the fine grid, the positive values the coarse grid. The viscosity amounts to $\nu_c = \frac{1}{6}$ (ls^2.lt), the body force to $(g_x)_c = 10^{-3}$ (ls.lt^2), and $U_0 = 0$ (ls.lt^{-1}).

Note that momentum is conserved, since $\sum_k \Delta V \rho u = 4\Delta V \rho_0 (U_0 + 2(g_x)_f \Delta t_f)$ and $\sum_k \Delta V \rho w = 0$ (ΔV denotes the volume of a cell, which is always equal to 1 in lattice-units, irrespective of the grid resolution).

It is also found that the average error over z is exactly equal to zero. This might explain the absence of an error in the channel flow case in Figure 5; error profiles are present while the channel flow develops to wards the parabolic profile, but sum to zero by means of the bounce-back rule at the wall (as also the errors are reflected). If we take the error for a specific z into consideration, we then find that the error in the velocity for each value of z develops in time following

$$\Delta u(z,t) = u(z,t) - (U_0 + (t - t_0)g_x) = F(z,t,\nu,\Delta x,\Delta t)g_x$$

which becomes approximately constant after a number of time steps (see Figure 13(a)). Consequently, the *relative* error in the velocity for a specific z, i.e. $\Delta u(z,t)/(U_0 + (t - t_0)g_x)$, decreases asymptotically in time, as denoted in Figure 13(b).

3.2.2. Body force normal to the grid transition interface. The same analysis can be performed for a flow normal to the grid transition interface, though the analytical expressions for the velocities after one single time step are more complex. The absolute error in the velocity and the density after one coarse time step can be expressed as

$$\begin{aligned} \Delta w(x, t + \Delta t_c) &= \frac{\mathscr{P}_1(g_z^3, \nu, \Delta t, \Delta x)}{\mathscr{P}_2(g_z^3, \nu, \Delta t, \Delta x)} g_z \\ \Delta \rho(x, t_0 + \Delta t_c) &= \frac{\mathscr{P}_3(g_z^2, \nu, \Delta t, \Delta x)}{F(g_z, \nu, \Delta t, \Delta x)} \rho_0 g_z \\ \Delta u(x, t_0 + \Delta t_c) &= 0 \end{aligned} \tag{4}$$

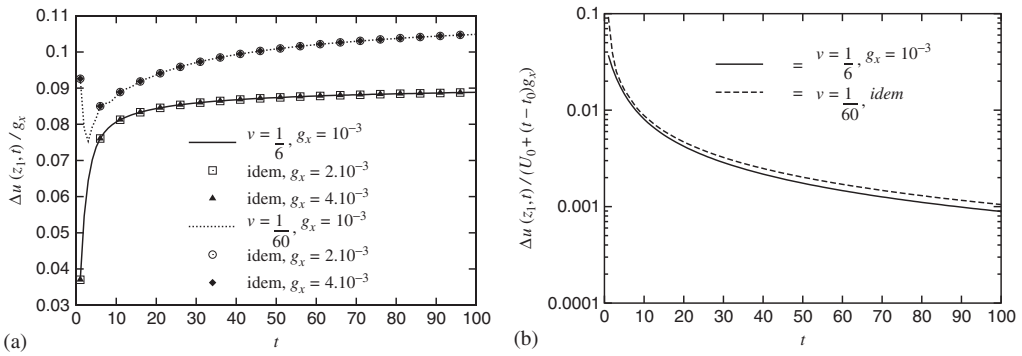

Figure 13. Development of the absolute and relative error in the velocity in time in the case of a uniform, accelerated flow field. The absolute error is divided by the body force, showing the linear relation between both quantities. All quantities are given in *coarse* lattice units: (a) absolute error; and (b) relative error.

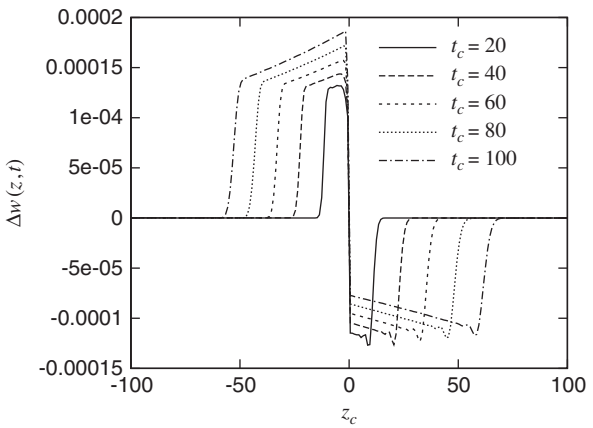

Figure 14. Profiles of the error in the velocity at the grid transition in the case of the accelerated uniform flow field. The negative z values represent the fine grid, the positive values the coarse grid. The post-processing step was applied to the velocities on the fine grid. The viscosity amounts to $\nu_c = \frac{1}{6}$ (ls^2.lt^{-1}), the body force to $(g_z)_c = 10^{-3}$ (ls.lt^2), and $W_0 = 0$ (ls.lt^{-1}).

where $\mathscr{P}_i(g_z^n, \nu, t, \Delta x, \Delta t)$ represents an nth-order polynomial and $F(g_z, \nu, \Delta t, \Delta x)$ a function, both functions of the body force, viscosity, time step size and grid spacing. From Equations (4) it follows that both velocity and density depend on viscosity and body force. Such errors are also found for the subsequent time steps, however, staggered profiles are obtained now. This result was also found in Section 3.1, confirming that a velocity normal to the grid transition interface introduces a staggered solution.

After performing the post-processing step on the fine grid, the error profiles show roughly a similar pattern as the previous test case in the sense that the error changes sign at the grid transition interface (see Figure 14). It should be noted, however, that the transport of both

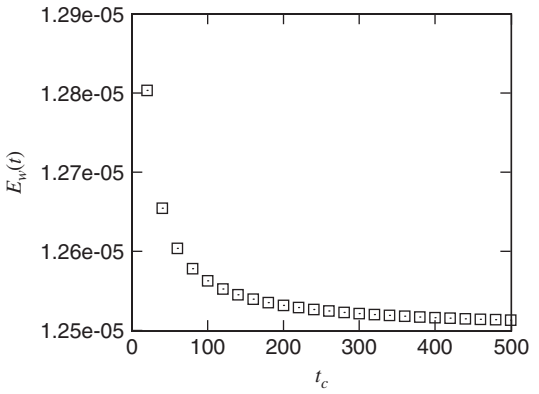

Figure 15. Average relative error around the grid transition in the case of an accelerated uniform flow field. The viscosity amounts to $\nu_c = \frac{1}{6}$ (ls^2.lt^{-1}), the body force to $(g_z)_c = 10^{-4}$ (ls.lt^2), and $W_0 = 0$ (ls.lt^{-1}).

errors is different; in Figure 12, the error seems to diffuse into the domain, while in Figure 14, the error shows to be of an advective nature.

In contrast to the previous case, the average relative error over each profile is not zero, but small and approximately constant. This average relative error is determined by averaging the error profiles from $z_i = -(t-t_0)/\Delta t \ldots (t-t_0)/\Delta t$, denoting the range of $n = 2(t-t_0)/\Delta t$ grid cells that are affected by the grid refinement technique on instant t. This error reads

$$E_w(t) = \frac{1}{n} \sum_{z_i} \frac{w(z_i, t) - (W_0 + (t-t_0)g_z)}{W_0 + (t-t_0)g_z}$$

which is plotted versus time in Figure 15.

