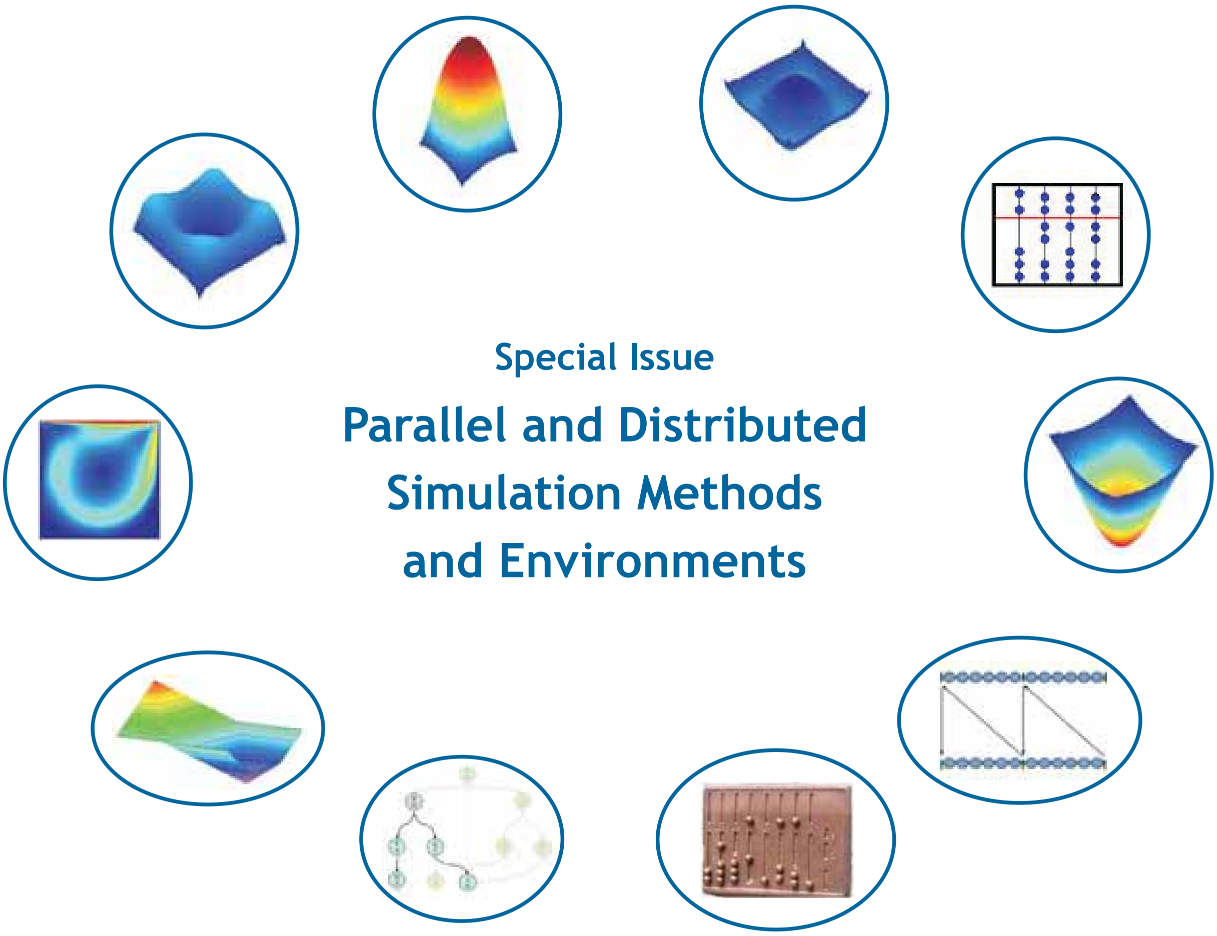
SNE SIMULATION NEWS EUROPE



Special Issue

Parallel and Distributed Simulation Methods and Environments

Volume 16 Number 2

September 2006, ISSN 0929-2268

Journal on Developments and Trends in Modelling and Simulation

Special Issue

Dear readers,

We are glad to present the first SNE Special Issue - a Special Issue on 'Parallel and Distributed Simulation Methods and Environments'. The idea for special issues was born in ASIM, the German Simulation Society. As there was and as there still is a need for state-of-the-art publications in topics of modelling and simulation, ASIM first tried to publish monographs on this subject. But publication of such books showed disadvantages: too slow production time, too high costs, and lack of publication issues. ASIM, seeking for alternatives, contacted ARGESIM with the idea of SNE Special Issues - while ARGESIM itself thought on Special Issues, because of lack in publication space in the regular SNE issues. Now, one year after the first contact, we can present the first Special Issue, edited by Thorsten & Sven Pawletta from University Wismar, Germany.

