Modeling Elastic Walls in Lattice Boltzmann Simulations of Arterial Blood Flow

Xenia Descovich[1,4*], Giuseppe Pontrelli[2], Sauro Succi[2], Simone Melchionna[3], Manfred Bammer[4]

[1]Vienna University of Technololgy, Institute of Analysis and Scientific Computing, Vienna, Austria; *xenia.descovich@tuwien.ac.at

[2]Consiglio Nazionale delle Ricerche, Istituto per le Applicazioni del Calcolo, Rome, Italy

[2]Consiglio Nazionale delle Ricerche, Istituto Processi Chimico-Fisici, Rome, Italy

[4]AIT Austrian Institute of Technology GmbH, Vienna, Austria

Simulation Notes Europe SNE 23(2), 2013, 93 - 100
DOI: 10.11128/sne.23.tn.10189
Received: Jan. 10, 2013 (Selected MATHMOD 2012 Postconf. Publ.); Revised Accepted: June 15, 2013;

Abstract. An essential part in the simulation of blood flow in arteries is the incorporation of the arterial elasticity by modeling the vessel wall and its interaction with the fluid inside the vessel. We suggest a simple approach for modeling elastic walls in lattice Boltzmann simulations of arterial blood flow that produces physically correct results. We have developed a simulation software that implements this approach combined with the lattice Boltzmann method and conducted numerical experiments on a generic vessel model. Preliminary results presented in this work are promising and encourage using this method for further simulations with real physiological parameters in medical applications.

Introduction

Cardiovascular diseases are the most common cause of death in industrialized countries [1]. Since experimental methods in the cardiovascular system are diffcult and limited, mathematical models and numerical methods to simulate the hemodynamic processes have gained importance in the past years. Research in that domain includes studies incorporating the whole arterial tree [2] as well as studies of only parts of it, e.g. a segment of an artery [3].

Since arteries are elastic and change in diameter depending on the blood pressure inside (which oscillates due to the periodic pumping of the heart), it is of particular importance to incorporate this elasticity in models of physiological flows in blood vessels.

Common numerical methods for blood flow simulations with elastic walls are complex. We present a simple method for modeling the blood flow in an artery and the elastic walls of the vessel. The flow field inside the vessel is computed by using the lattice Boltzmann method, which is a numerical approach for solving problems of computational fluid dynamics. In the lattice Boltzmann simulation, we include the developed model for the elastic walls. The model fulfills the essential properties of an elastic wall and respects the basic conservation laws.

This paper shows a short overview of the lattice Boltzmann method and its use in hemodynamics in Section 1. In Section 2, we present the method to model the elastic vessel wall. Numerical experiments presented in Section 3 show that the developed approach provides correct physical behavior. Section 4 gives an outlook on the possible application of the method in the simulation of blood flow in stented arteries.

1 Lattice Boltzmann Method

The lattice Boltzmann (LB) method is a mesoscopic approach based on the Boltzmann equation and can be used to solve various problems of computational fluid dynamics. It describes the dynamics of fictitious particles on nodes of a regular lattice.

The dynamics of the flow field are modeled by the

evolution of density distribution functions (also called `populations') $f_i(\mathbf{x},t) = f_i(\mathbf{x},\boldsymbol{c}_i,t),\ i = 1,\dots,q,$ which describe the probability of finding at time t a particle located at site $\mathbf{x}$ and traveling along the lattice in direction i with the speed $\boldsymbol{c}_i$. For a more detailed introduction to the LB method the reader is referred to [4] and [5].

1.1 LBGK model for blood flow simulation

The lattice Boltzmann method has been successfully applied to hemodynamics by many authors [6, 7, 3, 8], which has proven its value as an alternative to numerical methods based on the discretization of the Navier-Stokes equations of continuum mechanics.

For our simulations of the blood flow, we consider a so-called D2Q9 model (2 spatial dimensions, 9 directions i) and calculations are based on the lattice Boltzmann equation with single-time relaxation (LBGK approximation)

$$f_i(\mathbf{x}+\boldsymbol{c}_i\Delta t, t+\Delta t) - f_i(\mathbf{x},t) = \omega\Delta t\left(f_i - f_i^{eq}\right)(\mathbf{x},t) \tag{1}$$

with Δt being the temporal resolution and ω being the relaxation frequency. The right-hand side of (1) represents molecular collisions through a relaxation towards local equilibrium f_i^{eq} which is given by a second-order expansion of the Maxwell-Boltzmann equation. The fluid density ρ and the momentum $\rho\mathbf{u}$ are defined based on the distribution functions f_i:

$$\rho = \sum_{i=0}^{8} f_i \tag{2}$$

$$\rho\mathbf{u} = \sum_{i=0}^{8} f_i\,\boldsymbol{c}_i \tag{3}$$

Since we focus on large arteries, the flow in the vessel can be assumed to be Newtonian. Investigation of non-Newtonian flows in hemodynamics can be found elsewhere [9].