It is also found that momentum in the z direction is not *a priori* conserved . From the analysis of the first coarse time step, we can derive that the momentum in the z direction reads (in *fine* lattice units)

$$\sum_k \Delta V \rho w = 4\rho_0(W_0 + 2g_z\Delta t) + \rho_0 \left(\frac{2}{3} \frac{\Delta x \Delta t W_0 g_z}{\nu} - \frac{1}{2} \frac{\Delta t^3 g_z^2}{\Delta x} + \frac{1}{3} \frac{\Delta x \Delta t^2 g_z^2}{\nu} \right)$$

indicating an error with second- and third-order terms, which is also found for the subsequent time steps. For a finite computational domain, however, the error may sum to zero (with machine precision), as shown in the case of a simulation of an accelerated, uniform flow in a computational domain with periodic boundaries (see Figure 16).

3.2.3. Conclusion. It is found that the grid refinement technique introduces a staggered solution with small staggering amplitude on the fine grid for velocities normal to the grid transition interface. The staggering, however, disappears when a simple post-processing step is applied.

The presence of the staggering can be elucidated in a qualitative way; coarse grid particles propagate into the fine grid in step 3 (see Section 2.1), although a collision operator *corresponding to the coarse grid* was applied on these particles before. In other words, we may

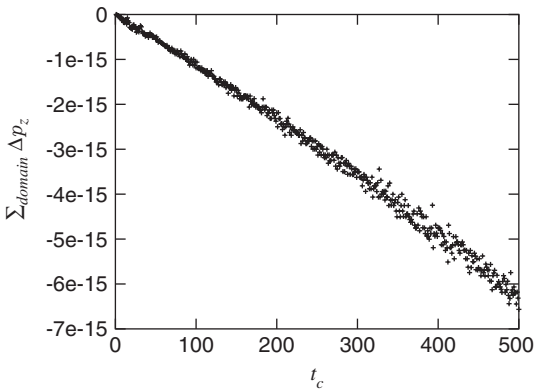

Figure 16. Change of total z-momentum (denoted by $\sum_{\text{domain}} \Delta p_z$) in the case of the accelerated, uniform flow in a computational domain with periodic boundaries. The simulation is performed on a grid of $(H_z)_c = 200$ (ls), the body force applied is $(g_z)_c = 10^{-3}$ (ls.lt^{-2}) and the viscosity amounts to $\nu_c = \frac{1}{6}$ (ls^2.lt^{-1}). Half of the grid is coarse, the other half is twice as fine.

say that these particles carry $t_0 + \Delta t_c$ information based on coarse time and length scales, but are used for the fine grid on the finer time steps $t_0 + \Delta t_f, t_0 + 2\Delta t_f, \ldots$ as well. A similar effect occurs in the case of fine grid particles propagating to the coarse grid, because the method implies that fine grid particles that enter the coarse grid do not collide anymore, but only propagate.

One could argue that the errors can be avoided by rescaling the non-equilibrium distribution. We found, however, that such a rescaling did not remove the errors, but even decreased the accuracy of the method to first order. This seems to be in contradiction to the findings in most references (e.g. in References [4, 8, 9, 11, 12], where the necessity of a rescaled non-equilibrium distribution has been shown for reaching second-order accuracy). Indeed it is shown in the previous sections that our method introduces an error with respect to the non-equilibrium distribution, since the numerical errors, and thus the velocities, depend on the viscosity (which, in the physical sense, is not the case for a uniform flow field). The errors, however, are found to be small. We therefore think that the methodology itself imposes the non-equilibrium distributions to be properly scaled. This issue, however, will be investigated in a more formal way in the near future, thereby following the methodology of Chen *et al.* [12].

A second reason why the average error over the total computational domain is found to be very small or even zero (depending on the orientation of the velocity considered with respect to the grid transition interface), could be the fact that numerical errors ultimately average to very small errors for flow problems with *confined* boundaries. Confined boundaries in lattice-Boltzmann simulations are mostly imposed by techniques that actually reintroduce particle distributions leaving the computational domain (i.e. bounce-back techniques, the free-slip algorithms and periodic boundaries), thereby also reflecting numerical errors. This probably explains why no numerical errors were observed in the case of 2-D channel flow, with velocities parallel to the grid transition interface (see Section 3.1).

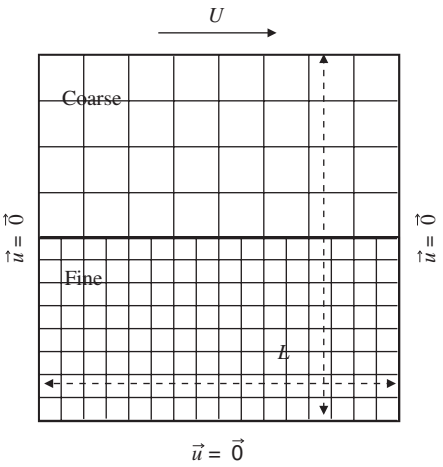

Figure 17. Grid setup for lid driven cavity flow. The *Re* number is defined as $Re = UL/v$.

3.3. *Two-dimensional, lid driven cavity flow*

The second test case is a lid driven cavity flow. This benchmark case consists of a closed, square, two-dimensional channel, with one side moving with a constant speed U (see Figure 17).

3.3.1. Earlier work. This case has been thoroughly studied numerically by several authors, see e.g. Reference [22] and references therein. Hou *et al.* [22] studied lid driven cavity flow at several *Re* numbers ranging from $Re = 100$ to 10^4, thereby using lattice-Boltzmann simulations on a uniform grid. The simulations focused, among others, on the positions of the vortices present in the cavity. The results presented in Reference [22] agreed well with earlier findings where other numerical techniques such as finite-difference and finite element methods were applied. At $Re = 100$, the centre of the primary vortex is located close to the upper right corner of the cavity, but moves closer to the centre when the *Re* number is increased. Next to the large, primary vortex, two other vortices are present in the lower left and lower right corners. At about $Re = 2000$, a fourth vortex starts to develop in the upper left corner. Lin and Lai [10] used the lid driven cavity flow at $Re = 1000$ as a benchmark case for testing their grid refinement technique. They used several locally embedded grids around the two vortices in the lower left and lower right corner and in the top left corner of the cavity, where the velocity gradients are high. Their results agreed quite well with the results of Hou *et al.* [22] and other authors.

3.3.2. Numerical setup. Because of the numerous publications on lid-driven cavity flow, also for lattice-Boltzmann simulations, we consider it as a good benchmark case to test our local grid refinement technique. We therefore perform a simulation at $Re = 1000$, using the same numerical setup as Hou *et al.*, but with a locally refined grid covering the lower half of the cavity (see Figure 17 and Table III).

Table III. Numerical input variables for lid-driven cavity flow at $Re = 1000$.

