Advances in Water Resources 32 (2009) 1767–1776

Contents lists available at ScienceDirect

Advances in Water Resources

journal homepage: www.elsevier.com/locate/advwatres

Multilayer shallow water flow using lattice Boltzmann method with high performance computing

Kevin R. Tubbs [a,1], Frank T.-C. Tsai [b,*]

[a] Donald W. Clayton Graduate Program in Engineering Science, Louisiana State University, 3418G Patrick F. Taylor Hall, Baton Rouge, LA 70803-6405, United States
[b] Department of Civil and Environmental Engineering, Louisiana State University, 3418G Patrick F. Taylor Hall, Baton Rouge, LA 70803-6405, United States

ARTICLE INFO

Article history:
Received 27 May 2009
Received in revised form 19 September 2009
Accepted 28 September 2009
Available online 4 October 2009

Keywords:
Lattice Boltzmann
Three-dimensional shallow water equations
High performance computing
BGK
Wind-driven circulation

ABSTRACT

A multilayer lattice Boltzmann (LB) model is introduced to solve three-dimensional wind-driven shallow water flow problems. The multilayer LB model avoids the expensive Navier–Stokes equations and obtains stratified horizontal flow velocities as vertical velocities are relatively small and the flow is still within the shallow water regime. A single relaxation time BGK method is used to solve each layer coupled by the vertical viscosity forcing term. To increase solution stability, an implicit step is suggested to obtain flow velocities. The main advantage of using the LBM is that after selecting appropriate equilibrium distribution functions, the LB algorithm is only slightly modified for each layer and retains all the simplicities of the LBM within the high performance computing (HPC) environment. The performance of the parallel LB model for the multilayer shallow water equations is investigated on CPU-based HPC environments using OpenMP. We found that the explicit loop control with cache optimization in LBM gives better performance on execution time, speedup and efficiency than the implicit loop control as the number of processors increases. Numerical examples are presented to verify the multilayer LB model against analytical solutions. We demonstrate the model's capability of calculating lateral and vertical distributions of velocities for wind-driven circulation over non-uniform bathymetry.

1. Introduction

In the last decade, the lattice Boltzmann (LB) method or LBM has become a powerful numerical method for simulating fluid flows [1]. The method is based on statistical physics and simulates fluid flow by tracking the evolution of distribution functions of fluid particles in discrete phase (velocity) space. The essential approach in the LB method lies in the recovery of macroscopic fluid flows from the microscopic flow behavior of the particle movement. The actual particles are not tracked, but their collective behavior is tracked through the mesoscopic evolution of particle distributions. The basic idea is to replace the nonlinear differential equations of macroscopic fluid dynamics by a simplified description modeled on the kinetic theory of gases [2]. The benefit of this description is that the LBM does not involve the solution of a global system of equations, but instead has locality, which makes it very suitable for parallel computing. Recently, this has become an important feature for numerical methods as high performance computing (HPC) systems are currently being designed to solve large-scale engineering problems.

The LBM was first developed to solve the equations of hydrodynamics governed by the Navier–Stokes equations based on the kinetic theory of gases described by the Boltzmann equation [3]. The method has been shown to be effective for simulating flows in complicated geometries on parallel computer architectures [4]. Furthermore, the method has become an alternative to other numerical methods, e.g. finite difference, finite element, and finite volume methods, in computational fluid dynamics. Due to its attractive features, recently the LBM has found a wide range of applications including wind-driven ocean circulation [5,6], discontinuous flows with shocks [7,8], and tidal flows on complex geometries with irregular bathymetry [9].

Even with increasing interest in using the LBM to solve the shallow water equations, its application is limited to two-dimensional planar problems [10–12]. When stratified horizontal velocities in depth are of interest, solving the depth-averaged shallow water equations is not sufficient. However, a full three-dimensional model of the Navier–Stokes equations is computationally expensive and does not yield much more information as the vertical velocities are relatively small and the flow is still within the shallow water flow regime. In order to take advantage of the shallow water equations while avoiding the drawbacks of the depth-averaged models, a multilayer system [13] was adopted in this study. The multilayer shallow water equations have been solved

* Corresponding author. Tel.: +1 (225) 578 4246; fax: +1 (225) 578 4945.
E-mail addresses: ktubbs2@lsu.edu (K.R. Tubbs), ftsai@lsu.edu (F.T.-C. Tsai).
[1] Tel.: +1 (225) 268 9529; fax: +1 (225) 578 4945.

0309-1708/$ - see front matter © 2009 Elsevier Ltd. All rights reserved.
doi:10.1016/j.advwatres.2009.09.008

by finite difference, finite element, and finite volume methods. Abgrall and Karni [14] developed a relaxation scheme to solve bi-fluid shallow water problems. The solution procedure has some similarities including propagation and relaxation; however, the present work deals with a single fluid and the solution procedure of the LBM is performed in the kinetic model. Audusse et al. [15] extended previous works [16,17] to model dam breaks using a finite volume solver. Their finite volume method is based on a continuous kinetic model to calculate the fluxes between cells. In this perspective, the LBM can be viewed as a discrete kinetic model. A detailed comparison of the LBM and continuous kinetic schemes can be found in [18]. Continuous kinetic schemes such as the gas kinetic scheme were found to be more memory efficient while the discrete kinetic LBM is computationally more efficient based on being three times faster. To the authors' best knowledge, the LBM has not been applied to the multilayer shallow water equations with HPC. Due the simplicity of the multilayer LB model, it can be an efficient numerical method for more complex flow and transport problems and can be easily extended to include, for example, turbulence models and quadratic friction laws while retaining the parallel advantages.

The objective of this study is to develop a parallelized LB model to solve the multilayer shallow water equations for three-dimensional shallow water flow problems on irregular boundary geometry and bathymetry. The LB method using a single relaxation time Bhatnagar–Gross–Krook (BGK) collision operator [19] (LBGK) was adopted to solve each layer coupled by the vertical viscosity forcing term. To increase solution stability, an implicit step is suggested to obtain flow velocities. The main advantage of using the LBM is that after selecting appropriate equilibrium distribution functions, the LB algorithm is only slightly modified for each layer and retains all the simplicities of the LBM within the HPC environment. We focus on the application of the multilayer shallow water equations to the problems of wind-driven circulation. We implement and investigate the parallel performance of the multilayer LB method on a shared memory HPC system, an AIX v5.3 constellation from IBM with 1.9 GHz IBM POWER5+ processors housed in the Center for Computation & Technology (CCT), Louisiana State University. We also demonstrate smaller case studies on a single workstation with a 3.0 GHz Intel® Core™2 Extreme quad core to illustrate parallel speedup on a single workstation. In this study, our LB code was written in Fortran 90 and parallelized with OpenMP using flow domain decomposition and cache optimization.

In this study, we use a parallel decomposition based on dividing the flow domain along the lateral flow direction, not based on the number of layers. A parallel decomposition based on layers would be challenging since the number of layers is always smaller than the number of nodes in the longitudinal or lateral directions. Furthermore, this decomposition would be difficult to balance workloads and require more communication between processors during the LB and implicit step. On the other hand, the decomposition along the lateral flow direction retains the inherent parallelism of the LBM by limiting the communication between processors. It also ensures that the vertical coupling between layers remains local to each processor, which is important for the implicit step.

The paper is organized as follows. In Section 2, the multilayer shallow water equations are introduced. Section 3 introduces the multilayer LB model and recovers the macroscopic equations up to second order accuracy through the Chapman–Enskog expansion. Section 4 discusses the parallel computing implementation of the multilayer LB model on shared memory systems. Section 5 presents numerical results to verify the model results against analytical solutions and demonstrate its ability to model stratified horizontal velocities. Section 6 concludes the study.

2. Multilayer shallow water equations

A multilayer model is considered by converting a three-dimensional shallow water flow problem into a number of coupled two-dimensional shallow water flow problems in layers as shown in Fig. 1. Based on the multilayer Saint–Venant system [13], the governing equations are similar to the traditional shallow water equations with additional terms for transferring momentum between the layers:

$$\frac{\partial h^{(\ell)}}{\partial t}+\frac{\partial\left(h^{(\ell)}u_i^{(\ell)}\right)}{\partial x_i}=0 \tag{1}$$

$$\frac{\partial\left(h^{(\ell)}u_i^{(\ell)}\right)}{\partial t}+\frac{\partial\left(h^{(\ell)}u_i^{(\ell)}u_j^{(\ell)}\right)}{\partial x_j}+\frac{\partial}{\partial x_i}\left(\frac{1}{2}gh^{(\ell)}\sum_{m=1}^{M}h^{(m)}\right)=v\left[\frac{\partial^2\left(h^{(\ell)}u_i^{(\ell)}\right)}{\partial x_j\partial x_j}\right]+F_i^{(\ell)},\quad \ell=1,2,\ldots,M \tag{2}$$