The editorial policy of SNE Special Issues is to publish high quality scientific and technical papers concentrating on state-of-the-art and state-of-research in specific modeling and simulation oriented topics in Europe, and interesting papers from the world wide modeling and simulation community. This Special Issue 'Parallel and Distributed Simulation Methods and Environments' (SNE 16/2), will be sent to all ASIM members - together with the regular SNE 16/1 (SNE 46), and sample copies will be sent to other European Simulation Societies; furthermore, it is available on basis of an individual subscription of SNE - SNE Special Issues are open for everybody, for publication and subscription (not only for ASIM). We think also on Special Issues publishing selected papers from EUROSIM conferences.

We hope, you enjoy this Special Issue, which presents state-of-the-art in parallel and distributed simulation, from theory with lookahead formulas via implementation with HLA and other systems to applications in ship desgin and blood flow simulation.

It is planned to publish a SNE Special Issue each year, for 2007 a Special Issue on 'Verification and Validation' (Guest Editor Sigrid Wenzel, University Kassel) is scheduled (SNE 17/2). I would like to thank all people who helped in managing this first Special Issue, especially the Guest Editors, Thorsten and Sven Pawletta from Wismar University.

Felix Breitenecker, Editor-in-Chief SNE; *Felix.Breitenecker@tuwien.ac.*at

Content

SNE Editorial Board

Felix Breitenecker (Editor-in-Chief), Vienna Univ. of Technology, *Felix.Breitenecker@tuwien.ac.at*
Peter Breedveld, University of Twenty, Div. Control Engineering, *P.C.Breedveld@el.utwente.nl*
Francois Cellier, ETH Zurich, Inst. f. Computational Science / University of Arizona, *fcellier@inf.ethz.ch*,
Russell Cheng, Fac. of Mathematics / OR Group, Univ. of Southampton, *rchc@maths.soton.ac.uk*
Rihard Karba, University of Ljubljana, Fac. Electrical Engineering, *rihard.karba@fe.uni-lj.si*
David Murray-Smith, University of Glasgow, Fac. Electrical & Electronical Engineering; *d.murray-smith@elec.gla.ac.uk*
Horst Ecker, Vienna Univ. of Technology. Inst. f. Mechanics, *Horst.Ecker@tuwien.ac.at*
Thomas Schriber, University of Michigan, Business School *schriber@umich.edu*
Sigrid Wenzel, University of Kassel, Inst. f. Production Technique and Logistics, *S.Wenzel@uni-kassel.de*

Guest Editors Special Issue *Parallel and Distributed Simulation Methods and Environments*
Thorsten Pawletta, *pawel@mb.hs-wismar.de*
Sven Pawletta, *s.pawletta@et.hs-wismar.de*
Res. Group Computational Engineering and Automation,
Wismar University, 23952 Wismar, Germany
WWW.MB.HS-WISMAR.DE/cea

SNE Contact
SNE-Editors / ARGESIM
c/o Inst. f. Analysis and Scientific Computation
Vienna University of Technology
Wiedner Hauptstrasse 8-10, 1040 Vienna, AUSTRIA
Tel + 43 - 1- 58801-10115 or 11455, Fax - 42098
sne@argesim.org; WWW.ARGESIM.ORG