	In coarse lattice units	In fine lattice units
U (ls.lt^{-1})	0.1	0.1
v (ls^2.lt^{-1})	$\frac{8}{625}$	$\frac{16}{625}$
L (ls)	128	256
Size coarse grid $H_x \times H_y$ (ls^2)	128×64	—
Size fine grid $H_x \times H_y$ (ls^2)	—	256×128

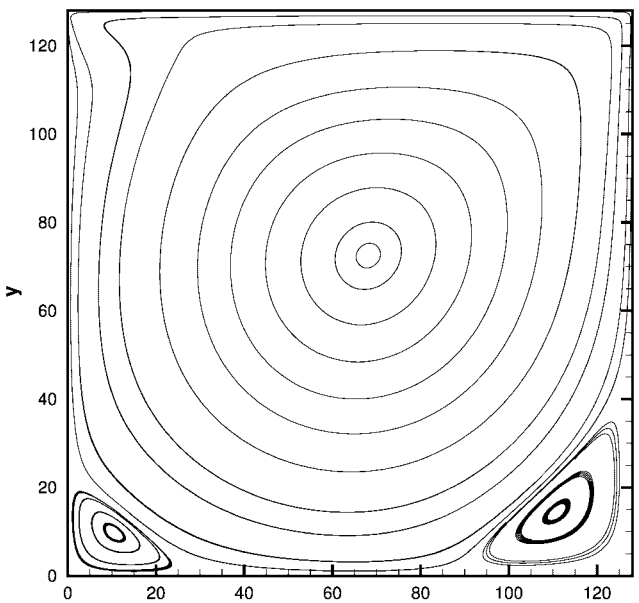

Figure 18. Streamline pattern in the case of a lid-driven cavity after post-processing for $Re = 1000$, following the numerical setup as described by Figure 17 and Table III.

This specific case is also studied by Lin and Lai [10] for their refinement technique, although they use a cavity of size $L_f = 255$ (ls) and a somewhat larger fine grid of size 255×137 (ls^2). The no-slip walls are imposed by the halfway-bounce-back method, and the velocity at the top of the cavity is imposed by adding an additional amount of momentum (viz. mass) to the reflected particles which is proportional to the boundary velocity [20]. Note that there are two singularities at the top corner points of the cavity; on the left and right walls, a zero-velocity is imposed, but at the same time, a certain velocity U is imposed on the same node. Such similarities cause numerical errors, but these errors appear to be small since a large grid is used.

3.3.3. Results. The streamline pattern of the simulation is presented in Figure 18. The post-processing step has been applied to eliminate the staggering (see Figure 19 for a detailed view of the bottom right corner before and after the post-processing step).

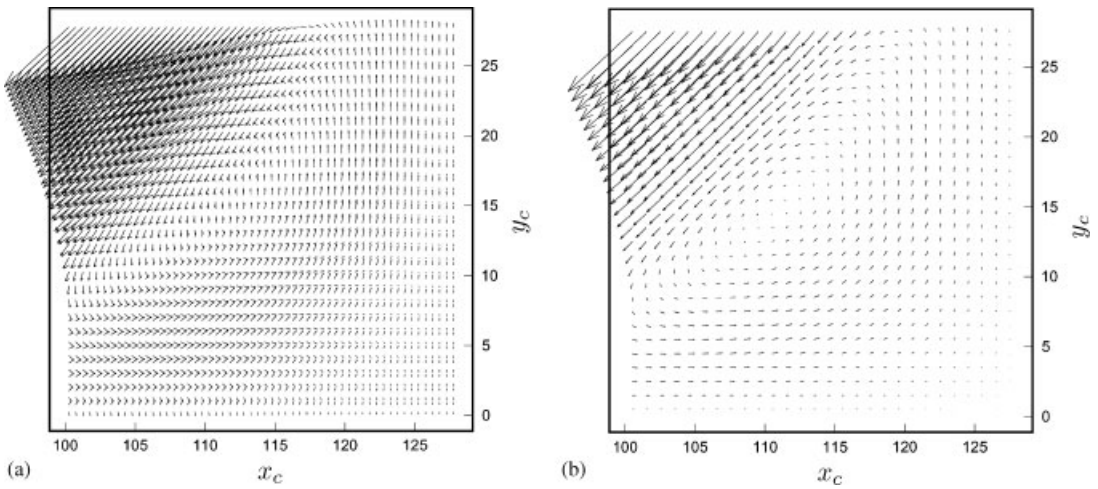

Figure 19. Vector field before and after the post-processing step of the flow in the bottom right corner of a lid driven cavity at $Re = 1000$: (a) before post-processing; and (b) after post-processing.

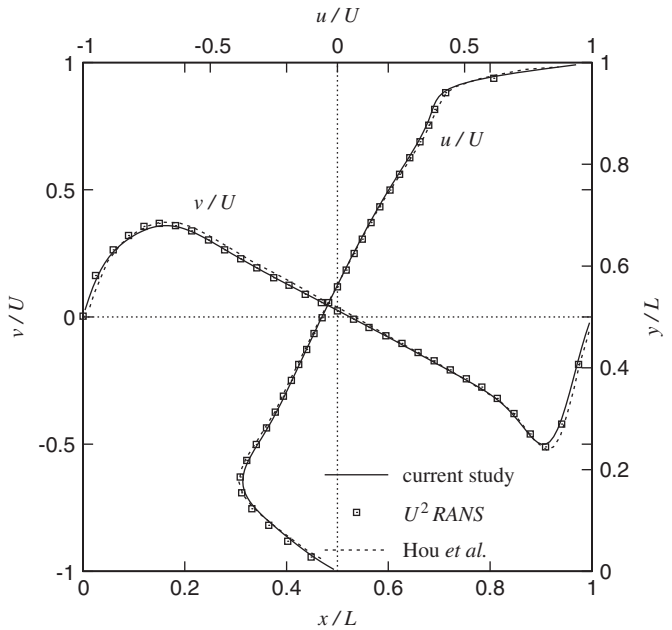

Figure 20. Velocity profiles in the x and y directions at the horizontal and vertical centre-lines of the cavity at $Re = 1000$ (U^2RANS finite-volume results taken from Reference [10]).

The velocity profiles at the horizontal and vertical centre-lines of the cavity are shown in Figure 20. The results from U^2RANS finite-volume calculations [10] and Hou *et al.* [22] are also presented in the figure for comparison. It is found that the results of the present

Table IV. Positions of the three vortices present in lid driven cavity flow at $Re = 1000$, compared with simulations by several other authors. The position is expressed in terms of x/L and y/L.

	x/L	y/L
	Central vortex	
Ghia *et al.*	0.5313	0.5625
Hou *et al.*	0.5333	0.5647
U^2RANS	0.5315	0.5669
Lin and Lai	0.5276	0.5669
Current study	0.5323 ± 0.0008	0.5663 ± 0.0008
	Lower left vortex	
Ghia *et al.*	0.0859	0.0781
Hou *et al.*	0.0902	0.0784
U^2RANS	0.0866	0.0748
Lin and Lai	0.0827	0.0787
Current study	0.0827 ± 0.0008	0.0766 ± 0.0008
	Lower right vortex	
Ghia *et al.*	0.8594	0.1094
Hou *et al.*	0.8667	0.1137
U^2RANS	0.8661	0.1102
Lin and Lai	0.8583	0.1142
Current study	0.8654 ± 0.0008	0.1128 ± 0.0008

study, the results of Hou *et al.* and the U^2RANS calculations agree very well. Small discrepancies between our calculations and those of Hou *et al.* may occur due to the use of different boundary methods; Hou *et al.* used the bounce-back method instead of the halfway bounce-back method for the no-slip walls, although the latter method is known to be more accurate [18]. In addition, Hou *et al.* applied the equilibrium distribution for imposing a velocity $(U, 0, 0)$ at the top of the cavity, which is not entirely correct. Another reason for the small differences might be the grid itself; Hou *et al.* used a uniform grid, consisting of 256×256 (ls^2) grid nodes, whereas in this study, the grid of the upper half of the cavity is twice as coarse. Note that the coarse grid at the upper part of the cavity apparently is sufficient to resolve the flow accurately. Another important finding is the fact that our grid refinement technique does not introduce any unphysical effect in v/U and u/U near the grid transition interface.