where $h^{(\ell)}$ is the local water height in layer ℓ, $u_i^{(\ell)}$ is the local velocity component in the i direction in layer ℓ, $F_i^{(\ell)}$ is the external force acting on layer ℓ, g is the gravitational acceleration, v is the kinematic viscosity, x_i is the Cartesian coordinate, and t is time. M is the total number of layers. The external force consists of the wind-driven forcing term $\left(F_{Wi}^{(\ell)}\right)$ (only for the top layer), the bed slope forcing term $\left(F_{Pi}^{(\ell)}\right)$, the vertical kinematic eddy viscosity term $\left(F_{\mu i}^{(\ell)}\right)$, the non-conservative pressure source term $\left(F_{NCi}^{(\ell)}\right)$ [13,15–17], and the Coriolis forcing term $\left(F_{Ci}^{(\ell)}\right)$ as follows

$$F_i^{(\ell)}=F_{Wi}^{(\ell)}+F_{Pi}^{(\ell)}+F_{\mu i}^{(\ell)}+F_{NCi}^{(\ell)}+F_{Ci}^{(\ell)} \tag{3}$$

$$F_{Wi}^{\ell}=\delta_{M\ell}\frac{\tau_{iz}^{W}}{\rho}=\delta_{M\ell}\frac{\rho_a}{\rho}C_W U_{Wi}W_s \tag{4}$$

$$F_{Pi}^{(\ell)}=-gh^{(\ell)}\frac{\partial z_b}{\partial x_i} \tag{5}$$

$$F_{\mu i}^{(\ell)}=-\kappa\delta_{1\ell}u_i^{(\ell)}+2\mu(1-\delta_{M\ell})\frac{u_i^{(\ell+1)}-u_i^{(\ell)}}{h^{(\ell+1)}+h^{(\ell)}}-2\mu(1-\delta_{1\ell})\frac{u_i^{(\ell)}-u_i^{(\ell-1)}}{h^{(\ell)}+h^{(\ell-1)}} \tag{6}$$

$$F_{NCi}^{(\ell)}=-\frac{gH^2}{2}\frac{\partial}{\partial x_i}\left(\frac{h^{(\ell)}}{H}\right) \tag{7}$$

$$F_{Ci}^{(\ell)}=\begin{cases}f_c h^{(\ell)}u_y, & i=x\\ -f_c h^{(\ell)}u_x, & i=y\end{cases} \tag{8}$$

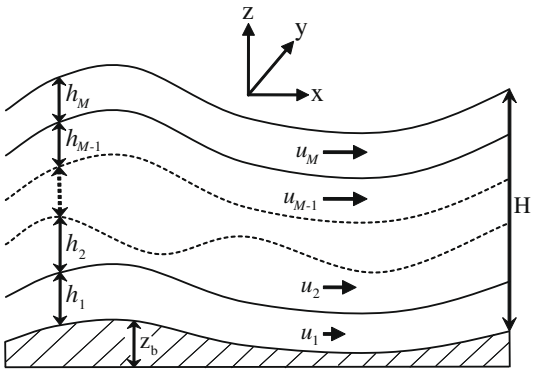

Fig. 1. Multilayer shallow water discretization.

where $\delta_{1\ell}$ and $\delta_{M\ell}$ are Kronecker delta functions; $f_c = 2\varpi \sin\varphi$ is the Coriolis parameter and ϖ is the rotation rate of the Earth and φ is the latitude; z_b is the bed elevation, τ_{iz}^W, is the wind stress in the i direction, ρ is the fluid density, ρ_a is the air density, C_W is the wind stress coefficient, U_{Wi} is the wind velocity components in the i direction, $W_s = \sqrt{\sum_i U_{Wi}^2}$ is the wind speed measured 10 m above the water surface, κ is the bottom friction coefficient, and μ is the vertical (kinematic) eddy viscosity. The water depth is the sum of local water heights of all layers, i.e. $H = \sum_{m=1}^{M} h^{(m)}$.

3. Lattice Boltzmann model for multilayer shallow water equations

3.1. BGK collision operator

The LB model is used to solve the shallow water equations for each layer. The LB equation for describing dynamics of local particle distribution functions in a discrete velocity field is [1,20]:

$$f_\alpha^{(\ell)}(\mathbf{x}+\mathbf{e}_\alpha\Delta t, t+\Delta t) = f_\alpha^{(\ell)}(\mathbf{x},t) - \frac{1}{\tau}\left(f_\alpha^{(\ell)}(\mathbf{x},t) - f_\alpha^{(\ell)eq}(\mathbf{x},t)\right) + \frac{\Delta t e_{\alpha i}}{6e^2}F_i^{(\ell)}(\mathbf{x},t) \tag{9}$$

where $f_\alpha^{(\ell)}$ is the particle distribution function moving along the α direction for the layer ℓ, $f_\alpha^{(\ell)eq}$ is the equilibrium distribution function (EDF) along the α direction for the layer ℓ, τ is the dimensionless relaxation time, $\mathbf{e}_\alpha = \{e_{\alpha i}\}$ is the streaming velocity along the α direction, $e = \Delta x/\Delta t$ is the lattice speed, Δx is the lattice size, Δt is the time step. The term $\left(f_\alpha^{(\ell)} - f_\alpha^{(\ell)eq}\right)/\tau$ is the Bhatnagar–Gross–Krook (BGK) collision operator [19], which represents changes in $f_\alpha^{(\ell)}$ due to particle collisions. F_i is the external force per unit mass in the shallow water equations [20]. The external forces are defined in macroscopic variables and their contribution distributed along the α directions. Eq. (9) consists of a streaming step and a collision step [2,20]. Based on Eq. (9), a multilayer LBGK model is constructed.

In each time step, the particle distribution functions arrive to their neighboring nodes at the same time through prescribed lattice connections. Therefore, the streaming velocity $\mathbf{e}_\alpha$ along the α direction is determined by the lattice connection and size. In this study, we consider a two-dimensional lattice with nine discrete velocities in planar direction, known as the D2Q9 lattice, shown in Fig. 2. The streaming velocity for D2Q9 is

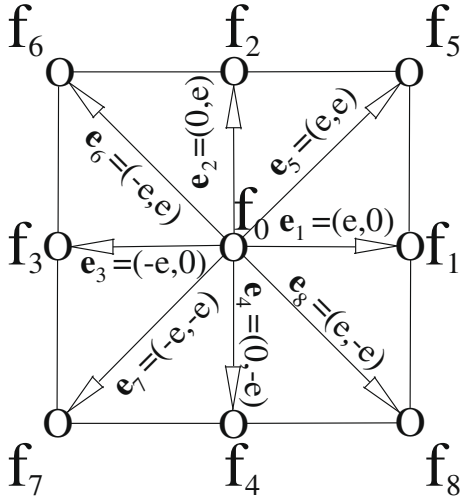

Fig. 2. D2Q9 lattice.

$$\mathbf{e}_\alpha = \begin{cases} (0,0) & \alpha = 0 \\ e\left[\cos\left(\frac{1}{4}(2\alpha-2)\pi\right), \sin\left(\frac{1}{4}(2\alpha-2)\pi\right)\right] & \alpha = 1,2,3,4 \\ \sqrt{2}e\left[\cos\left(\frac{1}{4}(2\alpha-9)\pi\right), \sin\left(\frac{1}{4}(2\alpha-9)\pi\right)\right] & \alpha = 5,6,7,8 \end{cases} \tag{10}$$

The local water height and flow velocity for each layer are calculated as the zeroth and first moments of the distribution functions:

$$h^{(\ell)} = \sum_\alpha f_\alpha^{(\ell)} \tag{11}$$

$$h^{(\ell)}u_i^{(\ell)} = \sum_\alpha e_{\alpha i} f_\alpha^{(\ell)} \tag{12}$$

The EDFs [1] applied to the multilayer shallow water equations are

$$f_\alpha^{(\ell)eq} = h^{(\ell)}\omega_\alpha\left(\frac{c_s^2}{e^2} + \frac{\mathbf{u}^{(\ell)}\cdot\mathbf{e}_\alpha}{e^2} + \frac{3}{2}\frac{\left(\mathbf{u}^{(\ell)}\cdot\mathbf{e}_\alpha\right)^2}{e^4} - \frac{1}{2}\frac{\mathbf{u}^{(\ell)}\cdot\mathbf{u}^{(\ell)}}{e^2}\right), \quad \alpha > 0$$

$$f_0^{(\ell)eq} = h^{(\ell)} - \sum_{\alpha=1}^{8} f_\alpha^{(\ell)eq} \tag{13}$$

where c_s^2 is known as the lattice squared speed of sound, which in LBM for shallow water equations is a numerical parameter that relates to the wave celerity, $C = \sqrt{gH}$, i.e., $c_s^2 = C^2/2 = gH/2$. It is noted that the weighting coefficients ω_α remain the same as in monolayer LBGK formulation since each layer has the same planar discretization. For D2Q9, the weighting coefficients are $\omega_\alpha = 1/3$, $\alpha = 1,2,3,4$ and $\omega_\alpha = 1/12$, $\alpha = 5,6,7,8$. The EDFs for the multilayer LBGK have the same form as the monolayer and follow the same constraints on the zeroth, first, second, and third moments for each layer:

$$\sum_\alpha f_\alpha^{(\ell)eq} = h^{(\ell)} \tag{14}$$

$$\sum_\alpha e_{\alpha i} f_\alpha^{(\ell)eq} = h^{(\ell)}u_i^{(\ell)} \tag{15}$$

$$\sum_\alpha e_{\alpha i}e_{\alpha j} f_\alpha^{(\ell)eq} = h^{(\ell)}\left(\delta_{ij}c_s^2 + u_i^{(\ell)}u_j^{(\ell)}\right) \tag{16}$$

$$\sum_\alpha e_{\alpha i}e_{\alpha j}e_{\alpha k} f_\alpha^{(\ell)eq} = h^{(\ell)}\frac{e^2}{3}\left(\delta_{ik}u_j^{(\ell)} + \delta_{jk}u_i^{(\ell)} + \delta_{ij}u_k^{(\ell)}\right) \tag{17}$$

To ensure that the multilayer LBGK model solves the multilayer shallow water equations with proper LB parameters, Appendix A utilizes the moments in Eqs. (14)–(17) and shows the recovery of the multilayer shallow water equations (1) and (2) up to second order by Chapman–Enskog multi-scale analysis, where we obtain

$$c_s^2 = \frac{1}{2}g\sum_{m=1}^{M} h^{(m)} = \frac{1}{2}gH \tag{18}$$

and

$$\nu = \frac{e^2}{3}\Delta t\left(\tau - \frac{1}{2}\right) \tag{19}$$

Similar recovery work for single-layer shallow water equations can be found in Zhou [20].