Editorial Info - Impressum
SNE Simulation News Europe ISSN 1015-8685 (0929-2268).
Scope: Development in modelling and simulation, benchmarks on modelling and simulation, membership info for EUROSIM and Simulation Societies.
Editor-in-Chief: Felix Breitenecker, Inst. f. Analysis and Scientific Computing, Mathematical Modelling and Simulation, ,
Vienna University of Technology, Wiedner Hauptstrasse 8-10,
1040 Vienna, Austria; *Felix.Breitenecker@tuwien.ac.at*
Layout: A. Breitenecker, ARGESIM TU Vienna / Linz;
Anna.Breitenecker@liwest.at
Printed by: Grafisches Zentrum TU, Wiedner Hauptstr. 8-10, A-1040, Wien
Publisher: ARGESIM and ASIM
ARGESIM, c/o Inst. for Analysis and Scientific Computation, TU Vienna,
Wiedner Hauptstrasse 8-10, A-1040 Vienna, Austria, and
ASIM (German Simulation Society), c/o Wohlfartstr. 21b, 80939München

© ARGESIM / ASIM 2006

Parallel Computation in Blood Flow Simulation using the Lattice Boltzmann Method

Daniel Leitner, Siegfried Wassertheurer, ARC Seibersdorf research GmbH, Vienna, Austria
{daniel.leitner, siegfried.wassertheurer}@arcsmed.a
Felix Breitenecker, Vienna University of Technology, Vienna, Austria
Felix.Breitenecker@tuwien.ac.at
Michael Hessinger, Andreas Holzinger, Medical University Graz
{michael.hessinger, andreas.holzinger}@meduni-graz.at

Lattice Boltzmann Models (LBM) are widely used to solve fluid mechanical problems in engineering and biomedical applications. First a brief introduction of LBM is given and an example model with three spatial dimensions is introduced. The model is relevant for blood flow simulation because it uses Reynolds and Womersley numbers found in hemodynamics with a realistic time dependent pressure gradient as a boundary condition. A big advantage of LBM is the possibility of easy parallelization. Therefore different approaches of implementations are discussed. To test parallelization, the example model is used as a benchmark. The simulation times are compared calculating the problem in parallel on one to four processors.

SNE 16/2, September 2006

Introduction

In the western industrial countries cardiovascular diseases are the most frequent cause of death. Therefore a lot of research is done to get a better understanding of the *cardiovascular system* (CVS). To simulate the CVS, various models of different accuracy are used and often coupled together to describe the circulation on different spatial and temporal scales [1].

In this work a LBM is used to simulate the blood flow in three spatial dimensions, solving the Navier-Stokes equation with the Lattice Bhatnagar-Gross-Krook (LBGK) method ([2, 3]). The main advantages of the LBGK method are that it is simple to implement and to parallelize which enables an efficient computation. Furthermore it is a bottom up approach. Thus the algorithm can be interpreted physically in every step, which makes the method very intuitive.

The calculations in computer fluid dynamics (CFD) and specially blood flow simulation in three spatial dimensions are very time consuming. Adequate computer systems often make use of multiple CPUs. Therefore it is fundamental for algorithms in CFD to support parallelization. In this work the LBGK method in three dimensions with 15 degrees of freedoms is tested under these aspects.

The LBGK method is tested on a Dell Precision 670 machine containing two Intel Xeon dual core processors with 2.8 GHz.

The test case simulates unsteady flow in a rectangle. The time dependent pressure gradient, fluid viscosity and the resulting Reynolds numbers lie in a range relevant for blood flow simulation.

The results of the simulation are in best accordance with the analytical results obtained by Womersley, see [4]. The example shows the benefits of the LBGK method for blood flow simulation and thus acts as a relevant example for comparing computation times of multiple CPUs.

1 The LBGK D3Q15 Model for Blood Flow Simulation

For simulating the flow field we use a LBGK model [2, 3], which is proved to be capable of dealing with pulsative flow within the range of Reynolds and Womersley numbers existing in large arteries [5, 6].

The LBGK model is based on a statistical description of a fluid in terms of the Boltzmann equation. The Boltzmann equation with single relaxation time is given by

$$\frac{\partial f}{\partial t} + \xi \cdot \nabla f = -\frac{1}{\lambda}(f - f^{eq})$$

This equation is discretised in the spatial domain, in velocity space and in time, yielding

$$f_i(x + c\cdot \mathbf{e}_i \Delta t, t + t) - f_i(x,t) = \\ -\frac{1}{\lambda}(f_i(x,t) - f_i^{eq}(x,t))$$

where $c = Dx/Dt$, Dx is the lattice grid spacing, and Dt the time step.

