



UvA-DARE (Digital Academic Repository)

High performance computational hemodynamics with the Lattice Boltzmann method

Axner, L.

Publication date

2007

Document Version

Final published version

[Link to publication](#)

Citation for published version (APA):

Axner, L. (2007). *High performance computational hemodynamics with the Lattice Boltzmann method*. [Thesis, externally prepared, Universiteit van Amsterdam].

General rights

It is not permitted to download or to forward/distribute the text or part of it without the consent of the author(s) and/or copyright holder(s), other than for strictly personal, individual use, unless the work is under an open content license (like Creative Commons).

Disclaimer/Complaints regulations

If you believe that digital publication of certain material infringes any of your rights or (privacy) interests, please let the Library know, stating your reasons. In case of a legitimate complaint, the Library will make the material inaccessible and/or remove it from the website. Please Ask the Library: <https://uba.uva.nl/en/contact>, or a letter to: Library of the University of Amsterdam, Secretariat, Singel 425, 1012 WP Amsterdam, The Netherlands. You will be contacted as soon as possible.

High Performance
Computational Hemodynamics
with the Lattice Boltzmann
Method



High Performance
Computational Hemodynamics
with the Lattice Boltzmann
Method

ACADEMISCH PROEFSCHRIFT

ter verkrijging van de graad van doctor
aan de Universiteit van Amsterdam
op gezag van de Rector Magnificus
prof. dr. D. C. van den Boom
ten overstaan van een door het college voor promoties ingestelde
commissie, in het openbaar te verdedigen in de Agnietenkapel
op dinsdag 18 December 2007, te 10:00 uur

door

Lilit Axner

geboren te Yerevan, Armenie

Promotiecommissie:

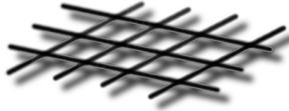
Promotor: prof. dr. P. M. A. Sloot

Co-promotor: dr. A. G. Hoekstra

Overige leden: Prof. Dr. J.H.C. Reiber
Prof. Dr. M.T. Bubak
Prof. Dr. J.A.E. Spaan
Prof. Dr. B. Chopard
Prof. Dr. P.A.J. Hilbers

Faculteit: Faculteit der Natuurwetenschappen, Wiskunde en Informatica

Section Computational Science



Universiteit van Amsterdam

The work described in this thesis has been carried out in the Section Computational Science of the University of Amsterdam, with financial support of:

- University of Amsterdam,
- Netherlands Organisation for Scientific Research – *Distributed Interactive Medical Exploratory for 3D Medical Images (NWO–DIME)* project (634.000.024),



Copyright © 2007 Lilit Axner.

ISBN 978-90-5776-170-6

Typeset with L^AT_EX 2_ε.

Printed by PrintPartners Ipskamp B. V., Enschede.

Author contact: labraham@science.uva.nl

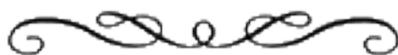
To my mother, Anna...

Contents

1	Introduction	1
1.1	High performance computing and computational hemodynamics	1
1.1.1	Computational hemodynamics	1
1.1.2	High performance computing	4
1.1.3	Merge of two worlds	6
1.2	Lattice Boltzmann method in computational hemodynamics	6
1.2.1	The model	8
1.2.2	Parallel Lattice Boltzmann method	9
1.2.3	Unstructured Lattice Boltzmann	10
1.3	Research overview and motivation	12
2	HemoSolve - A Problem Solving Environment for Image-Based Computational Hemodynamics	15
2.1	Introduction	15
2.2	The first step towards HemoSolve	15
2.2.1	Medical Data Segmentation	16
2.2.2	3D Editor and Mesh Generator	19
2.2.3	Hemodynamic Solver	19
2.2.4	Validation of Lattice Boltzmann Method	19
2.2.5	Flow Analyses	23
2.3	Examples	23
2.3.1	Aneurysm	23
2.3.2	Abdominal aorta	24
2.4	Discussions and Conclusions	24
3	Efficient LBM for Time Harmonic Flows	27
3.1	Introduction	27
3.2	Lattice Boltzmann Method	28

3.3	Constraint Optimization Scheme	28
3.3.1	Asymptotic error analysis	30
3.3.2	Experimental results	31
3.4	Stability of Time Harmonic LBM Simulations	35
3.5	Harmonic Flow in Human Abdominal Aorta	36
3.6	Regularized time-harmonic LBGK	37
3.7	Accuracy and Stability of RL-BGK	38
3.7.1	Numerical stability of R-LBGK	39
3.8	Conclusions	40
4	High Performance Lattice Boltzmann Computing	43
4.1	Introduction	43
4.2	Sparse LB Implementation	44
4.3	The Graph Partitioning Algorithms	44
4.4	The Performance Prediction Model	45
4.5	Results	49
4.5.1	Basic parameters	51
4.5.2	Performance Measurements of the Lattice Boltzmann application	54
4.6	Conclusions	55
5	Towards Decision Support in Vascular Surgery through Computational Hemodynamics	63
5.1	Introduction	63
5.2	HemoSolve and the new enhancements	64
5.2.1	Medical Data Segmentation	64
5.2.2	3D Editor and Mesh Generator	65
5.2.3	Hemodynamic Solver - Sparse Lattice Boltzmann Method	66
5.2.4	Flow Analyses	66
5.3	Carotid Artery with Severe Stenosis - with and without Bypass	67
5.4	Conclusion	68
6	Conclusion	73
6.1	Discussion	73
6.2	Future work	75
	Summary	77
	Nederlandse samenvatting	79
	Summary in Armenian	81

Acknowledgments	85
Publications	89
Bibliography	91
Index	101



Introduction



1.1 High performance computing and computational hemodynamics

1.1.1 Computational hemodynamics

Fluid dynamics is one of the major scientific disciplines describing the behavior of one of the main materials of nature: **Water**. The earth has liquid water on 70% of its surface, and the volume of blood of an average human is about 5 liters and fluids are engaged in almost all our daily activities. Thus fluid is an immediate requirement of life and deep knowledge of its behavior is a way to recognize and predict its influence on us and our environment.

Specifically, in the last decades there is a growing interest towards the details of the flow behavior of blood in the human vascular system, e.g. blood dynamics called *hemodynamics*.

The human cardiovascular system is very complex: The blood is pumped by heart, through the large arteries, such as the abdominal aorta, to the smaller diameter arteries called arterioles (see Fig. 1.1). These arterioles become capillaries and eventually venules, where the deoxygenated blood passes through veins back to heart, imposing a circular road-map through whole body [30]. Due to the pumping of the heart the blood pulsates over time, thus it is characterized as *time-harmonic*.

Atherosclerosis is the most common cardiovascular disease that affects the arterial blood vessel [4, 98, 100]. It is the formation of yellowish plaques containing fatty substances that are formed within the intima and inner media of arteries [101]. The plaque causes the hardening of the arteries and destructs or in some cases it completely blocks the normal time-harmonic blood flow (see Fig. 1.2 (right)). As described by M. Thubrikar [101], atherosclerosis begins as intimal lipid deposits in childhood and adolescence and in some arterial parts those lipid deposits convert into fibrous plaques by continued accumulation of lipid smooth muscle and connective tissue. In middle age, after 30 or more years since the beginning of the process, some of fibrous plaques produce occlusion, ischemia and clinical disease. It should be noted that only few of the lipid deposits convert into atherosclerotic lesions.

Atherosclerosis usually appears at the origins of tributaries, bifurcations and curvatures of the arteries. Moreover, it primarily affects the luminal side of them. This is

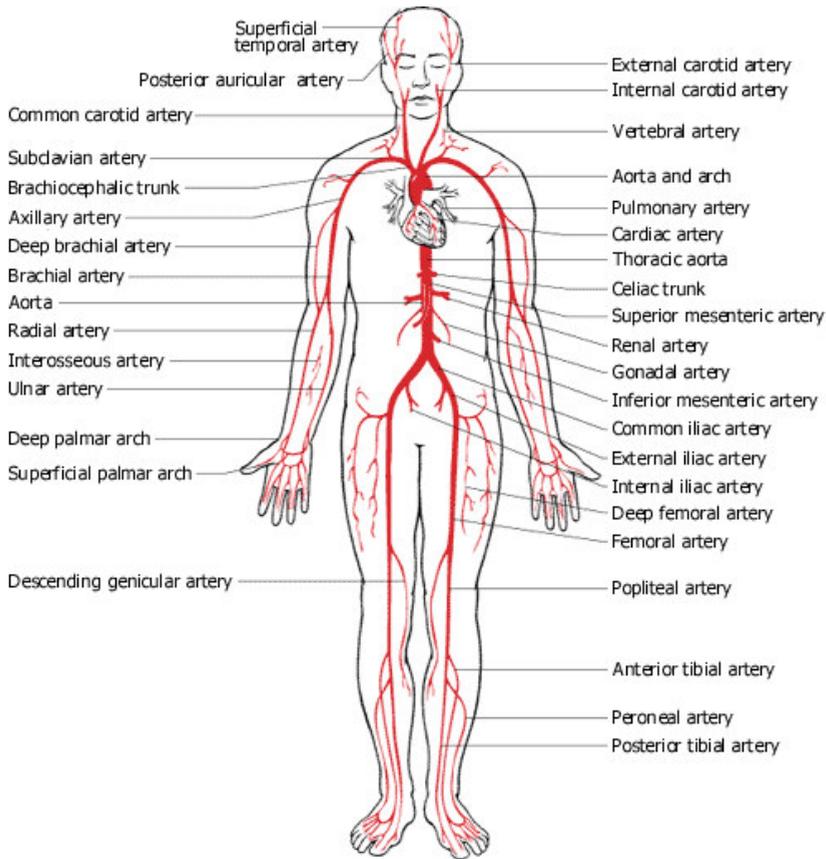


Figure 1.1: Human circulatory system, arterial branch. (The picture is obtained from <http://www.medem.com>)

explained by the distribution of wall stress through the thickness of the walls, e.g. the wall stress from pressure is the highest on the inner surface of the artery. Also the high stress concentrations at the branch origins produces a greater stretch at those locations.

The new approach to find the cause of atherosclerosis is to view the artery not only as a conduit of blood flow but also as a pressure vessel [101]. Thus two sources are considered:

- Stress concentration at branches
- Wall fatigue due to pulsatile blood pressure

The atherosclerotic deformations are characterized as slowly progressive and cumulative. For example the appeared soft plaque can suddenly rupture causing the formation of thrombosis. This, in turn will rapidly slow or stop the blood flow in approximately 5 minutes causing an infarction [109].

Aneurysm is a balloon enlargement of the aorta (see Fig. 1.2 (left)). Most commonly it appears in abdominal aorta and cerebral arteries (at the base of the brain, known as the Circle of Willis), rarely in ascending aorta and descending thoracic aorta (see Fig. 1.1) and sometimes it may cover several segments of the aorta. Aneurysm appears due to the same mechanism as atherosclerosis when the smooth muscle cells of the artery are unable to proliferate [101].

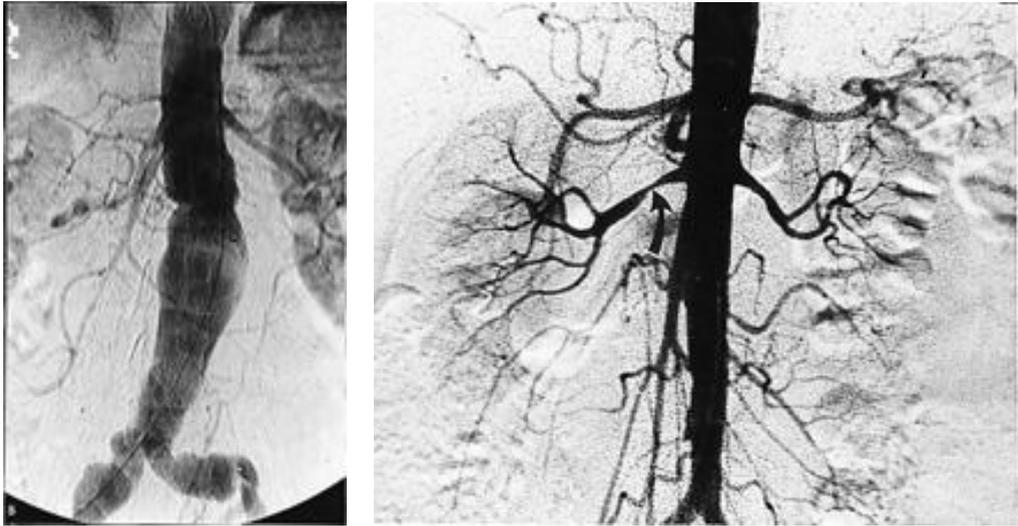


Figure 1.2: Aneurysm and stenosis in the abdominal aorta. (The pictures are obtained from <http://www.medcyclopaedia.com>)

The modeling of blood flow in human cardiovascular structures helps to obtain an extensive knowledge about its behavior and to create solutions for preventing the actual death caused by atherosclerotic diseases or aneurysms. It can be done by using the Navier-Stokes equation.

The Navier-Stokes equation was derived by Claude Louis M.H. Navier (1785-1836) and Sir George Gabriel Stokes (1819-1903). Although it is capable to describe such complex phenomena as solitons, von Karman vortex streets and turbulence, it is still hampered by two major limitations: the non-straight forward applicability to complex geometries and the high compute intensity.

One approach to overcome these limitations is to use cellular automata, more precisely lattice gas automata (LGA), to model the microscopic properties of the blood flow while exploring its macroscopic properties [90]. Historically the successor of LGA is the lattice Boltzmann method (LBM) and thus both methods are mesoscopic methods. The main idea behind LBM is the mapping of the average motion of the fluid/blood particles on the lattice. An introduction to the LBM method is given in the subsection 1.2. Another approach is the numerical discretion of Navier-Stokes equation. The finite element and the finite volume methods are examples of this approach.

In either way the final step is the integration of chosen approach into the state-of-the-art high performance computers. This we call *computational hemodynamics*.

In the further development of this thesis we use LBM as computational hemodynamics method. The locality of the LBM update scheme makes its parallelization straightforward while the optimal distribution/decomposition of the geometry over the multiple processors is the main bottleneck for high performance computing. This will be one of the problems addressed in later chapters.

1.1.2 High performance computing

Three dimensional simulations in computational hemodynamics as well as in any computational fluid dynamics problem require huge computer resources, both in memory and speed. Thus in CFD always the most powerful computers available have been used.

The term "Super computing" was first used by the New York World newspaper in 1920 about the IBM large custom-build tabulators for Columbia university [96]. Supercomputers first appeared in 1960s designed by Seymour Cray at Control Data Corporation. In 1970s most of the supercomputers were dedicated to vector computing. In 1980s and 1990s massive parallel processing systems with thousands of CPUs appeared. Nowadays, supercomputers typically are "off the shell" server-class microprocessors, e.g. computer clusters with commodity processors [3].

As of June 2007, the fastest machine is Blue Gene/L with 65.536 nodes, each with two processors, each with two data stream connectivity [3].

In this work we use two different types of CPU architectures: Vector machines and Streaming SIMD Extension2 (SSE2). We chose vector machines as they are known to produce the highest number of lattice updates per second while the architecture of SSE2 clusters is the best suited for the storage of the complex geometry data.

- Vector architecture

Vector processor is a CPU design that implements mathematical operations on multiple data elements simultaneously. This type of processors were first introduced by Westinghouse through Solomon project in 1960s [49]. The first successful implementation of vector processors was CDC STAR-100 and The Texas Instrument Advance Scientific Computer. These processors reached approximately 20 MFLOPS peak performance, however they took very long times decoding the vector instructions and to run the process. Next, Cray introduced the Cray-1 machine, where there were eight vector registers and the vector instructions were applied between registers, thus accelerating the processing. Also Cray-1 had a separate pipeline for addition/subtraction and separate for multiplication, thus applying the vector chaining technique. Cray-1 reached from 80 to 240 MFLOPS peak performance. Later different Japanese companies, one of which was NEC, started to produce similar vector machines. Due to the special super-scalar implementation, vector machines perform most efficient, when the amount of data to work on is significantly large [79].

The NEC SX-8 supercomputers are the latest generation of NEC SX-series of supercomputers. They are build from SMP machines each with 8 vector processors. More detailed specifications of the NEC SX-8 are given in Table 4.1 of Chapter 4. (See also [1]).

- Streaming SIMD Extension2

SSE2 is an addition to SIMD technique. SIMD processors are a single instruction, multiple data general purpose processors. They are used to achieve data level parallelism and are handling only data manipulation. The first widely used SIMD architecture supercomputers were Intel MMX x86 architectures [108].

The SSE2 extension is based on SIMD architecture, which also includes a set of cache-control instructions and a system of numerical format conversion instructions. The SSE2 was first introduced by Intel Pentium 4 processor. Moreover, in addition to SSE2 an instruction pipeline technique is used in Pentium 4 processors, which allows to reduce the cycle time of a processor and increase the instruction throughput. The hyper threading technology (HTT) that enables multiple threads to run simultaneously, improves reaction and response time, and increases the number of users a server can support is also combined in later models of Pentium 4 [108].

In this work we use the Dutch national compute cluster Lisa , based on 2Intel Xeon2 architecture at SARA Computing and Networking Service center [2]. The Intel Xeon cluster Lisa has an identical architecture to Pentium 4. More detailed specifications of this cluster are given in Table 4.2 of Chapter 4. (See also [2])

As was mentioned in previous subsection, one of the main issues in high performance implementation of computational hemodymanics is the optimal decomposition of the geometry, especially when the geometry is complex.

During the last decade a lot of decomposition algorithms have been developed for both the discretized Navier-Stokes (FEM,FVM) and the mesoscopic (LBM) methods. Examples of such algorithms are geometric schemes, spectral methods, combinatoric schemes, multilevel schemes, orthogonal bisectioning method, cell-based method etc. [15, 33, 47, 55, 110, 111]. All these schemes have their advantages and disadvantages. For example, geometric schemes use the geometric information of the data to find a good partitioning. They tend to be fast but produce worse quality partitions than those by multilevel schemes. Spectral methods are known to produce perfect partitions, however they are very expensive since they require the computation of the eigenvectors corresponding to the second smallest eigenvalue (Fiedler method) [62]. As for multilevel schemes, they tend to give quite reasonable, although not excellent partitions at a comparatively low cost. One of the known libraries that use multilevel graph partitioning is METIS [62]. The method as well as its advantages and disadvantages will be described further in the thesis.

1.1.3 Merge of two worlds

In vivo measurements of hemodynamic quantities give very detailed information about blood flow characteristics, however they are very difficult to obtain. Thus one of the challenges of computational hemodynamics is the implementation of numerical models in large datasets and the simulation/validation of flow dynamics.

The practice showed that the non-linear equations of fluid flows can only be solved analytically for special simplified cases. Those cases comprise deep physical insights, thus the following step is the application of these knowledge on the non-linear problems.

There have been several studies discussing the aspects of development of computational hemodynamics [21, 64, 98, 100, 118]. In all these studies it is explicitly pointed out that computational hemodynamics problems are known to be extensive and requiring large computational resources. Thus the transfer to high performance computing is a very natural step of development [16, 36, 119].

As mentioned by Taylor et al. [100], in order to realize the benefits of the merge of these two worlds we need to cite the major barriers and the ways to overcome:

1. Creation of complex hemodynamics models can be time consuming and a tremendous process - to model time-harmonic blood flow in the human vascular system one can use the Navier-Stokes equations and discretize them to create a computer algorithm.
2. Mesh generation from medical data is time consuming and complex - several mesh generation algorithms/software are created for these purposes.
3. The generated discrete problems can involve millions of equations for thousand of timesteps - the generated algorithmic blocks can be integrated into high performance computing [21, 90].

To these and other more detailed aspects we will refer in the further development of the thesis.

1.2 Lattice Boltzmann method in computational hemodynamics

In 1872 Ludwig Boltzmann derived the Boltzmann equation (Eq. 1.1) to describe the statistical distribution of particles in fluids.

$$\frac{\partial f}{\partial t} + \frac{\partial f}{\partial \mathbf{x}} * \frac{\mathbf{p}}{m} + \frac{\partial f}{\partial \mathbf{p}} \mathbf{F} = \frac{\partial f}{\partial t} \quad (1.1)$$

here m and \mathbf{p} are the mass and momentum of the fluid particle, $\partial \mathbf{x}$ and ∂t are the space and time steps respectively. $f(\mathbf{x}, \mathbf{p}, t)$ is the probability density function and \mathbf{F} is the force acting on the particles and making them propagate and collide [97].

One of the methods to simulate a Newtonian fluid flow is the lattice Boltzmann method (LBM) [17, 23, 25, 97] which can be derived from continuous Boltzmann equation [45,122]. In LBM models the fluid flow in the sense of particles performing propagation and collision over a discrete lattice. Some of the positive characteristics of LBM are e.g. its ability to incorporate complex boundary conditions. On the other hand it is hampered by slow convergence [42, 59, 106], caused by the demand of a low Mach number (Ma) (to suppress compressibility error) and the fulfillment of a Courant-Friedrich-Levy condition for numerical stability.

Historically LBM comes from lattice gas automata (LGA) [25,81,89] which are based on Boolean particle numbers and is a simplified model for particle dynamics. Although LGA is a good method for modeling fluid flow it suffers from the lack of Galilean invariance and stational noise due to its Boolean nature. In the transformation to LBM, the Boolean particle numbers are replaced by density distribution functions thus eliminating the statistical noise [7,97]. Moreover, from density distributions it is possible to derive the pressure field, hence there is no need to solve the Poisson equation as in the other CFD models. The further modification of lattice Boltzmann method is the approximation of its collision operator with the Bhatnagar-Gross-Krook (BGK) relaxation term [24,123]. It has been proved using Chapman-Enskog analysis it is possible to recover the Navier-Stokes equations from LBM. The chart in the Fig. 1.3 shows the complete tree from Newtonian flow to lattice Boltzmann model and to Navier-Stokes equations.

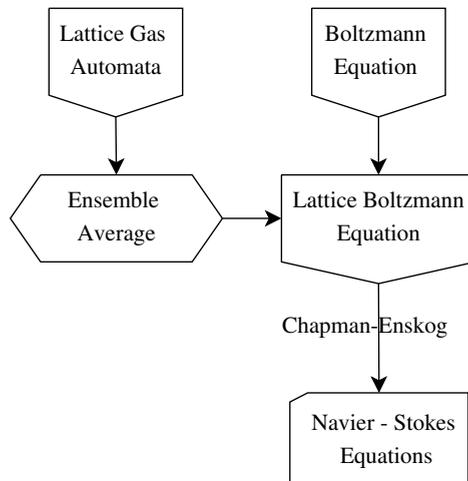


Figure 1.3: From Newtonian flow to discretized velocity lattice Boltzmann method and Navier-Stokes equations.

1.2.1 The model

The lattice Boltzmann BGK model can be described as

$$f_i(\mathbf{x} + \mathbf{e}_i, t + 1) - f_i(\mathbf{x}, t) = -\frac{1}{\tau} [f_i(\mathbf{x}, t) - f_i^{(eq)}(\mathbf{x}, t)] \quad (1.2)$$

Here \mathbf{e}_i the finite set of discrete velocities, τ the dimensionless relaxation parameter, $f_i(\mathbf{x}, t)$ the density distribution function and $f_i^{(eq)}(\mathbf{x}, t)$ the equilibrium distribution defined by

$$f_i^{(eq)} = \rho w_i \left(1 + \frac{\mathbf{e}_i \cdot \mathbf{u}}{c_s^2} + \frac{(\mathbf{e}_i \cdot \mathbf{u})^2}{2c_s^4} + \frac{\mathbf{u} \cdot \mathbf{u}}{2c_s^2} \right). \quad (1.3)$$

where w_i is a weighting factor, $c_s = 1/\sqrt{3}$ the speed of sound, ρ the hydrodynamic density determined by

$$\rho = \sum_i f_i = \sum_i f_i^{(eq)}, \quad (1.4)$$

and \mathbf{u} the macroscopic velocity determined by

$$\rho \mathbf{u} = \sum_i \mathbf{e}_i f_i = \sum_i \mathbf{e}_i f_i^{(eq)}. \quad (1.5)$$

The viscosity ν of the fluid is determined by

$$\nu = \frac{1}{6} \left(\tau - \frac{1}{2} \right). \quad (1.6)$$

All parameters in the model are in lattice units and $\delta x = \delta t = 1$. In general we consider three flow parameters that can reveal their influences on the flow characteristics: Those parameters are

- **The Reynolds number** Re - is the ratio between inertial and viscous forces

$$Re = \frac{\mathbf{u}D}{\nu} \quad (1.7)$$

- **The Mach number** Ma - is the ratio between fluid velocity and the speed of sound

$$Ma = \frac{|\mathbf{u}|}{c_s}. \quad (1.8)$$

- **The Womersley number** α - is the ratio between the time-harmonic flow frequency and viscous effects. α is considered only for time harmonic flows.

$$\alpha = R * \sqrt{\frac{\omega}{\nu}}. \quad (1.9)$$

In all these equations $D = 2R$ is a typical length scale of the flow problem, c_s is the speed of sound and ω is the flow frequency.

In all our further studies we first run our simulations for the simple cases and compare the results with analytical solutions such as the Womersley solution [117] for the time-harmonic flows. After being satisfied with the outcome we transfer our simulations to the real and more complex applications.