3.2. Multilayer LB algorithm

With the solution known at time n, the solution at time $n+1$ is calculated with an explicit treatment of quantities local to a layer, i.e., the left hand side of Eqs. (1) and (2), the kinematic viscosity term in Eq. (2), the wind-driven forcing term, the bed slope forcing term, the non-conservative pressure source term, and the Coriolis forcing term. The vertical viscosity forcing term ($F_{\mu i}^{(\ell)}$) can be interpreted as a friction term between layers. To increase the solution

stability, we treat this term implicitly. First, we solve Eq. (9) for each layer explicitly to obtain intermediate distribution functions $f_\alpha^{*(\ell)n+1}$, velocities $u_i^{*(\ell)n+1}$, and local water heights $h^{*(\ell)n+1}$:

$$f_\alpha^{*(\ell)n+1}(\mathbf{x}+\mathbf{e}_\alpha\Delta t) = f_\alpha^{(\ell)n}(\mathbf{x}) - \frac{1}{\tau}\left(f_\alpha^{(\ell)n}(\mathbf{x}) - f_\alpha^{((\ell)eq)n}(\mathbf{x})\right) + \frac{\Delta t e_{\alpha i}}{6e^2}F_i^{(\ell)n}(\mathbf{x}) \tag{20}$$

$$h^{*(\ell)n+1} = \sum_\alpha f_\alpha^{*(\ell)n+1} \tag{21}$$

$$u_i^{*(\ell)n+1} = \sum_\alpha e_{\alpha i} f_\alpha^{*(\ell)n+1}/h^{*(\ell)n+1} \tag{22}$$

Then, an implicit step is introduced to update macroscopic variables given as

$$\begin{bmatrix} h^{(\ell)n+1} \\ (hu_i)^{(\ell)n+1} \end{bmatrix} + \begin{bmatrix} 0 \\ F_{\mu i}^{(\ell)n+1} \end{bmatrix} = \begin{bmatrix} h^{*(\ell)n+1} \\ (hu_i)^{*(\ell)n+1} \end{bmatrix} \tag{23}$$

where the right hand side represents the intermediate solution given by the explicit LB step (Eqs. (20)–(22)) and is known. The first row in Eq. (23) relates to mass conservation, which is not affected by the vertical viscosity forcing. The second row in Eq. (23) relates to momentum that needs to account for the vertical viscosity forcing. The updating of flow velocities $u_i^{(\ell)n+1}$ at time $n+1$ based on the second row in Eq. (23) leads to an $M \times M$ tridiagonal linear system [13]:

$$\begin{bmatrix} a_1 & b_1 & 0 & \cdots & 0 \\ c_2 & a_2 & b_2 & \ddots & \vdots \\ 0 & c_3 & a_3 & \ddots & 0 \\ \vdots & \ddots & \ddots & \ddots & b_{M-1} \\ 0 & \cdots & 0 & c_M & a_M \end{bmatrix} \begin{bmatrix} u_i^{(1)n+1} \\ u_i^{(2)n+1} \\ u_i^{(3)n+1} \\ \vdots \\ u_i^{(M)n+1} \end{bmatrix} = \begin{bmatrix} (hu_i)^{*(1)n+1} \\ (hu_i)^{*(2)n+1} \\ (hu_i)^{*(3)n+1} \\ \vdots \\ (hu_i)^{*(M)n+1} \end{bmatrix} \tag{24}$$

where the matrix elements are

$$\begin{aligned} a_1 &= h^{(1)n+1} + \frac{2\mu\Delta t}{h^{(1)n+1}+h^{(2)n+1}} + \kappa\Delta t \\ a_\ell &= h^{(\ell)n+1} + 2\mu\Delta t\left(\frac{1}{h^{(\ell)n+1}+h^{(\ell+1)n+1}} + \frac{1}{h^{(\ell)n+1}+h^{(\ell-1)n+1}}\right), \\ &\quad \ell = 2,\ldots,M-1 \\ a_M &= h^{(M)n+1} + \frac{2\mu\Delta t}{h^{(M)n+1}+h^{(M-1)n+1}} \end{aligned} \tag{25}$$

$$b_\ell = h^{(\ell)n+1} + \frac{2\mu\Delta t}{h^{(\ell)n+1}+h^{(\ell+1)n+1}}, \quad \ell = 1,\ldots,M-1 \tag{26}$$

$$c_\ell = h^{(\ell)n+1} + \frac{2\mu\Delta t}{h^{(\ell)n+1}+h^{(\ell-1)n+1}}, \quad \ell = 2,\ldots,M \tag{27}$$

The above numerical procedure to advance the solutions of the multilayer LB algorithm from time n to time $n+1$ can be summarized in the following steps:

Step 1. Use the local water heights, $h^{(\ell)n}$, total water depth, H^n and velocities, $u_i^{(\ell)n}$ at time n to compute the EDFs, $f_\alpha^{(\ell)eq}$, from Eq. (13).
Step 2. Calculate the distribution functions, $f_\alpha^{(\ell)}$, using Eq. (20) and impose bounce-back boundary conditions for each layer to ensure conservation of mass and momentum at the impermeable walls.
Step 3. Calculate intermediate local water heights, $h^{*(\ell)n+1}$, total water depth, H^{*n+1} and velocities, $u_i^{*(\ell)n+1}$ using Eqs. (21) and (22).
Step 4. Update velocities $u_i^{(\ell)n+1}$ at time $n+1$ by solving the linear system in Eq. (24).
Step 5. Repeat Steps 1–4 for next time step.

4. High performance computing: OpenMP

To demonstrate the inherent parallel features of the LB model for high performance computing, this study implements OpenMP in a shared memory environment to test the scalability of the multilayer LB code. OpenMP parallelization is a straightforward extension of serial coding, based on the simple recipe: (i) add work distribution directives; (ii) carefully control variable scoping; (iii) if needed, choose the most suitable parallel loop scheduling; and (iv) if needed, take care of false sharing of variables among threads. Although this approach works with some very simple codes, most programs parallelized in this way do not scale beyond a small number of processors [21]. A computational advantage of the LBM is that implementing streaming and colliding steps leads to a computational algorithm that is very suitable for parallelization in both shared and distributed memory environments. Massaioli and Amati [21] implemented a two-dimensional LB algorithm using standard OpenMP directives and showed linear speedup to eight processors. The standard OpenMP directive uses an implicit control of the loop variable. In this study, we compare implicit and explicit loop control approaches in OpenMP for our multilayer LB algorithm. The latter allows for cache optimization by dividing the computational domain of each processor into sub-domains.

4.1. Cache optimization

Obtaining data from the main memory in every computational cycle is time-consuming and the CPU remains idle during this process. If the data were being accessed from cache in every computational cycle, time consumed by data transfer would be much less. To obtain good performance on cache-based computer architectures, the computational algorithms are required to divide the data (computational domain) into blocks (sub-domains) that can fit into cache and then be utilized repeatedly. Not all algorithms are amenable to this kind of cache optimization since data dependencies disallow updating sub-domains separately. The lattice Boltzmann algorithm has only nearest–neighbor data dependencies and is highly amenable to cache optimization.

4.2. Cache optimization for LBM using OpenMP

To implement the LBM algorithm efficiently with optimal use of cache, the grids need to be divided into subsections that fit in cache. Performing computations separately for each subsection, for several time steps, achieves cache optimization. This is possible since LBM computations are of a completely local nature in the collision step, the right hand side of Eq. (9), which includes the computation of the equilibrium distribution functions and the forcing terms:

$$f_\alpha^{\dagger(\ell)}(\mathbf{x},t) = (1-\omega)f_\alpha^{(\ell)}(\mathbf{x},t) + \omega f_\alpha^{(\ell)eq}(\mathbf{x},t) + \gamma F_i^{(\ell)}(\mathbf{x},t) \tag{28}$$

where $\omega = 1/\tau$ and $\gamma = \Delta t e_{\alpha i}/6e^2$. Computations in the collision step do not involve any dependencies between sub-domains. The streaming step is an assignment operation of an almost local nature due to nearest-neighbor dependencies:

$$f_\alpha^{(\ell)}(\mathbf{x}+\mathbf{e}_\alpha\Delta t, t+\Delta t) = f_\alpha^{\dagger(\ell)}(\mathbf{x},t) \tag{29}$$

There is more than one choice of domain decomposition for three dimensional data. The current parallel LB algorithm is implemented by decomposing the entire flow domain into several computational domains divided along one horizontal flow direction, i.e., the lateral flow direction, according to the number of processors. The choice of one horizontal flow direction is made to restrict data communication in one direction. This decomposition allows the LB equation to be calculated with minimal communication across processor do-

mains to retain the parallel benefits of the LBM. The main key is that the domain is not decomposed in the vertical (layer) direction. This allows the LB equation and the implicit step to remain completely local in layers to each processor and does not require the tridiagonal solver for Eq. (24) to be parallelized. An added benefit is that the computational domain of each processor is further divided into sub-domains to allow for cache optimization.