The particle distribution functions f_i evolve on a regular grid and represent particles travelling on the link $\mathbf{e}_i$ (Figure 1), thus $f_i\,(x, t)$ refers to the particle distribution on the lattice node x at time t on the link $\mathbf{e}_i$.
Note that $f_0\,(x, t)$ represents the particles resting at node x, thus i = {0,…, 14} in the D3Q15 LBGK method. In the name D3Q15, D3 is referring to the three spatial dimensions, Q15 to the 15 degrees of freedom in a node.

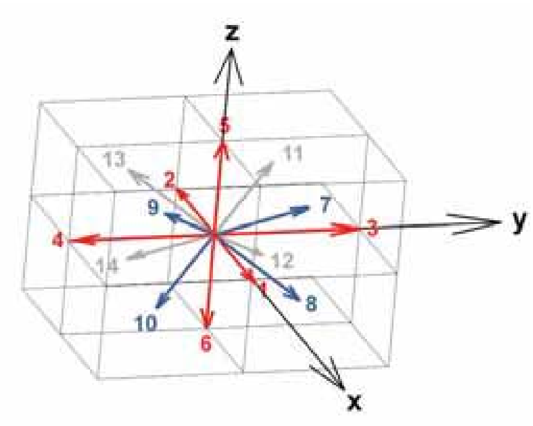


Figure 1: The velocity directions in the LBGK D3Q15 model

The equilibrium density distribution $f^{eq}(x, t)$ depends solely on the density $\rho\,(x, t)$ and the velocity $u(x, t)$ of a lattice node x. The density ρ and the velocity u are obtained from the density distribution function f_i:

$$\rho(x,t) = \sum_i f_i(x,t)$$

$$\rho(x,t)u(x,t) = \sum_i c\cdot \mathbf{e}_i f_i(x,t)$$

The equilibrium is defined as

$$f_i^{eq}(\rho,u) = \rho(\omega_i + 3\omega_i \mathbf{e}_i\cdot u + \\ + \frac{9}{2}\omega_i(\mathbf{e}_i\cdot u)^2 - \frac{3}{2}\omega_i u\cdot u)\cdot$$

with the weight coefficients

$$\omega_i = \begin{cases} 2/9, & i = 0 \\ 1/9, & i = 1,\dots,6 \\ 1/72, & i = 7,\dots,14 \end{cases}$$

The mass and momentum equations can be derived from the model via multiscale expansion as

$$\frac{\partial\rho}{\partial t} + \nabla\cdot(\rho\, u) = 0$$

$$\frac{\partial(\rho\, u)}{\partial t} + \nabla\cdot(\rho\, u u) = \\ -\nabla p + \nu(\nabla^2(\rho\, u) + \nabla(\nabla\cdot(\rho\, u)))$$

$$p = {c_s}^2\rho,\quad c_s = \frac{c}{\sqrt{3}},\quad \nu = (2\tau - 1)c^2\frac{Dt}{6}$$

where p is the pressure, c_s is the speed of sound, and ν is the kinematic viscosity. The mass and momentum equations are exactly the same as the compressible Navier-Stokes equation, if the density variation is small enough.

Thus the compressible Navier-Stokes equation is recovered in the incompressible limit. If the density fluctuation is assumed to be negligible, the incompressible Navier-Stokes equation can derived directly via the Chapman-Enskog procedure. Because of the expansion in the velocity term the lattice Boltzmann method is only applicable to low Mach number hydrodynamics.

2 Implementation Notes for Parallel Computation

The strictly local nature of the LBGK method enables an easy parallelization of the algorithm.
The pseudo code for a one processor machine can be easily formulated, as can be seen in the following code snippet:

```
while(running) {
  for each node {  calc kinetic equ }
  for each node {  calc equilibrium }
}
```

Furthermore note, that the discretised Boltzmann equation (see before) is normally formulated as kinetic equation ($\omega = 1/\lambda$):

$$f_i(x + c \cdot \mathbf{e}_i \Delta t, t + \Delta t) = (1-\omega) f_i(x,t) - \omega f_i^{eq}(x,t)$$

To adjust the method for multiple CPUs the set of nodes must be simply distributed on the threads, each running on one processor. In each calculated time step the threads must wait for each other two times:

```
while(running) {
  for each thread { calc kinetic
  equation for all nodes }
wait for all threads
  for each thread { calc equilibrium
  for all nodes }
wait for all threads
```

The way the nodes are distributed on the threads is important. Neighbouring nodes should be processed from one thread for optimal cache usage. Special care must be taken when there are complex boundary nodes which need more computation time. The nodes should be distributed in a way that every thread needs exactly the same time to calculate its nodes.