We finally compare our results as to the positions of the three vortices with the results of Ghia *et al.* [23], Hou *et al.* [22], and Lin and Lai [10], i.e. the large central vortex and the two smaller vortices at the lower right and left corner of the cavity. In the case of the results of Lin and Lai, only their case *b* is taken into consideration, since case *b* refers to roughly the same grid as in our simulation. The locations of the vortices are determined by examining the stream function and the streamline pattern, and can be found in Table IV. The accuracy of these locations is at least one order of magnitude smaller than the grid spacing, thus $\Delta(x/L) \approx \pm \frac{1}{1280} \approx \pm 0.0008$. It is clear that the results from our simulations agree well with the results of other simulations found in literature.

3.3.4. Conclusion. We conclude that the results of the current study agree very well with results obtained by other simulations found in literature. In spite of the post-processing step required, the grid refinement technique does not introduce significant numerical errors. This may imply that the solution on the fine grid contains accurate information, though this information is present as a staggered solution. Such a result was also found in Section 3.1 for 2-D channel flow normal to the grid transition interface. However, to guarantee that the grid refinement technique produces merely a staggered solution rather than time-dependent hydrodynamic fluctuations, a more severe test has to be performed. Such a test will be described in the next section.

3.4. Turbulent channel flow at $Re_\tau = 360$

The last benchmark case is turbulent channel flow for a in the literature thoroughly studied and well-documented Reynolds number. This three-dimensional flow, which is bounded by two parallel walls of infinite size, is forced by a pressure gradient or body force in the flow direction. The Reynolds number, based on the distance between the walls H_z and the centreline velocity U_c, should be roughly $Re_c > 3000$, in order to go beyond the transition from laminar to turbulent flow. The turbulence of this flow is mostly induced by the wall, where small structures are initiated. As the distance to the walls increases, the structures become larger.

The small-scale structures near the wall significantly affect the flow in the entire channel and must be resolved as accurately as possible. It is therefore necessary to use a fine grid near the wall. On the other hand, a coarser grid is sufficient near the centreline of the channel, where the size of the structures is large and strong hydrodynamical gradients are absent. It would therefore be a waste of computational effort to use a uniform, fine grid for the entire channel.

3.4.1. Earlier work. The first extensive direct numerical simulation (DNS) on turbulent channel flow at $Re_c = 6600$ was performed by Kim *et al.* [24]. This specific case, which can also be defined in terms of the wall shear velocity $Re_\tau = u_\tau H_z/\nu = 360$ with the wall shear velocity being dependent on the shear stress τ_s at the wall via $u_\tau \equiv (\tau_s/\rho)^{1/2}$, is still being used as reliable benchmark material. Kim *et al.* used a grid such that all scales could be resolved without any sub-grid model; grid cells with a small height near the wall and gradually increasing height closer to the centreline of the channel. They used a cosine function for the position of each cell as a function of the coordinate normal to the walls, i.e. $z_j^+ \equiv z u_\tau/\nu = 180(1 - \cos((j-1)\pi/(N-1)))$ (the '+' superscript denotes a quantity in *wall units*), where $N = 129$ is the number of grid cells in the normal direction, $z^+ = 0$ is the bottom of the channel and $z^+ = 360$ the top of the channel. This implies that the first grid cell has a height equal to $\Delta z^+ = 180(1 - \cos(\pi/128)) \approx 0.05$ and the two central grid cells a height $\Delta z^+ = 180 - 180(1 - \cos(63\pi/128)) \approx 4.4$. They also found that the grid size in the stream- and spanwise directions can be $\Delta x^+ \approx 12$(streamwise) and $\Delta y^+ \approx 7$(spanwise), since on average the size of the structures in these directions is much larger than in the normal direction. Another important numerical issue is the size of the computational domain, which is bounded by the no-slip boundary condition at the walls and periodic boundary conditions for the stream- and spanwise directions. Since periodic boundary conditions are used, one should take a sufficiently large domain, such that there is statistically hardly any correlation between two points positioned at a distance equal to half the depth and length of the domain.

Table V. Numerical settings for the turbulent channel flow simulation with the proposed grid refinement technique.

	In coarse lattice units	In fine lattice units
Size of the domain $H_x \times H_y \times H_z$ (ls^3)	$128 \times 64 \times 64$	$256 \times 128 \times 128$
Size of the fine grid, bottom channel (ls^3)	—	$256 \times 128 \times 14$
Size of the fine grid, top channel (ls^3)	—	$256 \times 128 \times 14$
Size of the coarse grid (ls^3)	$128 \times 64 \times 50$	—
Viscosity v ($ls^2.lt^{-1}$)	1.185×10^{-3}	2.370×10^{-3}
Body force g_x ($ls.lt^{-2}$)	1.388×10^{-6}	6.940×10^{-7}
Initial global density ρ_0 (ls^{-3})	1.0	1.0
Wall shear velocity u_τ ($ls.lt^{-1}$)	6.667×10^{-3}	6.667×10^{-3}
Number of time steps simulation $N_{t,\max}$ (lt)	1 200 000	2 400 000

Kim *et al.* therefore used a domain of $192 \times 160 \times 190$ grid cells, which corresponds to a size expressed in wall-units of $H_x^+ \times H_y^+ \times H_z^+ = 2350 \times 1150 \times 360$.

Concerning lattice-Boltzmann simulations of turbulent channel flow, several papers of Amati and co-workers can be found in literature [25–27]. In Reference [25], a large uniform grid of $432 \times 288 \times 144$ (ls^3) nodes was used at a Reynolds number $Re_c \approx 4800$. The lattice-Boltzmann code ran on a computer configuration containing 512 processors. It was found that the mean streamwise velocity in the viscous sublayer and the logarithmic layer could be reproduced very well with the LBM. In Reference [26], a finite-volume grid refinement technique (FVLBM) was applied for several grid lay-outs. One of these simulations roughly resembled the case studied by Kim *et al.* [24], although the grid setup was somewhat different. Fair results were obtained, although a low order interpolation scheme was used so that locality could be maintained. Locality favours parallel computing performance, but decreases the accuracy. In Reference [27], a turbulent channel flow was calculated for $Re_c = 6600$ on a uniform grid consisting of $256 \times 128 \times 128$ grid points. The size of this computational domain was actually too small to capture all turbulent quantities statistically well, but was large enough to obtain the lower-order quantities such as the mean streamwise velocity profile, the root-mean-square velocity fluctuations in the x-, y- and z-directions and the Reynolds shear stress (see Reference [28] for the so-called *minimal channel hypothesis*). For these quantities, good results were reported in Reference [27].