4.3. Parallel speedup and efficiency

Parallel performance is often evaluated by two factors: parallel speedup and efficiency. The parallel speedup is the ratio of computation times for one thread to the total number of threads (processors) used. The efficiency is the average speedup over the total number of threads. We use these two factors to evaluate the HPC in the following numerical simulations of wind-driven circulation in a square lake of dimensions 64,000 m × 64,000 m with a flat bottom. The multilayer LB method was implemented on a shared memory HPC system, an AIX v5.3 constellation from IBM with 1.9 GHz IBM POWER5+ processors. The initial water depth is 10 m. The lake is discretized into a grid of size $1024 \times 1024 \times 10$ corresponding to 10 layers and $\Delta x = \Delta y = 62.5$ m. The initial local water height is 1 m for each layer and the initial flow velocity is zero. The LBM parameters are $\tau = 0.501$, $\Delta t = 8$ s, and $e = 7.8125$ m/s. The physical parameters are $\tau_{iz}^W = 0.1\ \mathrm{N/m^2}$, $\rho = 1025\ \mathrm{kg/m^3}$, $\rho_a = 1.2\ \mathrm{kg/m^3}$, $C_W = 0.0015$, $\mu = 0.01\ \mathrm{m^2/s}$, $\kappa = 0.001$ m/s, and $f_c = 0\ \mathrm{s^{-1}}$. The wind direction is along the positive x direction. Therefore, the wind velocity is $U_{Wx} = 7.4536$ m/s and $U_{Wy} = 0$ m/s. The initial condition for the distribution functions are the EDFs in Eq. (13) with the static water ($\mathbf{u}^{(\ell)} = 0$) and the initial local water heights. Free-slip bounce-back boundary conditions are applied to the four vertical side walls of the lake. The simulations were run for 1000 time steps.

Table 1 shows the execution time, speedup and efficiency for the implicit and explicit loop controls that were tested up to 16 processors. It demonstrates the importance of cache optimization in LBM. Optimizing cache-use on each thread based on dividing the processor domains into sub-domains improves the speedup and efficiency. The explicit loop control implementation takes slightly longer than the implicit approach; however, the parallel performance of the implicit approach starts to break down past 12 processors. The speed up and efficiency of the implicit approach begins to decline after 8 threads while the explicit approach continues to scale up to 16 threads.

5. Numerical examples

5.1. Wind-driven circulation in rectangular lake

We verify the multilayer LB model using wind-driven circulation in a rectangular lake of dimensions 3400 m × 1400 m with a flat bottom. The initial water depth is 40 m, where an analytical solution for the horizontal velocity profile in depth is available [22]. The rectangular lake is discretized into 69×29 lattices in the planar direction corresponding to $\Delta x = 50$ m and $\Delta y = 50$ m. The vertical direction is discretized into five, 10 and 20 layers for each test case corresponding to an initial local water height of 8, 4, and 2 m, respectively. The LBM parameters are $\tau = 0.501$, $\Delta t = 2$ s, and $e = 25$ m/s. The bed friction is based on a linear friction law. The physical parameters are the same as in Section 4.3 except for the wind stress that is considered $\tau_{iz}^W = 0.1\ \mathrm{N/m^2}$ and $\tau_{iz}^W = 0.4\ \mathrm{N/m^2}$ for the two cases. Again, the wind direction is along the positive x direction. For $\tau_{iz}^W = 0.4\ \mathrm{N/m^2}$, the wind velocity is $U_{Wx} = 14.9071$ m/s and $U_{Wy} = 0$ m/s. The initial conditions are ap-

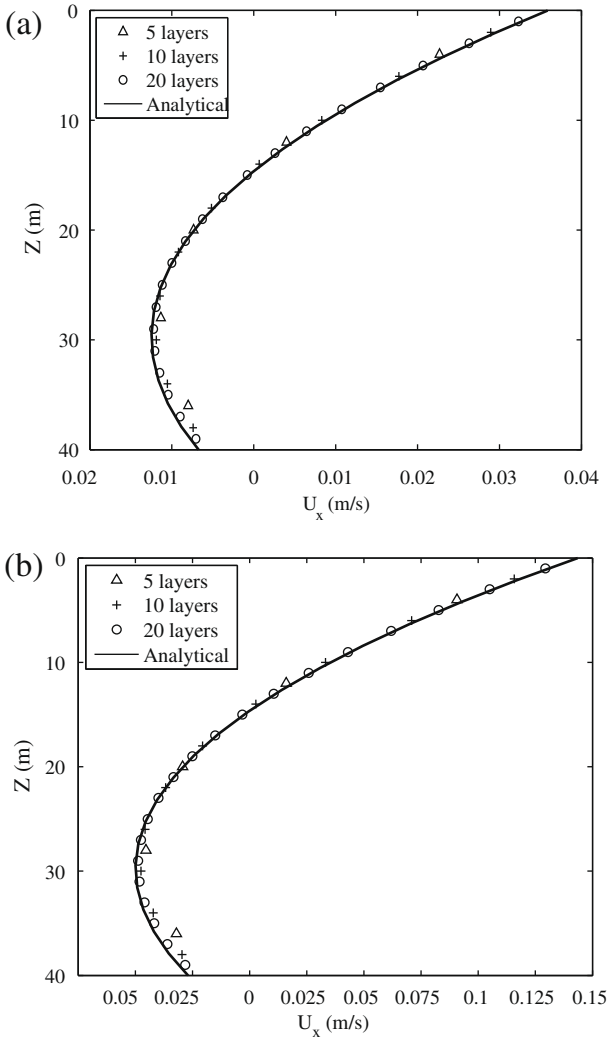

Fig. 3. Comparisons of numerical model prediction with analytical solution for: (a) $\tau_{iz}^W = 0.1\ \mathrm{N/m^2}$, and (b) $\tau_{iz}^W = 0.4\ \mathrm{N/m^2}$.

Table 1
Execution time (min), speedup, and efficiency for implicit and explicit loop control implementations on a $1024 \times 1024 \times 10$ grid.

Threads	Implicit loop control			Explicit loop control		
	Execution time (min)	Speedup	Efficiency	Execution time (min)	Speedup	Efficiency
1	40.408	1.00	1.00	45.978	1.00	1.00
2	20.260	1.99	0.99	22.156	2.00	1.00
4	10.096	4.00	1.00	11.698	3.99	0.98
8	5.468	7.39	0.92	6.005	7.54	0.94
12	4.335	9.32	0.78	3.946	11.65	0.97
16	6.306	6.40	0.40	3.076	14.94	0.93

plied by initializing the distribution functions to an EDF with the static water depth and local water heights, i.e. $f_\alpha^{(\ell)} = f_\alpha^{(\ell)eq}$, where $h^{(\ell)} = H^{init}/M$ and $\mathbf{u}^{(\ell)} = 0$. Free-slip bounce-back boundary conditions are applied to the four vertical side walls of the lake. The numerical simulation was carried out up to the establishment of a steady-state two-dimensional circulation. The u_x velocity profile at the center of the lake, $x = 1700$ m, $y = 700$ m, was compared against the analytical solution of Navier–Stokes equations assuming the surface slope and the horizontal velocity are constant in the longitudinal direction [22]. The vertical eddy viscosity is also assumed to be constant in the vertical direction. The multilayer LB solutions compare well to the analytical solutions of u_x profile for uniform wind stresses of 0.1 and 0.4 N/m^2, as shown in Fig. 3a and b. The corresponding maximum flow velocity occurred in the upper-most layer to the surface is 3.6 and 14.6 cm/s, respectively. This example shows that the multilayer LB model is capable of simulating wind-driven currents.

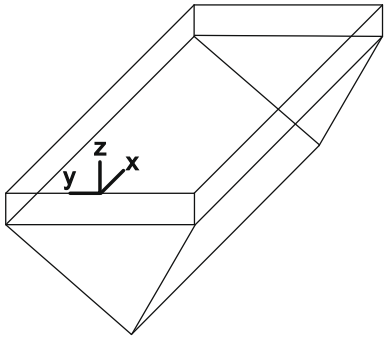

Fig. 4. Computational domain with triangular bathymetric profile.