In Fig. 1.4 is an example of a velocity flow profile in a straight circular tube with diameter D and length L and we simulate a time-harmonic flow over this tube. The flow is driven by the pressure gradient generated out of the pressure pulse(Fig. 1.4(b)) and applied on the inlet of the geometry. The flow is simulated over the period T which is discretized into multiple time-steps. Such simple test cases allow us to experiment with

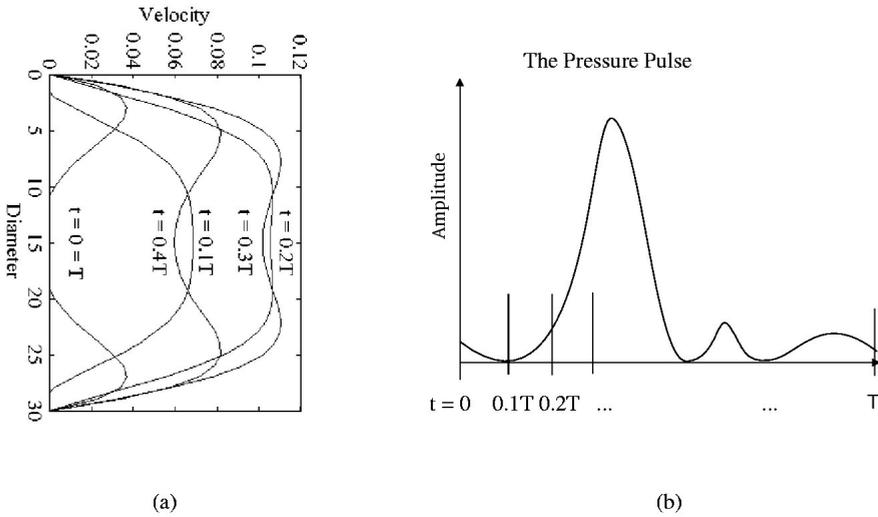


Figure 1.4: Flow in a straight circular tube: (a)- velocity profiles along the diameter D of the tube every 0.1 time-step during one period. (b) - a pressure pulse during one period T .

different boundary conditions, measure the accuracies and stabilities of simulations and validate the results.

1.2.2 Parallel Lattice Boltzmann method

As mentioned in section 1.1.3 three dimensional fluid flow simulations are known to be computationally extensive. LBM being one of them is not an exception. Kandhai et al. in 1998 [55] developed an efficient parallel LBM for 3D geometries, and studied a number of different geometric decompositions, such as orthogonal recursive bisection (ORB). Special attention has been paid to issues of load balancing.

In their work three different types of decomposition were compared with each other:

- Slice decomposition

- Box decomposition
- ORB decomposition

For the presenter geometry it was shown that in case of slice and box decompositions the workload was not balanced at all. The preference was given to ORB decomposition. Although the obtained results were 10 – 60% better than slice or box decomposition, ORB has a main disadvantage of the complicated communication scheme.

Later, in 1999 Dupuis et al. decomposed a two dimensional (2D) lattice by using a graph partitioning library called METIS (see Sec. 1.1.3) for any Lattice Gas application [33]. They proposed to represent the 2D lattice as a vector of cells instead of matrix, where every cell knows the indexes of its neighboring cells. This approach has two advantages. First it allows to exclude the unused areas from lattice, second the lattice traversal is simple to execute. It was concluded that the performance of this approach is a factor of two slower than the classical approach.

In the same year Satofuka and Nishioka [84] compared the three horizontal, vertical and checker board domain decomposition methods for LBM simulations. They concluded, that the vertical decomposition gives the highest speedup. In the case of horizontal decomposition the longer the number of lattice units in horizontal direction of each subdomain, the shorter the CPU time. The measurements were performed for 2D simulations and considering the outcome the vertical decomposition was applied for 3D simulations. Even better results of speed-up were obtained for the 3D case.

In 2005 Wang et al. [111] proposed another approach, the so called cell-based decomposition method. Unlike other methods, the cell-based method performs the load balance first to divide the total number of fluid cells into a number of subdomains, where the difference of fluid cells in each subdomain is either 0 or 1, depending on if the total number of fluid cells is a multiple of the processor numbers. This cell-based method recovers the interface rather than the load balance; it does not need iteration and gives an exact load balance. The performance of the results showed that it reached the theoretical parallel efficiency.

Detailed performance analysis on three SMP clusters were performed by Wu et al. [121] in 2006 based on a parallel multiblock method proposed by Yu et al. [124]. In this method the entire computational domain is divided into blocks, where each block is a segment in the x -, y -, and z -dimensions, respectively. Each block is assigned to a processor. The grid sizes in each block can be different. Only the communication along the border layers is necessary. The performance results showed that the large size problems scale well on the SMP clusters using the multiblock decomposition method.

1.2.3 *Unstructured Lattice Boltzmann*

Originally, LBM was formulated only on geometries that are discretized into uniform grids. Very soon, the potential of LB as an alternative method for complex fluid flow simulations became appreciated. There was a need to improve the method in order to apply it on more complex grids such as partially refined or nested grids. These types of

grids are especially useful to discretization of complex geometries, such as, in our case, the human vascular system.

First, the idea of making LBM applicable for non-uniform/unstructured grids was pointed out by Higuera in 1990 [48]. He proposed to use a coarse-grained distribution function which is defined at the center of the cell, which in its turn contains several nodes of the regular fine-grained grid.

In 1992 the *Finite-Volume LB method (FV-LBM)* was first introduced [17,75]. Main advantages of FV-LBM are the fact that the mass and momentum conservations remain well posed as in standard LBM and that it is able to cover the same Reynolds number range as standard LBM. However, its major disadvantages are the complicated implementation and loss of computational locality, which essentially increase the computational cost, especially for parallelization.

Next He, Luo and Dembo [46] modified LBM by adding a technique based on point-wise interpolation. In this new algorithm, the computational mesh is uncoupled from the discretization of momentum space and it can have an arbitrary shape. Collisions still take place on the grid points of the computational mesh. After a collision, the density distributions move according to their velocities. Although the density distributions at the grid sites now may not be exactly determined, they can always be calculated using interpolation. After interpolation, collision and advection steps are repeated [46]. Here the accuracy of LBM is of second order both in time and space and the error introduced by quadratic interpolation is more sufficient as it is small for any grid size. But due to the fact that in the scheme the interpolation distance has an impact on numerical viscosity, it increases the numerical error.

Later several modifications were applied on FV-LBM [22,74,77]. It has been applied on bilinear quadrilateral and triangular cells. Here the influence of interpolation distance on numerical error have been reduced. Moreover, in 2005 Stieber et al. applied the upwind discretization scheme on FV-LBM [95]. This scheme is based on the method of Peng et al. [74,77] but improves the stability and computational efficiency. However, in this scheme the linear reconstructional step is as expensive as the collision and propagation steps, which make the total cell update cost 50% higher.

Another adaptation of LBM is the *Multiscale LB schemes (M-LBM)* that have been presented by Filippova and Heanel [35]. The scheme is based on a coarse grid covering the whole integration domain. In a critical region, detected either by adaptation criteria or defined a priori, a finer grid is superposed to the basic, coarser grid. The calculation proceeds with large time steps accordingly to the coarser grid while on the finer grids several time-steps are performed to advance to the same time level [35]. This method is applied to LBGK. Here the collision step on the shared grid nodes is adjusted by correcting the non-equilibrium part of the particle distribution for the different time discretization and relaxation parameter on both grids. This assumption, however, is not entirely correct, because the non-equilibrium part depends on the relaxation parameter (in the case of LBGK scheme) and the time step. Without rescaling the non-equilibrium part, an error is introduced in the local components of the stress tensor.

A different modification of LBM is the *Finite Difference LB scheme (FD-LBM)* first pointed out in 2002 by Yu et al. [125]. This scheme is developed based on the fact that the common

LBM just one specific discrete representation of the Boltzmann equation. And as it does not use a different discretization of the space and time derivatives, one is free in choosing the type of grid. A multi-block method is developed and an accurate, conservative interface treatment between neighboring blocks is adopted, and demonstrated that it satisfies the continuity of mass, momentum, and stresses across the interface. The present multi-block method can substantially improve the accuracy and computational efficiency of the LBE method for viscous flow computations.

In 2000 Kandhai et al. applied FD-LBGK on nested grids [60]. The discrete velocity Boltzmann equation is solved numerically on each sub-lattice and interpolation between the interfaces is carried out in order to couple the sub-grids consistently. The computational domain is basically built up of a number of sub-domains which can have different grid resolutions. On each sub-domain the discrete velocity Boltzmann equation is solved and the adjacent sub-domains are coupled with each other by appropriate interpolation at the boundaries of the sub-domains. But again the presented finite-difference methods can accommodate only relatively smooth variations of the flow field because large deformations of the nonuniform mesh may result in numerical instabilities [73].

During the same period Tölke et al. [107] introduced implicit discretization and a non-uniform mesh refinement approach for the LBGK method. The results of the numerical examples shown in this work indicate that the use of implicit discretization methods for the LBGK equations may be of significant advantage at least for stationary flows of high Reynolds numbers in geometries of moderate complexity [107]. But stability and consistency issues of implicit methods for LBGK equations are far from being fully explored.

Recently a new idea of *Locally Embedded Grids (LEG)* is explored by Martin Rohde [82]. The idea of using the volumetric description of the grid for mass conservation is still applied here but no interpolation technique or rescaling of the non-equilibrium distribution is required to obtain accurate results. Furthermore, the technique is not restricted to a single-relaxation scheme, but can be used for any LB scheme, because the communication between the coarse and the fine grids takes place after the propagation step [82].

1.3 *Research overview and motivation*

The main goals of this thesis are

- to create a complete and enhanced computer-based simulation environment for image-based computational hemodynamics problems, using the LBM,
- to integrate it into a high performance computing environment,
- to apply it for visualization and understanding of biomechanical processes in complex vascular systems.

These will result to the creation of a comprehensive and efficient PSE that will answer to the "What if ..." questions concerning the biomechanical effects of complex vascular reconstruction processes, and that may also serve as a decision support system.

As was mentioned in section 1 during the last years the number of deaths caused by cardiovascular diseases has essentially increased, especially due to the vascular disorders caused by atherosclerotic diseases. The development of a computer-based problem solving environment (PSE) that provides all the computational facilities needed to solve a target class of problems is a necessity [50, 98].

For this type of PSE a major important component is a sophisticated time-harmonic flow simulator. As discussed in section 1.1.3 for fluid flow simulations besides traditional Navier-Stokes equations one can use the LBM method. However, as any CFD method it has its disadvantages, like slow convergence, pointed out in section 1.2. Thus the challenge for us is to create a scheme which will help to overcome this problem.

Moreover, the optimal parallelization of LBM is a developing issue and an interesting direction to investigate. Even though LBM is simple to parallelize due to the locality of its computations, there still can be complications connected with the discretization methods and workload imbalance. In section 1.1 the main concepts of high performance computing were introduced, while in section 1.1.3 their influences on the simulation model were discussed. The short description of existing parallelization schemes gives us clear indication that there are several ways of their optimization. Especially for the cases where the geometry discretization is inhomogeneous as described in section 1.2.3.

The thesis is constructed in the following way:

In the current chapter is a general introduction to the main concepts and a short presentation of the state of the art.

In Chapter 2 we describe the PSE that we have created for image-based computational Hemodynamics called HemoSolve. Also we give a complete evaluation of the LBM by comparing it to with the other CFD methods such as Finite Element Method (FEM).

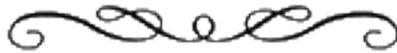
After evaluation of LBM, we continue with its improvement in the Chapter 3. We define a constraint optimization scheme which helps to speed-up its convergence. Also another improvement of is a new proposed variation of LBM called regularized LBM. A substantive comparison of these two methods is presented.

Next in Chapter 4 we integrate LBM into high performance computing. We show a complete set of experiments and comparisons of both discretization methods and performance aspects. Moreover, inspired by previous studies we develop a performance prediction model that can be used to evaluate the execution costs of the model.

In Chapter 5 we combine our results in a new and highly enhanced version of PSE, HemoSolve2.0.

At the end, in Chapter 6 we summarize the work presented in this thesis, draw our conclusions and discuss the future development possibilities.

*HemoSolve - A Problem Solving Environment for Image-Based Computational Hemodynamics**



2.1 Introduction

"A problem solving environment (PSE) is a computer system that provides all the computational facilities necessary to solve a target class of problems" [39, 50]. The target class of problems that we chose in our study is associated with cardiovascular diseases, a predominant cause of death [4, 98, 101]. In particular our attention is concentrated on vascular disorders caused by atherosclerosis. The goal of our PSE, which we call HemoSolve, is to provide a fully integrated environment for simulation of blood flow in patient specific arteries.

Because of the complex structure of the human vascular system it is not always obvious for surgeons how to solve the problem of bypass and/or stent placement on the deformed part, or to decide on specific treatment alternatives. Having a completely integrated computational hemodynamics environment like HemoSolve can serve as a pre-operational planning tool for surgeons, but also as a useful experimental system for medical students to enlarge their practical skills [93, 104]. It also serves as a environment for biomedical engineers that study e.g. new stent designs.

Moreover, HemoSolve is merged with Grid technology, thus offering a unified access to different and distant computational and instrumental resources [104]. This is one of the desirable abilities of PSEs in general [50].

2.2 The first step towards HemoSolve

In ref. [93] Steinman argues that a need exists for robust and user-friendly techniques that can help an operator turn a set of medical images into computational fluid dynamics

*This chapter is partly based on: L. Abrahamyan, J. A. Schaap, A. G. Hoekstra, D. Shamonin, F. M.A. Box, R. J. van der Geest, J. H.C. Reiber, P. M.A. Sloot. A Problem Solving Environment for Image-Based Computational Hemodynamics. In *LNCS 3514*, p.287, April (2005).

(CFD) input file in a matter of minutes. HemoSolve not only has this ability but also is a tool which allows to simulate pulsatile (systolic) flows in arteries.

The whole system consists of the following components (See Fig. 2.1):

1. Medical data segmentation to obtain arteries of interest;
2. 3D editing and mesh generation, to prepare for the flow simulation;
3. Flow simulation, computing of blood flow during systole;
4. Analysis of the flow, pressure, and stress fields.

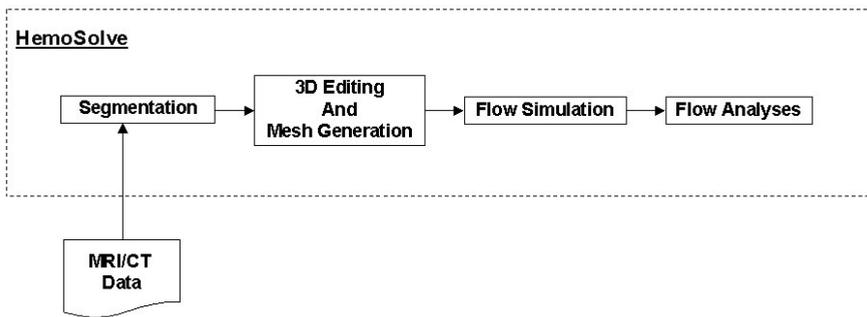


Figure 2.1: Functional design of HemoSolve.

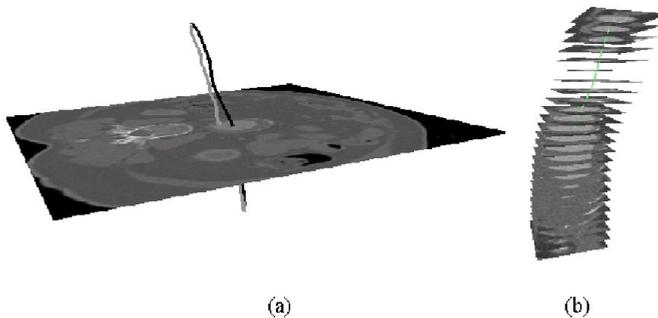


Figure 2.2: (a) - is the pathline and the centerline crossing an axial slice, (b) - is a stack of slices resampled along the centerline.

2.2.1 Medical Data Segmentation

The goal of the segmentation process is to automatically find the lumen border between the blood and non-blood, i.e. the vessel wall, thrombus or calcified plaque. The algorithm

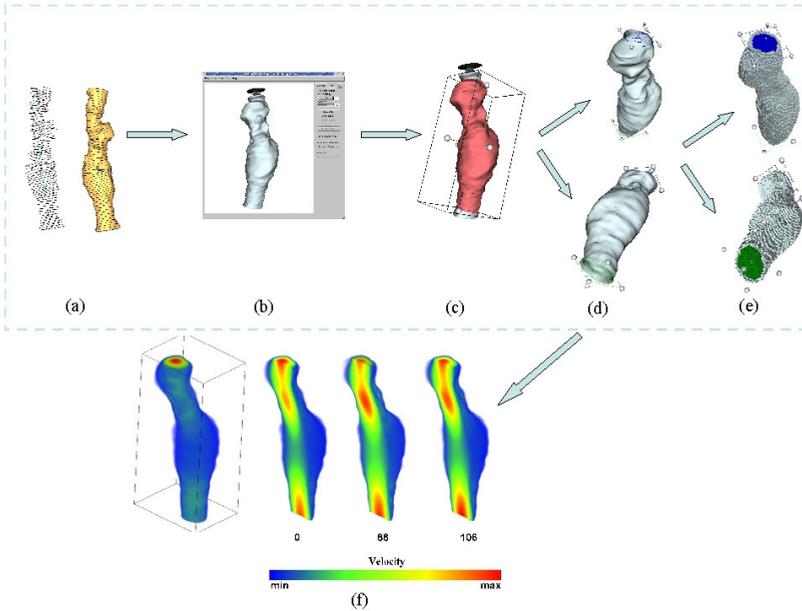


Figure 2.3: Aneurysm: main stages of HemoSolve. First segmentation of the raw medical data (a). Then the segmented data (b) is first cropped (c) and inlet/outlet layers are added (d) and the mesh is generated (e). Simulation results of created mesh are presented (f).

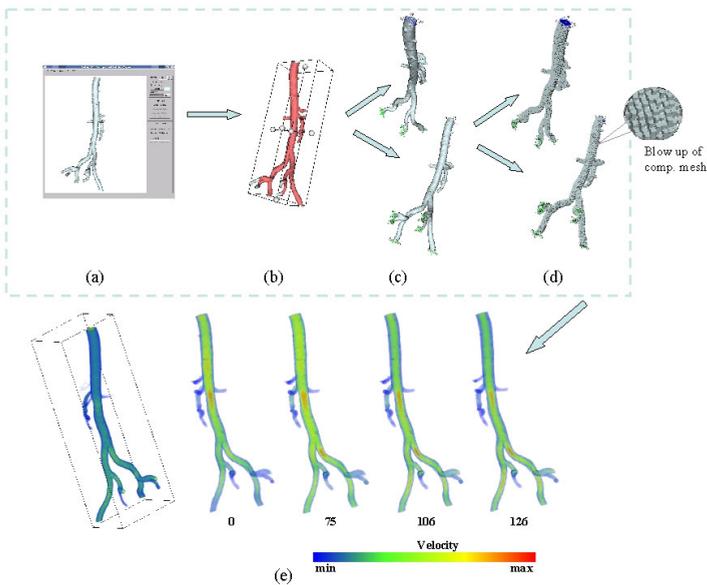


Figure 2.4: Abdominal aorta: main stages of HemoSolve. Segmented data (a) is first cropped (b) and inlet/outlet layers are added (c) and the mesh is generated (d). Simulation results of the created mesh are presented (e).

consists of three stages: In the first stage a wave front propagation algorithm is used to find an approximation of the centerline of the vessel. In the second stage the volumetric data is resampled into a stack of 2D slices orthogonal to the centerline. Then in each slice a contour delineating the lumen border is detected. Finally in the third stage, the stack of 2D contours is combined to form a 3D surface model, which will serve as input for the 3D editing tool.

- 3D Centerline extraction:

In order to get a first estimate of the centerline of the vessel we use the WaveProp method as described in [52] and [31]. WaveProp is based on the Fast Marching Level Set algorithm introduced by Sethian [86]. The principle is that an image is treated as an inhomogeneous medium through which a wave can propagate. The speed of the wave front is determined by a speed function which maps intensity values to speed values. The propagating wavefront creates a monotonous ascending function with its single minimum at the startpoint; with a steepest decent the shortest path from end point to startpoint is found.

In the 3D computer tomography angiography (CTA) datasets the user indicates with two points the beginning (proximal point) and end (distal point) of the vessel segment of interest. Then, a wave front is initiated at the proximal point and propagated through the vessel until it reaches the distal point marking all visited voxels as lumen (blood pool). This gives us a binary volume of the solid lumen. A distance image [29,83] is then calculated from the binary lumen volume containing for each voxel its distance to the background. Consequently, there will be a 3D ridge of high values in the distance image, which coincides with the exact path of the centerline of the solid blood pool. This ridge can be tracked as described in [31] using wave front Propagation and Backtracking, resulting in the centerline (Fig. 2.2 (a)).

- Per slice contour detection:

Perpendicular to the detected centerline 2D slices are extracted from the original CTA dataset (Fig. 2.2 (b)). In each of these transversal slices, a contour delineating the lumen is detected using the WaveProp algorithm similar as it was used to detect the centerline, but now with different speed functions and cost images. Contours delineating objects in an image usually follow the edge of that object. With the proper speed function, a wave propagating through an edge image will flow fast along the object edges and stall in solid parts of the object and in the background. The backtrack method can be used to obtain a contour from one point on the edge to another point on the edge, following that particular edge.

- 3D Surface model:

The stack of 2D transversal contours can be used to obtain per-slice information, such as diameter, circumference and area of the 2D contour. For the flow simulations however, a 3D surface model is needed. Therefore, the stack of 2D contours is converted into a 3D surface model by connecting each of the contour points of one slice to the closest contour point in the next slice (Fig. 2.3 (a)).

2.2.2 3D Editor and Mesh Generator

3D editing is the second component after the segmentation. The 3D stereoscopic image can easily be maintained in this user-friendly editing tool. Here surgeons and students can execute their experimental visualization studies on realistic arterial geometries. They can crop parts of the artery, where important factors in the study of hemodynamics exist, with the help of a clipping instrument. They can add inlet and outlet layers on the end-points of the arterial geometry and can enhance it with structures like bypasses or stents. Also this component allows them to define the geometrical features (e.g. width, length, placement positions) of these structures. Thus, the 3D editing tool allows surgeons and students to mimic the real surgical processes.

The final stage of this component is mesh generation. The prepared arterial geometry, including aneurysms, bifurcations, bypasses and stents, is converted into a computational mesh in several minutes. The mesh could be coarse or fine depending on the wish of user.

The mesh is then ready to be used in flow simulators.

2.2.3 Hemodynamic Solver

As a computational hemodynamic solver the Lattice Boltzmann Method(LBM) is used in HemoSolve: In this solver the flow is time-harmonic and after simulation the pressure, velocity, and shear stress fields during one full harmonic period are produced and can be visualized. LBM receives the input geometry mesh from the 3D editing tool.

LBM is a mesoscopic method based on a discretized Boltzmann equation with simplified collision operator [97]. Here the flow is considered Newtonian. We have shown that LBM is capable of solving hemodynamic flows in the range of Reynolds and Womersley numbers of [http://www.google.com/ interest](http://www.google.com/interest) [9, 13, 14]. To run the simulator, except the input data file from 3D editing tool, one should define several patient-specific free parameters such as the Reynolds number.

2.2.4 Validation of Lattice Boltzmann Method

[†] The complexity of the human vascular system and the time-harmonic character of the blood flow make it very difficult to analyze and predict the behavior of flow after surgical interference. In order to exactly mimic this behavior in HemoSolve, well-established flow solvers need to be used.

In order to show that LBM is a comprehensive method to simulate time-harmonic blood flow in the human vascular system we compare it with Finite Element Method (FEM). Several studies of comparison between LBM and FEM methods have shown promising results [40, 54]. Although, in all those studies, whether in 2D or 3D geometries, the fluid flow has been time-independent.

[†]The geometry and FEM data are kindly provided by Dr. Rod Hose and Dr. Adam Jeays, Department of Medical Physics and Clinical Engineering, University of Sheffield.

We compare LBM and FEM solvers for a time-harmonic flow in realistic 3D geometry. As an application we choose to analyze the blood flow in the superior mesenteric artery (SMA). SMA is the part of artery that starts from the anterior surface of the abdominal aorta (AA), distal to the root of the celiac trunk (see Fig. 1.1). Although atherosclerotic disease is rare in this part of the artery its occlusion at SMA leads to the death of patient in the 80% of cases [80].

The analysis between these two methods have been done through:

- comparison of *velocity* profiles at three different cross sections along AA and SMA
- comparison of *pressure* profiles at the outlet of SMA
- fixation of a time and place of flow characteristics such as *vortex* formation next to the bifurcation

In Fig. 2.5 (left) the surface of the superior mesenteric artery is shown.

The data is obtained from MRI scan and a triangular mesh is generated to be used in the FEM solver [53]. The 3D mesh for FEM consists of total 100465 nodes, with 650 nodes on the AA inlet, 295 on the AA outlet and 207 on the SMA outlet. From this mesh a voxel mesh is generated for LBM using the 3D editing and mesh generation tool with 694 nodes on the AA inlet, 316 on the AA outlet and 224 on the SMA outlet. FLOTRAN (ANSYS Inc.) is used as a FEM solver for simulating the 3D Navier-Stokes equations as (see Jeays et al. [53]). As the artery is relatively large the following flow parameters are applied in FEM: the density of the blood ρ is 1000kgm^{-3} , Newtonian viscosity ν is $4\text{mPa}\cdot\text{s}$, the cardiac cycle duration T is 0.86s with five milliseconds time-steps δt and the blood is considered incompressible. The maximum velocity u of 0.8m/s at a peak systole in the aorta, combined with the diameter D of 0.0165m , gives a maximum Reynolds number (Re) of 3300 [53]. In order to apply the same flow condition in the LBM solver we have converted all the parameters into dimensionless units and applied a constraint optimization scheme (see sec. 3.3). This resulted in $D = 30$ lattice points, $T = 17200$ and $\nu = 0.0065$ with $Re = 3300$ for $\alpha = 11$.

As inlet/outlet boundary conditions we applied the same as described in Jeays et al. [53]. The velocities (created using Womersley's solution [117]) are specified at the proximal AA opening, the pressure waveform (created using the Westerhof model [114]) is specified at the distal AA opening and a free flow boundary condition is applied at the outflow of SMA.

In Fig. 2.6 we depicted the velocity profiles at peak systole in the last cardiac cycle executed, when beat to beat convergence had been achieved, along the cutting plane shown in Fig. 2.5 (right). Here we clearly see the expected flow behavior and the formation of vortex in the top of SMA just below the outer wall.