3.4.2. Numerical setup. We now return to our own grid refinement technique, implemented for the turbulent channel flow case at $Re_c = 6600$. For comparison, we perform the same simulations as described by Amati *et al.* [27], thereby making the central region of the channel twice as coarse as their original grid. The mean streamwise velocity profile, the velocity fluctuations and the Reynolds stresses are compared in order to study the effect of the grid refinement technique on the final results. We also compare our results with the results of Kim *et al.* [24], but we emphasize that we do not intend to perform a perfect DNS; our main interest is to investigate the performance of the refinement technique.

The settings for the computation are shown in Table V and Figure 21. The grid consists of three parts, i.e. a fine grid attached to the bottom wall of the channel, a coarse grid in the centre and a fine grid attached to the top wall. The height of both fine grids equals

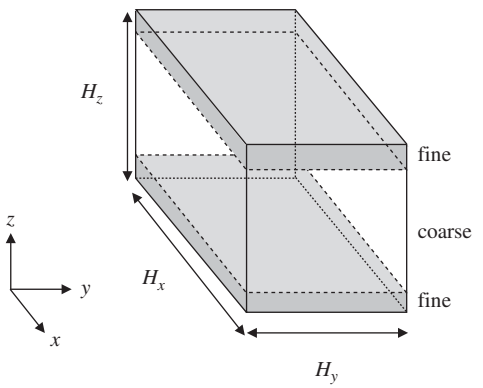

Figure 21. Schematic representation of the computational domain, its sizes, the grid setup and the coordinate system, used for the turbulent channel flow simulation as indicated in Table V.

14 fine grid cells (the grid transition interface is at $z^+ \approx 40$ in wall units), so that the smallest structures are covered by the fine grids. The fine grid cells have a size $\Delta z^+ \approx 2.8$, hence the first grid node is located at position $z^+ \approx 1.4$ since the halfway bounce-back method is used for both no-slip walls.[‖] For $z^+ > 40$, the grid cells have a size of $\Delta z^+ \approx 5.6$, which is slightly too large for resolving the scales near the grid transition interface. From the fact that the mean streamwise velocity profile is logarithmic and the production and dissipation of turbulent kinetic energy being approximately equal, one can derive that the local Kolmogorov length scale in the logarithmic layer follows $\eta_k u_\tau / \nu = (\kappa z^+)^{1/4}$ [29], with $\kappa \approx 0.4$ representing the Von Karman constant. This results in a normalized Kolmogorov length scale of $\eta_k u_\tau / \nu \approx 2$ for $z^+ = 40$, which is too small for the large grid cells to be resolved entirely (as a rule of a thumb, a grid cell can be roughly twice as large as the Kolmogorov length scale). The lack of resolution near the grid transition interface on the coarse grid may therefore cause deviations from the data of Amati *et al.*

The grid refinement technique reduces the total time, necessary for this simulation, by 73% compared to the uniform grid calculations performed by Amati *et al.* [27]. The reduction of required memory is 68%.

The parallel code is run on four cluster nodes, each containing two AMD Athlon 1800+ CPU's and 1 GB of memory. The parallel efficiency is found to be higher than 95%. It takes about 330 h (i.e. 2 weeks) for calculating a time span of $t^+ \equiv N \Delta t u_\tau / H_z = 20$, hence the total simulation (including the startup) takes 1650 h, corresponding to 10 weeks. The minimal memory required amounts to $19 \times (128 \times 64 \times 50 + 2 \times 256 \times 128 \times 14) \times 4 \approx 100$ MB.

The startup of the turbulent channel flow needs some extra attention. The flow field is initiated from zero velocity by developing a Poiseuille flow with a centreline velocity of $U_{\max} = 0.1$ (ls.lt^{-1}), a body force $g_x = 8 U_{\max} \nu / H_z^2$ (ls.lt^{-2}) and preferably a high viscosity, so that the parabolic profile develops quickly. Then, the flow is disturbed on each grid node by superimposing small random fluctuations on the local velocities $\mathbf{u}(\mathbf{x}, t)$ of 10% of the original velocity (this step is only performed during one time step on $t = 0$ (lt)). The viscosity and

[‖] For real DNS, one should use at least three grid cells within the first wall unit.

body force are then set to $\nu = 3.95 \times 10^{-4}$ (ls^2.lt^{-1}) and $g_x = 1.93 \times 10^{-8}$ (ls.lt^{-2}) respectively, and are increased every $t^+ \approx 5$ for a time span of $t^+ = 20$, thereby keeping the Reynolds number, $Re_\tau \equiv u_\tau H_z/\nu = (g_x H_z^3/(2\nu^2))^{1/2} = 360$, constant. This procedure is necessary, since applying a large body force and viscosity at once results in too large fluctuations in the streamwise direction during the startup phase, being up to $\pm 100\%$ of the local velocity. On top of that, we use an alternative lattice-Boltzmann scheme [15, 16] which is more stable than the 19-speed LBGK scheme. After the startup phase, the simulation is continued (with the settings as indicated in Table V and the alternative lattice-Boltzmann scheme) for about $t^+ \approx 20$, which means approximately 400 000 coarse time steps for this specific simulation. The mean streamwise velocity $\langle u_x \rangle$, the root-mean-square velocity fluctuations $\langle u'_\alpha u'_\alpha \rangle^{1/2}$ and the Reynolds shear stress $\langle u'_x u'_z \rangle$, all being a function of z, are monitored until a quasi-steady-state situation has developed. Also the total shear on both walls, i.e. $2\nu/H_z(\sum_{x,y} u_x(x, y, 1)/\Delta z + \sum_{x,y} u_x(x, y, 127)/\Delta z)$, is compared with the applied body force g_x, which should sum to zero in steady-state flow.

After reaching a quasi-steady-state flow, the simulation is run for a dimensionless timespan of $t^+ = 60$. Similar to Amati *et al.*, the 19 speed LBGK scheme is applied here. Samples are taken every $\Delta t^+ = 0.5$, so that an ensemble average can be calculated from $H_x \times H_y \times 60/0.5 \approx 4 \times 10^6$ samples on the fine grid and 10^6 samples on the coarse grid.

Finally, a simulation similar to the latter one is performed, however, the alternative lattice-Boltzmann scheme of Somers [16] is applied now. This simulation may show the applicability of our local grid refinement technique for other lattice-Boltzmann schemes.

3.4.3. Results. A snapshot of the flow in a cross-section of the channel is shown in Figure 22. From this vector field, it is clear that at least no significant unphysical behaviour or discontinuities can be found near the grid transition interface.

The mean streamwise velocity can be found in Figure 23, where the profile is compared with the results of Amati *et al.* [27] and the two models for the viscous sublayer and the logarithmic layer. The DNS results of Kim *et al.* [24] are added to the figure for completeness. The figure shows hardly any difference between the results for a uniform, fine grid (Amati *et al.*) and the results of our simulation on the locally refined grid. This indicates that the grid refinement technique does not significantly influence the mean streamwise velocity profile. On top of this, we have simulated the same channel on a uniform, but *coarse* grid, whereof the results are also shown in Figure 23. These results, compared to those of the locally refined grid, clearly show that the grid refinement technique does produce more accurate hydrodynamics on the finer grid. This is not that obvious, since no interpolations in space and time have been applied for the communication from the coarse to the fine grid.