1.2 Elastic walls

In simulations of blood flow, it is important to consider the compliance of the vessel. Therefore, a model for the vessel wall has to be developed that describes its spatial displacement as it interacts with the flow dynamics. Fang et al. [6], for example, uses a parametrization of the vessel wall with special treatment for curved boundaries.

The method has been successfully applied to unsteady flow in elastic tubes in two dimensions (see [10]). However, the method is very complex in three dimensions where the vessel wall is described by means of surfaces.

For our approach, we have developed a simple method to model the elastic vessel which does not need a parametrization of the wall. It is based on the method of Leitner [3] and acts strictly locally like the LB method itself. By this, the complexity of the algorithm is not increased and the method can be used for simulations in two and three dimensions.

In the method of Leitner [3], lattice nodes can have two different states: fluid, representing the blood inside the vessel, and solid, describing the vessel wall. The compliance of the wall is modeled by changing the type of a node – from solid to fluid in the case of expansion and vice versa in the case of contraction of the vessel. This change of node type, which models the displacement of the wall, is dependent on the local pressure of the surrounding fluid nodes. To avoid a rupture of the vessel wall, Leitner [3] uses a cellular automaton (CA) with rules to update the wall in every simulation time step.

In the following section, we present an improved method to model the elastic wall that does not require the use of cellular automata.

2 Improved Method to Model Elastic Vessel Wall

Our approach for the elastic wall is based on the work of Leitner [3] which uses node type changes in a lattice as mentioned above to model the displacement of the wall. In order to ensure mass conservation in the model, we developed methods that redistribute the populations when a node changes its type. Node type changes are based on the local fluid properties and are performed by means of pressure thresholds, which will be explained below.

2.1 Wall modeling

We consider a computational domain of dimesions $N_x \times N_y$, where N_x and N_y are the number of lattice nodes (equally spaced) in direction x and y, respectively. Each node (x, y) can have two different states: fluid or solid. Solid nodes represent the tissue of the vessel, fluid nodes represent the blood inside the vessel. The initial configuration of solid and fluid nodes is gained by reading the data from a PGM binary image. Contrary to the modeling described by Leitner [3], the wall of the vessel is not situated on the solid nodes but is imagined to be located between the last fluid and first solid node in a given direction. All nodes that are not fluid are by default solid. Thus, the problem of rupture of the vessel wall does not occur and the approach does not require the use of CA.

In the following, the terms *destruction* and *creation* of nodes are used to signify the state change of a node. One must be aware that in this type of modeling, nodes are neither destroyed nor created – the geometrical domain being fixed of dimensions $N_x \times N_y$ – but that nodes change only their type. The state change `fluid to solid' will be termed as *destruction* and `solid to fluid' as *creation*.

Created fluid nodes need to be initialized with values of the density ρ and the velocity $\mathbf{u}$ (corresponding to the momentum $\rho\mathbf{u}$). This is done by averaging the populations from the fluid nodes surrounding the new fluid node (one average for each direction i) and assigning these values to the new node. The values of ρ and $\mathbf{u}$ can be computed based on the assigned values of the distribution functions f_i by using (2) and (3). Compared to the method of Leitner [3], who initializes new fluid nodes with an equilibrium distribution function, this approach includes also the non-equilibrium part of the populations, which is not negligible for nodes in proximity of the wall.

A mass conservation problem arises when nodes are created or destroyed in a way that the total number of fluid nodes changes. Mass is a priori not conserved as mass is added when initializing a new fluid node or subtracted when a fluid node is destroyed. Leitner [3] does not refer to this problem, and it is not clearly evident how it is circumvented. In order to ensure mass conservation in our approach, we have developed methods that rescale the populations in a part of the domain when a node type change occurs.