As shown in Fig. 2.5 we compare the obtained velocity profiles along the three A, B and C lines after simulations by both solvers. The flow profiles at regions A and B are shown in Fig 2.7 (left) and (right) respectively and for region C in Fig 2.8.

We see a very good agreement between velocity profiles of FEM and LBM. Our measurements show that the difference between velocities next to the walls is maximum 0.078m/s while in the middle it is about 0.018m/s . Moreover, we see that the maximum

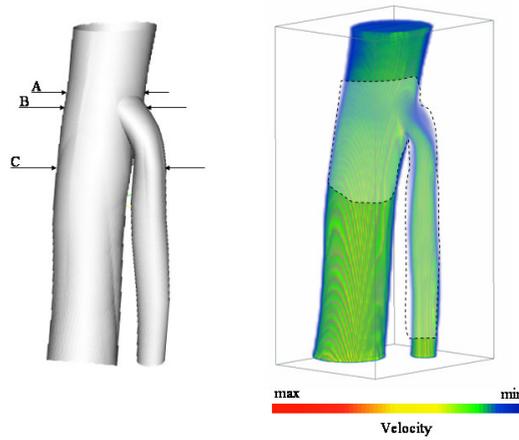


Figure 2.5: The surface of the superior mesenteric artery (left) and a simulated velocity profile in it(right) at $Re = 3300$ with $\alpha = 11$. Here red and blue are the high and low velocities respectively. A , B and C are the cutting planes along which we look into the velocity profiles in detail further on.

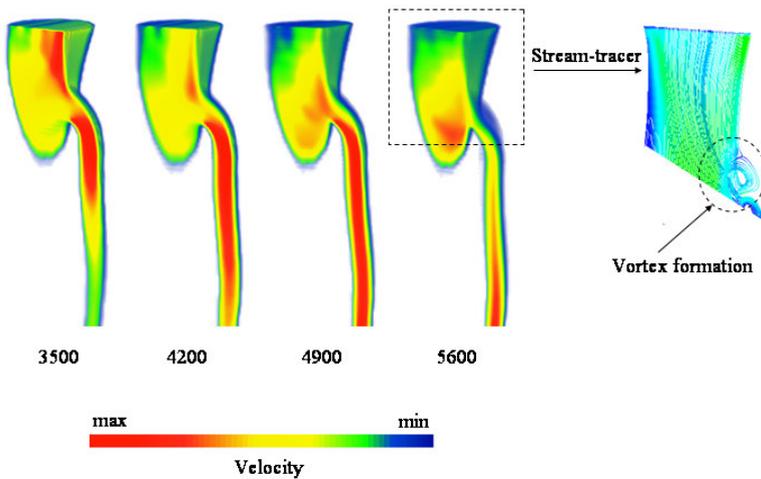


Figure 2.6: The simulated velocity profiles depicted during one systole every 700 time-step (left) and the stream-lines showing the formation of vortex (right).

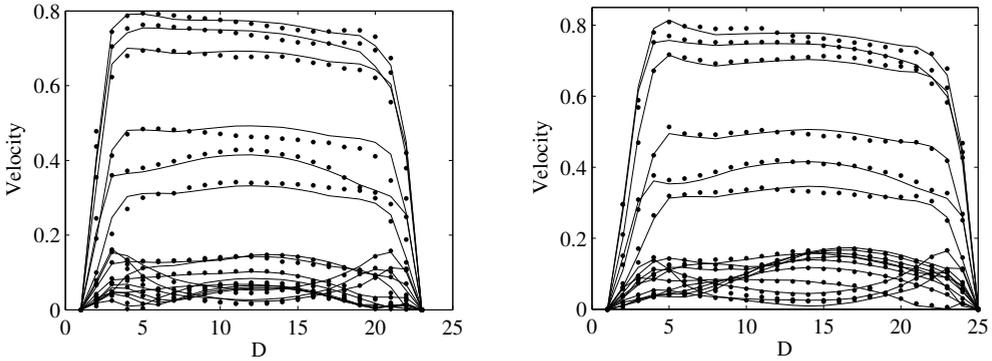


Figure 2.7: Comparison of the velocity profiles between LBM (bullets) and FEM (solid lines) at the region A(left) and B (right)(see Fig. 2.5) for every 0.05s timestep. Velocities are presented in m/s .

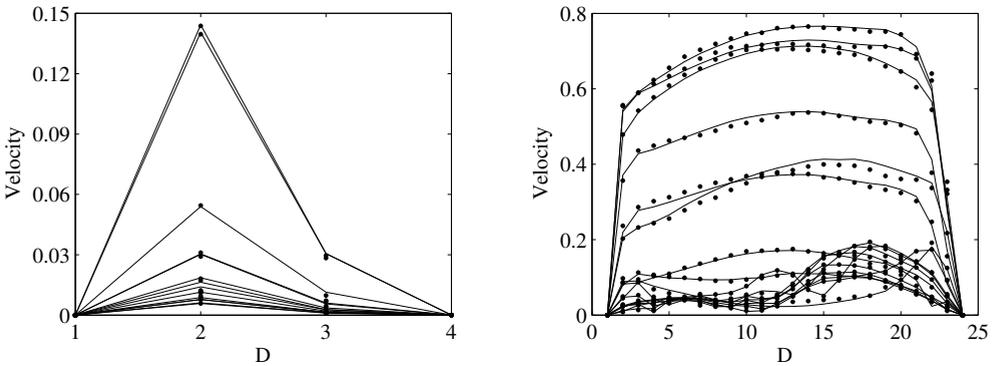


Figure 2.8: Comparison of the velocity profiles between LBM (bullets) and FEM (solid lines) for SMA (left) and AA (right) at the region C (see Fig. 2.5) every 0.05s timestep. Velocities are presented in m/s .

velocity in Fig. 2.7, e.g. before the bifurcation, is tilted to the left while in Fig. 2.8 (right), e.g. after bifurcation, it is near the right side. This indicates a spiraling of the flow in AA. The same effect we see during the flow visualization. This is one of the two interesting characteristics of the flow noted by Jeays et al. [53] for FEM comparable to our observations for LBM. Furthermore, we noted that the flow in the SMA is directed from the bifurcation towards the outer wall, which causes the creation of a vortex at the highest level of the SMA just below the outer wall (see Fig. 2.6). This vortex formation is due to the angle that the SMA forms with the AA [53].

In Fig. 2.9 we have also plotted the pressure profile at the outlet of SMA for both methods during one complete heart beat. Here we also see a very good agreement between both methods.

We have shown a good agreement between velocity as well as pressure profiles of FEM and LBM at all three cross sections. We have also presented the similarities of

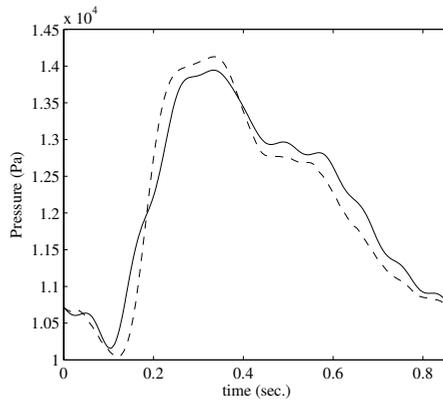


Figure 2.9: Comparison of the pressure profiles between LBM (dash line) and FEM (solid line) at the outlet of SMA during one systole. Pressure is presented in Pa .

flow characteristics, such as the precise timestep and coordinates of vortex formation, for both methods. Thus, based on these arguments we conclude that LBM is a quite eligible method to use as a solver of Navier-Stokes equation for simulations of time-harmonic blood flows in human vascular system.

2.2.5 Flow Analyses

In order to analyze the blood flow in arteries its velocity, pressure and shear stress profiles need to be examined. Several methods exist for it and among them visualization of the flow is one of the advanced methods that helps to understand the meaning and behavior of flow better. Also visualization techniques are different and can show different features of flow. One of the visualization techniques we use in HemoSolve is based on simulated pathline visualization [92].

2.3 Examples

As an application example of using HemoSolve we present two case studies with complex geometries representing parts of the human vascular system:

- Aneurysm in the upper part of abdominal aorta;
- Whole abdominal aorta.

2.3.1 Aneurysm

We consider the case of an aneurysm (ballooning out of the artery) in the upper part of the aorta. First the medical data of the upper part of a patient's abdominal aorta with an aneurysm is segmented by applying the segmentation algorithm (Fig. 2.3 (a,b)). Then

the segmented part which includes the aneurysm, is transferred into the 3D editing tool where the user crops the structurally interesting part (Fig. 2.3 (c)) and defines inlet and outlet layers (Fig. 2.3 (d)). These layers are easy to control, that is to change the plane of their position by simply moving the normal vector in the middle of each layer or to change the size by movement of corner points. In this example there is one inlet at the top and one outlet at the bottom layer. Finally the mesh is generated depending on the constructed geometry (Fig. 2.3 (e)). The mesh is then used as input data for the CFD solver. The presented flow is the velocity profile of blood flow simulated by LBM method. The Reynolds number applied to this blood flow is 500. The size of the generated mesh is $146 \times 73 \times 55$ lattice points and the simulation time is about 20 minutes on 16 processors. As a result three frames during the cardiac systole are captured (Fig. 2.3 (f)).

2.3.2 Abdominal aorta

Next we consider the full lower abdominal aorta down to the bifurcation. Again the medical data is segmented by applying the segmentation algorithm (Fig. 2.4 (a)). Then the same steps as in the first example are applied (Fig. 2.4 (b)), except for the outlet layers that are six now and are in different planes (Fig. 2.4 (c)) and the sizes of generated meshes are bigger (Fig. 2.4 (d)). The ready mesh then used as an input data for CFD solver. The presented flow is the velocity profile of blood flow simulated by LBM method with the same Reynolds number 500. The size of the generated mesh is $355 \times 116 \times 64$ lattice points and the simulation time is 85 minutes on 16 processors. As a result three frames during the cardiac systole are captured (Fig. 2.4 (e)).

2.4 Discussions and Conclusions

The field of image based hemodynamics needs integrated PSEs, especially to enhance the preparation of computational meshes, and to allow non-specialists to enlarge their practical skills. HemoSolve consists of several components that make it a complete system. Thus biomedical engineers, surgeons or novice surgeons can take raw medical data from a patients' vascular system and after several simple steps within a quite short period get completed flow fields (velocity, pressure, shear stress) which they can analyze with different visualization tools. One of this tools is the personal space station (PSS) which supports 3D visualization and interaction [126]. Moreover with the help of the 3D editing tool potential users can add bypasses or stents to vessels and examine the blood flow profile in them. The biomechanists of University of Amsterdam and Leiden University Medical Center have already used HemoSolve in their scientific research. As was reported by Zudilova and Slood [126], two different projection modularities: a virtual reality and a desktop have been tested and compared for vascular reconstruction simulation systems. These studies have been conducted based on the visualization components of HemoSolve. They have reported that the results of user profiling showed that the combination of the virtual reality and desktop capabilities within the same environment is the best solution for potential users.

Another study have been conducted by Twente University to estimate the usefulness of environments like HemoSolve for the university students. It has been reported that virtual learning environments hold great promise as a tool for medical education [70].

An important feature of HemoSolve is also the fact that it is integrated into Grid environment. It is an environment that supports a uniform access to distributed data (images), compute power and, visualization and interaction [102–104]. If the data (CT scan in the hospital), simulation (high performance computing center) and visualization (student desktop) are situated in different geographical places, Grid is the environment to combine them. That is why the fact of HemoSolve being integrated into Grid is one of its important attributes.

In order to estimate the efficiency of HemoSolve we compare its main features with the requirements of users from PSEs in general. Those characteristics are [39,50] :

- Simple human-computer interaction - It is enhanced by graphical user interface which is easy accessible even for inexperienced users [126].
- Complete and accurate numerical models - Numerical model used in this PSE is LBM [9] which is complete and quite accurate for simulation of blood flow in human vascular system.
- Parallel and distributed computing environment - The solvers in PSE are fully parallelized [57] .
- Geographically distributed data - This image-based PSE is completely integrated into a Grid environment, which gives huge abilities not only to distribute the data but also to do simulations and diagnoses by grid computing [104] .
- Usefulness for university students - The potential users of this PSE are considered novice surgeons who can first practice their knowledge by doing an operation on PSE and afterward apply their experience on patients.

We conclude that HemoSolve is a well defined, easy applicable environment for image-based hemodynamics research of time-harmonic blood flow in the human vascular system.

*Efficient LBM for Time Harmonic Flows**



3.1 Introduction

The lattice Boltzmann method (LBM) is a well recognized method in computational fluid dynamics and has attracted much attention [17, 23, 25, 97]. It is widely used in simulations of fluid flow in complex geometries such as fluid flow through porous media e.g. [26, 56, 63] or time-harmonic blood flow e.g. [9, 10, 27, 34, 43]. However, LBM is hampered by slow convergence [42, 59, 106], caused by the demand of low Mach number (Ma) (to suppress compressibility error) and the fulfillment of the Courant-Friedrich-Levy condition for numerical stability. In order to meet these constraints the number of time-steps for reaching steady state needs to be large, and so is the execution time. For this reason we must carefully choose simulation parameters that, given flow properties and a target simulation error, minimize the execution time. We call this the constraint optimization problem for LBM simulations.

In this chapter we propose a solution to the constraint optimization problem for time-harmonic flows. For validation of the constraint optimization scheme we compare results of time-harmonic flow simulations with analytical Womersley solutions [117]. We also perform stability analysis for a range of Reynolds (Re) and Womersley (α) numbers.

Moreover, it is known that L-BGK suffers from numerical instabilities at high Reynolds numbers (Re). Several solutions have been proposed to improve the method in this respect [6, 66]. In this chapter we follow another route, we choose to use the regularization scheme proposed by Latt and Chopard [68]. In this RL-BGK scheme, the simulated kinetic variables are submitted to a small correction (the regularization), after which they only depend on the local density, velocity, and momentum flux. This correction is immediately followed by the usual L-BGK collision term. The effect of the regularization step is to eliminate non-hydrodynamic terms, known as "ghost variables", and to enforce a closer relationship between the discrete, kinetic dynamics and the macroscopic Navier-Stokes equation.

We study the performance of the RL-BGK through a numerical simulation of the time-harmonic Womersley [117] flow for 3D geometries. Further, we analyze the accuracy of

*This chapter is partly based on: L. Axner, A. G. Hoekstra, and P. M. A. Sloot. Simulating time-harmonic flows with the lattice Boltzmann method. In *Phys Rev. E*, 75,3, p.036709, (2007), and partly on: L. Axner, J. Latt, A. G. Hoekstra, B. Chopard, and P. M. A. Sloot. Simulating time-harmonic flows with the Regularized L-BGK Method In *Int. J. Mod. Phys. C*, 18,4, p.661, (2007).

the scheme by comparison of the simulation results with analytical solution. We also present stability analysis of the model for some benchmark flows.

3.2 Lattice Boltzmann Method

We apply the three dimensional 19-velocity (D3Q19) model [123] for time harmonic flows [9]. The fluid flow is quasi-incompressible and all simulations in this chapter (except those in subsection 3.5) are performed on a straight circular tube with rigid walls. On the walls we use Bouzidi boundary conditions (BBC) [19]. For the experiments on the straight tube we use periodic inlet/outlet boundary conditions and the flow was driven by a time-harmonic body force [10]. As we confirmed earlier for time-harmonic flows [11] BBC is more stable and accurate than the bounce back on links boundary condition, especially for high Mach numbers (Ma). For the simulations we use a previously developed highly efficient parallel code. [55].

We define an average simulation error (\widetilde{Er}) as

$$\widetilde{Er} = \frac{1}{T} \sum_t \frac{\sum_x |\mathbf{u}_{th}(\mathbf{x}, \mathbf{t}) - \mathbf{u}_{lb}(\mathbf{x}, \mathbf{t})|}{\sum_x |\mathbf{u}_{th}(\mathbf{x}, \mathbf{t})|}, \quad (3.1)$$

where $\mathbf{u}_{th}(\mathbf{t})$ is the analytical Womersley solution, $\mathbf{u}_{lb}(\mathbf{t})$ is the simulated velocity, and T is the number of time steps per period. For \mathbf{u}_{th} we use the Womersley [117] solution. For flow in an infinite tube driven by pressure gradient $Ae^{i\omega t}$,

$$\mathbf{u}_{th} = \frac{AR^2}{\nu} \frac{1}{i^3 \alpha^2} \left\{ 1 - \frac{J_0(\alpha i^{3/2} x)}{J_0(\alpha i^{3/2})} \right\} e^{i\omega t}, \quad (3.2)$$

where R is the radius of the tube, x is defined as $x = r/R$, $\omega = \frac{2\pi}{T}$ is the circle frequency, J_0 is the zero order Bessel function, ν is the viscosity and α is the Womersley number as defined in Eqs 1.6 and 1.9 Here $D = 2R$ is the diameter of the tube.

As an example, we show in Fig. 3.1 simulation results together with analytical solutions, for $Re = 10$, $\alpha = 6$ and $Re = 3000$, $\alpha = 16$ respectively. Other examples, and more detailed comparisons can be found in [8–10].

The agreement in the case of low Re number is good while for $Re = 3000$ the agreement is less good, especially near the walls. The simulation error in the first case is $\widetilde{Er} = 0.002$, while for the second case is $\widetilde{Er} = 0.092$.

3.3 Constraint Optimization Scheme

In LBM simulations for time-harmonic flows one must specify four free parameters: the diameter D , the period T , the relaxation parameter τ (or related viscosity ν) and the Mach number Ma . The choice of these four parameters not only influences the accuracy of the method, but also other features like stability, convergence and execution time. We will now address the question how to optimally choose these parameters. First we specify Re

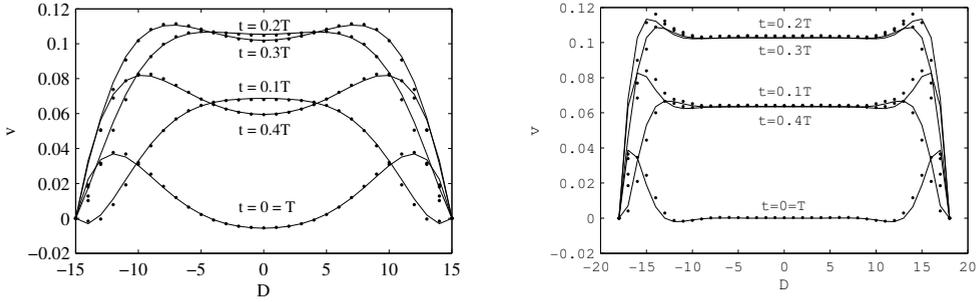


Figure 3.1: Comparison of simulated velocity profile (dots) with analytical Womersley solution (solid line).

and α of the harmonic flow that we wish to simulate. With only these two constraints we have an undetermined system where we can freely choose two simulation parameters. In order to fix them, we apply the following rule. We specify a next constraint, the simulation error that we want to achieve, reducing the number of free parameters to one. Finally, we demand that the remaining simulation parameter should be chosen such that the execution time is minimized. This still leaves us with many ways to select the simulation parameters that we use in the minimization procedure. We have chosen to take τ and Ma as the free parameters that will be fixed by the simulation error constraint and the execution time minimization constraint. These in turn determine the discretizations in time and space.

The Reynolds number Re is defined by Eq. 1.7, and the Mach number Ma is defined in Eq. 1.8.

Combining these equations we get

$$D = \frac{Re}{c_s} \frac{v}{Ma} \quad (3.3)$$

and combining Eqs. 1.9 and 3.3 results in

$$T = \frac{\pi Re^2}{2c_s^2 \alpha^2} \frac{v}{Ma^2} \quad (3.4)$$

Eqs. 3.3 and 3.4 express the spatial and temporal discretizations as a function of two constraints, and the remaining free parameters v and Ma . Here \mathbf{u} can be both negative and positive.

We need to realize that we have to enforce a minimum discretization D_{min} (to guarantee that the geometry is discretized at a minimum accuracy to capture the details of the flow) and T_{min} a minimum required period (due to the Nyquist sampling theorem stating a minimum number of sample points to represent a periodic signal). Due to these minimum discretizations, we can enforce a few constraints on v and Ma . Using Eqs. 3.3 and 3.4 we find

$$\nu \geq \frac{D_{min}c_s}{Re}Ma, \quad (3.5)$$

and

$$\nu \geq \frac{2T_{min}c_s^2\alpha^2}{\pi Re^2}Ma^2, \quad (3.6)$$

Moreover, due to reasons of stability of LBM simulations we must demand that $\nu \geq \nu_{min}$ (e.g. $\tau > \tau_{min}$) [94]. The minimum allowed viscosity, for small Ma , is then determined by this stability constraint. For larger Ma the minimum allowed viscosity is determined by the linear constraint of Eq. 3.5, for $Ma > Ma^*$ the minimum viscosity is determined by the quadratic constraint of Eq. 3.6. Here Ma^* is defined as the intersection between constraints (3.5) and (3.6),

$$Ma^* = \frac{\pi D_{min}Re}{2T_{min}c_s\alpha^2}. \quad (3.7)$$

Typically Ma^* is large, and we only need to worry about the linear constraint of Eq. 3.5. Our task is now to determine \widetilde{Er} in the allowed simulation region, find contours of constant \widetilde{Er} , and minimize the execution time along those contours. One approach would be to measure \widetilde{Er} . Another is to study the asymptotic error behavior and try to obtain analytical expressions. We will follow both approaches.

3.3.1 Asymptotic error analysis

We assume three sources of error, due to spatial and temporal discretization and due to the compressibility error. The errors due to spatial and temporal discretizations are of first or second order, depending on the boundary conditions [11]. The compressibility error is known to be of second order in Ma [51,72]. We write the error as

$$\widetilde{Er} = \frac{k_x^{(n)}}{D^n} + \frac{k_t^{(n)}}{T^n} + k_m(Ma^2) \quad (3.8)$$

where $k_x^{(n)}$, $k_t^{(n)}$ and k_m are coefficients that depend on details of the flow problem but do not depend on spatial or temporal discretization. n equals 1 or 2 depending on the boundary conditions (for BBC $n = 2$). By substituting Eqs. 3.3 and 3.4 into 3.8 and using Eq. 1.6 we find

$$\widetilde{Er} = \zeta_x^{(n)} \frac{Ma^n}{\left(\tau - \frac{1}{2}\right)^n} + \zeta_t^{(n)} \frac{Ma^{2n}}{\left(\tau - \frac{1}{2}\right)^n} + \zeta_m(Ma^2) \quad (3.9)$$

where $\zeta_x^{(n)} = k_x^n c_s^n / Re^n$, $\zeta_t^{(n)} = k_t^n (2c_s^2 \alpha^2)^n / (\pi Re^2)^n$ and $\zeta_m = k_m$. Next demand that \widetilde{Er} has a constant value ε . From Eq. 3.9 we can now derive an iso-error contour for τ as a function of Ma and ε ,

$$\tau = Ma \left(\frac{\xi_x + \xi_t Ma^2}{\varepsilon - \xi_m Ma^2} \right)^{1/n} + \frac{1}{2}. \quad (3.10)$$

The execution time T_{exec} for LBM for time-harmonic flows can be written as

$$T_{exec} = N_p T_p, \quad (3.11)$$

where N_p is the number of periods needed to achieve a stable time-harmonic solution and $T_p = T t_{iter}$ is the execution time per period, with t_{iter} the execution time for one LBM iteration. Finally, $t_{iter} = D^3 t_{node}$, where t_{node} is the time spent to update one node in the lattice. For our simulations we assume that we have D^3 nodes. From previous experiments [11] we know that N_p hardly depends on other parameters and for our case we assume that it is constant. Thus Eq. 3.11 can be written as

$$T_{exec} = N_p T D^3 t_{node} = C_t T D^3, \quad (3.12)$$

where C_t is a constant. If we substitute Eqs. 3.3 and 3.4 into 3.12 and use Eq. 3.9 we get

$$T_{exec} = C_t \frac{\pi Re^5}{2c_s^5 a^2} \frac{1}{Ma} \left(\frac{\xi_x + \xi_t Ma^2}{\varepsilon - \xi_m Ma^2} \right)^{4/n}. \quad (3.13)$$

Independent of n , T_{exec} goes to infinity for Ma decreasing to 0 (because T then goes to infinity, see Eq. 3.4) and for Ma reaching the value $\sqrt{\frac{\varepsilon}{\xi_m}}$ (because ν , and therefore D goes to infinity, see Eq. 3.3 and 3.10) T_{exec} has a minimum between these two limiting values. Later we will compare our experimental results to these analytical solutions. Plots of the asymptotic iso-error contours are shown in Fig. 3.2. Here we assume $k_x = k_t = 1$ and $k_m = 0.05$, and second order boundary conditions, i. e. $n = 2$.

Using Eq. 3.13 we have plotted in Fig. 3.3 $\ln(T_{exec}/C_t)$ as a function of Ma along the analytical iso-error curves. Each of the contours has a minimum point which corresponds to the optimal value of Ma for a certain \bar{Er} . For example for $\bar{Er} = 0.005$, T_{exec} has its minimum at $Ma = 0.18$ and as Ma increase towards $\sqrt{\frac{\varepsilon}{\xi_m}}$, T_{exec} grows to infinity.

3.3.2 Experimental results

We performed three sets of experiments: $Re = 50$, $\alpha = 6$, $Re = 200$, $\alpha = 6$ and $Re = 1200$, $\alpha = 16$. We measured the error on every timestep for a range of values of Ma and τ and from that we compute the average error \bar{Er} and its standard deviation of \bar{Er} . In Table 3.1 the standard deviations for $\bar{Er} = 0.0032$, 0.0054 and 0.009 are shown. Note that \bar{Er} is measured at each time step and averaged over the period.