The root-mean-square velocity fluctuations are shown in Figure 24. This figure shows that our simulations agree fairly well with the results of Amati *et al.* The fluctuations in the streamwise (u_x) direction are almost similar, though we have slightly higher values in the central region of the channel.

The fluctuations in the spanwise (u_y) direction also agree fairly well with Amati *et al.*, however, small dips are observed here near the transition from the coarse to the fine grid at $2z/H \approx 0.22$. It is obvious that these dips are introduced by the grid refinement technique. We think that this unphysical effect might also be present for the streamwise fluctuations, since both u_x and u_y velocities are parallel to the grid transition interface and, as we have seen in

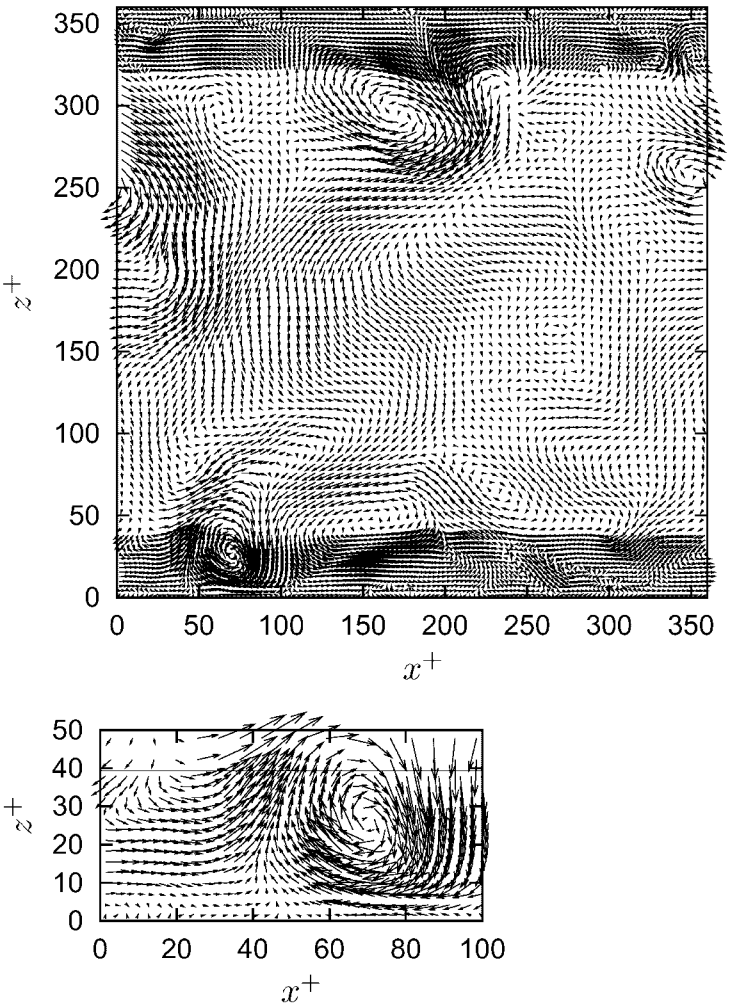

Figure 22. Snapshot of the flow in a cross-section of a turbulent channel. The top figure shows the entire cross-section, the lower figure the bottom left part. The line in the lower figure represents the transition interface between the fine and the coarse grid.

Section 3.1, behave different from the velocity normal to the grid transition interface. However, such deviations cannot be observed in the streamwise profile, probably because such small unphysical effects are obscured by the large gradients in the streamwise fluctuations profile.

The fluctuations in the normal (u_z) direction need special attention here, because these velocities are normal to the grid transition interface. Such velocities show a staggered profile, as we have seen with previous test cases described in this article. Figure 25(a) shows normal velocity profiles u_z versus z for different (x, y) in the flow at $t^+ = 60$, Figure 25(b) represents the normal velocity versus time at one specific position on the fine grid. The amplitude of this staggering is significant compared to the local velocity and the local root-mean-square velocity

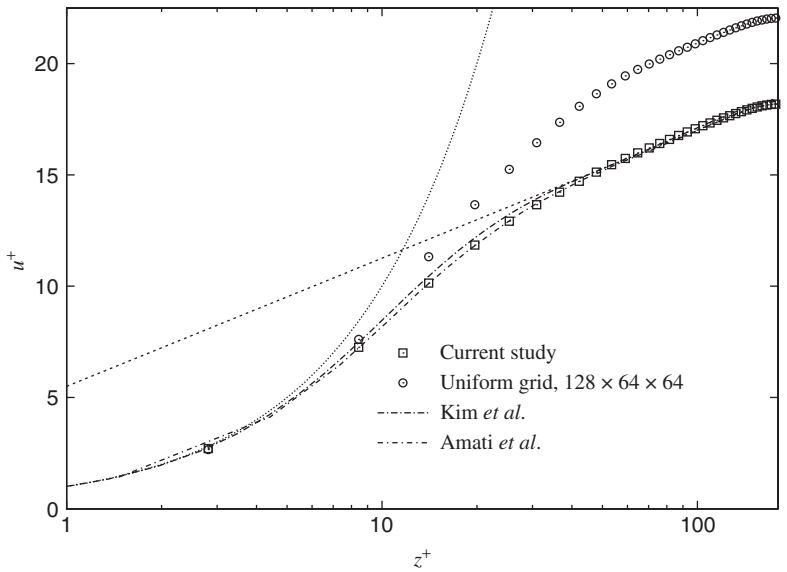

Figure 23. Dimensionless mean streamwise velocity $u^+ = \langle u_x \rangle / u_\tau$ versus the distance from the wall in the case of turbulent channel flow. The grid in the range of $z^+ = 0$–40 is twice as fine as the grid in the centre of the channel. The results of Amati *et al.* [27] and Kim *et al.* [24] are also shown for comparison. The results of a uniform, coarse grid clearly show the effect of refinement near the wall. The dashed lines represent the models for the viscous sublayer and the logarithmic layer.

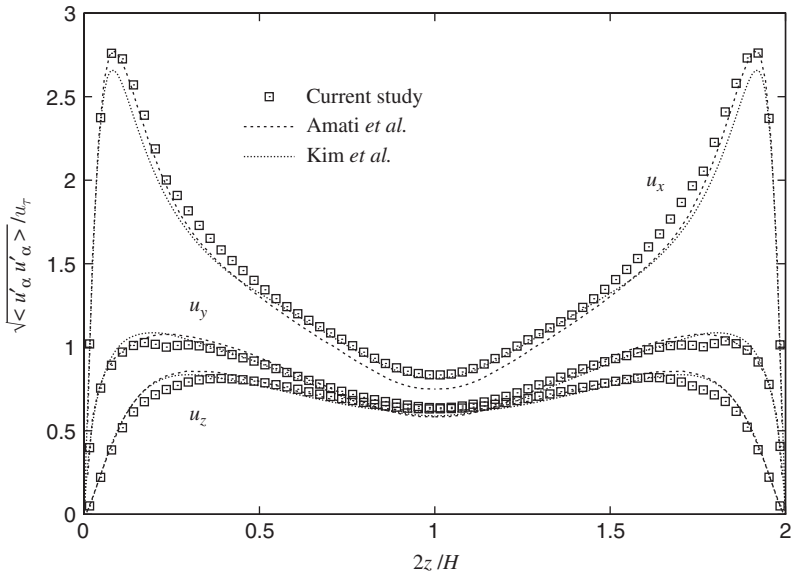

Figure 24. Dimensionless root-mean-square velocity fluctuations $\langle u'_\alpha u'_\alpha \rangle^{1/2} / u_\tau$ for all directions versus the (dimensionless) distance to the wall. The results of Amati *et al.* [27] and Kim *et al.* [24] are also shown for comparison.