2.2 Population rescaling methods

Mass conservation is imposed each time fluid nodes change their type (from solid to fluid in the case of expansion and vice versa in the case of contraction of the vessel). Below, we present two methods to rescale the populations.

Local Rescaling. The local rescaling takes into account only the nearest neighbors of the node changing its state. At expansion, mass (density) is redistributed in the following way: Let (x, y) be a node changing its state from solid to fluid and $\sum \rho_{nb}$ the sum of the densities at fluid nodes neighboring node (x, y). After initializing the new
fluid node (x, y), its populations and those from the neighboring fluid nodes are rescaled by the factor $\frac{\sum \rho_{nb}}{\sum \rho_{nb} + \rho(x,y)}$. Density and velocity are computed based on these new populations. In this way, mass is conserved. The scaling factor is equal to the fraction $\frac{\text{old local mass}}{\text{new local mass}}$ and is smaller than 1. This approach models the transfer of mass to a new fluid node from the neighboring nodes.

In a similar way, mass is redistributed locally when a fluid node is destroyed, i.e., when a node changes its state from fluid to solid (contraction). The scaling factor is again $\frac{\text{old local mass}}{\text{new local mass}}$ and is in this case greater than 1. All fluid nodes surrounding the disappearing fluid node are rescaled by this scaling factor. By this procedure, mass is transfered from the destroyed node to the neighboring fluid nodes.

The local rescaling inuences the flow field in the vessel only locally. Depending on the fluid viscosity and the vessel geometry, it takes a certain time until this perturbation of the flow field is damped in the simulation.

Rescaling 'by columns'. This method of rescaling takes into account the whole column of nodes - the vessel can be considered as a sequence of `rings' adjoint to each other – in which a node type change occurs. When a node next to the wall changes its state (from solid to fluid or vice
versa), the populations of every node in the same column are rescaled to ensure mass conservation (a rescaling factor similar to the one above is used.)

2.3 Pressure thresholds

Node type changes occur depending on the local pressure surrounding a given node. In our model, pressure thresholds are assigned to each node, increasing with the radius of the vessel segment. Nodes that are further away from the center of the vessel have a higher threshold that has to be exceeded for an outwards displacement of the wall, i.e., for changing the type of the neighboring solid node to fluid.

A linear relationship between the pressure p and the radius R is assumed, similar to the one of the pulmonary blood vessels (see [11]):

$$p = p_0 + \alpha(R - R_0) \tag{4}$$

Here, α is a compliance constant. R_0 is the radius when the transmural pressure is zero and p_0 the pressure at a node located at distance R_0 from the center of the vessel. The linear pressure-radius relationship is a good approximation for large arteries [10].

Pressure thresholds are computed based on this linear relationship and assigned to each node. Since the vessel is embedded in a lattice of fixed dimensions, there are nodes that can never become fluid considering that there is a maximum expansion of the vessel. To all those nodes, a pressure threshold exceeding the allowed pressure range is assigned. This prevents that, in the simulation, the vessel expands more than is physiologically possible.

Our model uses the precomputed thresholds to simulate wall displacement by appropriate node type changes. The following shall exemplarily explain our procedure of node type changes based on the pressure thresholds. Let (x, y) be a solid node neighboring a fluid node $(x, y-1)$ at the upper wall of the vessel. The wall is imagined to be located between those two nodes. Let $p_t(x, y)$ be the pressure threshold of node (x, y). If the pressure at node $(x, y-1)$ exceeds $p_t(x, y)$, i.e. $p(x, y-1) > p_t(x, y)$, the node (x, y) becomes fluid, i.e., the vessel expands. Conversely, let us suppose that the nodes (x, y) and $(x, y-1)$ are both fluid. If $p(x, y) < p_t(x; y-1)$, node (x, y) changes its state from fluid to solid, i.e., the vessel contracts.

3 Simulation and Preliminary Results

We implemented a simulation software for the lattice Boltzmann algorithm combined with our elastic wall model using the programming language C. The program includes the rescaling methods and the pressure threshold algorithm described above. Using our software program, we conducted numerical experiments to show the feasibility of our approach.

3.1 Numerical experiments

The main objective of the numerical experiments presented here is to prove that the model provides correct physical behavior. Simulation with real physiological conditions will be performed in a later work.