Fig. 3.4 shows the experimental iso-error curves (markers) for $Re = 50$, $\alpha = 6$. These experimental results suggest that the $Ma - \tau$ correlation is linear. In the limit of small Ma Eq. 3.10 can be written as

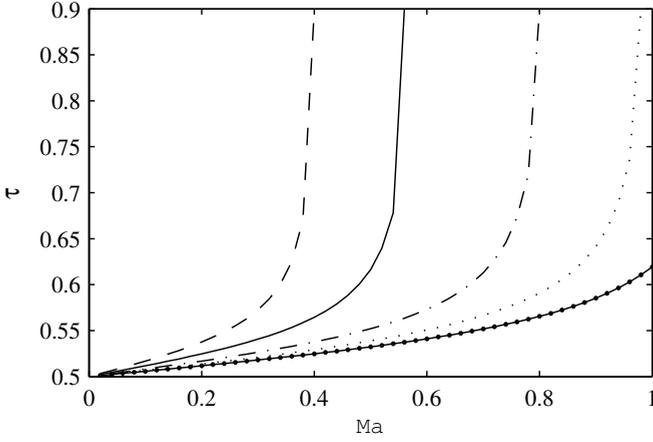


Figure 3.2: Iso-error contours in the (Ma, τ) plane for $Re = 50$ and $\alpha = 6$ at $\widetilde{Er} = 0.005$ (- -), 0.01 (-), 0.02 (- · -), 0.03 (· · ·) and 0.04 (- - -).

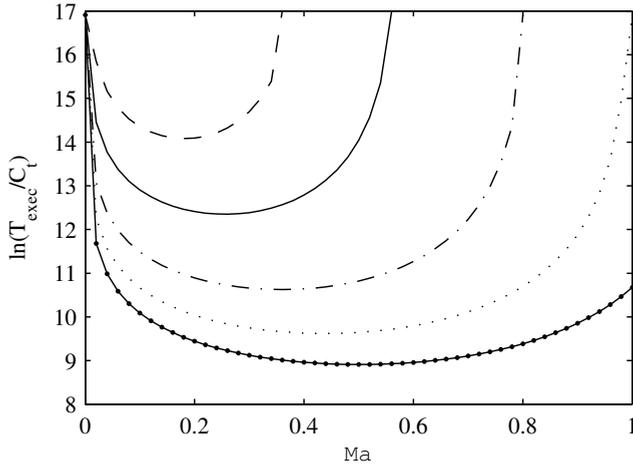


Figure 3.3: $\ln(T_{exec}/C_t)$ as a function of Ma along iso-error contours for $Re = 50$ and $\alpha = 6$ at $\widetilde{Er} = 0.005$ (- -), 0.01 (-), 0.02 (- · -), 0.03 (· · ·) and 0.04 (- - -).

Table 3.1: The standard deviation for several \widetilde{Er} 's.

Re	α	Ma	τ	Er	Dev
50	6	0.2	0.7	3.2×10^{-3}	2×10^{-4}
200	6	0.4	0.9	5.4×10^{-3}	4×10^{-4}
1200	16	0.2	0.7	9×10^{-3}	7×10^{-4}

$$\tau = \left(\frac{\xi_x}{\varepsilon} \right)^{1/n} Ma + \frac{1}{2}. \quad (3.14)$$

Fitting the experimental data with Eq. 3.14 results in a different coefficient k_x for each \widetilde{Er} . To improve this, we fit the experimental data with the analytical iso-error curves (solid lines) for the complete Eq. 3.10 for $n = 2$. We find a very good agreement between experimental and analytical results. Now the values of k_x, k_t , and k_m are the same for all \widetilde{Er} . The same fitting has been done for $Re = 200, \alpha = 6$ and $Re = 1200, \alpha = 16$.

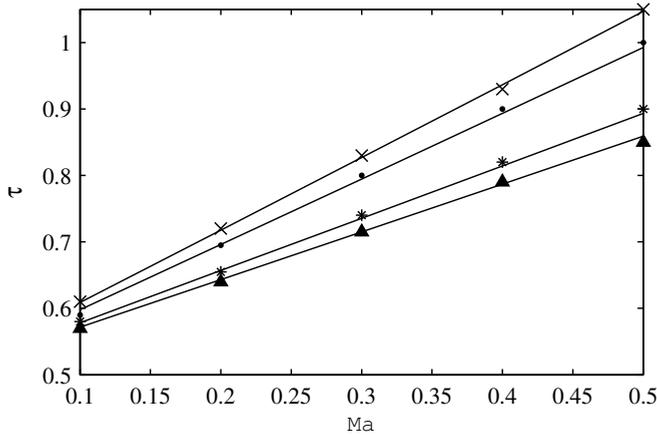


Figure 3.4: Iso-error contours in the (Ma, τ) plane of both analytical and experimental errors $\widetilde{Er} = 0.0026$ (crosses), 0.0032 (bullets), 0.005 (asterics) and 0.006 (triangles) for a range of τ and Ma , for $Re = 50$ and $\alpha = 6$.

In Table 3.2 we show the resulting values of the k_x, k_t , and k_m coefficients.

Table 3.2: The values of spatial k_x , temporal k_t and compressibility error k_m coefficients for $Re = 50, 200, 1200$ and correspondingly $\alpha = 6, 6, 16$.

Cases	k_x	k_t	k_m
Re = 50 $\alpha = 6$	4.8	1.9	< 0.0001
Re = 200 $\alpha = 6$	20	10	< 0.001
Re = 1200 $\alpha = 16$	240	120	< 0.01

In order to evaluate the influence of the spatial, temporal and compressibility errors on the simulation results we fit the obtained data shown in Table 3.2 to Eq. 3.8. The magnitudes of the D and T when $Re = 50, \alpha = 6$, and $Ma = 0.2$ are 38 and 650 respectively for $\widetilde{Er} = 0.0032$. We substitute these parameters into Eq. 3.8 and observe that \widetilde{Er} is mostly caused by the spatial discretization and the influence of the compressibility error is almost unnoticeable (data not shown). This agrees with the result of Shi et al. [88] that

the simulation error is almost not influenced by compressibility error. The results confirm the second order behavior in time and space as demonstrated in [8,60]. The values of k_m in the Table 3.2 are quite small and are the threshold ones, for higher values the fitting breaks down and for smaller ones it does not improve. We measured the standard deviation of the fitting error and for this specific case they are small, e.g. for space discretization it is approximately 0.006. This shows that we have a good agreement between analytical and experimental results.

In Fig. 3.5 we compare the analytical and experimental results of T_{exec} .

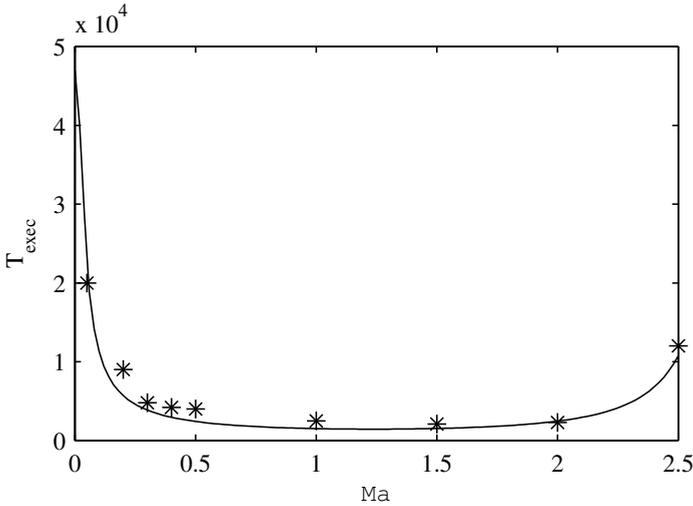


Figure 3.5: Comparison of T_{exec} as a function of Ma for $Re = 50$, $\alpha = 6$ for $\tilde{Er} = 0.0032$ (*) with the analytical T_{exec} (solid line).

The analytical curves are obtained from Eq. 3.13, where $C_t = 40 * t_{node}$. The t_{node} is the time spent to update one node in the lattice and in our case is 2×10^{-7} seconds.

Note that even for very large Ma numbers we accurately reproduce the theoretical Womersley solutions. When inspecting the results as a function of time, the velocity in the tube is close to the theory for both and small and large Ma numbers. This is surprising, as we always assume that $Ma \ll 1$ in order to suppress compressibility errors. We believe that our benchmark, a straight 3D tube may be too simple and allows for such large Ma without deteriorating the results. As is well known, a time-independent 2D Poiseuille flow is an exact solution of the Lattice Boltzmann Equations [44]. If the boundary conditions are chosen in a correct way, the LBM will be exact in that case for any Ma number. We expect, but did not check this explicitly, that this may also hold for time harmonic 2D Womersley flow, and that the results can be carried over to three dimensions. However, in three dimensions the symmetry is slightly broken due to the discretization of the tube. Due to our constraints, a large Ma number means an extremely fine spatial discretization (see Eq. 3.3). This may be the reason that we are capable to run our simulations quite accurate for such large Ma numbers. When going to realistic geometries, as

the aorta, we expect that we must keep the Ma number at more familiar values, i.e. < 1 .

Both plots in Fig. 3.5 have similar behavior, that is for a specific parameter set there exists a minimum T_{exec} . One should be aware of this behavior in order to choose optimal simulation parameters with minimum T_{exec} under the given constraints.

As a conclusion we confirm that using the constraint optimization scheme it is possible to find a parameter set that gives the minimum T_{exec} for a desired \widetilde{Er} . Also in our asymptotic error analysis we have shown the second order behavior of \widetilde{Er} . From detailed comparisons of experimental and analytical results we showed that the $Ma - \tau$ correlation is linear and observed that the error is not influenced by compressibility error. This may be due to the complete symmetry of the geometry, for real cases we expect to see a relevant influence of the compressibility error.

3.4 Stability of Time Harmonic LBM Simulations

Numerical stability of LBM has been studied by many authors e.g. [66,76,94,120]. These studies are mainly performed assuming uniform, time independent background flow. The stability of LBM depends on three conditions [94]. First, the relaxation time τ must be ≥ 0.5 corresponding to positive shear viscosity. Second, the mean flow velocity must be below a maximum stable velocity and third, as τ increases from 0.5 the maximum stable velocity increases monotonically until some fixed velocity is reached, which does not change for larger τ .

In our experiments we fix Ma ($u = 0.1$) and for a range of α we push Re to its highest possible values by decreasing ν . Divergence of momentum profiles is considered to be a definite sign of instability in the system. In order to have a large range of Re number we chose three different cases $D = 24, 36$, and 48 .

In Fig. 3.6 the highest attained Re numbers as a function of α are plotted. We observe the growth of the stability limit for Re with increasing α . As we can see from the plots the maximum Re we reached for static flows is 2300 for $D = 48$. This indicates that the system is still stable for $\tau = 0.506$ and $u = 0.1$. This is comparable to the 2D results obtained by Lallemand and Luo [66].

The threshold values of Re for time-dependent flows are much higher, e.g. $Re = 4000$ for $D = 48$ as $\alpha = 16$.

It is known that the transition into turbulence appears at high Re numbers. The transition starts at $Re_\delta \sim 850$, where Re_δ is the Reynold number based on the Stokes layer thickness [27]. Moreover, experimental results of Shemer [87] indicate that the threshold value of transition into turbulence in a slowly pulsating pipe flow is $Re = 4000$. These results confirm that we observe numerical instabilities in this laminar regime.

In the graph for larger values of α we see almost linear behavior. The interesting part is when the value of α is ≤ 6 while the viscous forces are dominating. Here the limiting magnitude of Re can be quite small. This non-linear behavior for small values of α is remarkable and not understood at all.

With these measurements we confirm that for time-harmonic flows it is possible to reach high Re numbers especially with second order wall boundary conditions. From our

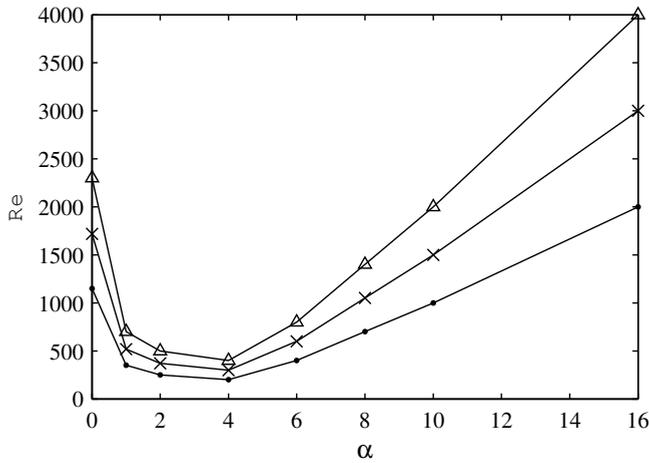


Figure 3.6: The threshold values of Reynolds number for a certain range of Womersley numbers for $u_{max} = 0.1$, $D = 24$ (-●-), 36 (-x-) and 48 (-△-). The stability regions are below the lines.

experiments we observe that, especially for small $\alpha = 2, 3$ with $Re = 10$, when bounce back on links (BBL) boundary conditions were used, the system was less stable.

In Fig. 3.7 we plot the minimum v as a function of $D^2\omega$ in order to compare the minimum τ_s for all $D = 24, 36$, and 48 cases. We see a complete overlap of obtained profiles, which implies that the stability depends on the magnitude of $D^2\omega$.

We also note that the obtained large Re numbers are possible only for the idealized tube. For real cases it is already difficult to reach stability for small Re numbers. In future work we will extend and investigate in more detail all these observations. Currently we do not have a good explanation for the remarkable non-linear behavior we observe in Fig. 3.7.

3.5 Harmonic Flow in Human Abdominal Aorta

We applied a constraint optimization scheme in the simulation of harmonic blood flow in the lower abdominal aorta [10]. The two constraints, $Re = 600$ and $\alpha = 4$, are typical values in the lower abdominal aorta of a person in resting condition [100]. For this simulation we used first order boundary conditions on the walls and time-harmonic pressure difference on inlet/outlet layers. We performed the experiments of a constraint optimization scheme for BBL. We used the resulting values for the simulation of blood flow in the lower abdominal aorta. The errors associated with this boundary condition are at least three times higher than in the case of BBC [11]. For the given Re and α constraints we had to choose $Ma = 0.1$ in order to keep the $\bar{Er} = 10\%$ while minimizing the execution time. Here the relaxation parameter $\tau = 0.514$. The characteristics of observed flowfields together with backflows above the bifurcations are in a good agreement with the results obtained in [10, 100]. In Fig. 3.8 we show one case of our observations near the main

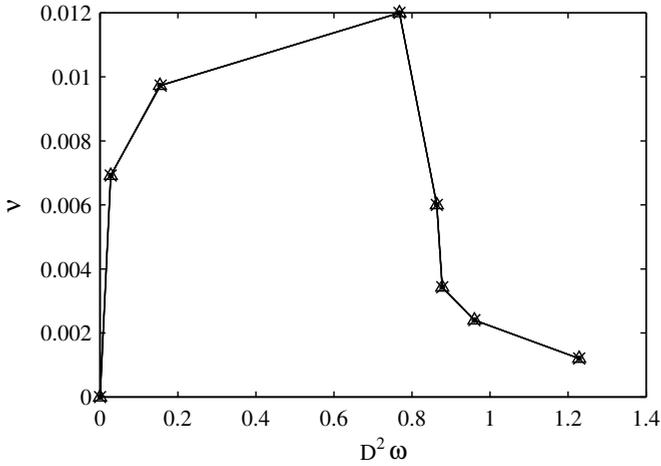


Figure 3.7: The minimum ν as a function of $D^2\omega$ for the given range of Reynolds and Womersley numbers for $u_{max} = 0.1$, $D = 24$ (-●-), 36 (-x-) and 48 (-△-).

bifurcation.

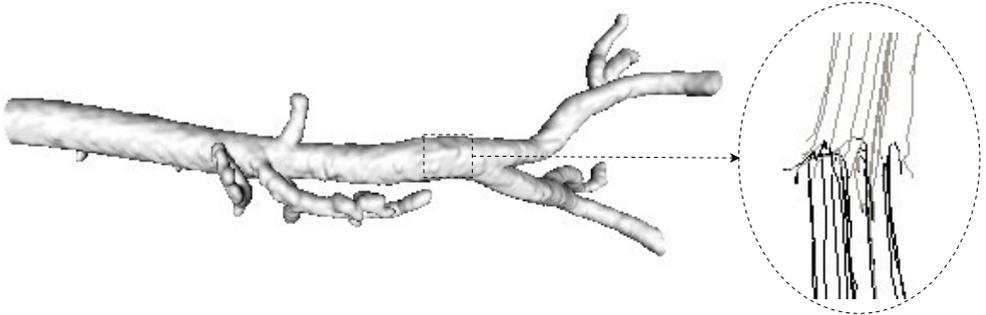


Figure 3.8: Lower abdominal aorta for $Re = 600$ and $\alpha = 4$. Velocity profile shown by streamlines (black - backflows, gray - forward flows).

3.6 Regularized time-harmonic LBGK

In the L-BGK method, the dynamics of the discrete distribution function f_i is described by means of a relaxation to a local equilibrium term (see Eq. 1.2). Thus the evolution of the f 's depends only on the macroscopic particle density (ρ) and fluid velocity (\mathbf{u}), and the non-equilibrium values of the distribution functions, $f_i^{(neq)} = f_i - f_i^{(eq)}$. This becomes clear when the dynamics is cast into the following form:

$$f_i(\mathbf{x} + \mathbf{e}_i, t + 1) = \left(1 - \frac{1}{\tau}\right) f_i^{(neq)}(\mathbf{x}, t) + f_i^{eq}(\rho(\mathbf{x}, t), \mathbf{u}(\mathbf{x}, t)). \quad (3.15)$$

It is well known that Eq. 1.2 can be related to the Navier-Stokes equation for a fluid through a multi-scale, Chapman-Enskog expansion of the f_i 's. This procedure uses an approximation of the non-equilibrium term as $f_i^{(neq)} \approx f_i^{(1)}$, where

$$f_i^{(1)} = -\frac{\tau w_i}{c_s^2} Q_{i\alpha\beta} \partial_\alpha \rho u_\beta, \quad (3.16)$$

In this expression, we have introduced the symmetric stress tensor $Q_{i\alpha\beta} = c_{i\alpha} c_{i\beta} - c_s^2 \delta_{\alpha\beta}$, defined in terms of the Kronecker symbol $\delta_{\alpha\beta}$. Here w_i is the weight factor of corresponding lattice directions. Under the same approximation, the stress tensor $\Pi_{\alpha\beta}^{(neq)} = \sum_i f_i^{(neq)} c_{i\alpha} c_{i\beta}$ can be related to the velocity gradients as follows:

$$\Pi_{\alpha\beta}^{(neq)} \approx \Pi_{\alpha\beta}^{(1)} = -c_s^2 \tau / (\partial_\alpha (\rho u_\beta) + \partial_\beta (\rho u_\alpha)). \quad (3.17)$$

The key idea of the regularized BGK model is to exploit the symmetries both of Q and $\Pi^{(neq)}$ to express $f_i^{(neq)}$ fully in terms of the stress tensor. Indeed, by using the symmetry $Q_{i\alpha\beta} = Q_{i\beta\alpha}$ Eqs. 3.16 and 3.17 can be combined to obtain

$$f_i^{(1)} = \frac{w_i}{2c_s^4} Q_{i\alpha\beta} \Pi_{\alpha\beta}^{(neq)}. \quad (3.18)$$

This concludes the derivation of the regularized BGK model. In this new model, the term $f_i^{(neq)}$ is replaced by $f_i^{(1)}$, and Eq. 3.15 becomes:

$$f_i(\mathbf{x} + \mathbf{e}_i, t + 1) = (1 - \frac{1}{\tau}) f_i^{(1)}(\Pi(\mathbf{x}, t)) + f_i^{eq}(\rho(\mathbf{x}, t), \mathbf{u}(\mathbf{x}, t)). \quad (3.19)$$

All the macroscopic variables, Π , \mathbf{u} and ρ , are computed locally from the particle distribution functions f_i . All the terms contained in Eq. 3.19 can be written out properly so as to obtain a numerical model just as efficient as the original L-BGK.

More details on the regularized LB model, and a discussion of its relation to Multiple relaxation time (MRT) models, can be found in Ref. [68].

Here also we compare the velocity profiles of the simulated flow with the analytical Womersley solution [117].

3.7 Accuracy and Stability of RL-BGK

We use the three dimensional 19-velocity (D3Q19) model [123] for time harmonic flows [9]. The fluid flow is quasi-incompressible in a straight tube with rigid walls. On the walls we use Bouzidi boundary conditions (BBC) [19] and bounce back on links (BBL). The test case presented here is a $36 \times 38 \times 38$ tube with body force driven flow and periodic inlet/outlet boundary conditions. We studied multiple test-cases with a range of Reynolds (Re) and Womersley (α) numbers. We compared the velocity profiles for every $0.1 \times T$ time-step with analytical Womersley solutions and measured the \widetilde{Er} according to Eq. 3.1.

In Fig. 3.9 we show the comparison of the velocity profiles for $Re = 300$, $\alpha = 16$ and $u_{max} = 0.1$ (i.e. Mach number $Ma = 0.2$) velocity case.

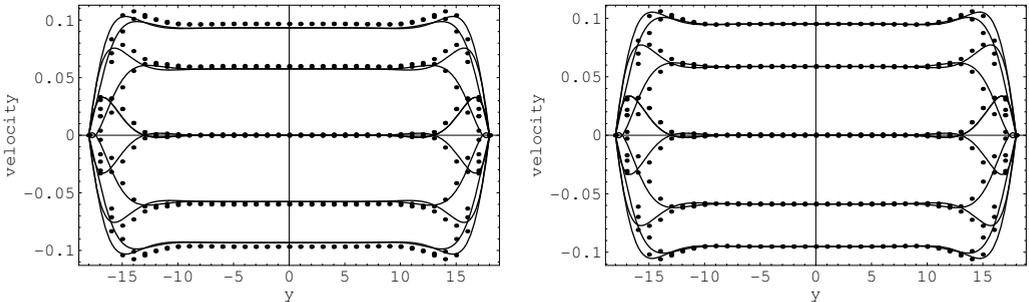


Figure 3.9: Comparison of velocity profiles of simulated (dots) L-BGK (left) and RL-BGK (right) schemes with analytical Womersley (solid lines) solutions for flow in the tube with diameter $D = 36$, $\alpha = 16$ and $Re = 300$ at every $0.1 \times T$ timestep.

From Fig. 3.9 we see that the agreement for the RL-BGK scheme (right) is better in comparison with L-BGK (left). From our simulations we observe that in case of BBC the improvement of accuracy is not as pronounced as in case of BBL. For L-BGK in case of BBL the error $\widetilde{Er} = 9 \times 10^{-3}$ while for RL-BGK $\widetilde{Er} = 3.2 \times 10^{-3}$. Table 3.3 lists \widetilde{Er} for a range of Re and α for both RL-BGK and L-BGK (see sec. 3.3).

Table 3.3: Simulation error \widetilde{Er} for a range of Re and α .

Re	50	100	600	1200	3050
RL-BGK					
$\alpha = 6$	0.003	0.004	0.005	0.02	0.05
$\alpha = 10$	0.003	0.003	0.005	0.01	0.03
$\alpha = 16$	0.002	0.003	0.004	0.01	0.02
L-BGK					
$\alpha = 6$	0.01	0.01	0.02	0.06	0.2
$\alpha = 10$	0.008	0.009	0.01	0.03	0.1
$\alpha = 16$	0.007	0.009	0.01	0.03	0.07

From Table 3.3 we can see that in case of BBL \widetilde{Er} is approximately three times smaller for the RL-BGK scheme. Thus, if we use the RL-BGK scheme aiming at the same accuracy as L-BGK, the execution time will be lower by a factor of $(\sqrt{3})^3 \approx 5$. From our measurement we observe that the execution time of RL-BGK in comparison with L-BGK is larger by $\approx 3\% - 4\%$.

3.7.1 Numerical stability of R-LBGK

Numerical stability of the L-BGK has been an issue for many authors [51, 88, 94]. We measured the numerical stability of the RL-BGK scheme. In these measurements we fix

Ma for $u = 0.1$, and we push Re to its highest possible value for a range of α . For a 3D tube we fix the diameter to $D = 36$ lattice points (see 3.3). We applied two different boundary conditions: Bounce back on links (BBL) and BBC boundary conditions on the walls. We also performed the same stability measurements on a real geometry, a human abdominal aorta with $D_{max} = 18$, and BBL on walls for time harmonic flow [9]. We consider the simulation to be unstable when the total momentum in the system diverges.

In Fig. 3.10 we compared the stability limits of L-BGK with RL-BGK for the same simulation parameters. As we can see from Fig. 3.10 the regularized method is more stable than L-BGK. In case of the tube with BBC boundary conditions the improvement is not as obvious as in case of BBL. With the use of BBL boundaries, RL-BGK simulations could reach a Reynolds number five times higher than L-BGK on a stationary flow, and two times higher on time-harmonic flows. For abdominal aorta the improvement is from three to four times.

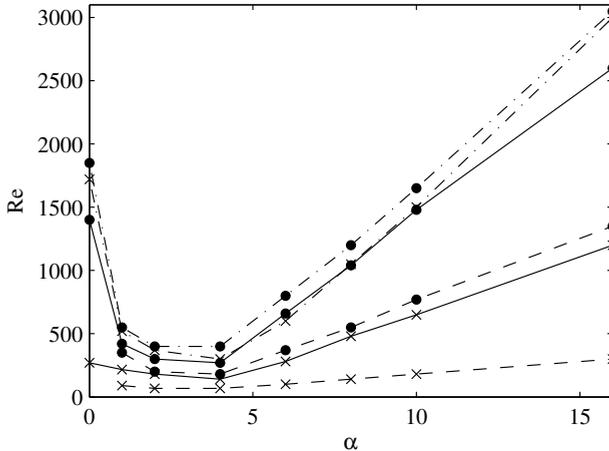


Figure 3.10: Comparison of the threshold values of Reynolds number for RL-BGK(●-) and L-BGK(-x-) schemes for simple tube with $D = 36$ for BBC (- · -) and for BBL(—), and abdominal aorta with $D_{max} = 18$ for BBL(- -). The range of Womersley number is from 1 to 16 for $u_{max} = 0.1$.