5.2. Wind-driven circulation in rotating and non-rotating basins

In this example, we demonstrate the multilayer LB model by simulating the wind-driven circulation with and without rotation over laterally varied bathymetric cross sections. We consider a triangular bathymetry shown in Fig. 4 and two Gaussian bathymetric profiles. The initial water depth for the triangular bathymetry has a minimum of 3 m and maximum of 20 m. The Gaussian bathymetric profiles have initial water depths given by

$$H_0(y) = 8 + 12\exp\left[-\left(\frac{y}{2000}\right)^2\right] \tag{30}$$

and

$$H_0(y) = 8 + 8\exp\left[-\left(\frac{y - 3D/10}{2000}\right)^2\right] + 12\exp\left[-\left(\frac{y + 3D/10}{2000}\right)^2\right] \tag{31}$$

where D is the width of the basin. The coordinate system for the triangular and Gaussian bathymetries is based on Fig. 4, where the x-axis is at the center of the channel. The y-axis is laid along the closed boundary at $x = 0$. For each numerical example, the numerical domain consists of a longitudinally uniform basin 100 km long and 10 km wide for triangular and 15 km wide for Gaussian bathymetric profiles. Grid size is 250 m along the x and y directions. Ten vertical layers are used.

In the LBM formulation, the computational domain is covered by $401 \times 41 \times 10$ lattices for the triangular bathymetric profile and $401 \times 61 \times 10$ lattices for the Gaussian bathymetric profiles with $\tau = 0.501$, $\Delta x = 250$ m, $\Delta t = 12.5$ s, and $e = 20$ m/s. The free-slip condition was used for all closed boundaries. The initial water in the basin was static and a wind stress was increased linearly during the first six simulated hours. After six hours, the wind was constant. The wind stress acted along the positive x direction

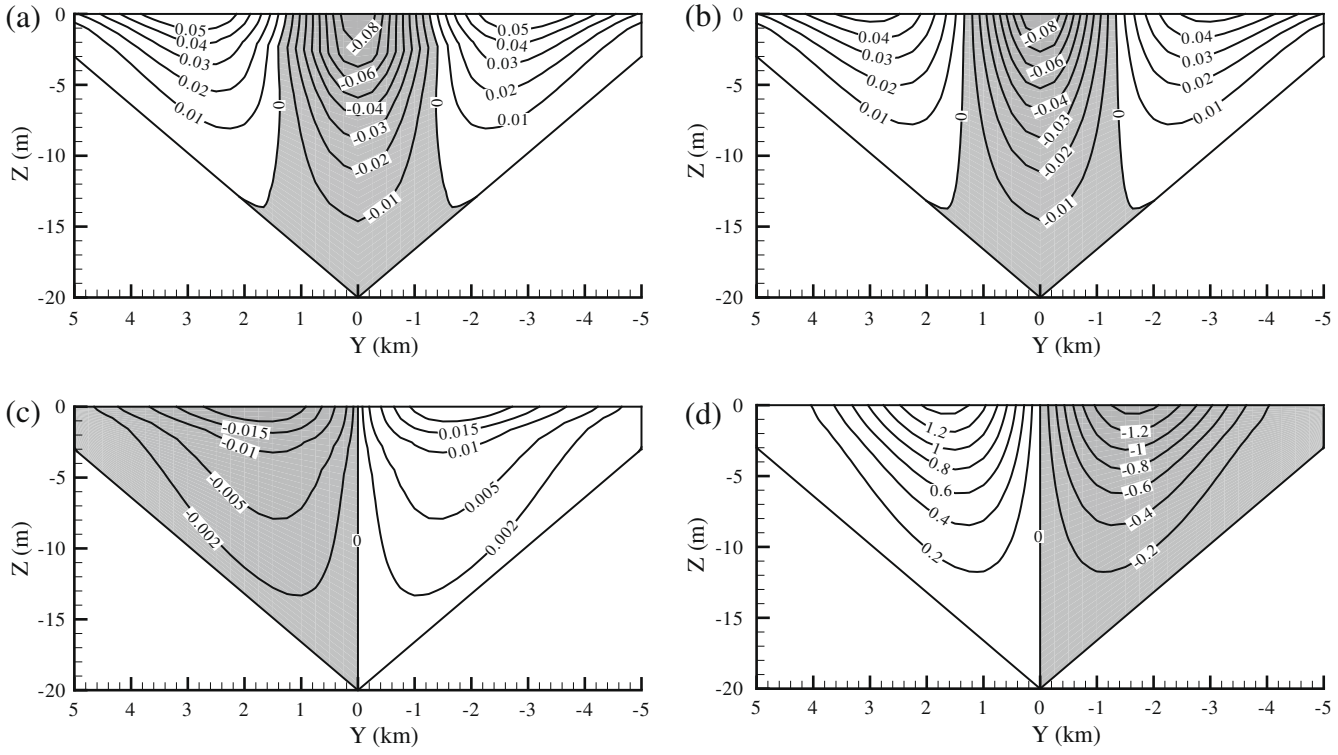

Fig. 5. Contours of u_x velocity (m/s) at (a) $x = 50$ km, and (b) $x = 98$ km and u_y velocity ($\times 10^{-2}$ m/s) at (c) $x = 50$ km, and (d) $x = 98$ km for a non-rotating system with the triangular bathymetric profile. The gray areas represent negative velocities.

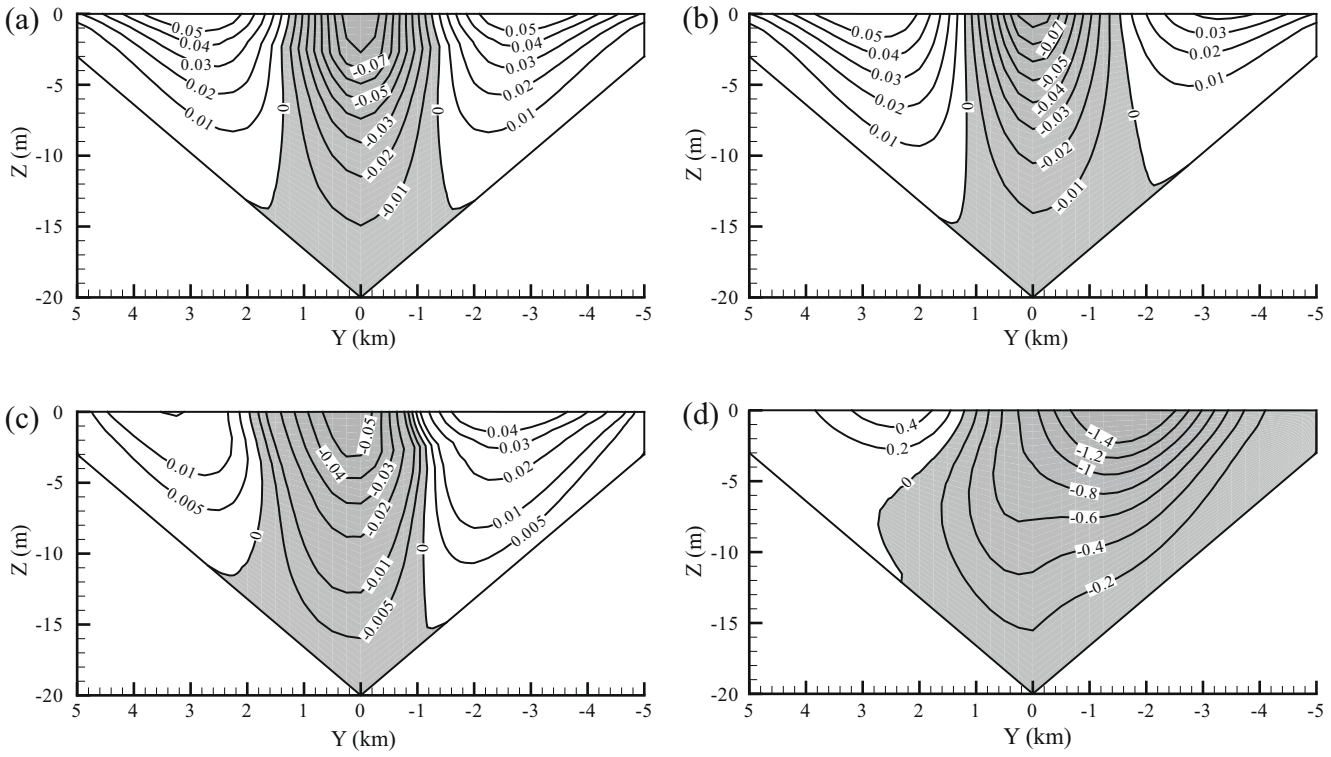

Fig. 6. Contours of the u_x velocity (m/s) at (a) $x = 50$ km, and (b) $x = 98$ km and u_y velocity ($\times 10^{-2}$ m/s) at (c) $x = 50$ km, and (d) $x = 98$ km for a rotating system with the triangular bathymetric profile. The gray areas represent negative velocities.

and blew uniformly throughout the domain. The numerical simulations were run up to 2 days after the wind stress was constant with the establishment of a steady state velocity field occurring after about 1 day. The numerical simulations were run on a single workstation with a 3.0 GHz Intel® Core™2 Extreme quad core processor.