For our experiments, we consider a straight channel with flat walls modeling the vessel. We impose periodic boundary conditions in direction of the channel to model an infinite long tube. This condition will be replaced by inlet/outlet boundary conditions in a later work. At the wall, we impose bounce-back boundary conditions which represent a no-slip condition (i.e., the fluid velocity at the wall is zero.) Furthermore, the pressure is increased `manually' at a certain time, just enough to provoke an increase of the channel radius of one unit (expansion), and at a later time decreased again, just enough to induce a decrease of the radius of one unit (contraction). Thus, we investigate the behavior of the flow field for only one cycle of expansion and subsequent contraction (of the whole channel). The uniform pressure increase/decrease is used only to test the method. In a later stage, it will be replaced by an oscillating pressure gradient at the inlet simulating the periodic pumping of the heart.

For the simulation, the following physical parameters are used: viscosity $\nu = \frac{1}{3}$, maximum velocity $U_{max} = 0.01$, and initial density $\rho_0 = 1.0$. All parameters and variables are normalized and thus dimensionless. The computational domain is fixed with 200 × 100 nodes.

As mentioned above, a straight channel is considered having the macroscopic variables ρ (and thus p) and $\mathbf{u}$ at nodes with the same y-coordinate equal, respectively. Furthermore, all nodes with the same y-coordinate have the same pressure threshold. By this, the flat wall of the straight channel is displaced at all nodes with the same y-coordinate (i.e., one layer of nodes) within a single time step.

To induce an expansion in our model, the pressure is increased manually by adding an amount Δp to the pressure of all fluid nodes at a certain time t_{add} and then decreased again at time $t_{sub} > t_{add}$. We chose Δp so that the wall is moved by only one layer at the lower and upper boundary. Due to the forced increase (decrease) of pressure at t_{add} (t_{sub}), mass increases (decreases) by a value proportional to Δp at t_{add} (t_{sub}).

In order to check whether the implemented method provides correct physical behavior, the following steps are executed:

1. Wait for fully developed flow (until time t_0).
2. At time t_0 ($= t_{add}$), when flow is fully developed, add mass by adding a small value Δf_i to the populations f_i at each node. This corresponds to an increase of pressure by $\Delta p = c_s^2 \Delta\rho$ in each node, with $\sum_{i=0}^{8} f_i$ and $c_s = 1/\sqrt{3}$ being the speed of sound. Then wait for the flow to be fully developed (until time t_1).
3. At time t_1, expansion occurs because the pressure thresholds at the wall are exceeded due to the increase of pressure at time t_0. The channel is expanded by one layer at the upper and lower wall, respectively. The populations f_i are rescaled so as to ensure mass conservation. Then wait for fully developed flow (until time t_2).
4. At time t_2 ($= t_{sub}$), reduce mass by subtracting Δf_i from the populations f_i at each node. This corresponds a decrease of pressure. Wait again until flow is fully developed (until time t_3).
5. At time t_3, contraction occurs because the pressure has fallen below the pressure threshold due to the decrease of pressure at time t_2. The radius of the channel is reduced by one unit, so one layer of nodes is destroyed at the lower and upper wall, respectively. The populations f_i are rescaled so as to ensure mass conservation.

In this procedure, we ensure that mass is conserved any time except when mass is added (step 2) or subtracted (step 4) by proper rescaling. The results of this approach are presented hereafter.

3.2 Simulation results

Comparison between analytical solution and simulation result. The analytical solution of a 2-dimensional fully developed steady flow in a channel of width H, driven by a constant pressure gradient F and with constant viscosity ν (Newtonian flow assumed) is given by the following formula (Poiseuille flow):

$$\mathrm{u}_{\mathrm{analytical}}(y) = \frac{-F(y - y_{\mathrm{lower}})(y - y_{\mathrm{upper}})}{2\nu} \qquad (5)$$

Herein, y_{upper} and y_{lower} denote the y-coordinates of the upper and lower boundary, respectively. Thus, $H = (y_{\mathrm{upper}} - y_{\mathrm{lower}})$. The pressure gradient F is given by

$$F = \frac{8\nu U_{max}}{H^2} \qquad (6)$$

We compared our simulation output to the analytical solution using the same parameters. Figure 1 displays the velocity profile of the exact solution and the result of the numerical simulation of a symmetric straight channel. It can be observed that the computed velocity profile (blue crosses) and the analytical solution by Poiseuille (green dashed line) coincide.