For the computation a data structure is needed to store the densities, equilibria and information about neighborhood. Basically there are three approaches for the representation of the states: lattice, list, and object.

Lattice Implementation

The simplest approach is to store all states in a three - dimensional lattice and a three-dimensional array, respectively.
Two lattices are needed. In one time step the new values are calculated from an input lattice A into a lattice B, in the next step from B to A and so on, see Figure 2. There is no need to store extra information about neighborhood because the position of the data in the lattice and therefore the position in memory is known.

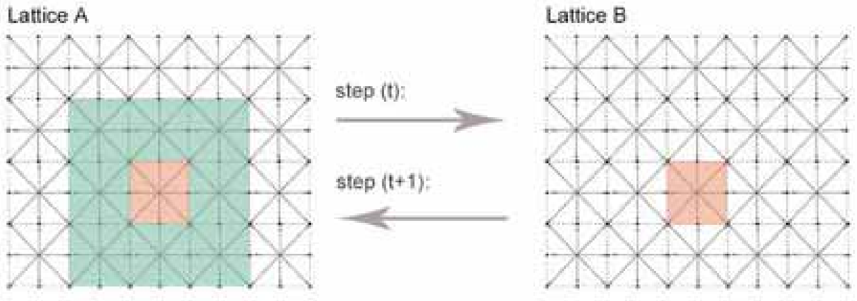


Figure 2: Implementation with two lattices.

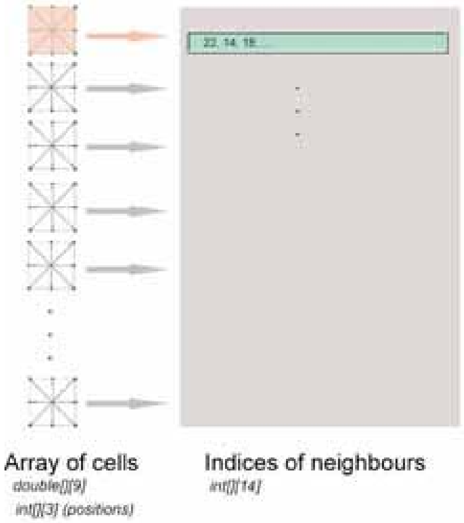


Figure 3: Implementation with a list.

List Implementation

A major drawback of the method with lattices is when small structures lie in a big volume, like arteries in tissue. Most of the nodes are boundary nodes and there is only a small percentage of fluid nodes.
The idea is to calculate and store only relevant nodes, which are fluid nodes and no slip nodes neighbouring them. The data of these relevant nodes are written into a list. Two sets of states are stored for every node, old states and new states. In addition extra information about neighbourhood is needed, see Figure 3. Thus the indices of the neighbours must be stored in a table for each node in the list.

Further the positions of the nodes in space are needed, therefore three additional values must be stored for every node. Note that in the implementation with lattices this information is provided in a natural way.

Implementation with Objects

A more intuitive but slightly slower and more storage demanding approach is to represent every node as an object. The states and positions in space are stored within the object. Neighbourhood is realized by storing the references to the neighbours, see Figure 4.

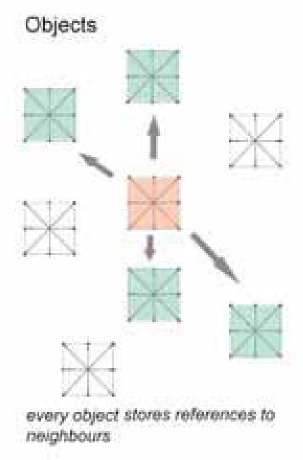


Figure 4: Object oriented implementation.

This needs more memory on 64 Bit machines because a reference needs 64 Bits while an integer used for an index of a list needs only 32 Bits of memory.

The object oriented approach has been used in this work, because it simplifies experiments with new node types.