Another interesting behavior observed in Fig. 3.10 is the non-linear behavior of the stability limit for small α 's. This phenomena is similar to the one observed in Fig. 3.6 which we currently cannot explain and needs to be investigated further.

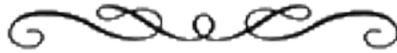
3.8 Conclusions

We have presented results of simulation of time-harmonic flows with LBM. We have defined a constraint optimization scheme and proved that using this scheme it is possible to find a parameter set that gives the minimum execution time for a predefined simulation error. We performed both asymptotic error analysis and numerical experiments. We

conclude that the simulation error is hardly influenced by compressibility error. From stability measurements of the system, we found a unique dependence of the minimum allowed viscosity as a function of $D^2\omega$. This needs further study. Finally, we applied the constraint optimization scheme on our simulations of blood flow in the lower abdominal aorta. We have performed examinations of flow profiles and showed a qualitative agreement with experimental results.

We also have presented numerical simulation results of time-harmonic Womersley flow in a 3D tube by applying the regularized L-BGK scheme. We compared the velocity profiles from simulations with the analytical Womersley solution. We showed that the accuracy associated with RL-BGK is higher than with L-BGK by comparing the simulation errors. This implies that if we use the RL-BGK scheme with the same accuracy as L-BGK the execution time will decrease significantly (see Table 3.3). We also presented the stability measurements for the RL-BGK scheme where we reached higher Reynolds numbers than in case of the L-BGK scheme. For a 3D time-harmonic flow we achieved three to five times higher Reynolds numbers which is comparable with the results obtained in Ref. [68] for 2D cavity flow. The algorithm can be used to more efficiently calculate 3D time-harmonic flows like blood flow in abdominal aorta.

*High Performance Lattice Boltzmann Computing**



4.1 Introduction

The lattice Boltzmann method (LBM) is a well established scheme in Computational Fluid Dynamics (CFD) [22, 97, 123] and has proven to be a promising and reliable solver for fluid flow simulations in complex geometries such as porous media, fissured rocks and geometries from medical applications [9, 10, 24, 26–28, 34, 56, 63]. However, for such large and complex geometries the simulations are known to be computationally intensive. This is why there is an ongoing interest towards optimal LBM simulation codes and efficient parallelization strategies [11, 32, 33, 57, 58, 67, 69, 78, 85, 105, 113]. Key to achieve an optimum in parallel performance is the correct choice of the domain decomposition method, to preserve the workload balance and to minimize the interprocessor communications.

A new LBM approach based on sparse matrix linear algebra is proposed by Schulz et al. [85]. Here the computational domain is mapped on an unstructured grid, where the geometrical ordering of nodes is rearranged and a new index list of only fluid nodes is composed. Thus each fluid node has an adjacency index list of its neighboring fluid nodes, while the solid nodes are completely eliminated. This method requires storage of three one-dimensional arrays: two for density distributions (as a double buffering is used) and the other one for neighbor node indices. Next, domain decomposition has been performed for this scheme by using the METIS partitioning library [5, 61, 62]. With this parallelized sparse-LBM approach memory consumption is essentially minimized and a highly optimized parallel performance can be achieved.

In this chapter we extend this initial parallelized sparse-LBM approach by performing detailed analysis of domain decomposition methods within the METIS library and their comparison to a one-dimensional decomposition. This is done in terms of estimation of the number of edge-cuts (to find the amount of data that needs to be communicated between processors) and an estimation of the amount of fluid nodes per processors (to

*This chapter is based on: L. Axner, J. Bernsdorf, T. Zeiser, P. Lammers, J. Linxweiler and A.G. Hoekstra Performance evaluation of a parallel sparse Lattice Boltzmann solver (submitted for publication in *J. of Comp. Phys.*)

assess the load imbalance). Further we analyze performance results in terms of parallel scalability, and find the sources of loss of parallel efficiency, as well as predict the behavior of the solver for other geometries and/or on other systems using a detailed performance prediction model. This model includes both details of the parallel architecture (single node performance, point-to-point communication) and details of the domain decomposition. The sources of loss of efficiency are estimated in terms of fractional overhead functions [37,38].

In section 4.2 we give a short overview of our sparse-LB implementation, section 4.3 describes the general idea behind partitioning methods. In section 4.4 we develop the performance prediction model and in section 4.5 we show the results of the parallel performance on different machine architectures and compare with the scalability prediction model. In section 4.6 we present our conclusions.

4.2 Sparse LB Implementation

The lattice Boltzmann method is said to be very efficient and easy to implement. But in most cases, a simple full matrix implementation is used, where not only the fluid and boundary nodes, but also the solid fraction is allocated in computer memory. Depending on the geometry, this is a considerable waste of resources, and also of CPU-cycles due to the long loops over solid nodes in order to reach the fluid ones. In the framework of a simple full-matrix implementation, the density distribution array for the whole bounding box is allocated in memory. This results in $2 * 19 * l_x * l_y * l_z$ REAL numbers for the D3Q19 model for a $l_x * l_y * l_z$ lattice [18,112].

Well-known methods from sparse matrix linear algebra were first applied to the lattice Boltzmann method by Schulz et al. [85]. Here the storage of the density distribution is only for the fluid nodes and thus the complete domain is mapped on an unstructured grid. Instead of geometrical ordering of those fluid nodes, a new index list is composed, where each node has the index list of its neighboring fluid nodes. In this method all data is stored in 1D arrays, where only $2 * N * 19$ REAL numbers are stored for the density distribution (N is the number of fluid nodes) and $N * 18$ INTEGERS for the adjacency list. This approach allows to save a considerable amount of memory.

4.3 The Graph Partitioning Algorithms

The goal of the graph partitioning algorithms is to partition large and complex graphs such that the edge-cut is minimized (and thus the related communication overhead) while trying to keep the number of nodes on the sub graphs as balanced as possible (thus minimizing load imbalance). In sparse LBM the computational domain is mapped on a graph, where graph vertexes represent the fluid nodes and edges are the links connecting the fluid nodes. From this graph the 1D array of fluid nodes and adjacency index list, as described in section 4.2, are constructed.

In previous studies, the METIS graph partitioning library was used to compute partitions from the graphs [33,85]. However, in neither of these studies one can find in-

formation or an evident objective for the chosen function from the multiple partitioning functions of METIS.

The METIS partitioning library contains different functions for graph and mesh partitioning. METIS reduces the size of the original graph by collapsing vertices and edges, partitions a smaller graph and then uncoarsens it to construct a partition for the original graph [5, 61, 62]. In this way it produces high quality partitions and in addition is extremely fast. The partitioning functions are mainly based on the combination of two algorithms: modified Kernighan-Lin and Fiduccia-Mattheyses.

The Kernighan-Lin starts with an initial bipartition of the reduced size graph such that each half contains roughly half of the weight of the original graph and each further iteration it searches a subset of vertices/nodes from each part of the graph, such that their swapping will create a partition with smaller edge-cut [62]. The algorithm continues until it cannot find any two such subsets. The Kernighan-Lin algorithm takes $O(|E|\log|E|)$ time while Fiduccia-Mattheyses takes $O(E)$ [61], where E is the number of edge-cuts.

Fiduccia-Mattheyses can work with multiple partitions instead of two. The main achievement of this algorithm is its ability of analyzing the effect of moving a single vertex. That is, it can compute the change of the number of edge-cuts, caused by the movement of the current vertex from one partition to another, which is called gain. The algorithm repeats by selecting the vertex with the largest gain from the larger partition and moves it to another partition. To prevent the same vertex from moving in the same direction the algorithm marks the vertex [62].

In METIS the Kernighan-Lin partitioning algorithm is modified by the gain computing quality of the Fiduccia-Mattheyses algorithm. Moreover, the coarsening, partitioning and uncoarsening phases are also enhanced with a number of improvements. Thus, depending on the demands of the fluid flow solver one can adjust the parameters of the partitioning functions to achieve an approximate balance between maximum load balance and minimum communication amount. Moreover, METIS allows to use a (heavily) weighted graph, i.e. a graph with labels on nodes and/or edges, to strengthen the different features of partitions. The weights on edges of the graph can be used to strengthen the connection between nodes thus to help to minimize the communication amounts. The weights on the nodes can be used to strictly preserve the load balance between partitions.

We use two different functions of the METIS partitioning library: Multilevel K-way and Multilevel Recursive Bisectioning (RB). The difference is that the *multilevelRB* approach computes a k-way partitioning by performing recursive bisectioning on all phases of the multilevel graph partitioning. On the other hand, the *multilevelK-way* performs coarsening only once, then the coarsest graph is directly partitioned into k parts and finally the uncoarsening phase is also performed only once, during which a k-way partitioning refinement algorithm is applied to improve the quality of the partitions [62].

4.4 The Performance Prediction Model

In order to analyze the parallel scalability of the sparse LB solver and to find the sources of loss of parallel efficiency we develop a performance prediction model. Using this model

it is also possible to predict the parallel performance of the solver for other geometries. Inspired by the work of Fox et al. [38], who introduced the concept of fractional communication overheads, we write the equation for $T_p(N)$, the execution time on p processors for a problem containing N fluid nodes as

$$T_p(N) = \frac{T_1(N)}{p} + T_{comm}(N), \quad (4.1)$$

where T_{comm} is the communication time. Note, that we assume that the communication time on each processor is assumed to be the same, and the computational time on each processor can be written as $T_1(N)/p$. This will be discussed in more detail below.

Now we can immediately write for the parallel efficiency $\varepsilon_p(N)$

$$\varepsilon_p(N) = \frac{T_1(N)}{p * T_p(N)} = \frac{1}{1 + f_{comm}}, \quad (4.2)$$

where f_{comm} is the fractional communication overhead defined as

$$f_{comm} = \frac{p * T_{comm}(N)}{T_1(N)}. \quad (4.3)$$

We note that if we express the parallel execution time as

$$T_p(N) = \frac{T_1(N)}{p} + \sum_i T_i(N), \quad (4.4)$$

where $T_i(N)$ are all overheads that can be identified, the efficiency can be expressed as

$$\varepsilon_p(N) = \frac{1}{1 + \sum_i f_i}, \quad (4.5)$$

with f_i a fractional overhead defined as

$$f_i = \frac{p * T_i(N)}{T_1(N)}. \quad (4.6)$$

Next we will develop a detailed performance model for an application of size N . We assume that N discretizes some computational domain, and that the computation on each grid point is a stencil based operation, i.e. only information from neighboring grid points is needed to update the grid point itself. Parallelization is achieved by dividing the domain into p sub-domains, where each sub-domain has n_j points and

$$N = \sum_{j=1}^p n_j. \quad (4.7)$$

Note, that the n_j are not necessarily equal to N/p , i.e. we assume load imbalance, and therefore we expect to find some form of overhead induced by this load imbalance. Next, due to the stencil operation the amount of communication per domain is determined by the boundary of the sub-domain, which we denote by δn_j . Finally, we assume

that the total execution time per processor is determined by computation on the domain n_j followed by communication on the boundary δn_j . This means that we exclude for now the possibility of latency hiding (a technique that may improve the performance a lot, and that could be considered when the fractional communication overheads become large) [115].

Now call t_j the execution time on processor j and write this as a summation of computation time (t_{comp}) and communication time (t_{comm}).

$$t_j = t_{comp}(n_j) + t_{comm}(\delta n_j) \quad (4.8)$$

The execution time on p processors is now determined by the slowest processor, i.e.

$$T_p(N) = \max_j \{t_j\}, \quad (4.9)$$

and

$$T_1(N) = t_{comp}(N). \quad (4.10)$$

Without any further assumption write

$$T_p(N) = \frac{t_{comp}(N)}{p} + \max_j \{t_j\} - \frac{t_{comp}(N)}{p}, \quad (4.11)$$

and

$$\varepsilon_p(N) = \frac{1}{1+f} \quad (4.12)$$

with a single fractional overhead defined as

$$f = \frac{p * \max_j \{t_j\}}{t_{comp}(N)} - 1. \quad (4.13)$$

To proceed we need to find approximations for $T_p(N)$. We define an upper bound to the execution time and use it as an approximation written as

$$T_p(N) = \max_j \{t_j\} \leq \max_j \{t_{comp}(n_j)\} + \max_j \{t_{comm}(\delta n_j)\} \quad (4.14)$$

The computational time is linear in the number of grid points. If we call τ_{comp} the time required to execute one grid point, and realize that due to caching and memory layout of the data τ_{comp} actually depends on the number of grid points we find that

$$t_{comp}(n) = n * \tau_{comp}(n). \quad (4.15)$$

For communication we assume a linear model for point-to-point communication, i.e. to send m density distributions requires a time $\tau_{setup} + m * \tau_{send}$. τ_{setup} is the time needed to initialize the communication and τ_{send} is the time needed to send one byte of information. Each processor j will send and receive data to and from d_j other processors. This number d_j depends on the details of the partitioning of the computational domain. To each of the d_j processors an amount of e_{jk} density distributions (with $k = 1 \dots d_j$ and 8

bytes per distribution) is communicated. Moreover, the total number of grid points that are sent to other processors is the edge-cut e_j (in bytes) which is

$$e_j = \sum_{k=1}^{d_j} e_{jk}. \quad (4.16)$$

In stencil based operations the communication is always an exchange operation, that is, an amount of data is sent to a processor, and the same amount is received back again. We assume that this is implemented as two synchronous blocking point-to-point communication routines. With all these definitions we can now write a closed expression for the communication time from processor j as

$$t_{comm}(\delta n_j) = 2 \sum_{k=1}^{d_j} (\tau_{setup} + e_{jk} \tau_{send}) = 2d_j \tau_{setup} + 2e_j \tau_{send}. \quad (4.17)$$

We assumed that the point-to-point communication between any pair of processors in a parallel computer is equal. In section 4.5 we will refine this further, by distinguishing between two types of point-to-point communication.

From the partitioning tool METIS we can get all information, for each processor, on the number of nodes n_j , the degree of connectivity d_j and the edge-cut e_j .

With all these definitions we finally find

$$T_p(N) = n_{max} \tau(n_{max}) + \max_j \{2d_j \tau_{setup} + 2e_j \tau_{send}\} \quad (4.18)$$

By casting Eq. 4.18 in the form of Eq. 4.4 we find

$$T_p(N) = \frac{N}{p} \tau(N) + n_{max} \tau(n_{max}) - \frac{N}{p} \tau(N) + \max_j \{2d_j \tau_{setup} + 2e_j \tau_{send}\} \quad (4.19)$$

The last term in Eq. 4.19 is due to communication and gives rise to a fractional communication overhead, just like in Eq. 4.3. The second and third terms are in fact a mix of two effects, namely load imbalance and the potential change of the speed of a single processors due to the partitioning. We can separate them by writing

$$n_{max} \tau(n_{max}) - \frac{N}{p} \tau(N) = n_{max} (\tau(n_{max}) - \tau(N)) + \tau(N) \left(n_{max} - \frac{N}{p} \right) \quad (4.20)$$

Now the processor speed effect and load imbalance are clearly separated. The first term on the right hand side of Eq. 4.20 is due to the processor speed effect, whereas the second is due to load imbalance.

This means that we find three fractional overheads:

- a fractional communication overhead

$$f_{comm} = \frac{p * \max_j \{2d_j \tau_{setup} + 2e_j \tau_{send}\}}{N * \tau(N)}; \quad (4.21)$$

- a fractional load imbalance overhead

$$f_l = \frac{p * n_{max}}{N} - 1; \quad (4.22)$$

- a fractional processor speed overhead

$$f_s = \frac{p * n_{max} (\tau(n_{max}) - \tau(N))}{N * \tau(N)} = \frac{p * n_{max}}{N} \left(\frac{\tau(n_{max})}{\tau(N)} - 1 \right). \quad (4.23)$$

The communication overhead and fractional load imbalance overhead are always equal or larger than zero, giving rise to a loss of efficiency. However, this is not the case for the fractional processor speed overhead. On cache based microprocessor the speed of the processor is typically faster for small problem sizes, when the problem fits completely in cache. If the original problem size N does not fit in cache and the decomposed problem size n_{max} does, we find that $\tau(n_{max})/\tau(N) < 1$ and therefore $f_s < 0$, resulting in an increase of the parallel efficiency. This effect is well known and appears in the literature as super linear speedup, i.e. the situation where a measured speedup is larger than p , or equivalent, the efficiency $\varepsilon_p > 1$. Our analysis reveals that this happens when the summation over all fractional overheads is smaller than zero, or in this case, when $-f_s > f_{comm} + f_l$.

Note that this is all based on the upper bound estimation of the execution time (see Eq. 4.14), so we expect to overestimate the execution time and therefore underestimate the parallel efficiency.

4.5 Results

We perform an extended set of experiments to assess the efficiency of the partitioning algorithms, both for complex and simple geometries. The geometries we use are the human abdominal aorta (AA), a porous media with high solid fraction (PM) and a straight square channel (SSC) (see Fig. 4.1). For all three geometries we have chosen three different domain sizes containing $5 * 10^6$, 10^5 and $5 * 10^4$ fluid nodes respectively. Thus in total we have nine experimental data sets. On each of these data sets we apply both the *MultilevelK-way* and *MultilevelRB* partitioning algorithm and examine their behavior. Due to special treatment of the inlet/outlet boundaries in the current version of our LBM solver we prefer to keep the nodes of first three layers at the inlets and outlets completely in one partition. To achieve this we put high weights on edges of those layers and put low weights on the rest of the edges throughout the complete graph. We compare the amount of edge-cuts between partitions and the amount of fluid nodes per partition. We also compare these two algorithms with 1D bisection (operated on the original 3-D domain by allowing only flat interfaces) for SSC. Finally the execution time for all nine sets

of data is measured on two different platforms: a NEC SX-8 vector machine [1] (the characteristics are given in Table 4.1), and a PC cluster [2] (the characteristics are given in Table 4.2), using from 1 up to 128 processors. In order to be able to compare the results with the performance prediction model and to find the sources of loss of parallel efficiency, we need first to evaluate the basic parameters connected with the parallel architecture (single node performance, point-to-point communication) and with the parallel decomposition (load imbalance, edge-cuts, degree distribution).

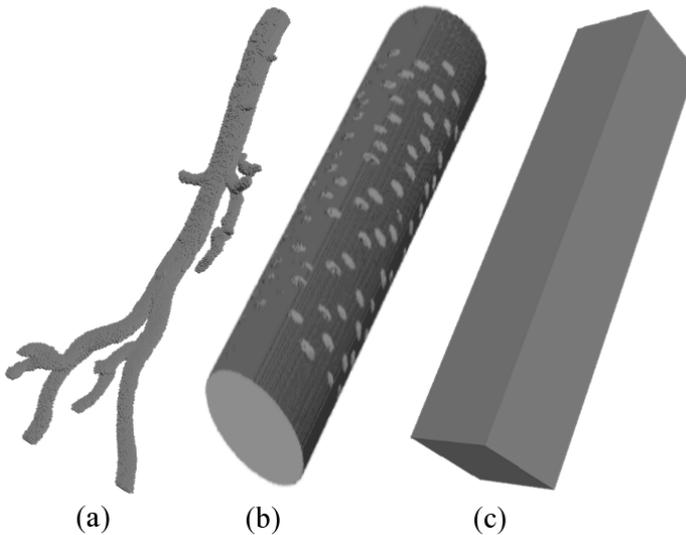


Figure 4.1: (a)- Abdominal aorta, (b) - porous media, (c) - straight square channel.

Table 4.1: NEC SX-8 specifications (see: <http://www.hpce.nec.com/>).

	Multi-Node	Single-Node	
	2-512 nodes	1 node	
	SX-8R/M	SX-8R/A	SX-8R/B
Central Processing Unit			
Number of CPUs	8 - 4096	4 - 8	1 - 4
Logical Peak Perf.	299.2GF - 153.1TF	149.6GF - 299.2GF	37.4GF - 149.6GF
Vector Peak Perf.	281.6GF - 144TF	35.2GF - 140.8GF	35.2GF - 281.6GF
Main Memory Unit			
Memory Archit.	Shared and Distr.	Shared	
Capacity	64GB - 128TB	32GB - 256GB	32GB - 128GB
Peak Data Transf. Rate	288TB/s	563.2 GB/s	281.6 GB/s
Internode Crossbar Switch (IXS)			
Peak Data Transf. Rate	16GB/s per node bi-directional	-	

Table 4.2: Specifications of the 2IntelXeonTM Lisa cluster (see: <http://www.sara.nl/userinfo/lisa/description/index.html/>).

Compute nodes	680			
	Number	Memory	L2 cache	InfiniBand
	255	4GB	2MB	Yes
	260	2GB	1MB	Yes
	165	4GB	2MB	No
Speed	3.4 GHz			
Memory	Extended Memory 64 Technology (EM64T)			
Total Peak Performance	8.5 TF/s			
Disk Space	15 TB SGI Infinite Storage TP9300S disk array for home file system			
	Topspin's InfiniBand Network			
Bandwidth	800 MB/s			
Latency	<6 μ sec			

4.5.1 Basic parameters

1. **Single processor speed** - We first measure $t_{comp}(N)$ by running multiple simulations with a sufficient range of data sizes on a single processor and using Eq. 4.15 to extract $\tau_{comp}(N)$. Our experimental results (Fig. 4.2) show that on the PC cluster $\tau_{comp}(N)$ is almost constant if $N < 5 * 10^3$ and $N > 10^5$ and increases in between. Thus we have divided the complete set of measurements into three regions. For small and large values of N , $\tau_{comp}(N)$ is assumed to be constant, and in the middle region we assume that $\tau_{comp}(N)$ depends linearly on N . As mentioned in section 4.4, $\tau_{comp}(N)$ depends on the cache size of a specific processor. For the PC cluster the cache size is 1Mbyte. From our computations we know that the memory consumption per lattice node is $2 * 8 * 19 = 304$ bytes (for 19 distributions). Thus if the number of lattice nodes per processor is $N < 3 * 10^3$, the data will completely fit into cache.

It is well-known that vector machines such as the NEC SX-8 perform best for large data sizes. The experimental results show a rather smooth decrease of $\tau_{comp}(N)$ as N increases. Assume that the execution time depends linearly on N (e.g. a constant time to fill up the vector units, and then a time proportional to N to produce all results), then the execution time can be written as $a * N + b$. Using Eq. 4.15 we find $\tau_{comp}(N) = a + b/N$, and this function we use to fit the measurements.

In Fig. 4.2 and 4.3 we have plotted $\tau_{comp}(N)$ for the LBM solver on a single processor as a function of the number of fluid nodes for PC cluster and the NEC SX-8 machine respectively. We see a good agreement between measurements, together with the fitting results. A clear discontinuity due to the cache size for the PC cluster at $N = 3 * 10^3$ data size can be observed. In Fig. 4.3 we clearly see that the bigger the problem size the better the performance. We have fitted the experimental results to the models as described above, and the results are summarized in Table 4.3.

In the figures, the seemingly misfit of measurements and fitting results between $N = 3 * 10^3$ and $N = 10^5$ data sizes is due to the linear interpolations.

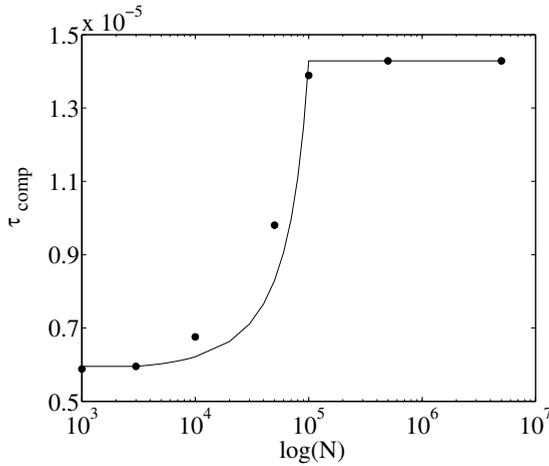


Figure 4.2: Single processor performance on the PC cluster as a function of data size; (bullets) are the measurements, (line) is a fit to the data, using the performance model presented in the sec. 4.4.

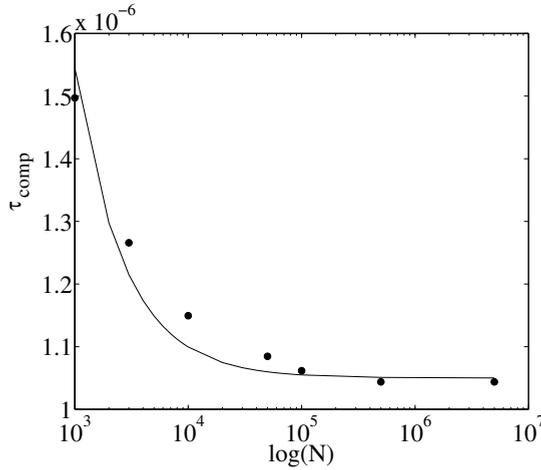


Figure 4.3: Single processor performance on the NEC SX-8 vector machines as a function of data size; (bullets) are the measurements, (line) is a fit to the data, using the performance model presented in the sec. 4.4.

2. **Partitioning** - In Fig. 4.4 we show the SSC and the AA data sets partitioned using the *multilevelRB* and *multilevelK-way* algorithms. We see that the *multilevelRB* creates more sliced cuts between partitions, while the partitions of *multilevelK-way* are less structured and have curved cuts. However, for both cases the average number of fluid nodes per processor is very close to N/P . Also, we measured the execution times of both methods for two different 10^5 and 10^6 problem sizes (see section 4.5). The measured execution times for *multilevelK-way* partitioning are equal 0.9 and 1.3 minutes respectively while for *multilevelRB* they are 2.1 and 2.5

minutes (both executed on a single processor of a PC cluster).