The first example is for the non-rotating case, in which the bathymetric profile was triangular with minimum depth of 3 m, maximum depth of 20 m. The physical parameters are $\tau_{iz}^{W} = 0.03\ \mathrm{N/m^2}$, $\rho = 1025\ \mathrm{kg/m^3}$, $\rho_a = 1.2\ \mathrm{kg/m^3}$, $C_W = 0.0015$, $\mu = 0.004\ \mathrm{cm^2/s}$, $\kappa = 0.0025$ m/s, and $f_c = 0\ \mathrm{s^{-1}}$. The wind stress was applied in the positive x direction. The wind velocity is $U_{Wx} = 4.0825$ m/s and $U_{Wy} = 0$ m/s. The bed friction is formulated using a linear friction law. Fig. 5 shows the u_x and u_y distributions at $x = 50$ km and $x = 98$ km planes. As shown in Fig. 5a and b, water in the shallow region flows (with positive u_x) in the direction of the wind at all depths. Water flows (with negative u_x) in the opposite direction of the wind in the central part of the channel. The magnitude of the flow is highest near the surface and decreases with the depth as expected due to bottom friction. Fig. 5c and d shows a divide of the transverse flow, u_y at $y = 0$. The symmetric distributions are due to neglecting the Coriolis force (non-rotating).

To further demonstrate the multilayer LBM, we added the effect of the Earth's rotation with the Coriolis parameter, $f_c = 10^{-4}\ \mathrm{s^{-1}}$ to the previous case. The u_x distributions in Fig. 6a and b show similar lateral variability to those in the non-rotating case. The difference is seen at $x = 98$ km plane where u_x is symmetric for the non-rotating case, but asymmetric for the rotating case. The spatial distributions of u_x at $x = 50$ km and $x = 98$ km planes for both cases were consistent with those obtained by Sanay and Valle-Levinson [23]. Due the Coriolis force, the u_y distributions in Fig. 6c and d are not symmetric.

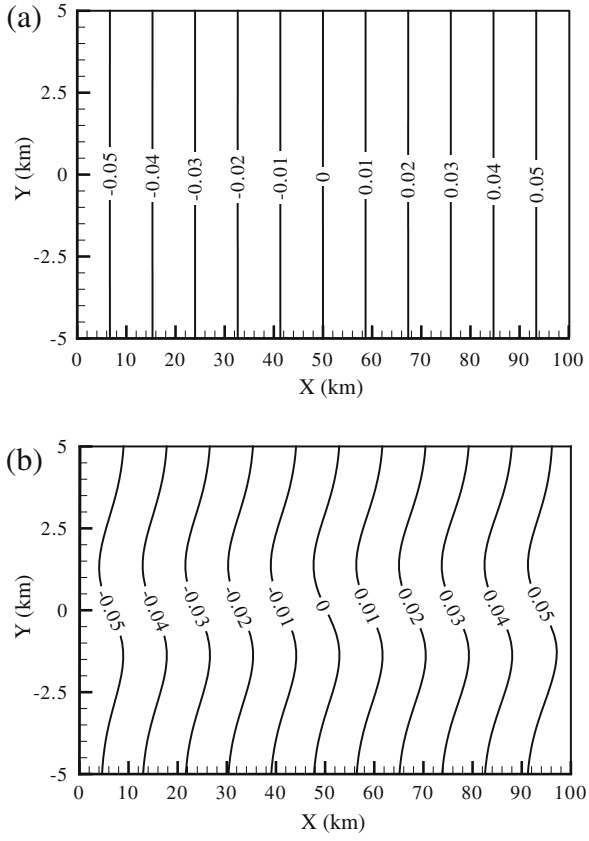

Fig. 7. Contours of the water surface elevation (m) for (a) non-rotating, and (b) rotating cases for the triangular bathymetric profile.

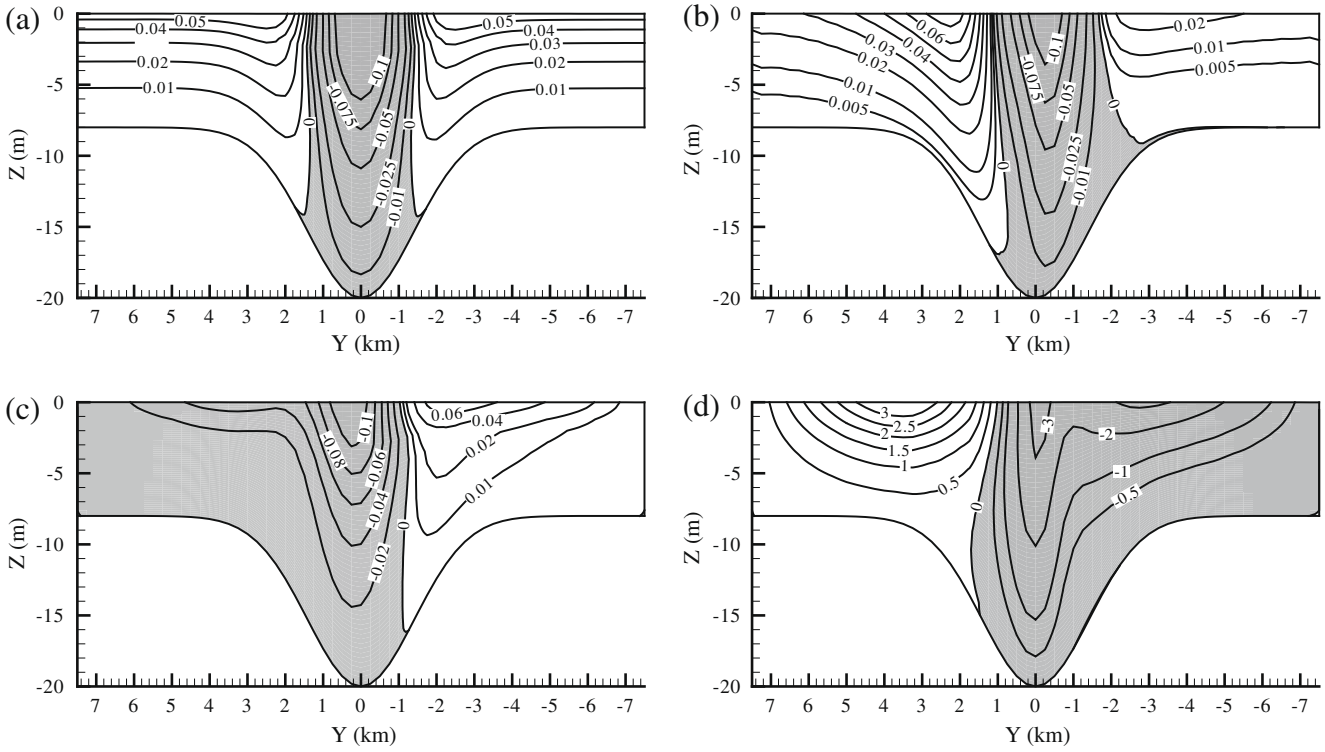

Fig. 8. Contours of u_x velocity (m/s) at (a) $x = 50$ km, and (b) $x = 98$ km and u_y velocity ($\times 10^{-2}$ m/s) at (c) $x = 50$ km, and (d) $x = 98$ km for the Gaussian bathymetric profile given in Eq. (30). The gray areas represent negative velocities.

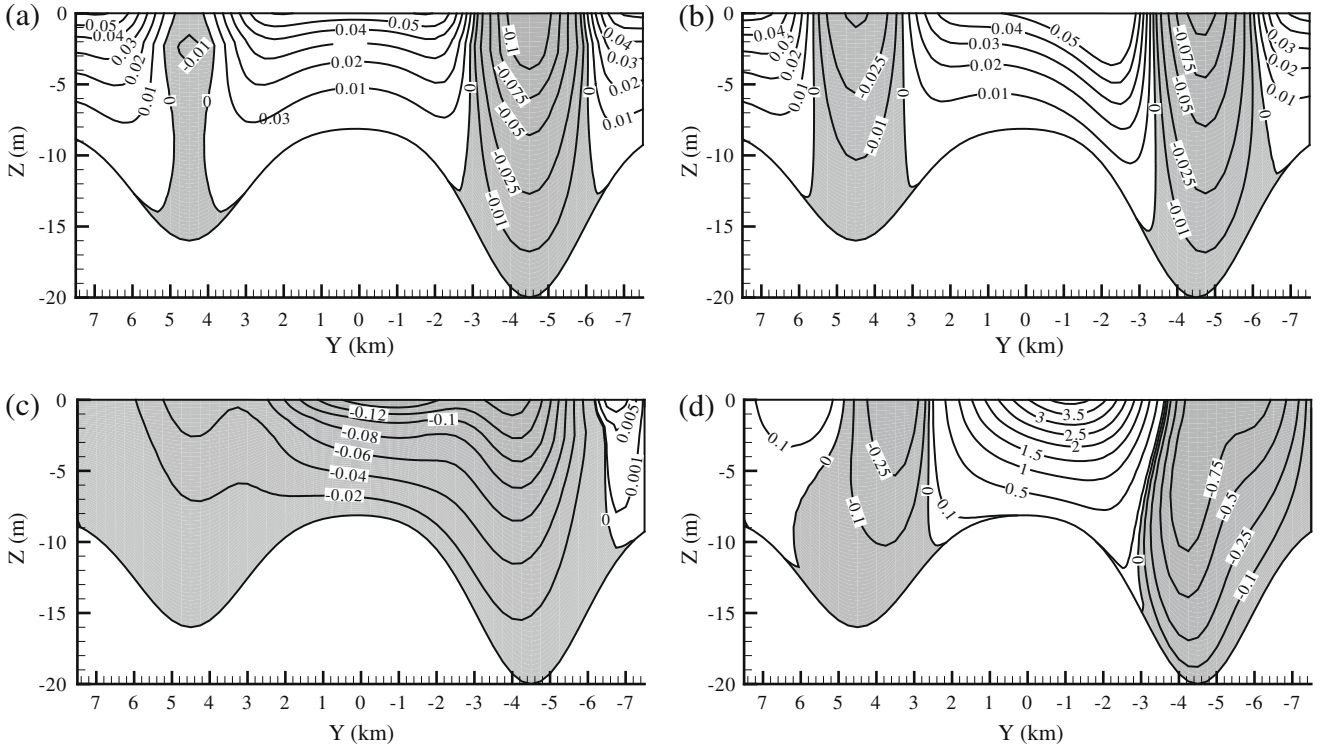

Fig. 9. Contours of u_x velocity (m/s) at (a) $x = 50$ km, and (b) $x = 98$ km and u_y velocity ($\times 10^{-2}$ m/s) at (c) $x = 50$ km, and (d) $x = 98$ km for the Gaussian bathymetric profile given in Eq. (31). The gray areas represent negative velocities.