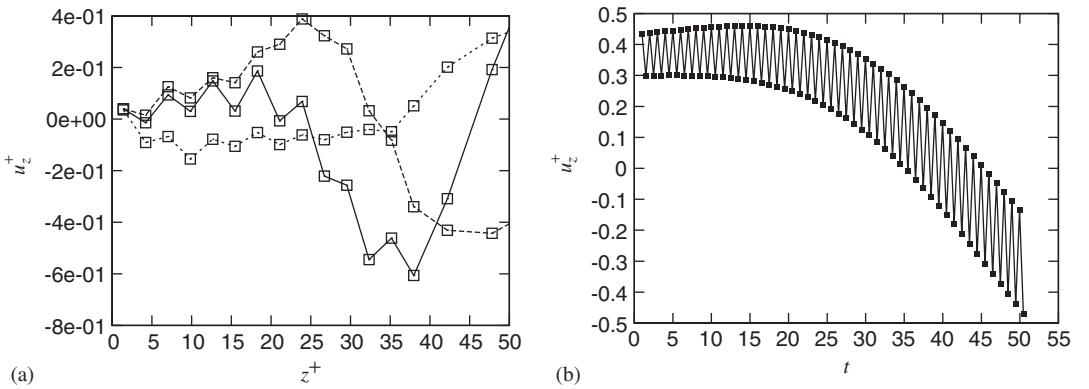

Figure 25. Normal velocities u_z^+ versus z^+ near the bottom wall for three different positions (x, y) in the flow on $t^+ = 60$ (a) and u_z^+ versus time at one specific position on the fine grid (b). The refined grid is located between $z^+ = 0$ and 40.

fluctuations. The influence of this staggering on the turbulence, however, turns out not to be of great importance for the first- and second-order turbulence statistics. In the staggered solution, u_z fluctuates around the desired values with alternating minima and maxima, which change position every time step on the fine grid. The profile for the normal velocity fluctuations exhibits somewhat smaller values near the grid transition interface, which may originate from the grid refinement technique or the lack of resolution in the logarithmic layer for $z^+ > 40$.

The Reynolds stress $\langle u_x' u_z' \rangle$ is shown in Figure 26. It is clear that the simulation results are in good agreement with the result of Amati *et al.* No significant dips are found near the transition between both grids, which is to be expected since the profiles in Figure 24 neither show such dips in the streamwise and normal directions. The total shear stress, which is the sum of the Reynolds stress and the viscous stress, is also plotted in the figure. Its profile should be a straight line and follow the theoretical shear stress profile $\tau_t = -u_\tau^2(1 - 2z/H_z)$, which is to a large extent the case in our simulations.

The simulations by means of the alternative lattice-Boltzmann scheme of Somers [16] instead of the 19-speed LBGK scheme, show small deviations from the results presented above. The largest deviation can be found in the mean streamwise velocity profile, amounting to +3% in the centre part of the channel. The velocity fluctuations for all directions and the Reynolds stress $\langle u_x' u_z' \rangle$ show deviations less than 2%. These small deviations may be caused by: (i) the combination of the lattice-Boltzmann scheme used and our grid refinement technique, as the velocity and density errors are scheme dependent [19]. (ii) the numerical differences between both lattice-Boltzmann schemes. In the current work, however, this issue is not further investigated.

3.4.4. Conclusion. We may conclude that even for a complex flow such as turbulent channel flow, the grid refinement technique proves to be useful. Most hydrodynamical properties studied here are captured quite well, though the technique appears to have some influence on the results. Some deviations, however, might also originate from the lack of resolution near the grid transition interface on the coarse grid, which results in a reduced turbulence intensity.

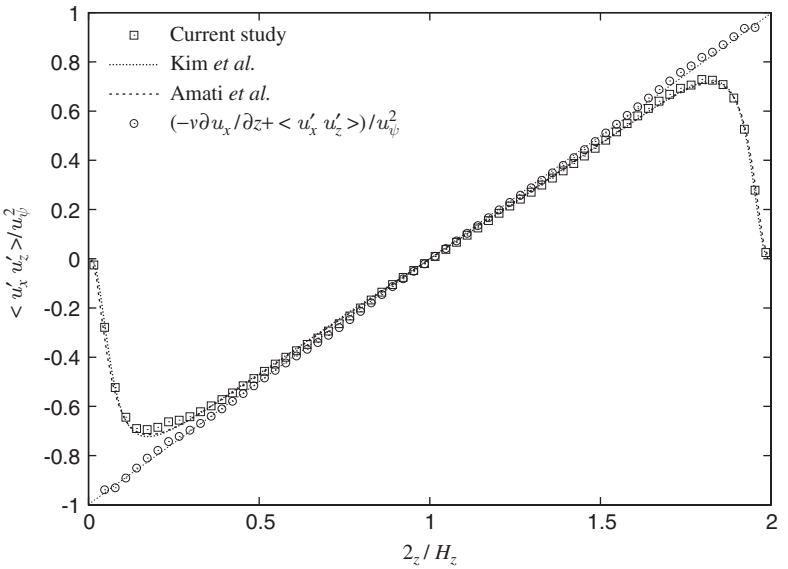

Figure 26. Dimensionless Reynolds stress $\langle u_x' u_z' \rangle / u_\tau^2$ versus the distance from the wall. The results of Amati *et al.* [27] and Kim *et al.* [24] are also shown for comparison, together with the total shear stress and the theoretical shear stress (straight dotted line).

These deviations can be diminished by using embedded grids with more than two levels of refinement, however, we did not perform such a simulation in the present study.

4. CONCLUSIONS AND OUTLOOK

An alternative local grid refinement technique is proposed for lattice-Boltzmann schemes. The technique can be considered as a member of a class of methods that use locally embedded uniform grids. These techniques use patches of refined, uniform grids, superimposed on the main coarse grid. On top of that, time is also discretized such that $\Delta x/\Delta t = 1$ on each grid. In this way, the standard lattice-Boltzmann equation can be used on all grids, since the set of lattice velocities $\mathbf{c}_i$ (both the directions and the size) remains the same. Moreover, such techniques diminish the computational effort effectively, because both time and space are locally refined, which decreases the required computational time even more.

The proposed method is different from other locally embedded uniform grid techniques in the way particle distributions propagate from one grid to another. In most methods, particle distributions are treated as densities residing on grid nodes, and are interpolated (in time and space) and rescaled on the nodes shared by both grids subsequently. In the case of the method described in this paper, the particles are considered as mass, propagating from coarse grid cells to fine grid cells and vice versa. In this way, mass is conserved in a natural way. The method does not contain any interpolation or rescaling of particle distributions, and is therefore simple to implement.