To get a better idea of the load imbalance, in Fig. 4.5 we have plotted the standard deviation of the distribution of the number of fluid nodes on processors as a function of p . From the measurements of standard deviation we hardly observe a difference between partitionings using *multilevelRB* and *multilevelK-way* on all geometries, except for SSC for a very large number of partitions. This is due to the fact that *multilevelRB* performs bipartitioning along the shortest latitude of the geometry until it reaches the minimum possible number of layers per partition and continues bipartitioning along the longitude which causes a high load imbalance for $p = 128$. For complex geometries we prefer to follow the advice of Karypis et al. [62] and use *multilevelRB* for less than eight partitions while apply *multilevelK-way* for a larger number of partitions. Moreover, we have compared these algorithms with a 1D bisection and noted that both METIS algorithms performed better. That is, even though 1D bisectioning divides the domain into equal parts, after a certain number of partitions the deviation of number of fluid nodes per partition becomes quite high from the average number of fluid nodes per partition.

3. **Point to point communication** - This overhead depends on the specifics of the machine architectures and the details of the edge-cut and degree of connectivity of the partitions. Moreover, most modern machines have more than one processor per node, and therefore we must also distinguish between on-board communication (with both processors on the same node) and off-board communication (between processors on different nodes). In our case this applies only to the NEC SX-8 which has eight processors per node.

To measure the point-to-point communication we run a "pingpong"-type communication with different data sizes between two processors. Next we fit the data to a linear function and obtain τ_{setup} and τ_{send} . The results are shown in Figs 4.6 and 4.7. Moreover, in Fig. 4.7 we see a clear difference between communication times for on-board and off-board measurements. Thus in our further predictions for the NEC SX-8 for $p \leq 8$ we take the on-board values and for $p > 8$ the off-board ones. We also note that for $N = 1Kbyte$ a discontinuity appears. This is a buffering effect in the MPI communication routines. From the fits we extract values of τ_{setup} and τ_{send} , which are shown in Table 4.4.

Next we consider the edge-cuts per processor obtained from the partitioning algo-

Table 4.3: Single processor performance.

<i>Platform</i>	N	$\tau_{comp}(N)(sec.)$
PC cluster	$< 3 * 10^3$	$0.58 * 10^{-5}$
	$> 10^5$	$1.42 * 10^{-5}$
	Otherwise	$0.58 * 10^{-5} + \frac{1.42 * 10^{-5} - 0.58 * 10^{-5}}{10^5 - 3 * 10^3} * (N - 3 * 10^3)$
NEC SX-8	For all data sizes	$1.05 * 10^{-6} + 4.95 * 10^{-4} / N$

rithms. For complex geometries like AA or PM both the *multilevelK-way* and *multilevelRB* behave almost similarly as shown in Fig. 4.8. The standard deviations of edge-cuts are in the order of 25%. Considering the fact that the *multilevelK-way* is faster but the partitions of *multilevelRB* are more structured we follow the advice of Karypis et. al [62] and use *multilevelRB* if $p \leq 8$ and the *multilevelK-way* otherwise. However, in the case of SSC after certain number of partitions (e.g. 64 in Fig. 4.8) the standard deviation of edge-cuts produced by *multilevelRB* is in the order of 30% while in the case of the *multilevelK-way* it is 40%. Thus for simple geometries like SSC we prefer to use *multilevelRB* as the communication time associated with it will be less. We also conclude that the difference between these two partitioning functions is mainly in the number of edge-cuts rather than load imbalance.

Table 4.4: Point to point communication overhead.

<i>Platform</i>	$N(\text{bytes})$	$a(\text{seconds})$	$b(\text{second/bytes})$
PC cluster	< 1024	$6.34 * 10^{-6}$	$4.77 * 10^{-9}$
	> 1024	$1.38 * 10^{-5}$	$1.54 * 10^{-9}$
NEC SX-8 (On board)	< 1024	$2.15 * 10^{-6}$	$1.45 * 10^{-10}$
	> 1024	$2.90 * 10^{-6}$	$6.09 * 10^{-11}$
NEC SX-8 (Off board)	< 1024	$5.02 * 10^{-6}$	$2.64 * 10^{-10}$
	> 1024	$1.40 * 10^{-5}$	$1.86 * 10^{-10}$

4.5.2 Performance Measurements of the Lattice Boltzmann application

After measuring all the parameters required for the performance model we will now present the performance measurements for the LBM simulations. First we show detailed measurements of the total communication times and compare them to the model expressed in Eq. 4.17, and next we present the total execution time of the LBM simulations as a function of N and p .

Using the point-to-point communication overheads we compute the total communication times from the communication model and compare them with the measured ones in Figs. 4.9 and 4.10. We observe a reasonable agreement between measurements and predictions on either machine. The maximum difference is for the SSC case, which for the PC cluster it is about 28% while for the NEC SX-8 it is 40%. As for the AA and PM the agreement is better, especially for the NEC SX-8. Similar behavior was observed for the cases with smaller number of fluid nodes (data not shown).

Next, we measure the total execution times for all geometries with all three data sizes, and also compute the execution times from performance model (Eq. 4.19). The results shown in Figs. 4.11 and 4.12 are for the AA case. For the other two geometries we find comparable results (data not shown). The overestimation of the execution times for the biggest data size is about 5%, while for the smallest data size for $p = 128$ it is about 40% on the PC cluster and 10% on the NEC SX-8. As was mentioned in section 4.4, this is due

to the fact that all the computations of prediction model are based on the upper bound estimations.

We also plot in Figs. 4.13 and 4.14 the execution times in term of Lattice Updates per second (LUP/s) for all data sizes on both architectures. In Fig. 4.13 we see the super linear speed-up due to the "cache effect" on the PC cluster for the smallest geometries with $5 * 10^4$ fluid nodes. On NEC SX-8 (Fig. 4.14), as was expected, the best performance we get for the largest data sizes, of almost 75% peak performance for 128 processors.

In Figs. 4.15 and 4.16 we show the efficiencies as a function of p . Here the under-estimation of the prediction model is approximately 4%. Thus the agreement between measurements and performance model is quite high. Also in Fig. 4.16 we see a steep decline of efficiencies for the medium and small data sizes. This is due to the fact that on vector machines for such small problem sizes we see more the single processor effect (as we partition the problem, the single processor performance goes down, see Fig. 4.3) and therefore a large positive fractional processor speed overhead (Eq. 4.23). For small problem sizes on the PC cluster the situation is reversed. Here the fractional processor speed overhead is negative due to the caching effect, and we observe efficiencies much larger than one (see Fig. 4.15).

In Fig. 4.17 we have plotted fractional communication and fractional load imbalance overheads (Eqs. 4.21 and 4.22) as a function of p . From the plot it is clear that for large p the loss of the efficiency due to the load imbalance can be as large as 18%, while the loss due to communication is about 7% on the NEC SX-8 and 20% on the PC cluster. For a small number of processors the fractional load imbalance overhead is negligible as compared to fractional communication overhead. Thus from Fig. 4.17 we conclude that for a small number of processors the best way to improve the efficiency is the implementation of latency hiding [115] to decrease the communication overhead. For a large number of processors we propose to use a better partitioning algorithm to improve the load balance. By comparing the fractional load imbalance overheads as a function of number of fluid nodes we also observed that the bigger the data size the easier it is to achieve good load balance (data not shown). Therefore we can try to extrapolate our results to much larger data sizes. If we assume that the single processor speed overhead for large data sizes is zero, the point-to-point communication overhead decreases as $1/N^{2/3}$ and for a 10^8 data size geometry we decrease the load imbalance by 10% we can achieve almost 95% peak performance.

All these comparisons confirm that the performance model has a good accuracy and that it is eligible for predicting the performance for any size/type of geometry, if one knows the estimated fractional overheads.

4.6 Conclusions

In this chapter we have developed a performance prediction model and tested it by comparing the computed performance results with time complexity measurements of our parallel sparse lattice Boltzmann solver. In order to interpret the results, we defined three sources of parallel efficiency loss, i.e. fractional processor speed overhead, frac-

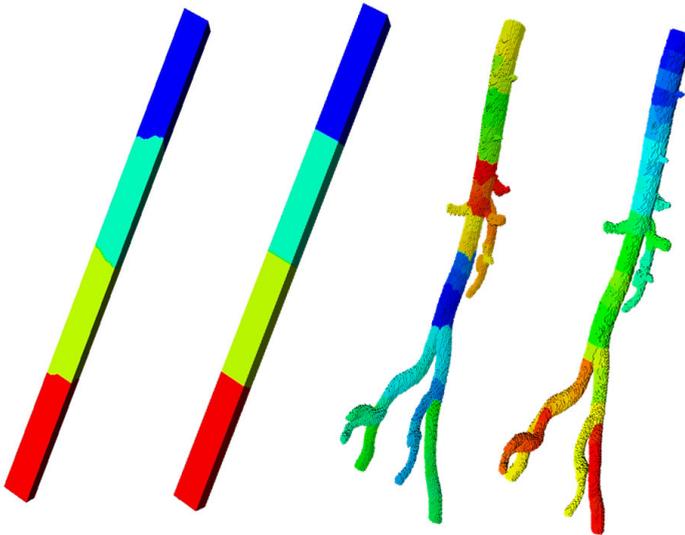


Figure 4.4: (a)- Straight square channel with four partitions created with the *RB* method, (b)- as (a) but now with the *K – way* method, (c)- Abdominal aorta, 128 partitions created with the *RB* method and (d)- as (c) but now with the *K – way* method.

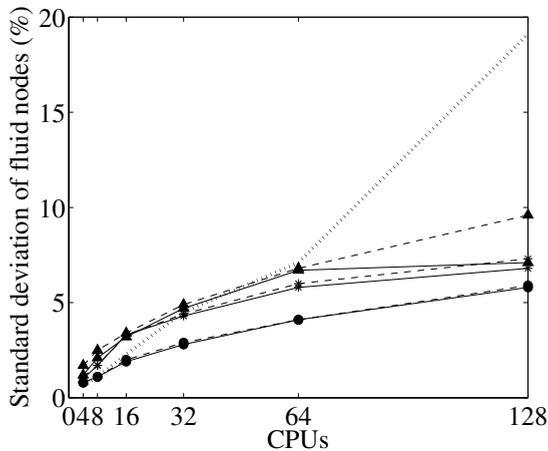


Figure 4.5: The standard deviation of fluid nodes per processor for the largest data size (i.e. with $N = 5 * 10^6$) for AA (bullets), PM (stars) and SSC (triangles) with both *RB* (solid lines) and *K-way* (dashed lines). 1D bisection (dotted line) is only for *SSC* geometry.

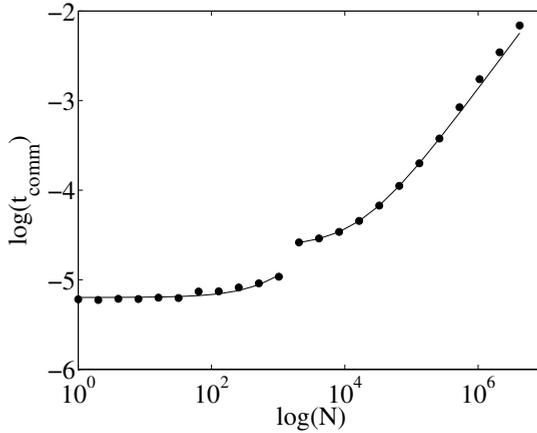


Figure 4.6: Point-to-point communication time (seconds) as a function of data size (bytes) on the PC cluster. The bullets are the measured data and the solid line is a fit to the data.

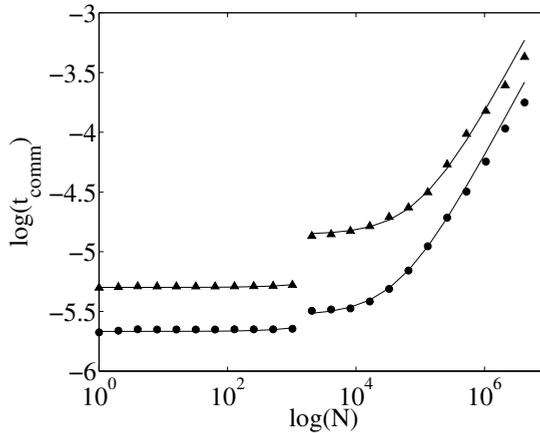


Figure 4.7: Point-to-point communication time (seconds) as a function of data size (bytes) on the NEC SX-8 vector machines. Triangles are the measured off-board point-to-point communication times, bullets are the measured on-board point-to-point communication times and the solid lines are the fit to the data.

tional communication overhead and fractional load imbalance overhead and measured them for all nine cases and two architecturally different supercomputers: a NEC SX-8 vector machine and a PC cluster.

For the three different geometries with three different data sizes we have performed a graph partitioning by using two different functions of the METIS graph partitioning library. We compared the partitioning results and conclude that with simple geometries is it better to use the *multilevelRB* method rather than *multilevelK-way*, while for complex ones the combination of both will give optimal solution [62].

The measurements show that for the largest data sizes we obtain almost 75% peak

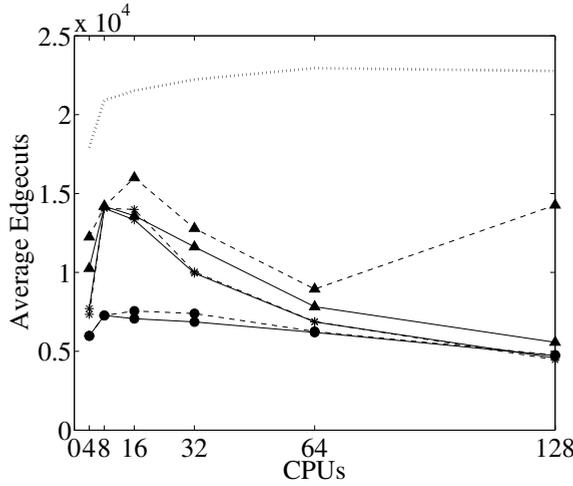


Figure 4.8: The average number of edge-cuts per processor for the largest data size(i.e. with $N = 5 * 10^6$) for AA(bullets), PM(stars)and SSC (triangles) with both RB (solid lines) and K-way (dash lines). 1D bisection (dotted line) is only for SSC geometry.

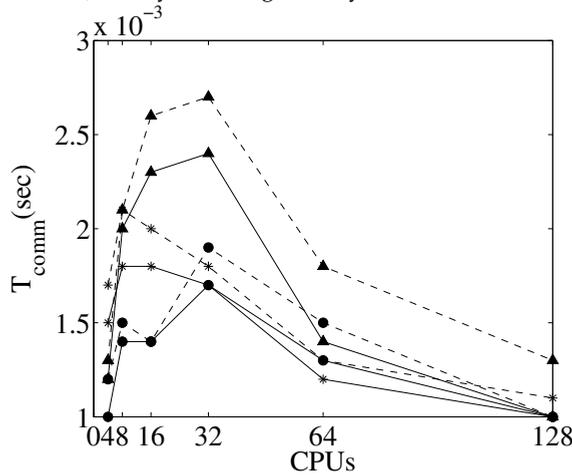


Figure 4.9: Total communication time as a function of p on the PC cluster for the largest geometries(i.e. with $N = 5 * 10^6$). The solid lines are the measurements and the dashed lines are model predictions from Eq. 4.17. The bullets are for AA, the stars for PM and the triangles for SSC.

performance on NEC SX-8 vector machine. Moreover with the help of the performance model we estimate that the efficiency on the NEC SX-8 can be improved if one chooses more robust partitioning algorithms as the efficiency loss for large p is mostly due to load imbalance. The comparison of these measurements with prediction model gives us 5% error, which shows that the developed model can be used for scalability prediction of parallel performance for any geometries.

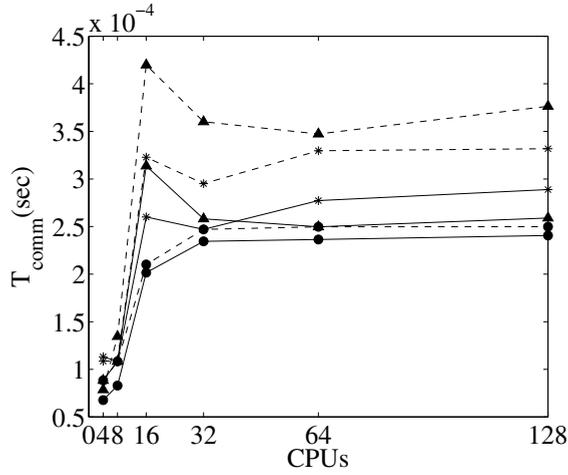


Figure 4.10: Total communication time as a function of p on the NEC SX-8 for the largest geometries (i.e. with $N = 5 * 10^6$). The solid lines are the measurements and the dashed lines are model predictions from Eq. 4.17. The bullets are for AA, the stars for PM and the triangles for SSC.

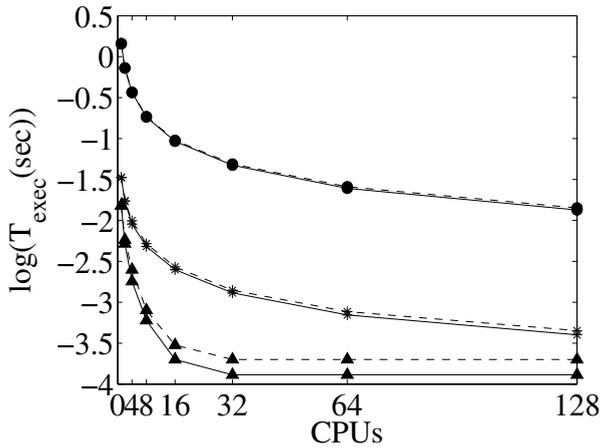


Figure 4.11: Execution time as a function of p on the PC cluster for AA for all three problem sizes. The solid lines are measurements and dashed lines are predictions from the performance model. The bullets are for $N = 5 * 10^6$, the stars for $N = 10^5$ and the triangles are for $N = 5 * 10^4$.

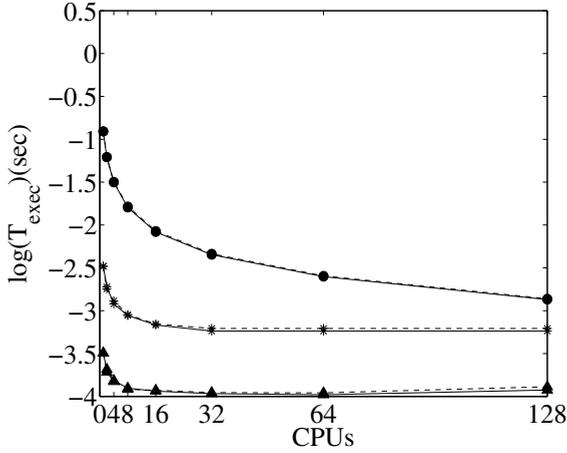


Figure 4.12: Execution time as a function of p on the NEC SX-8 vector machines for AA for all three problem sizes. The solid lines are measurements and dashed lines are predictions from the performance model. The bullets are for $N = 5 \times 10^6$, the stars for $N = 10^5$ and the triangles are for $N = 5 \times 10^4$.

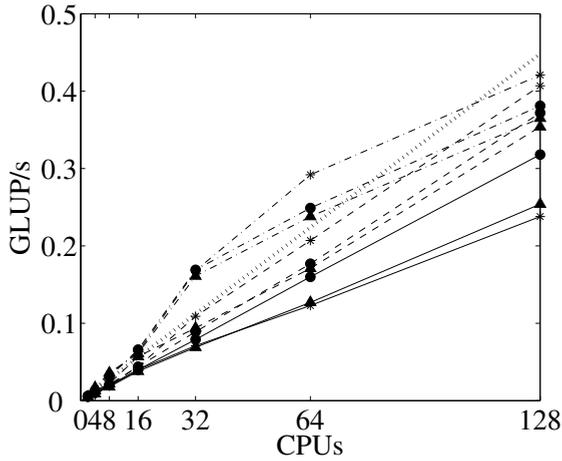


Figure 4.13: Lattice Updates per second as a function of p on the PC cluster for $N = 5 \times 10^6$ (solid lines), $N = 10^5$ (dashed lines) and $N = 5 \times 10^4$ (dotted lines). Bullets are AA, stars are PM and triangles are SSC. The dotted line indicates the ideal peak performance.

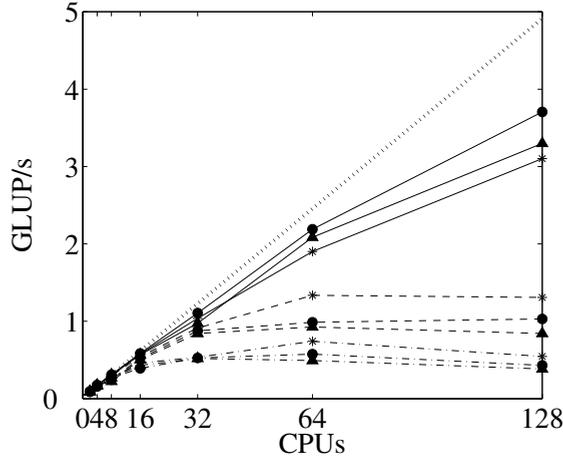


Figure 4.14: Lattice Updates per second as a function of p on the NEC SX8 machines for $N = 5 * 10^6$ (solid lines), $N = 10^5$ (dashed lines) and $N = 5 * 10^4$ (dotted lines). The bullets are AA, the stars are PM and the triangles are SSC. The dotted line indicates the ideal peak performance.

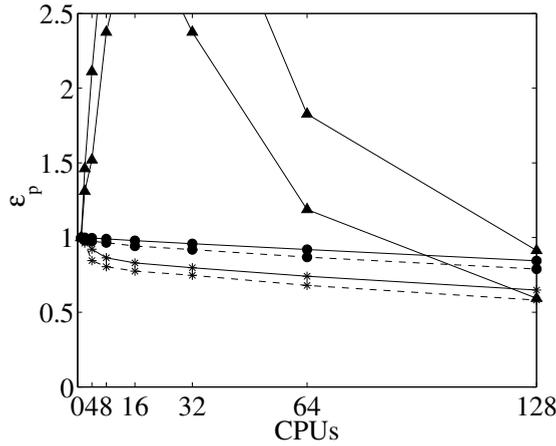


Figure 4.15: Efficiencies as a function of p on the PC cluster. The solid lines are measurements and dashed lines are predictions from the performance model. The bullets are for $N = 5 * 10^6$, the stars for $N = 10^5$ and the triangles are for $N = 5 * 10^4$.

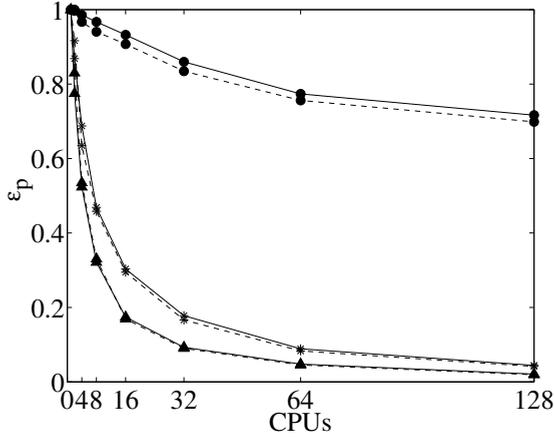


Figure 4.16: Efficiencies as a function of p on the NEC SX-8 vector machines. The solid lines are measurements and dashed lines are predictions from the performance model. The bullets are for $N = 5 * 10^6$, the stars for $N = 10^5$ and the triangles are for $N = 5 * 10^4$.

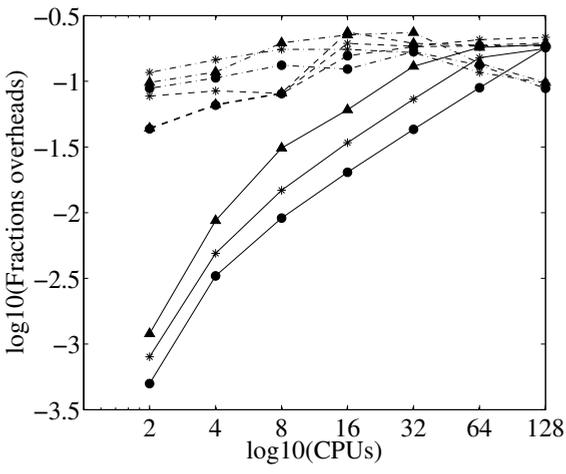


Figure 4.17: (a)- Fractional overheads for the largest data size (e.g. $N = 5 * 10^6$). Bullets are AA, stars are PM and triangles are SSC. The solid lines are the load imbalance overheads, dashed lines and dotted-dashed lines are the communication overheads on the NEC SX-8 and the PC cluster respectively.

*Towards Decision Support in Vascular Surgery through Computational Hemodynamics**



5.1 Introduction

Nowadays, arterial diseases are a major cause of death [4,101,116]. Their consequences are vascular system deformations and associated disorders of normal blood circulation in the human vascular system. In order to understand the vascular mechanics and find adequate solutions to these problems the development of a well-established problem solving environment is a major step. "A problem solving environment (PSE) is a computer system that provides all the computational facilities necessary to solve a target class of problems" [39,50].

In previous chapter 2 we introduced HemoSolve - a PSE for image-based computational hemodynamics. The objective of HemoSolve is to provide an environment for simulation of blood flow in patient specific arteries. Here further development of HemoSolve is presented. The new techniques integrated into its major components make it a helpful and easy-applicable e-learning tool which also could be used as a decision support system in pre-operational planing [65,98,99].

The human vascular system is a very complex structure and the treatments like bypass and/or stent placement on its deformed parts is not always a straight forward procedure for surgeons. An enhanced decision support system like HemoSolve can serve as a pre-operational planing tool for surgeons and a useful experimental system for medical students for training purposes [93,104].

As its ancestor, the new version of HemoSolve is also merged with Grid technology which allows an access to different and distant computational and instrumental resources [104]. This is one of the desirable features of PSEs in general [50].

We first describe the construction of HemoSolve and describe the new enhancements of its major components. Next we provide an example of its applicability on a case study of a carotid artery with a severe stenosis. We mimic a surgical procedure by adding a

*This chapter is based on: L. Axner, H. A. Marquering, A. G. Hoekstra, R. J. van der Geest, J. H.C. Reiber and P. M.A. Sloot. Decision Support through Computational Hemodynamics in Vascular Surgery (submitted for publication in *Physics in Medicine and Biology*)

bypass to the stenosed artery, and we simulate a time-harmonic blood flow before and after the bypass placement and compare the obtained velocity profiles.