3 Simulation Results

The example models are boxes of 20*20 nodes with a length of 20, 100, 200 nodes. The boxes are surrounded with no-slip nodes describing the walls. At the top and bottom of the box a special boundary condition is applied for describing the time dependent pressure gradient. For more information about boundary conditions for LBGK methods the reader may refer to [2]. An approach to model elastic walls of arteries is given in [7].
The results of the simulation are presented in Figure 5. They are in best accordance with analytic solutions presented by Womersley.

The simulation is done on a Dell Precision 670 containing two Intel Xeon dual core processors with 2.8 GHz with Windows XP SP2 and J2SE 5 Update 7.

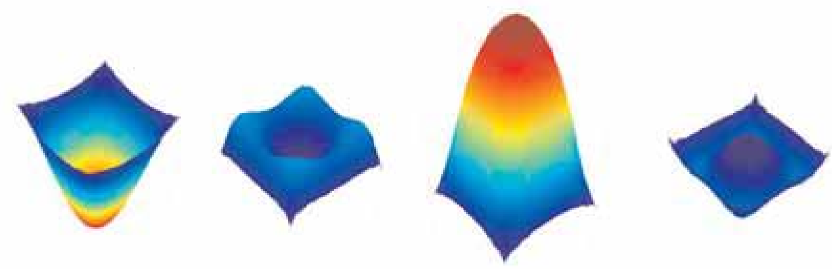


Figure 5: Velocity profiles of unsteady flow.

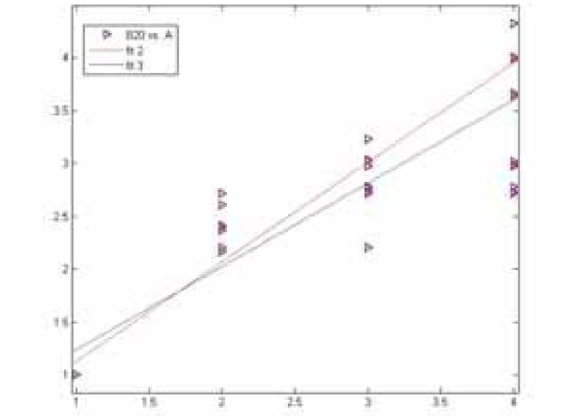


Figure 6: Simulation times of a 20*20*20 box (8000 nodes).

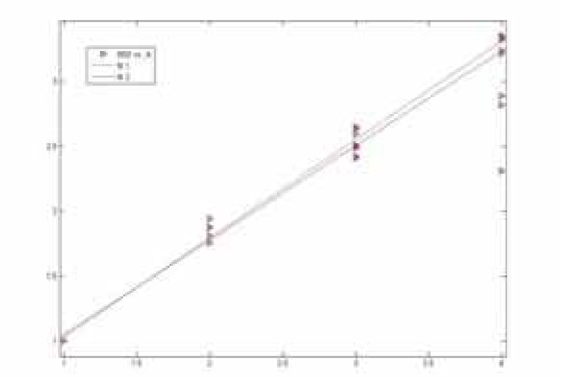


Figure 7: Simulation times of a 20*20*50 box (20000) nodes.

Time is measured over 100 time steps. The simulation is done 20 times on one to four processors. The run times are compared to the calculation time of one processor. Thus ideally two processors should work exactly with twice the perfomance as one.
The blue line in Figures 6-8 is the regression line of all sample points, the red line is the regression line of all points except the three worst results.

Simulation of the Box with 20*20*20 nodes: The smallest experiment with only 8000 nodes works very well and scales nearly linear. Eye catching is that two CPUs work have more than twice the performance than one CPU (see Figure 6). An explanation for this is the architecture of the computer. When two threads are used they run on the cells of only one CPU, while the second CPU can serve the operation system.

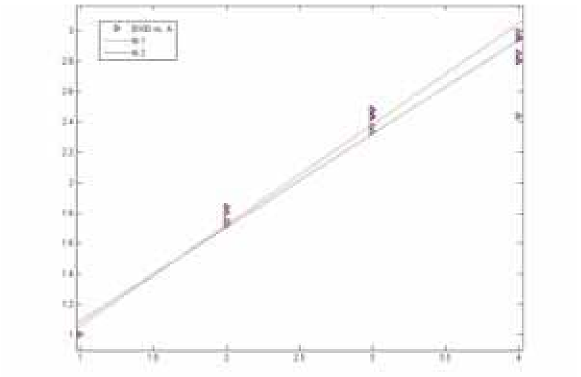


Figure 8: Simulation times of a 20*20*100 box (40000) nodes.