The simulation time for both cases was 20.67 min on a single core and 5.07 min on four cores of a single workstation, demonstrating the expected 4 times speedup.

An important feature of the solution caused by the inclusion of the Earth's rotation is the free surface elevation distribution in the domain as shown in Fig. 7. In the non-rotating case, the surface ele-

vation distribution was only a function of along-channel direction. The across-channel barotropic pressure gradient expected in wind-driven flow over flat-bottom rotating systems is nearly a linear function of y location [24]. The combination of the Coriolis effect and laterally varying bathymetry produced surface elevation contours with stronger lateral variability compared to the non-rotating case. This is due to the spatial variability in the longitudinal flow and vertical mixing dictated by the laterally varying bathymetry. The spatial distribution of the simulated surface elevation was consistent with that obtained by Sanay and Valle-Levinson [23] and Glorioso and Davies [24].

The second numerical example considers a Gaussian bathymetric profile with initial water depth given by Eq. (30). The initial water depth has a minimum of 8 m and a maximum of 20 m located at the center of basin. The physical parameter values remain the same as in the triangular bathymetry case with the Coriolis parameter, $f_c = 10^{-4}\ \mathrm{s}^{-1}$. The Gaussian profile produces a channel-shoal combination resulting in the flow patterns shown in Fig. 8. The water flows in the direction of the wind for shallow regions and in the opposite direction of the wind in the central part of the channel as shown in Fig. 8a and b. The asymmetry near the closed boundary is more prevalent for this bathymetric profile (see Fig. 8b). The u_y distributions shown in Fig. 8c and d are asymmetric because of the Coriolis force. The magnitude of u_y at $x = 98$ km plane is much larger than that at $x = 50$ km plane.

The third numerical example considers a bathymetric profile with two Gaussians and initial water depth given by Eq. (31) with a minimum of 8 m. The maximum depth is 20 m located at $y = -3D/10$ km, and a local maximum depth is 16 m located at $y = 3D/10$ km. Using the same physical parameter values as in the previous Gaussian bathymetry case, the flow patterns are shown in Fig. 9. Similarly, the water flows in the direction of the wind in the shallow regions and flows in the opposite direction of the wind in both deeper channels as shown in Fig. 9a and b. The u_y is very small at $x = 50$ km plane (Fig. 9c) while it is very strong at $x = 98$ km plane (Fig. 9d).

The simulation time for the second and third numerical examples was 32.89 min on a single core and 8.12 min on four cores of a single workstation, which also demonstrates the expected 4 times speedup.

6. Conclusions

This study has shown the potential of using a multilayer LB model for simulating three-dimensional shallow water flows in a high performance computing environment. A single relaxation time BGK method was used to solve each layer coupled by the vertical viscosity forcing term. To increase solution stability, an implicit step is suggested to obtain flow velocities.

The influence of wind stress, vertical viscosity forcing, bottom friction and bathymetry were tested in this study. The numerical results of flow velocities for wind-driven circulation in a rectangular lake with a flat bottom agree well with the analytical solutions [22]. Moreover, the multilayer LB model was tested over non-uniform bathymetry with and without the effects of the Earth's rotation to calculate lateral and vertical distributions of the velocities. The simulated wind-driven circulation patterns, consisting of downwind flow over the shoals and upwind flow in the channel along the entire domain, were consistent with other studies [23,24].

This study has demonstrated parallel performance of the multilayer LB model using OpenMP. The parallel decomposition along only on the horizontal flow directions has two advantages: (1) It retains the inherent parallelism of the LBM for each layer; and (2) it retains the locality of the tridiagonal solver over layers with respect to threads. The study has shown that the use of the explicit loop control with cache optimization is important in maintaining linear speedup as the number of processors increases.

Acknowledgements

The study was supported in part by the Department of the Interior, US Geological Survey under Grant No. 05HQGR0142. The first author was also supported by the US National Science Foundation under Grant No. DGE-0504507. The views and conclusions contained in the document are those of the authors and should not be interpreted as necessarily representing the official policies, either expressed or implied, of the US Government. The authors acknowledge the constructive reviews provided by three anonymous reviewers.

Appendix A. Recovery of multilayer shallow water equations

To recover the multilayer shallow water equations without the forcing term, $f_\alpha^{(\ell)}(\mathbf{x} + \mathbf{e}_\alpha \Delta t, t + \Delta t)$ is expanded around $(\mathbf{x}, t)$ using the Taylor series expansion:

$$f_\alpha^{(\ell)}(\mathbf{x} + \mathbf{e}_\alpha \Delta t, t + \Delta t) = f_\alpha^{(\ell)}(\mathbf{x}, t) + \sum_{n=1}^{\infty} \frac{d^n f_\alpha^{(\ell)}(\mathbf{x}, t)}{dt^n} \frac{\Delta t^n}{n!} \tag{A1}$$

where $d_t = \partial_t + \mathbf{e}_i \cdot \nabla$ is the total derivative with respect to time. For multi-scale analysis, we adopt the Chapman–Enskog expansion as follows:

$$\partial_i = \varepsilon \partial_{1i} \tag{A2}$$

$$\partial_t = \varepsilon \partial_{t1} + \varepsilon^2 \partial_{t2} \tag{A3}$$

Furthermore, we perturb $f_\alpha^{(\ell)}$ around $f_\alpha^{(\ell)eq}$ in terms of Knudsen number ε [25]:

$$f_\alpha^{(\ell)} = f_\alpha^{(\ell)eq} + \varepsilon f_\alpha^{(\ell)(1)} + \varepsilon^2 f_\alpha^{(\ell)(2)} + \varepsilon^3 f_\alpha^{(\ell)(3)} + O(\varepsilon^4) \tag{A4}$$

where $f_\alpha^{(\ell)(k)}$ are the perturbation terms. Introducing Eqs. (A1)–(A4) into Eq. (9) and grouping terms of the same order in ε, the differential equations up to second order are

$$O(\varepsilon): \quad \Delta t \left\{ \frac{\partial}{\partial t_1} + e_{\alpha i} \frac{\partial}{\partial x_{1i}} \right\} f_\alpha^{(\ell)eq} = -\frac{1}{\tau} f_\alpha^{(\ell)(1)} \tag{A5}$$

$$O(\varepsilon^2): \quad \Delta t \left(\frac{\partial}{\partial t_2} f_\alpha^{(\ell)eq} + \left\{ \frac{\partial}{\partial t_1} + e_{\alpha i} \frac{\partial}{\partial x_{1i}} \right\} f_\alpha^{(\ell)(1)} \right) + \frac{\Delta t^2}{2} \left(\frac{\partial}{\partial t_1} + e_{\alpha i} \frac{\partial}{\partial x_{1i}} \right)^2 f_\alpha^{(\ell)eq} = -\frac{1}{\tau} f_\alpha^{(\ell)(2)} \tag{A6}$$

Taking $\sum_\alpha$ Eq. (A5), $\sum_\alpha e_{\alpha i}$ Eq. (A5), $\sum_\alpha$ Eq. (A6), and $\sum_\alpha e_{\alpha i}$ Eq. (A6), one can obtain

$$O(\varepsilon): \quad \frac{\partial M_0^{(\ell)}}{\partial t} + \frac{\partial M_{1i}^{(\ell)}}{\partial x_i} = 0$$
$$\frac{\partial M_{1i}^{(\ell)}}{\partial t} + \frac{\partial M_{2ij}^{(\ell)}}{\partial x_j} = 0 \tag{A7}$$

$$O(\varepsilon^2): \quad \frac{\partial M_0^{(\ell)}}{\partial t_2} = 0$$
$$\frac{\partial M_{1i}^{(\ell)}}{\partial t_2} = \frac{\partial}{\partial x_{1j}} \left[\Delta t \left(\tau - \frac{1}{2} \right) \left(\frac{\partial M_{2ij}^{(\ell)}}{\partial t_1} + \frac{\partial M_{3ijk}^{(\ell)}}{\partial x_{1k}} \right) \right] \tag{A8}$$

where

$$M_0^{(\ell)} = \sum_\alpha f_\alpha^{(\ell)}, \quad M_{1i}^{(\ell)} = \sum_\alpha f_\alpha^{(\ell)} e_{\alpha i}$$
$$M_{2ij}^{(\ell)} = \sum_\alpha f_\alpha^{(\ell)} e_{\alpha i} e_{\alpha j}, \quad M_{3ijk}^{(\ell)} = \sum_\alpha f_\alpha^{(\ell)} e_{\alpha i} e_{\alpha j} e_{\alpha k} \tag{A9}$$