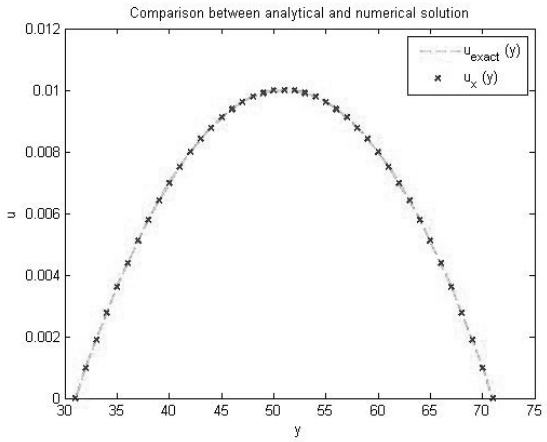

Figure 1: Velocity: Comparison between analytical solution and numerical result.

Approach for testing the modeling of elasticity using local rescaling. We conducted a simulation in the straight channel using the local rescaling method to show that it follows expected physical behavior. The procedure described below is only an artificial setup to enforce expansion and contraction through pressure increase/decrease for conducting this simulation experiment.

For this simulation, the problem of expansion and contraction has been separated in the parts explained in Section 3.1 and local rescaling has been used. Figure 2 shows the total mass (equal to the sum of the density ρ at each fluid node of the lattice) over time. It can be seen that mass is conserved at expansion (occurring at time $t = 3101$) and contraction (occurring at time $t = 4276$). After the wall has reached its initial position again, i.e., after having performed the steps 1 to 5 cited above, the value of the total mass is the same as the initial value. However, to reach the same value of the total mass as at the beginning, Δf_i in step 4 has to be multiplied by $\frac{H}{H+2}$, H being the width of the initial channel. This is due to the fact that the diameter (width) of the channel increases by 2 after expansion, i.e., there are two layers of fluid nodes more than before expansion. Thus, Δf_i, being subtracted at every node, has to be reduced by this proportionality factor to take into account the increase of nodes.

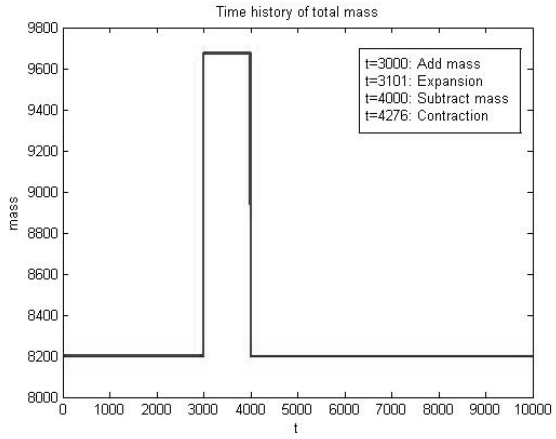

Figure 2: Total mass as a function of time.

Figure 3 depicts the density at a fluid node next to the wall resulting from this simulated experimental setup. It can be observed that the density oscillates after expansion or contraction due to the propagation of the local perturbation throughout the channel (oscillation period) created by the local rescaling.

The duration of the oscillations is inuenced by the fluid parameters (viscosity). The value of ρ returns to the initial value after one cycle (expansion and subsequent contraction). Due to the local rescaling of the populations at the changed nodes and their neighbors, the value of ρ decreases at expansion and increases at contraction.

The resulting velocity component u_x at a fluid node next to the wall is displayed in Figure 4. Small oscillations occur at expansion and contraction due to the local perturbation of the flow field. After one cycle, the value of u_x returns to the initial value (when flow is fully developed).

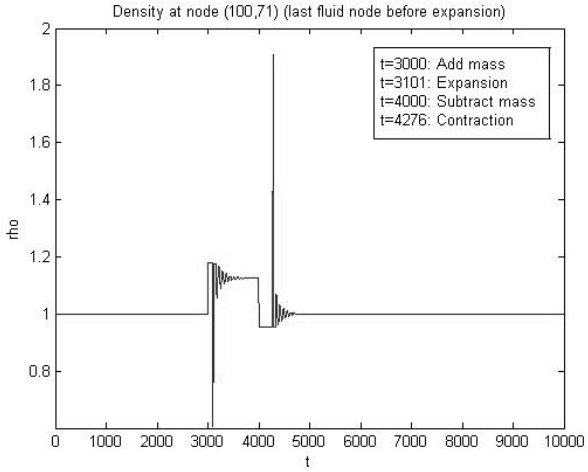

Figure 3: Density at fluid node next to the upper wall as a function of time (local rescaling used).