The threads on the two cells can use the cache in an optimal way. In this small example four threads are most of the times really four times faster than one thread.

Simulation of the Box with 20*20*50 nodes: The slightly bigger example with 20000 nodes shows a different behaviour. Because the simulation time is longer, more influences from the operation system affect the simulation time. Further performance is lost because the cache usage is not as good as in the smaller model. This results in slower simulation times than in the first example. Four threads are only three times faster than one thread (see Figure 7).

Simulation of the Box with 20*20*100 nodes: The biggest example with 40000 nodes shows similar behaviour than the model with 20000 nodes. Four threads have nearly the same performance as three threads (see Figure 8). The reason for this behaviour is that system services take a lot of time from one processor and therefore all other threads have to wait.

4 Summary

The LBGK D3Q15 method is described, which is a widely used method for fluid mechanical application. This work mainly focuses on the application of this method to hemodynamics and parallelization. A relevant example for blood flow simulation is described and the LBGK D3Q15 method is used for the calculation of the problem.

The model is used as a benchmark to test the ability of parallelization of the method. Different approaches of implantation are discussed. The object oriented approach is favoured.

For the computation a Dell Precision 670 workstation containing two Intel Xeon dual core processors with 2.8 GHz is used. The method scales linearly as expected for small models (8000 nodes). For larger models (20000 or 40000 nodes) the simulation time is strongly influenced by services of the underlying operation system when working with four threads.

References

[1] D. Leitner, J. Kropf, S. Wassertheurer, F. Breitenecker: *Lattice Boltzmann Methode zur Simulation vom Strömungsverhalten in Arterien*. In (U. Rüde et al., eds.) Proc.18th Symp.Simulationtechnique ASIM 2005, Frontiers in Simulation, SCS Publishing House, Erlangen, 2005; pp 768-774.

[2] S. Succi: *The Lattice Boltzmann Equation for Fluid Dynamics and Beyond*. Oxford University Press, 2001.

[3] D.A. Wolf-Gladrow: *Lattice-Gas Cellular Automata and Lattice Boltzmann Models - An In-troduction*. Lecture Notes in Mathematics. Springer, 2002.

[4] D.A. McDonald: *Blood Flow in Arteries*. Edward Arnold, 1974.

[5] A. M. Artoli, D. Kandhai, H. C. Hoefsloot, A. G. Hoekstra, P. M. A. Sloot: *Lattice bgk simulations of flow in a symmetric bifurcation*. Future Gener. Comput. Syst., 20(6):909-916, 2004.

[6] A.M. Artoli, A.G. Hoeksta, P.M.A. Sloot: *Simulation of a systolic cycle in a realistic artery with the lattice boltzmann bgk method*. Int. J. Mod. Phys. B, (17):95-98, 2003.

[7] D. Leitner, S. Wasssertheurer, M. Hessinger, A. Holzinger: *A Lattice Boltzmann Model for pulsative blood flow in elastic vessels*. Elektronik und Informationstechnik, heft 4, 152-155, 2006.

Corresponding author: Daniel Leitner
Daniel Leitner, Siegfried Wassertheurer,
ARC Seibersdorf research GmbH, Biomedical Engineering,
Floragasse 7, 1040 Vienna, Austria;
{daniel.leitner, siegfried.wassertheurer}@arcsmed.at

Felix Breitenecker, Vienna University of Technology,
Inst. for Analysis and Scientific Computing,
Wiedner Hauptstr. 8, 1040 Vienna, Austria;
fbreiten@osiris.tuwien.ac.at

Michael Hessinger, Andreas Holzinger, Medical University Graz, Auenbruggerplatz 2/4, 8036 Graz, Austria
{michael.hessinger, andreas.holzinger}@meduni-graz.at

Received: May 15, 2006
Revised: July 10, 2006
Accepted: July 20, 2006