5.2 HemoSolve and the new enhancements

HemoSolve is a 3D image-based PSE that allows to simulate time-harmonic flow in the human vascular system. It consists of the following components:

1. Medical image segmentation to determine the morphology of the arteries of interest;
2. 3D editing and mesh generation, to prepare for the flow simulation;
3. Flow simulation, computing of blood flow during systole;
4. Analysis of the flow, pressure, and stress fields.

As pointed out in ref. [93] there is a high demand for well-defined and user-friendly techniques that can turn a set of medical images into a computational fluid dynamics (CFD) input file in a matter of minutes. This is the goal we pursue in our further development of HemoSolve. The main changes, as compared to the previous version of HemoSolve have been conducted on its 3D editing and mesh generation and flow simulation components (See Fig. 5.1).

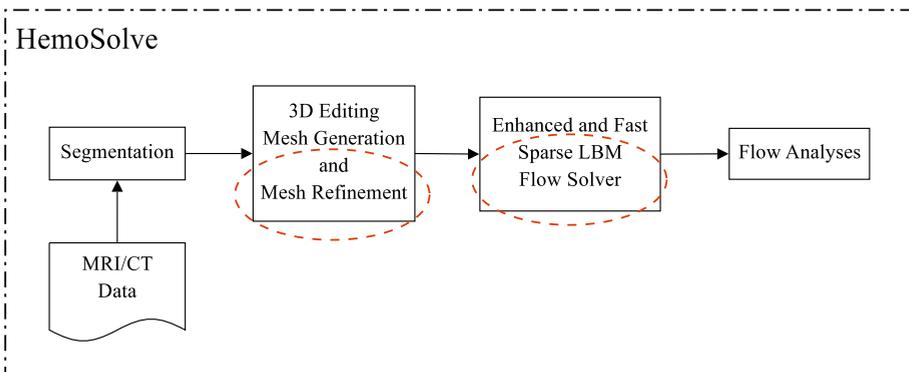


Figure 5.1: Functional design of HemoSolve.

5.2.1 Medical Data Segmentation

[†]The segmentation algorithm is similar to the one described in the section 2.2.1, except the the middle stage which is modified according to Marquering et al. [71]and divided in two parts: Longitudinal contour detection and Transversal contour detection. That is a

[†]Implemented at Department of Radiology, Leiden University Medical Center

stretched Curved Multi Planar Reformatted (CMPR) image is constructed which consists of a stack slices orthogonal to the detected centerline. On four longitudinal views of the CMPR image, the lumen contours along the running length of the vessel are automatically detected using a MinCost algorithm, which determines the minimal cost path on a gradient image. The intersection points of the longitudinal contours with the transversal slices are subsequently used as attraction points in the third step which consists of the lumen contour detection on the transversal slices of the CMPR image.

Longitudinal contour detection:

The 3D volume is resampled to produce a CMPR image in which the centerline is stretched to a straight line. By removing the curvature, the CMPR representation provides a good overview of the vessel in a single image by removing the its curvature. To make the contour detection insensitive to image noise we use information of the complete 3D shape of the vessel. We produce four longitudinal cut planes of this CMPR volume that are perpendicular to the CMPR transversal planes and pass through the center of the CMPR image, and thus through the centerline of the vessel. On these four views the longitudinal contour detection is carried out. The vessel borders are detected using the model-guided Minimum Cost Approach [91]. The contour is defined as the minimal cost path that travels from top to bottom of a cost function of the intersection image that contains a combination of spatial first- and second- derivatives of the longitudinal cut planes. The two vessel boundaries in a longitudinal plane are detected simultaneously such that a well-defined interface at one side can support the vessel border on the other side. The longitudinal contour detection of these four slices results in eight longitudinal contours.

To exclude calcified plaques from the lumen, an intensity based detection of calcified plaques is applied to exclude calcified plaques from the lumen. Dependent on the distance between the calcified plaques and the lumen, the gradient-based penalty function is adjusted.

Transversal contour detection:

The longitudinal contours are used to steer the contour detection on the transversal slices of the CMPR slices: The longitudinal contours produce a series of intersection points for each transversal slice. These points specify the region of interest for the transversal minimal cost contour detection and they function as attraction points with an adjustable strength to attract the contour detection to areas close to these points. The contours in single transversal images can also be manually corrected by adjusting a part of the contour or by moving an attraction point.

5.2.2 3D Editor and Mesh Generator

The second component of HemoSolve is a user-friendly 3D editing tool where a 3D stereoscopic image can easily be maintained. After loading of artery, the parts of interest can be selected and cropped from the original geometry with the help of a clipping

instrument. Then the remaining arterial geometry can be enhanced with structures like bypasses and/or stents, and inlet and outlet layers can be added on its end-points. Here the geometrical measurements (e.g. width, length, placement positions) of these structures are easy adjustable. Thus the 3D editing tool has all the necessary functionality to mimic a real surgical procedure.

The last stage is the 3D mesh generation from the prepared arterial geometry including bypasses and stents. The resolution of the mesh is also easy adjustable. The resulting mesh then serves as an input for the flow simulation.

In the previous version of HemoSolve, due to the limitations of the flow solver, only structured meshes were possible to create and use. Thus, during the generation of a structured mesh out of a very complex geometry, like the human abdominal aorta with its side branches, the major part of the time, work and memory were consumed on the filling-in of the complete structures hexagon. While in the current version due to the enhancement of the flow solver, HemoSolve is capable to treat also unstructured meshes. Thus the 3D editing tool is now capable to create unstructured meshes by saving 4/5 of the computational time and 80% of memory consumption.

5.2.3 Hemodynamic Solver - Sparse Lattice Boltzmann Method

The computational hemodynamic solver used in HemoSolve is the sparse lattice Boltzmann method(LBM) based on the sparse matrix linear algebra [12, 85]. We have shown that LBM is capable to cover a wide range of Reynolds (Re) and Womersley (α) numbers [9, 13]. The model is described in detail in section 4.2

For the application described in the next section the sparse-LBM flow simulator uses bounce back on links as a wall boundary condition and pressure inlet and free-flow outlet boundary conditions on the inlet/outlet layers. The flow solver receives as an input the geometry mesh from 3D editing tool and after specification of patient-specific free input parameters like the Re and α the flow is simulated.

5.2.4 Flow Analyses

There are several methods to examine the simulated blood flow. One of them is visualization of its velocity, pressure and/or shear stress profiles. Different visualization techniques show different features of flow. In HemoSolve we use a technique based on simulated pathline visualization [92]. The flow profiles are visualized during one period at each time-step.

5.3 Carotid Artery with Severe Stenosis - with and without Bypass

As a case study to demonstrate the applicability of the enhanced HemoSolve we have chosen to use a carotid artery with a severe stenosis obtained from a patient[‡]. The carotid artery (artery that supplies head and neck with blood) for this patient specific data has a severe stenosis (narrowing) due to atherosclerosis. For clearness of the visualization of results we choose to use only its deformed branch in our simulations.

First the CT image of one branch of patient's carotid artery with stenosis is segmented by applying the segmentation algorithm (Fig. 5.2 (a)). Then the segmented part which includes the stenosis, is transferred into the 3D editing tool (Fig. 5.2 (b)) where the user adds a bypass (Fig. 5.2 (b) and (c)), crops the structurally interesting part (Fig. 5.2 (d)) and defines inlet and outlet layers (Fig. 5.2 (e)). The placement of bypass is easy to control, by moving the top, middle and bottom manipulation points to change its position and length. The placement of inlet/outlet layers is also easy to control, that is to change the plane of position with simply moving the normal vector in the middle of the layer or to change the size just by movement of corner points of the rectangle. In this example there is one inlet at the top and one outlet at the bottom layer. The last step is the unstructured mesh generation (Fig. 5.2(f)). The mesh is then used as input for the LBM flow simulator.

We assume that the blood flow characteristics in the stenosed artery with the bypass placement are recovered to the same qualities as the flow in a healthy carotid artery. The maximum Re of a healthy patient in the carotid artery is known to be 1100 [41], thus we apply the same Re in the simulation of blood flow in the geometry with bypass. The size of the generated mesh is 584x46x40 lattice points. Choosing the maximum $Ma = 0.1$, $\alpha = 7$ and the rest of the parameters according the constraint optimization scheme (see sec. 3.3) we run the simulation with pressure boundary condition on the inlet layer and free flow on the outlet layer (see sec. 5.2.3). Next, after convergence, we obtain the velocity profile at the inlet of this geometry and apply it as an inlet boundary condition for blood flow simulation in the carotid artery with severe stenosis (without bypass). The rest of the parameters are chosen the same way.

In Figs. 5.3 and 5.4 six frames during one period are captured for both cases.

Our measurements show that the Re above stenosis part (region A in Fig. 5.3) is extremely low and about 36 for the geometry with bypass placement at the same place (region B in Fig. 5.4) the maximum $Re = 1100$. Thus, the bypass placement over the stenosis successfully recovers the expected flow profile through complete carotid artery.

Next, in Figs. 5.5 and 5.6 we plot the velocity profiles at the above stenosis (across B and A regions respectively as indicated in Figs. 5.3 and 5.4) part as a function of time and diameter of geometry for both cases. We see that the peak velocity of the carotid artery with bypass is about 16 times higher than the one without bypass.

[‡]The data is kindly provided to us by Dr Aad van der Lugt/Erasmus MC Rotterdam and Henk A. Marquering, Laboratorium voor Klinische en Experimentele Beeldverwerking (LKEB)

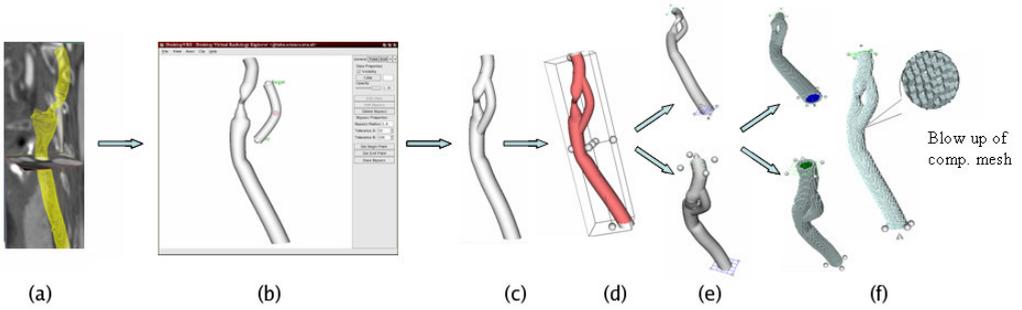


Figure 5.2: Carotid artery with severe stenosis. Main stages of HemoSolve: Segmentation of the raw medical data (a), load of the segmented data into 3D editing tool (b) and bypass placement (c), cropping of the segmented data (d), adding inlet/outlet layers (e) and generation of an unstructured mesh (f).

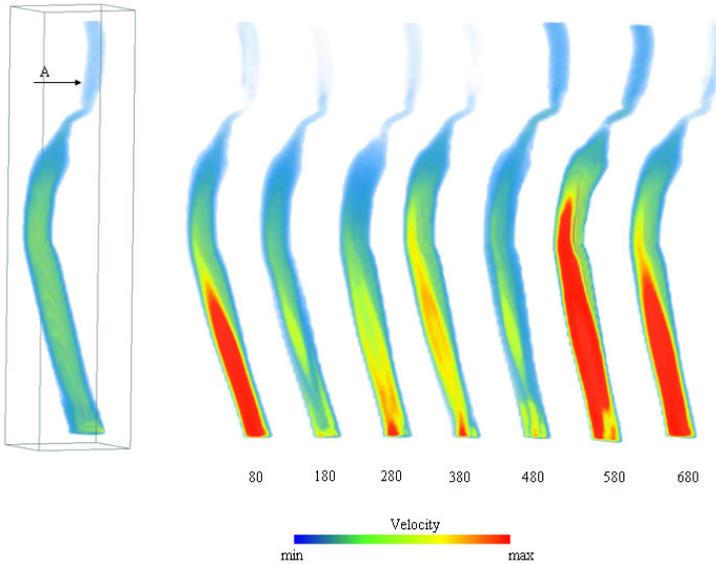


Figure 5.3: Velocity profile in lattice unites of carotid artery with severe stenosis every 100 time-steps during one period. Here A is the region where we look into velocity profiles in details in Figs. 5.5 and 5.6

5.4 Conclusion

In this chapter we have presented a new version of HemoSolve - a problem solving environment for image based computational hemodynamics. We have described in detail the new enhancements of its major components and explained their advantages. For the given example our measurements showed that we gained in the simulation speed about 9 times. To confirm its usability we have mimicked a real surgical procedure of

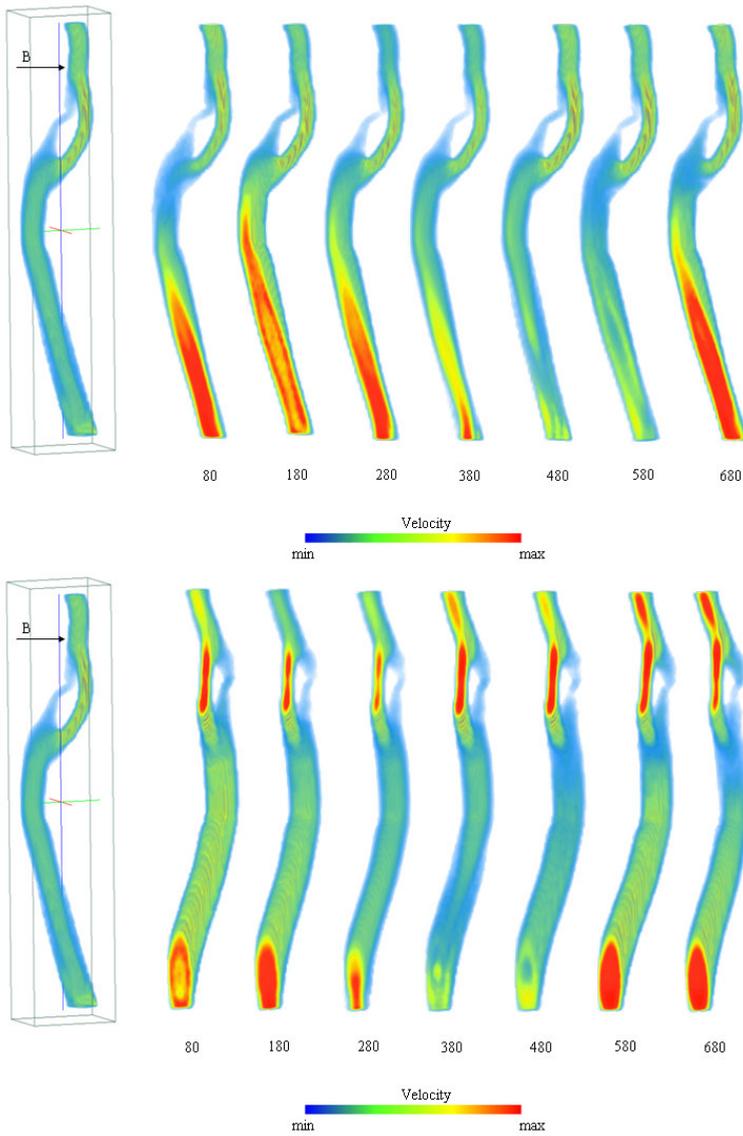


Figure 5.4: Velocity profile in lattice unites of carotid artery with severe stenosis after bypass placement every 100 time-steps during one period from bypass(top) and stenosis(bottom) sides. Here B is the region where we look into velocity profiles in details in Figs. 5.5 and 5.6

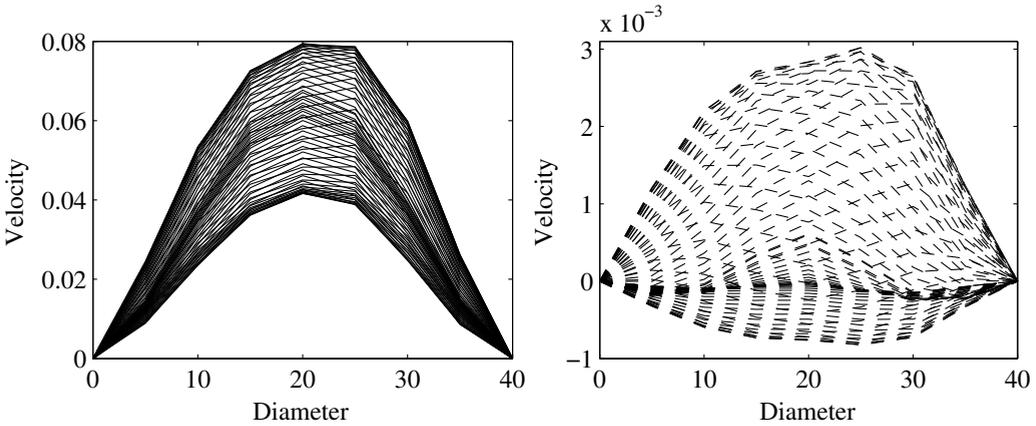


Figure 5.5: Velocity profiles as a function of diameter of carotid artery with (left) and without (right) bypass printed every 10th time-step.

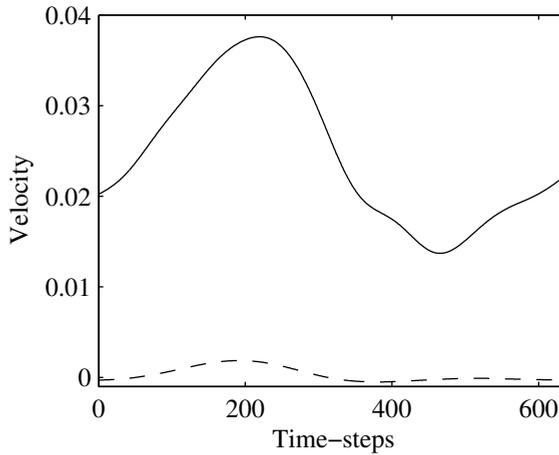


Figure 5.6: Velocity profiles as a function of time of carotid artery with (solid lines) and without (dashed lines) bypass during one period.

bypass placement on a patient specific carotid artery with severe stenosis. Next we have examined the obtained time-harmonic flow profiles. Thus from the raw medical data of patients vascular system, after corresponding simple steps biomedical engineers and novice surgeons can obtain a ready time-harmonic flow fields and analyze them with visualization tools.

The new enhancements of HemoSolve make it not only highly sophisticated but also fast and considerably less memory consumable PSE. Moreover, HemoSolve as a well-defined PSE satisfies its main requirements, e.g. it has a simple human-computer interaction graphical interface [126], it uses complete and accurate flow solver LBM for simulation of time-harmonic blood flow. It is integrated into parallel and distributed computing

environment [13] and also merged with Grid environment [104].

As conclusion, HemoSolve is a well-established decision support and user-friendly training system for novice surgeons and biomedical engineers for image-based hemodynamics research of time-harmonic blood flow in the human vascular system.

Conclusion



6.1 Discussion

Houstis and Rice [50] have mentioned that over the past 60 years there has been an enormous increase both in the speed of computer hardware and in algorithmic power. With the increase of the problem sizes, the demand for speeding up the simulation is also increasing. There is also a need for high level and powerful problem solving environments (PSE). The main idea behind a PSE is to combine powerful hardware and smart algorithms to extract fast solutions for a target class of problems. As pointed out "Future PSEs will use essentially all the resources of future computing technologies, e.g. net-centric facilities, dynamic and collaborative computations, knowledge discovery and learning, software reuse, human-machine interaction, etc."

Our first step in this direction was to create a PSE (called HemoSolve) for image-based computational hemodynamic problems with the aim to transform medical data through information to knowledge. HemoSolve is user-friendly complete system, that allows novice students to enlarge their practical skills by mimicking surgery. The raw medical data of a patient's vascular system can be obtained and after several steps, in a short period of time, final results can be analyzed and visualized within the PSE. As CFD solver in HemoSolve the lattice Boltzmann method is used.

Although HemoSolve is completely parallelized, optimizations were still needed. Therefore our next goal was to speed-up the complete environment. For this reason we have concentrated on the improvement of each component separately. It is worthwhile to mention that HemoSolve is completely integrated into a Grid environment, thus supporting seamless integration of data (CT scan in the hospital), simulation (high performance computing center) and visualization (student desktop) tools.

The first and major component to improve was the hemodynamic solver. Although LBM is a well-established method it is still hampered with slow convergence due to the method specific constraints. We have developed a constraint optimization scheme that allows to choose the simulation parameters in a way to get a minimum execution time. The novel idea behind this scheme is that the simulation error is one of the constraints, and can be defined a-priori. Good agreement between theoretical and experimental results proved the correctness of the scheme. In our simulations we have also managed to reach quite high Reynolds numbers for Womersley flow in a 3D straight tube. This is difficult to achieve for stability reasons due to the minimum relaxation parameter limit. Using this

scheme as an example for a real application we have simulated a time-harmonic blood flow in a human abdominal aorta.

Also, as LBM is a relatively new method, we have decided to do another validation of this method by comparing it with FEM. As an application we have used the time-harmonic blood flow in the superior mesenteric artery with peak Reynolds number of 3300. Although the Reynolds number is quite high, we have achieved a very good agreement between both solvers. When examining velocity profiles, we have observed all the flow characteristic similarities, such as vortex formation next to the bifurcation or the spiraling of the flow in the abdominal aorta.

Another improvement of time-harmonic LBM is the integration of the regularized scheme into the collision term of LBM. Significant improvements of accuracy and stability were obtained.

The next components of HemoSolve to improve were mesh generation and parallelization methods. The fact that LBM was applied on structured voxel meshes was considered as one of the main disadvantages. Within the algorithm not only fluid but also solid nodes had to be saved and iterated, which consumed a major part of the execution time and the memory. The new sparse-LBM scheme, where only fluid nodes are saved, appeared to be an enhanced scheme which increased the simulation speed enormously and allowed to treat data of over eight orders of magnitudes.

For the data distribution of parallel sparse-LBM we have used the METIS data partitioning library. In contrast to other partitioning methods, the functions of the METIS library are able to partition large data sizes, e.g. $5 * 10^6$ fluid nodes in only 2 – 3 minutes. Our experiments showed that the efficiency loss due to the load imbalance between partitions is about 18% for $5 * 10^6$ data size on 128 processors/partitions and less for smaller number of partitions. Thus compared to the influence of e.g. fractional communication overhead, which is approximately 7% for any number of processors.

In order to evaluate the loss of efficiency of parallel sparse-LBM due to the different fractional overheads, we have developed a performance prediction model. This model allows us to preliminary estimate the efficiency, provided load imbalance, single processor performance and point-to-point communication overheads.

All those improvements gave a significant speed-up of HemoSolve. For example, the time consumed for time-harmonic blood flow simulation in the carotid artery with bypass placement implemented with the latest version of HemoSolve was about nine times shorter than the time consumed by the previous version. Typically our experiments with enhanced HemoSolve showed that the same experiment which used to take three hours of execution time now takes only about 20 minutes. Thus now we are able to analyze the outputs much faster, even for very large problem sizes. The demonstrated final experiment of bypass placement on the carotid artery with severe stenosis using HemoSolve is a clear example of mimicking a real surgery.

Thus, we have achieved the main goals posed at the beginning of this thesis: we have created a complete and enhanced computer-based simulation environment for image-based computational hemodynamics problems using the LBM, integrated it into high performance computing environment, and applied it for visualization and understanding of biomechanical processes in complex vascular systems.

We have presented LBM as a robust and accurate method for hemodynamic problems. In all our experiments we have shown a good agreement between LBM and analytical or other numerical solutions. We have managed to simulate time-harmonic blood flows in a large human arterial tree with very high Reynolds numbers, that are expected to cause instabilities due to the low viscosity limitations of LBM. All the simulations were performed for incompressible, Newtonian flows. Therefore, the questions whether LBM is eligible to use for non-Newtonian flows and will it be robust enough to apply for simulations of fluid-structure interactions to reconstruct the real flow physiology in human vascular systems need still to be answered.

We have also developed an enhanced PSE that has all the necessary tools to mimic a surgical operation, such as bypass placement. Some experiments have been conducted to explore its usefulness for students as a learning tool. But whether PSE like HemoSolve would be useful for the training of surgeons or whether it can serve as a decision support system to use just before the operation of real patient is still an open issue.

And last but not the least, the main question "What is the role of computational hemodynamics for preoperational planing..." remains unanswered.

6.2 Future work

There are several possibilities to continue this research both from a numerical point of view as well as from the application view point.

- Data distribution and load imbalance - Although in this research the obtained results of using the METIS partitioning library are quite satisfactory, we have still estimated that there is maximum of 18% impact of load imbalance on the efficiency. Thus there is a need of more sophisticated partitioning algorithms that will be comparably fast but with better load balancing.
- Grid refinement - Our results of simulations of blood flow in the complete abdominal aortic tree have shown that in order to be able to analyze the flow profiles in the narrow side-branches one needs to have a very fine mesh. The refinement of the complete geometry will increase the execution time of simulations enormously. Therefore the incorporation of partial refinement algorithm into HemoSolve is the next logical step.
- Wall elasticity - Until now, for all the medical geometries in all the examples throughout this thesis the walls of arteries considered to be rigid. This is one of the major gaps that separate us from reality. Implementation of the elasticity of the walls is a major step. One of the possible algorithms to do this is the spring LBM [20].

The implementation of the above mentioned points may bring the LBM a step forward towards the simulation of non-Newtonian flows. It may also help to realize whether we need to use more advance numerical methods or will LBM be sufficient to use in implementing the fluid-structure interactions for real flow simulations. Consideration of blood cells transportation and their possible deformation in the flow is also an open

issue. Another interesting question is: can we be satisfied with the results obtained from simulations of incompressible flows or do we need an immediate switch to compressible ones?

All those are the questions and discussions that still need substantial answers.