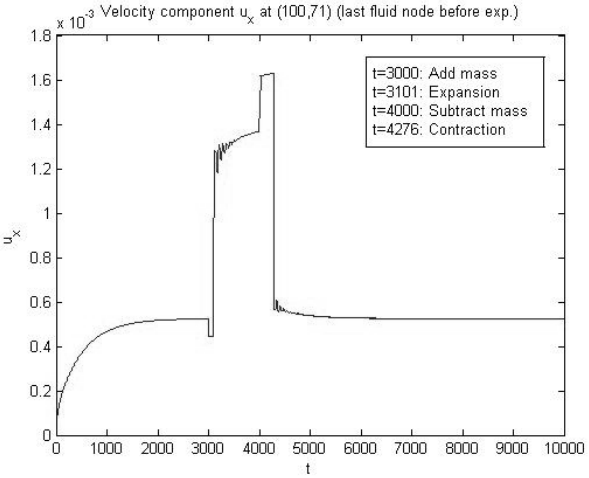

Figure 4: Velocity component u_x at fluid node next to the upper wall as a function of time (local rescaling used).

The simulation results presented above show expected physical behavior. Mass is conserved and initial values of the density and the velocity are recovered after one cycle of expansion and subsequent contraction. The oscillatory transient lasts a few hundred time steps, which is an expected duration for the chosen viscosity. As described, the method works strictly locally as does the lattice Boltzmann method itself. This allows straight-forward implementation of the method in lattice Boltzmann simulations and makes it suitable for parallel computation.

Approach for testing the modeling of elasticity using rescaling 'by Columns'. For this simulation, the same steps 1–5 cited in Section 3.1 have been performed and rescaling `by columns' has been used. The time history of the total mass shows the same behavior as the one for the local rescaling method. Contrary to the local rescaling method, where contraction (as well as expansion) occurs at both the upper and lower wall at the same time step in the simulation, the contraction of the channel when using rescaling `by columns' is split into two steps. First, contraction at the lower boundary occurs followed by a contraction at the upper boundary a few hundred time steps later. This is related to the order in which the lattice nodes are processed in the implementation (here, from lower boundary to upper boundary). Since rescaling `by columns' does not affect only the nearest neighbors of a node becoming fluid (as in the local rescaling method), but the populations of all nodes in the same column, a node type change at the lower wall affects the flow field at the upper wall in the same time step. As a consequence, the condition for expansion or contraction at the upper wall is not necessarily fulfilled anymore as soon as a node type change (and thus rescaling of the whole column) has occurred at the lower wall.

Figure 5 displays the density at a fluid node next to the wall. It can be observed that the density oscillates much less after expansion (contraction) compared to the oscillations occurring with the local rescaling method as local perturbations are limited. Furthermore, oscillations are completely absent at contraction. The value of ρ returns to the initial one after one cycle (expansion and subsequent contraction) as expected.

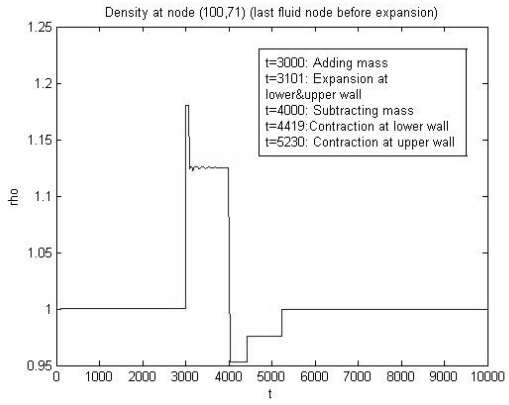

Figure 5: Density at fluid node next to the upper wall as a function of time (rescaling `by columns' used).

The time history of the velocity component u_x at a fluid node next to the wall, depicted in Figure 6, exhibits a similar shape as the time history of u_x when using local rescaling, but without oscillations at expansion and contraction. After one cycle, the initial value of u_x is recovered.