Introducing Eq. (A9) into Eqs. (A7) and (A8), the differential equations for the first and second orders become

$$O(\varepsilon): \quad \frac{\partial h^{(\ell)}}{\partial t_1} + \frac{\partial h^{(\ell)} u_i^{(\ell)}}{\partial x_{1i}} = 0$$
$$\frac{\partial h^{(\ell)} u_i^{(\ell)}}{\partial t_1} + \frac{\partial h^{(\ell)} u_i^{(\ell)} u_j^{(\ell)}}{\partial x_{1j}} = -\frac{\partial}{\partial x_{1i}}(h^{(\ell)} c_s^2) \tag{A10}$$

$$O(\varepsilon^2): \quad \frac{\partial h^{(\ell)}}{\partial t_2} = 0$$
$$\frac{\partial h^{(\ell)} u_i^{(\ell)}}{\partial t_2} = \frac{\partial}{\partial x_{1j}}\left[\Delta t\left(\tau - \frac{1}{2}\right)\frac{\partial}{\partial t_1}\left(\delta_{ij} h^{(\ell)} c_s^2 + h^{(\ell)} u_i^{(\ell)} u_j^{(\ell)}\right)\right] + \frac{\partial}{\partial x_{1j}}\left[\Delta t\left(\tau - \frac{1}{2}\right)\frac{\partial}{\partial x_{1k}}\left(\frac{e^2}{3}(\delta_{jk} h^{(\ell)} u_i^{(\ell)} + \delta_{ik} h^{(\ell)} u_j^{(\ell)} + \delta_{ij} h^{(\ell)} u_k^{(\ell)})\right)\right] \tag{A11}$$

Selecting the relaxation time

$$\tau = 0.5 + 3\frac{\nu}{\Delta t e^2} = 0.5 + 3\frac{\Delta t \nu}{\Delta x^2} \tag{A12}$$

and the squared speed of sound

$$c_s^2 = \frac{1}{2} g \sum_{m=1}^{M} h^{(m)} = \frac{1}{2} g H \tag{A13}$$

Eqs. (A10) and (A11) become

$$O(\varepsilon): \quad \frac{\partial h^{(\ell)}}{\partial t_1} + \frac{\partial h^{(\ell)} u_i^{(\ell)}}{\partial x_{1i}} = 0$$
$$\frac{\partial h^{(\ell)} u_i^{(\ell)}}{\partial t_1} + \frac{\partial h^{(\ell)} u_i^{(\ell)} u_j^{(\ell)}}{\partial x_{1j}} = -\frac{\partial}{\partial x_{1i}}\left(\frac{g h^{(\ell)} \sum_{m=1}^{M} h^{(m)}}{2}\right) \tag{A14}$$

$$O(\varepsilon^2): \quad \frac{\partial h^{(\ell)}}{\partial t_2} = 0$$
$$\frac{\partial h^{(\ell)} u_i^{(\ell)}}{\partial t_2} = \nu \frac{\partial^2 h^{(\ell)} u_i^{(\ell)}}{\partial x_{1j} \partial x_{1j}} + \underbrace{\nu \frac{3}{e^2}\frac{\partial}{\partial t_1}\left[\frac{\partial}{\partial x_{1i}}\left(\frac{g h^{(\ell)} \sum_{m=1}^{M} h^{(m)}}{2}\right) + \frac{\partial h^{(\ell)} u_i^{(\ell)} u_j^{(\ell)}}{\partial x_{1j}}\right]}_{\mathbf{I}} \tag{A15}$$

The term **I** represents the numerical error. Substituting the second equation of (A14) into the term **I**, it becomes the second derivative of $h^{(\ell)} u_i^{(\ell)}$ with respect to t_1, which is small compared to the first derivative and is neglected. Combining the first and second order terms in Eqs. (A14) and (A15), the multilayer shallow water equations are recovered:

$$\frac{\partial h^{(\ell)}}{\partial t} + \frac{\partial h^{(\ell)} u_i^{(\ell)}}{\partial x_i} = 0$$
$$\frac{\partial h^{(\ell)} u_i^{(\ell)}}{\partial t} + \frac{\partial h^{(\ell)} u_i^{(\ell)} u_j^{(\ell)}}{\partial x_j} + \frac{\partial}{\partial x_i}\left(\frac{g h^{(\ell)} \sum_{m=1}^{M} h^{(m)}}{2}\right) = \nu \frac{\partial^2 h^{(\ell)} u_i^{(\ell)}}{\partial x_j \partial x_j} \tag{A16}$$

References

[1] Chen S, Doolen GD. Lattice Boltzmann method for fluid flows. Ann Rev Fluid Mech 1998;30:329–64.
[2] Succi S. The lattice Boltzmann equation for fluid dynamics and beyond. Oxford: Oxford University Press; 2001.
[3] McNamara GR, Zanetti G. Use of the Boltzmann equation to simulate lattice-gas automata. Phys Rev Lett 1988;61(20):2332–5.
[4] Kandhai D, Koponen A, Hoekstra AG, Kataja M, Timonen J, Sloot PMA. Lattice-Boltzmann hydrodynamics on parallel systems. Comput Phys Commun 1998;111(1–3):14–26.
[5] Zhong L, Feng S, Lou D, Gao S. Wind-driven, double-gyre, ocean circulation in a reduced-gravity, 2.5-layer, lattice Boltzmann model. Adv Atmos Sci 2006;23(4):561–78.
[6] Salmon R. The lattice Boltzmann method as a basis for ocean circulation modeling. J Marine Res 1999;57:503–35.
[7] Zhou JG, Causon DM, Mingham CG, Ingram DM. Numerical prediction of dam-break flows in general geometries with complex bed topography. J Hydraul Eng 2004;130(4):332–40.
[8] Zhou JG. Lattice Boltzmann simulations of discontinuous flows. Int J Modern Phys C: Comput Phys Phys Comput 2007;18(1):1–14.
[9] Thömmes G, Sea M, Banda MK. Lattice Boltzmann methods for shallow water flow applications. Int J Numer Methods Fluids 2007;55(7):673–92.
[10] Banda MK, Thömmes MSG. Lattice Boltzmann simulation of dispersion in two-dimensional tidal flows. Int J Numer Methods Eng 2009;77(6):878–900.
[11] Li Y, Huang P. A coupled lattice Boltzmann model for advection and anisotropic dispersion problem in shallow water. Adv Water Resour 2008;31(12):1719–30.
[12] Li Y, Huang P. A coupled lattice Boltzmann model for the shallow water-contamination system. Int J Numer Methods Fluids 2009;59(2):195–213.
[13] Audusse E. A multilayer Saint–Venant model. Discrete Contin Dynam Syst, Series B 2005;5(2):189–214.
[14] Abgrall R, Karni S. A relaxation scheme for the two-layer shallow water system. In: Hyperbolic problems: theory, numerics, applications, Part II, 2008. p. 135–44.
[15] Audusse E, Bristeau MO, Decoene A. Numerical simulations of 3D free surface flows by a multilayer Saint–Venant model. Int J Numer Methods Fluids 2008;56(3):331–50.
[16] Audusse E, Bristeau MO, Decoene A. 3D free surface flows simulations using a multilayer Saint–Venant model: comparisons with Navier–Stokes solutions. Numer Math Adv Appl 2006:181–9.
[17] Audusse E, Bristeau M-O. Finite-volume solvers for a multilayer Saint–Venant system. Int J Appl Math Comput Sci 2007;17(3):311–20.
[18] Guo Z et al. A comparative study of the LBE and GKS methods for 2D near incompressible laminar flows. J Comput Phys 2008;227(10):4955–76.
[19] Bhatnagar PL, Gross EP, Krook M. A model for collision processes in gases. I. Small amplitude processes in charged and neutral one-component systems. Phys Rev 1954;94(3):511–25.
[20] Zhou JG. Lattice Boltzmann methods for shallow water flows. Springer; 2004.
[21] Massaioli F, Amati G. Achieving high performance in a LBM code using OpenMP. Presented at fourth European workshop on OpenMP, Roma, 2002.
[22] Shankar NJ, Cheong HF, Sankaranarayanan S. Multilevel finite-difference model for three-dimensional hydrodynamic circulation. Ocean Eng 1997;24(9):785–816.
[23] Sanay R, Valle-Levinson A. Wind-induced circulation in semienclosed homogeneous, rotating basins. J Phys Oceanogr 2005;35(12):2520–31.
[24] Glorioso PD, Davies AM. The influence of eddy viscosity formulation, bottom topography, and wind wave effects upon the circulation of a shallow bay. J Phys Oceanogr 1995;25(6):1243–64.
[25] Takashi A. Derivation of the lattice Boltzmann method by means of the discrete ordinate method for the Boltzmann equation. J Comput Phys 1997;131(1):241–6.