It is shown that the method introduces errors that depend on the orientation of the flow with respect to the orientation of the grid transition interface. These errors depend on the viscosity, applied body force, the grid spacing and the time step. Analysis of these errors, in the case of a uniform flow field, reveals that the errors are small. The errors are found to be largest in the case of a flow perpendicular to this interface, where a staggered solution is obtained on the fine grid. After a post-processing step (i.e. space averaging), however, the results turn out to be accurate. We think that, although no rescaling of the non-equilibrium distribution is performed, such a rescaling takes place by the methodology itself. These thoughts are supported by that fact that, even for a complex, time-dependent flow such as turbulent channel flow, good first- and second-order turbulence statistics are obtained, which only slightly differ from a lattice-Boltzmann simulation obtained on a fine, uniform grid. Especially in the region near the walls of this flow, high velocity gradients can be found which cannot be resolved accurately by a first-order accurate scheme (see e.g. Reference [26]). An extensive formal description, however, is needed and will be derived in the near future.

In general, it can be concluded that the method proves to be a useful tool for different kinds of flow problems where local grid refinement is required. Besides the fact that no spatial interpolation is needed and mass is conserved, the method proves to be very useful when other lattice-Boltzmann schemes than the LBGK schemes have to be used, since no rescaling of the non-equilibrium distribution is required.

It should be realized that the use of cubic grids is not desired in all flow problems. In the case of wall bounded turbulence, such as turbulent channel flow, the resolution in the stream- and spanwise directions can actually be much coarser than in the normal direction; thus a stretched grid would be more appropriate here. However, numerous flow problems exist where the choice for locally embedded grids is most useful, such as the turbulent flow around a cylinder, a building or a revolving impeller in a mixing vessel. We finally want to point out that the method can easily be transformed into a lattice-Boltzmann scheme for mass transport, thereby inheriting its advantage of being mass conservative.

REFERENCES

1. Mei R, Shyy W. On the finite difference-based lattice Boltzmann method in curvilinear coordinates. *Journal of Computational Physics* 1998; **143**:426–448.
2. Kandhai D, Soll W, Chen S, Hoekstra A, Sloot P. Finite-difference lattice-BGK methods on nested grids. *Computer Physics Communications* 2000; **129**:100–109.
3. Nannelli F, Succi S. The lattice-Boltzmann equation in irregular lattices. *Journal of Statistical Physics* 1992; **68**(3/4):401–407.
4. Chen H. Volumetric formulation of the lattice Boltzmann method for fluid dynamics: basic concept. *Physical Review E* 1998; **58**(3):3955–3963.
5. Peng G, Xi H, Duncan C. A finite volume scheme for the lattice Boltzmann method on unstructured meshes. *Physical Review E* 1999; **59**(4):4675–4682.
6. He X, Luo L-S, Dembo M. Some progress in lattice Boltzmann method: part I. Nonuniform mesh grids. *Journal of Computational Physics* 1996; **129**:357–363.
7. He X, Doolen GD. Lattice Boltzmann method on a curvilinear coordinate system: vortex shedding behind a circular cylinder. *Physical Review E* 1997; **56**(1):434–440.
8. Filippova O, Hänel D. Grid refinement for lattice-BGK models. *Journal of Computational Physics* 1998; **147**: 219–228.
9. Filippova O, Hänel D. Acceleration of lattice-BGK schemes with grid refinement. *Journal of Computational Physics* 2000; **165**:407–427.
10. Lin C-L, Lai YG. Lattice Boltzmann method on composite grids. *Physical Review E* 2000; **62**(2):2219–2225.
11. Dupuis A, Chopard B. Theory and applications of an alternative lattice Boltzmann grid refinement algorithm. *Physical Review E* 2003; **67**:066707.

12. Chen H, Filippova O, Hoch J, Molvig K, Shock R, Teixeira X, Zhang R. Grid refinement in lattice Boltzmann methods based on volumetric formulation. *Physica A* 2005, in press.
13. Succi S, Filippova O, Smith G, Kaxiras E. Applying the lattice Boltzmann equation to multiscale fluid problems. *Computational Science and Engineering* 2001; **3**(6):26–37.
14. d'Humières D, Ginzburg I, Krafczyk M, Lallemand P, Luo L-S. Multiple-relaxation-time lattice Boltzmann models in three dimensions. *Philosophical Transactions of the Royal Society*, *Series A* 2002; **360**(1792): 437–451.
15. Eggels JGM, Somers JA. Numerical simulation of free convective flow using the lattice-Boltzmann equation. *International Journal of Heat and Fluid Flow* 1995; **16**:357–364.
16. Somers JA. Direct simulation of fluid flow with cellular automata and the lattice-Boltzmann equation. *Applied Scientific Research* 1993; **51**:127–133.
17. Qian YH, d'Humières D, Lallemand P. Lattice BGK for the Navier–Stokes equation. *Europhysics Letters* 1992; **17**(6):479–484.
18. Rohde M, Kandhai D, Derksen JJ, Van den Akker HEA. Improved bounce-back methods for no-slip walls in lattice-Boltzmann schemes: theory and simulations. *Physical Review E* 2003; **67**:066703.
19. Rohde M. Extending the lattice-Boltzmann method: novel techniques for local grid refinement and boundary conditions. *Ph.D. Thesis*, 2004.
20. Ladd AJC. Numerical simulation of particulate suspensions via a discretized Boltzmann equation. Part I. Theoretical foundation. *Journal of Fluid Mechanics* 1994; **271**:285–309.
21. Manwart C, Aaltosalmi U, Koponen A, Hilfer R, Timonen J. Lattice-Boltzmann and finite-difference simulations for the permeability for three-dimensional porous media. *Physical Review E* 2002; **66**:016702.
22. Hou S, Zou Q, Chen S, Doolen G, Cogley AC. Simulation of cavity flow by the lattice Boltzmann method. *Journal of Computational Physics* 1995; **118**(2):329–347.
23. Ghia U, Ghia KN, Shin CT. High-Re solutions for incompressible flow using the Navier–Stokes equations and a multigrid method. *Journal of Computational Physics* 1982; **48**:387–411.
24. Kim J, Moin P, Moser R. Turbulence statistics in fully developed channel flow at low Reynolds number. *Journal of Fluid Mechanics* 1987; **177**:133–166.
25. Amati G, Succi S, Piva R. Massively parallel lattice Boltzmann simulations of turbulent channel flow. *International Journal of Modern Physics C* 1997; **8**(4):869–877.
26. Amati G, Succi S, Benzi R. Turbulent channel flow simulations with a coarse-grained extension of the lattice Boltzmann method. *Fluid Dynamics Research* 1997; **19**:289–302.
27. Amati G, Succi S, Piva R. Preliminary analysis of the scaling exponents in channel flow turbulence. *Fluid Dynamics Research* 1999; **24**(4):201–209.
28. Jimenéz J, Moin P. The minimal flow unit in near-wall turbulence. *Journal of Fluid Mechanics* 1991; **225**: 213–240.
29. Pope SB. *Turbulent Flows*. Cambridge University Press: Cambridge, 2001.