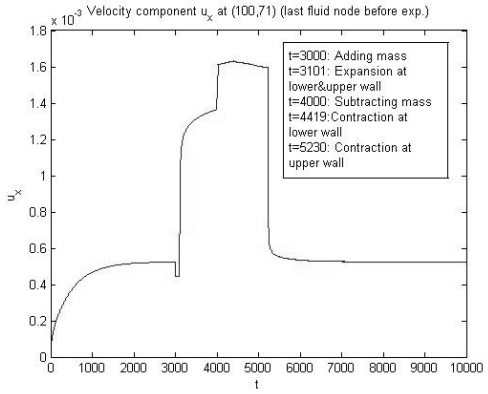

Figure 6: Velocity component u_x at fluid node next to the upper wall as a function of time (rescaling `by columns' used).

Although the rescaling `by columns' provides expected physical behavior (mass conservation, recovery of the initial values after one cycle) and minimizes local perturbation, it exhibits the drawback that node type changes at one wall boundary affect the flow field within the whole channel without propagation latency. Since this specific effect does not correspond to real fluid dynamics, the method can only be used in special cases where this side effect is negligible. Therefore, the rescaling `by columns' method minimizes perturbations in the simulation but lacks the generality of the local rescaling method.

4 Outlook

The aim of the future work is the simulation of blood flow in stented arteries. A stent is a wire metal mesh inserted into a blood vessel to prevent its occlusion. Due to the geometry and the different elasticity of the stent, the behavior of the blood flow changes and thus, turbulences can occur. It is assumed that those turbulences can cause a renewed narrowing of the vessel, so-called *in-stent restenosis*, which is a pathobiologic process prevalently occurring after stent implantation [12].

The modeling of elastic walls presented in this work has the advantage that it can also be applied to stented arteries. For the modeling, we consider three different types of nodes: `fluid`, representing the blood inside the vessel, `tissue`, describing the tissue of the vessel, and `stent`, representing the stent. Since the stented part is stiffer than the rest of the vessel, nodes of type `stent` will have higher pressure thresholds. This enhanced model will be implemented in our simulation software and further elaboration of our approach will be reported in a later work.

5 Conclusion

We presented a simple approach for modeling elastic walls in lattice Boltzmann simulations of blood flow in arterial segments. The described method provides correct physical behavior of vessel walls interacting with the simulated blood flow. The preliminary results are promising and encourage extending our approach to allow simulations of stents placed in arteries to investigate their inuence on the flow field.

Acknowledgement

This research was accomplished within the context of the project `BioCompatible Materials and Applications – BCMA' initiated by the AIT Austrian Institute of Technology GmbH. It was partly funded by AIT and the federal state of Lower Austria and co-financed by the EC (EFRE).

References

[1] Eurostat. *Health statistics – Atlas on mortality in the European Union*. Offce for Offcial Publications of the European Communities, Luxembourg; 2009.

[2] Kropf J. *Multiscale Blood Flow Modelling and Simulation*. Ph.D. thesis, Vienna University of Technology; 2007.

[3] Leitner D. *Simulation of Arterial Blood Flow with the Lattice Boltzmann Method*. Ph.D. thesis, Vienna University of Technology; 2007.

[4] Succi S. *The Lattice Boltzmann Equation*. Oxford University Press, Oxford; 2001.

[5] Wolf-Gladrow D. *Lattice-Gas Cellular Automata and Lattice Boltzmann Models: An Introduction*. Springer, 2000.

[6] Fang H, Wang Z, Lin Z, Liu M. Lattice Boltzmann method for simulating the viscous flow in large distensible blood vessels. *Physical Review E*. 2002; 65(5):51925

[7] Artoli A. *Mesoscopic computational haemodyna-mics*. Ph.D. thesis, University of Amsterdam; 2008.

[8] Melchionna S, Bernaschi M, Succi S, Kaxiras E, Rybicki F, Mitsouras D, Coskun A, Feldman C. Hydrokinetic approach to large-scale cardiovascular blood flow. *Computer Physics Communications*. 2010; 181(3):462–472

[9] Pontrelli G, Ubertini S, Succi S. The unstructured lattice Boltzmann method for non-Newtonian flows. *Journal of Statistical Mechanics: Theory and Experiment*. 2009; 6:06005

[10] Hoekstra A, van't Hoff J, Artoli A, Sloot P. Unsteady flow in a 2D elastic tube with the LBGK method. *Future Generation Computer Systems*. 2004; 20(6):917–924

[11] Fung Y. *Biomechanics: Circulation*. Springer Verlag; 1997.

[12] Lowe H, Oesterle S, Khachigian L. Coronary in-stent restenosis: current status and future strategies. *Journal of the American College of Cardiology*. 2002: 39(2):183–193