Summary



The aim of the work in this thesis was to create a complete and sophisticated computer-based simulation environment that may also serve as a decision support system for vascular reconstruction. We have defined a PSE for image-based computational hemodynamics, where as CFD solver we used LBM. We investigated optimization methods for the LBM flow solver. We also merged the PSE into a high performance computing environment. The intension was to improve its performance and scalability. We proposed optimization models and analyzed the results on a wide range of experiments.

Chapter 1 is an introduction to the work. There we first have showed the relevance of our research by introducing the main concepts. We have briefly described the actual formation, cause and influence of atherosclerosis and aneurysms in humans. Next we gave a short description of the high performance computing concept and discussed its influence on optimized solutions of complex systems. Finally we have described the LBM as a fluid dynamics method, showing its ability to be parallelized and provide references to examples of earlier work where it has been applied on unstructured/nested grids of complex geometries.

In Chapter 2 we have described a problem solving environment; HemoSolve for image based computational hemodynamics. We discussed its main components in detail showing it as a robust image-based tools. Next we concentrated on the FEM and LBM hemodynamic solvers used in HemoSolve and compared them with each other. We have proved the eligibility of LBM as an enhanced CFD solver for 3D time-harmonic fluid flows. This was confirmed by the experimental results of 3D time-harmonic blood flow simulations in the abdominal aorta, superior mesenteric artery and an aortic part with large aneurysm. We have concluded by pointing out the advantages of HemoSolve as an easy applicable, user-friendly and robust image-based problem solving environment.

After establishment of LBM as a well-defined hemodynamic solver we went one step further and examined the possibilities of its optimization in Chapter 3. LBM has several constrains to fulfill the demand of stability and accuracy. Those constraints are causing a slow convergence, which in its turn decreases the performance of the complete PSE. Thus we have defined a constraint optimization scheme which allows us to choose the simulation parameters in a way, that will speed-up the convergence of LBM. In this chapter

we have described the theory of the scheme in detail and compared it with the experimental results. The good agreement between two of them allowed us to also examine the stability of the system for very high Reynolds numbers. Also, using the constraint optimization scheme we have simulated the time-harmonic blood flow in a human abdominal aorta. Next, we have tried to improve the accuracy of LBM by using a new regularized scheme, e.g. we introduced a small correction to the collision term of LBM. We have tested the scheme for time-harmonic flows by comparing it with usual LBM and by checking the numerical stability for high Reynold numbers. We conclude that RL-LBGK is more accurate and stable than LBGK.

The fulfillment of optimizations of the LBM method itself led us to the optimization of its parallelization schemes. In Chapter 4 we have described a sparse-LBM method which allows us to work with only the fluid part of the complete data domain and ignore the solid fractions. Thus saving the memory consumption and also execution time. Here we used the graph partitioning library METIS for data decomposition. The functions we used from METIS are based on the combination of modified Kernighan-Lin and Fiduccia-Mattheyses algorithms. Next, we have defined a performance prediction model which allows us to determine the execution time of the system on the parallel machines a-priori. This model takes into account such fractional overheads as the single processor performance, load imbalance and communication overheads. The comparison of our experimental measurements with theory sowed us very satisfactory results with only 5% prediction error.

Finally, in Chapter 5 we have combined all our improvements of LBM and the parallelization scheme into our PSE, making it a complete decision support system. We have shown its enhancements and speed-up by simulating time-harmonic blood flow in human carotid artery with a severe stenosis. Next we have mimicked a real surgical operation by bypass placement over the stenosis and again simulating the blood flow. Our results were quite satisfactory as to comparison with blood flow characteristics in the carotid artery of healthy patient.

Chapter 6 concluded our work by discussions of the thesis and discussing the future developments.

Nederlandse samenvatting



De belangrijkste doelstelling van het in dit proefschrift beschreven onderzoek was het ontwikkelen van een compleet, geavanceerd, computergebaseerd trainings- en beslissingsondersteuningssysteem dat een verbinding legt tussen gegevens en kennis door middel van informatie. Daartoe hebben we een Problem Solving Environment (PSE) gedefinieerd voor beeldgebaseerde hemodynamica, gegeven een op een Lattice-Boltzmann Methode (LBM) gebaseerde techniek voor het oplossen van problemen in de vloeistofdynamica (Computational Fluid Dynamics, of CFD). Ter verbetering van de efficiëntie en de schaalbaarheid hebben we ons bovendien intensief gericht op parallellisatie en optimalisatie van de LBM. We hebben een aantal verschillende optimalisatiemodellen voorgesteld, en geanalyseerd op basis van een ruim scala aan experimenten. De optimalisaties hebben geleid tot een snelheidsverbetering die onze beeldgebaseerde PSE in staat stelde chirurgische operaties op realistische wijze te simuleren.

Hoofdstuk 1 bevat een algemene introductie tot ons werk. Allereerst tonen we het belang aan van ons onderzoek door middel van een overzicht van de belangrijkste concepten die aan de basis liggen van het proefschrift. We geven een korte beschrijving van het ontstaan, de oorzaken, en de gevolgen van slagaderverkalking (atherosclerose) en vaatverwijding (aneurysma). We definiëren het probleem als een complex systeem, en stellen voor een oplossing te vinden door middel van het modelleren en simuleren van bloedstromen in het menselijke vaatstelsel. Verder geven we een korte beschrijving van het gebied van parallel rekenen en bespreken haar invloed op geoptimaliseerde oplossingen voor complexe systemen. Tot slot beschrijven we LBM als een model voor vloeistofdynamica, met inbegrip van haar geschiktheid voor parallellisatie, en geven voorbeelden van eerder onderzoek dat LBM heeft toegepast op ongestructureerde en geneste grids van complexe geometrieën.

Hoofdstuk 2 geeft een gedetailleerde beschrijving van de belangrijkste componenten van de door ons ontwikkelde PSE, genaamd HemoSolve. Bovendien vergelijken we twee verschillende, in HemoSolve gebruikte oplossingsmethoden voor problemen in de vloeistofdynamica (LBM en FEM, of Finite Element Method), waarbij we de geschiktheid van LBM aantonen als een oplossingstechniek voor 3-dimensionale tijdharmonische vloeistofstromen. We bevestigen onze theoretische discussie door middel van numerie-

ke experimenten met 3D tijdharmonische simulaties van bloedstromen in verschillende delen van het vaatstelsel. Op basis van deze resultaten concluderen we dat HemoSolve een eenvoudig te gebruiken en robuust beeldgebaseerd PSE systeem is.

Nadat we hebben aangetoond dat LBM een geschikte oplossingsmethode voor hemodynamische problemen is, hebben we in Hoofdstuk 3 de mogelijkheden ter optimalisatie van LBM onderzocht. LBM heeft een aantal randvoorwaarden die de eis van stabiliteit moeten waarborgen. Deze randvoorwaarden zorgen voor een langzame convergentie, en derhalve voor een verminderde efficiëntie van onze gehele PSE. We hebben daarom een methode ter optimalisatie van de randvoorwaarden ontwikkeld die ons in staat stelt de simulatie parameters zodanig te kiezen dat de convergentie van de LBM wordt versneld. In dit hoofdstuk hebben we de theorie van deze methode in detail beschreven, en vergeleken met experimentele resultaten. De hoge mate van overeenkomst tussen de theorie en de numerieke resultaten stelde ons in staat de stabiliteit van ons systeem te testen voor zeer hoge Reynoldsgetallen. Op basis van onze optimalisatiemethode van de randvoorwaarden hebben we de tijdharmonische bloedstroom in een menselijke buikslagader gesimuleerd. Daarnaast hebben we geprobeerd de nauwkeurigheid van LBM te vergroten door een kleine correctie van de LBM collisionparameter. We hebben deze aanpassing getest voor tijdharmonische stromen, en - voor hoge Reynoldsgetallen - vergeleken met de gebruikelijke LBM parameter instellingen. We hebben daarbij vastgesteld dat onze aangepaste versie (RL-LBGK) nauwkeuriger en stabielere is dan de originele versie (LBGK).

Om significante snelheidsverbeteringen van de LBM methode zelf te bereiken hebben we ons vooral gericht op het optimaliseren van de toegepaste parallele rekenmethoden. In Hoofdstuk 4 beschrijven we een nieuwe sparse-LBM methode die ons in staat stelt het solide deel van het volledige data domein te negeren, en ons te beperken tot het fluïde deel. Deze aanpak, gecombineerd met het gebruik van een nieuwe graafpartitioneringsbibliotheek voor data distributie (METIS), bespaart op het geheugengebruik, en resulteert in een snellere executietijd. De METIS functionaliteit die we hebben toegepast is gebaseerd op aangepaste Kernighan-Lin en Fiduccia-Mattheyses algoritmen. Daarnaast hebben we een schattingsmodel ontwikkeld dat ons in staat stelt de executietijd van het systeem vooraf te bepalen. Dit model bevat kostenparameters voor de rekestijd op een enkele processor, verschillen in de werkdruk tussen processoren onderling, en interprocescommunicatie. Experimentele resultaten tonen aan dat ons schattingsmodel resulteert in een maximale fout van slechts 5%.

Tot slot worden in Hoofdstuk 5 alle ontwikkelde verbeteringen van het LBM algoritme geïntegreerd in ons PSE systeem. We tonen de verbeteringen in nauwkeurigheid en snelheid aan door middel van een simulatie van een tijdharmonische bloedstroom in een ernstig vernauwde menselijke halsslagader. Verder hebben we een chirurgische bypass operatie nagebootst, en achtereenvolgens de bloedstroom in de halsslagader opnieuw gesimuleerd. Het eindresultaat is zeer positief in vergelijking met de bloedstromen in de halsslagader van een gezonde patiënt.

Hoofdstuk 6 besluit het proefschrift met een algemene discussie en een beschrijving van mogelijke richtingen voor toekomstige onderzoeken in dit vakgebied.

Summary in Armenian



Ամփոփում

Այս աշխատության նպատակն է ստեղծել մի ամբողջական, զարգացած և առաջադեմ միջավայր, որը կկարողանա ծառայել որպես որոշումների կայացման ապավենիչ համակարգ՝ հիվանդի դեֆորմացված զարկերակային համակարգ վերականգնողական գործունեության համար: Այսինքն, մենք սահմանել ենք **խնդիրների լուծման միջավայր** (ԽԼՄ), որը հիմնված է պատկերահիմունքային հաշվողական հեմոդինամիկայի վրա և որտեղ, որպես հաշվողական հեղուկ դինամիկայի (ՀՀԴ-ի) մեթոդ, կիրառել ենք **Ցանցային Բոյցմանի մոդելը** (ՑԲՄ):

Հաջորդ քայլում հետաքննել ենք ՑԲՄ-ի օպտիմիզացման տարբեր մեթոդներ: Դրանից հետո միայն ամբողջ սահմանված ԽԼՄ-ն ներմուծել ենք *բարձր արդյունավետության սուպերկոմպյուտերային համակարգ*՝ դրանով իսկ բարձրացնելով ԽԼՄ-ի արդյունավետությունն ու կիրառելիությունը:

Այսպիսով, նախ առաջարկել ենք օպտիմալ մոդելով զինված համակարգ, ապա հետազոտել ենք դրա արդյունավետությունը: Վերջում, բազմաթիվ փորձարկումների միջոցով, քննարկել ենք ստացված արդյունքները:

Գլուխ I-ն այս աշխատության նախաբանն է, որտեղ մենք մատնանշել ենք այս աշխատության կարևորությունն ու ծանոթացրել հիմնական տերմիններին: Նաև հակիրճ ներկայացրել ենք մարդու արյունատար անոթային համակարգում աթերոսկլերոզի և անևոթիզմի առաջացման ու զարգացման պատճառներն ապա, բացատրել ենք հիվանդի վրա դրանց ազդեցության հետևանքները:

Հաջորդ ենթաբաժնում ներկայացրել ենք, թե ինչ ասել է *բարձր արդյունավետության սուպերկոմպյուտերային համակարգ* և քննարկել ենք այդ

համակարգի դրական ազդեցությունը բարդ խնդիրների լուծման օպտիմիզացման վրա:

Այս գլուխն եզրափակել ենք՝ նկարագրելով ՅՔՄ-ն որպես հեղուկ դինամիկայի մեթոդ ապա, ցուցադրել ենք դրա զուբախեռ-ծրագրավորման հնարավորություններն և վերջում մատնանշել ենք նախկին աշխատանքներն, որտեղ ՅՔՄ-ն կիրառվել է բարդ խնդիրների, այսինքն՝ անկանոն և հյուսվածքային գրաֆների վրա:

Գլուխ II-ում նկարագրել ենք ԽՄՄ-ն, որը հիմնված է պատկերահիմունքային հաշվողական հեմոդինամիկայի վրա և այն անվանել ենք ՀեմոՍոլվ: Ապա քննարկել ենք ՀեմոՍոլվի հիմնական բաղադրամասերն, ապացուցելով դրանց մատչելիությունն որպես պատկերահիմունքային հետազոտիչ գործիք: Հետո կենտրոնացել ենք համակարգում օգտագործվող ՅՔՄ-ի վրա և այն համեմատել ենք մեկ այլ հաշվողական հեղուկ դինամիկայի՝ ***Որոշիչ Էլեմենտային մեթոդի*** հետ:

Այսպիսով, ապացուցել ենք ՅՔՄ-ի լիազորությունն որպես ՀՀԴ մոդել՝ եռաչափ համակարգում ժամանակի ընթացքում պուլսացվող արյան հոսքի մոդելավորման համար: Այս պնդումը հաստատել ենք որովայնային աորտայում, մեզենտերիկ և կարոտիդ զարկերակներում արյան հոսքի մոդելավորման հաշվողական փորձարկումներով:

Այս գլուխն եզրափակել ենք ցուցադրելով ՀեմոՍոլվ-ի առավելություններն՝ որպես ճկուն, հարմար կիրառելի և մատչելի պատկերահիմունքային ԽՄՄ:

ՅՔՄ-ն որպես լիազորված ՀՀԴ մեթոդ հաստատելուց հետո մեր հաջորդ քայլը դրա կատարելագործման հնարավորությունների հետազոտումն է, որը մենք նկարագրել ենք գլուխ III-ում:

ՅՔՄ-ն ունի մի քանի սահմանափակումներ, որոնց բավարարումով է պայմանավորված այս մեթոդի կայունությունն ու ճշտությունը: Մյուս կողմից այս սահմանափակումները հանդիսանում են ամբողջ ԽՄՄ-ի դանդաղեցման պատճառ: Այսպիսով, մենք որոշեցինք ստեղծել մի *սահմանափակումների օպտիմիզացման սխեմա*, որը կթույլատրի ընտրել ՅՔՄ-ին բավարարող պարամետրերն այնպես, որ արագացվի ամբողջ համակարգի գործունեությունը, միաժամանակ պահպանելով ՅՔՄ-ի կայունությունն ու ճշտությունը:

Այս գլխում նախ մանրամասնորեն նկարագրել ենք մեր ստեղծած օպտիմիզացման սխեման, որից հետո դրա փորձարկումներից ստացված արդյունքները համեմատել ենք թեորիայի հետ: Ստանալով արդյունքների և թեորիայի բարձր համապատասխանություն, հետազոտել ենք նաև մեթոդի կայունությունը բարձր կարգի Ռեյնոլդի թվերի դեպքում: Ապա, կիրառելով սահմանափակումների օպտիմիզացման սխեման, մոդելավորել և հաշվարկել ենք մարդու որովայնային աորտայում պուլսացվող արյան հոսքը:

Հաջորդ քայլում փորձել ենք բարձրացնել ՅՖՄ-ի ճշտությունը՝ ներմուծելով մեկ այլ նորագույն կարգավորիչ սխեմա, այսինքն կատարելով փոքր ուղղում ՅՖՄ-ի մասնիկների բախումն արտահայտող հավասարման մեջ: Այսպիսով, ստեղծել ենք **կարգավորիչ ՅՖՄ** (ԿՅՖՄ):

Այս սխեման փորձարկել ենք պուլսացվող հոսքի համար՝ համեմատելով սովորական ՅՖՄ-ի հետ և կրկին ստուգել ենք դրա կայունությունը բարձր կարգի Ռեյնոլդի թվերի դեպքում: Ստացված արդյունքների հիման վրա եզրակացրել ենք, որ ԿՅՖՄ-ն ավելի կայուն ու ճշտգրիտ է, քան ՅՖՄ-ն:

ՅՖՄ-ի այս օպտիմիզացման սխեմաների բավարարումն ու հաջողությունը մեզ ուղղորդել են դեպի զուգահեռ ծրագրավորման սխեմայի օպտիմիզացմանը, որը մենք նկարագրել ենք գլուխ IV-ում:

Այստեղ նախ բացատրել ենք **մասնակի-ՅՖՄ-ն**, որը թույլատրում է աշխատել ամբողջ տվյալների համակարգի միայն հեղուկ մասնիկների հետ և անտեսել պինդ մասնիկներն՝ այսպիսով խուսափելով կոմպյուտերային հիշողության և ծրագրի կատարման ժամանակի ավելորդ վատնումից: Դրանից հետո նկարագրել ենք ՄԷՏԻՄ գրաֆների մասնատման գրադարանն, որը կիրառել ենք գրաֆների միջոցով արտահայտված տվյալների մասնատման համար: Մեր ընտրած ՄԷՏԻՄ-ի ֆունկցիաները հիմնված են մոդիֆիկացված Կերնիզան-Լին և Ֆիդուչիա-Մաթեյասի ալգորիթմների համախմբության վրա:

Հաջորդ քայլում մենք սահմանել ենք **աշխատանքի ցուցանիշների կանխատեսման** մի մոդել, որը հնարավորություն է տալիս նախապես կանխատեսել ծրագրի կատարման ժամանակը զուգահեռ-կոմպյուտերային համակարգերի վրա: Այս մոդելը հաշվի է առնում այնպիսի հանգամանքներ, ինչպիսիք են մեկ պրոցեսորի աշխատանքի ժամանակատարության ցուցանիշը, տվյալների բաշխման անհամասեռությունն ու միջպրոցեսորային կոմունիկացիաների ժամանակատարությունը:

Գլուխը սահմանափակել ենք մեր փորձարկումների և թերիայի համեմատությամբ՝ ցուցադրելով բավականին բավարար արդյունքներ և ստանալով կանխատեսման միայն 5% սխալ:

Վերջապես, գլուխ V-ում, ՀեմոՍոլվ-ի մեջ համախմբել ենք ՅՖՄ-ի բոլոր բարելավումներն ու զուգահեռացման սխեմաները՝ դարձնելով ՀեմոՍոլվ-ը մի ամբողջական որոշումների կայացման ապավենիչ համակարգ: Համոզվելու համար ՀեմոՍոլվ-ի կատարելագործության մեջ՝ ներկայացրել ենք մի վիրտուալ օպերացիայի կատարման փորձարկում. նախ մոդելավորել ենք հիվանդի նեղացված (ստենոզիտով) կարոտիդ զարկերակում պուլսացվող արյան հոսքն ապա, վիրտուալ շունտավորման օպերացիա կատարելուց հետո, կրկին մոդելավորել ենք՝ այժմ նաև շունտի միջով հոսող արյան հոսքն՝ ակնկալելով տեսնել առողջ մարդու արյան հոսքին բնորոշ արդյունքներ:

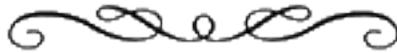
Օպերացիայից հետո հիվանդի վերականգնված զարկերակի և առողջ անձի զարկերակի պուլսացվող արյան հոսքերի բնութագրերի համեմատության բարձր համապատասխանությունն ապացուցեց ՀեմոՍոլվի արդյունավետությունն ու արագագործությունը:

Գլուխ VI-ում ամփոփել ենք այս աշխատությունը՝ քննարկելով ամբողջ թեզն և առաջարկել ենք ապագա զարգացման մի շարք հնարավորություններ:

Acknowledgments

Ralph Fiennes: 'The people I consider successful are so because of how they handle their responsibilities to other people, how they approach the future, people who have a full sense of the value of their life and what they want to do with it.

Success.. is all about being able to extend love to people... not in a big, capital letter sense but in the everyday. Little by little, task by task, gesture by gesture, word by word.'



I would like to open my acknowledgment with the words of gratitude to my promoter Prof. Peter M.A. Sloom. Peter, thank you very much, first of all for replying to my very first mail from Sweden asking if there was any PhD position in your group. Your reply was with challenge and since then I have tried my best to impress you with my knowledge and scientific abilities. I hope I have managed :). I also want to thank you for all your help, from the beginning until the end of my PhD. Knowing that in case of need of help you will always support me, helped me to succeed in many quests. Thank you!

Next, I want to thank my co-promoter Alfons G. Hoekstra. Alfons, it was a pleasure to work with you. You are a great teacher, scientist, you knew when I was stuck and needed a push and you were always ready to give me a hand. Thank you for your enormous help. I was happy to have a supervisor like you as I learned a lot from you and also know how great to have a supervisor with high demands. Now I am ready to work with any tough boss ;).

My next special thanks are for my friend and great professor Bastien Chopard. Bastien I will never forget our nice walks in Krakow and the visit to salt-mines, it was an honor to have you as a guest in Armenia, and I was truly amazed by the beauty of Chateau de Chillon and fairy nature around it, and the tasty Molokhov (if I remember the name correctly) of course. Also my special thanks to Jonas Latt.

I would also like to thank my coordinator in Sweden (KTH, Nada) Lennart Edsberg and the Swedish Institute organization that made my study in Sweden possible and thus new horizons have opened to me.

I am vary grateful to Prof. Hans Reiber and Rob van der Geest for all the help during collaborations within DIME project. And thank you Jorrit and Henk for an easy and fast collaboration.

Another truly amazing experience was my collaboration with several German groups within the International lattice Boltzmann Consortium. I would like to thank Joerg for developing this idea, for his hospitality during my visit to NEC Research Labs. Europe, for many many beer gardens and the Armenian dinner. Special thanks to Guntram (I wish I could work with you at least for a half year, I would learn so much from you), George (your father's bike helped me to discover the wonderful nature around Bonn), Richard and Suzy (for entertaining weekends). Also I would like to thank Dr. Guy Lonsdale and Dr. Achim Basermann for their hospitality.

My amazing experience continued in HLRS with Peter Lammers in Stuttgart. Peter we worked three days, for more than 12 hours per day, around the same table, but I think those were the most productive days ever, that made even Joerg happy ;). And I've appreciated a lot that at the end of the day you still have found energy to show me the beautiful non-touristic Stuttgart.

Next thanks are for Thomas Zeiser. If there is any information in the world that one wants to know about supercomputers, ask Thomas. Thomas it was a great experience to work with you.

What impressed me a lot is the group of Prof. Manfred Krafczyk. It is like one complete family and I was happy to be their guest even for couple of days. For a great help I want to thank Sebastian, Jan and Soeren and of coarse Manfred and Jonas for making it possible.

At the end of my research I had another nice collaboration with a group from Sheffield University for which I would like to thank Dr. Rod Hose, Dr. Pat Lawford and Adam Jeays. And of coarse my warm regards goes to Sara who was like me a visitor in Germany from Sheffield University. During these two visiting periods we managed to become friends, share thoughts and discussions both over lattice Boltzmann and over life in general.

Even though I had a lot of collaboration during my research and multiple visits to other groups, the most of the time I have spent in my office at Science Park with my great office-mates Qiu and Maxim. Although I have tried my best not to disturb them with much noise but I assume three people in a small room can cause problems (although we never had one). Guys you have never complained but still I want to thank you for your patience. It seemed as if we were synchronized in work and in breaks for little chats.

I also would like to thank all the people in and outside of our SCS group: Breannan (happy to be your next wall office neighbor), Alfredo (and Heidi of coarse), Drona, Evghenii, Derek, Alessia, Mike, Stefan, Nicolas, Alfonso, Yves, Jordi, Thomas, Jiangjun, Carlos, Paul, Michael (with Anya), Rob (for advice when I needed one), Simon, Thomasz (with Anja), Eric L., Gokhan, Syed, Dick, Marian, Jaap (was great to have a chat over a beer in Geneva), David, Dorien, Virginie, Amanda, Gohar, Zimming (always useful advice I got from you), Viktor, Wibi, Silvia (especially at the end ;)). Special thanks to Russians: Roman (hope you are managing the grapes-hunt by now with your fork and knife), Denis (for all the help with visualizations), Dmitri, Vladimir, Kostya, Lev and Ilya. Guys my special thanks to you for always being ready to give me a hand when I was moving from house to house and that has happened 5 times. Also it was a pleasure to bicycle with you in Netherlands. Special thanks to Zeger for taking my huge suitcase

into your slim sport-car :). Kamil, you started to help me from the first days, by translating the dutch mails, until the last days, giving hints for the thesis writing format. And of course, thank you Artoli, I don't know how I would manage in Amsterdam if you wouldn't teach me to cycle. Also, your advice, when I have just arrived to continue your work, helped me during all these four years. I am very grateful to you.

A big huge thank you is for Frederik Benjamin Hitipeuw well known as Erik ;). Dear Erik what we all were going to do without you? You are our problem solving person from bureaucratic questions until "no coffee" ones. In the mornings, making a couple of jokes, while taking a cup of coffee, was enough to keep my and (I am sure others will agree with me) all poor PhD student's mood up for the complete day.

One important component for comput(er or ational) scientist is probably the computer ;) and supercomputers in my case. When it has a problem the only ones that can help are network administrators. So I am very thankful to them especially Stephan, Ruud and Adri and also special thanks to Willem Vermine from SARA supercomputing center.

My dear Elena and Frank the first day I met you I realized that I have a family here. I am very grateful to you for being my best friends that cared about me so much. You were the ones sharing with me my happy and not so happy days. Thanks to you I became rich with two more amazing friends.

Valeriu, our friendship started at the first day in Sweden, where we studied in the same group and continues until now and hopefully forever. It was a great coincidence that you also found a PhD position in Amsterdam, CWI. I know that even in a very difficult moment I can always count on you, thank you for that.

I met also amazing people out of work. Those people helped me to understand and partially to get integrated into local society. Those people filled my days in Amsterdam with joy and happiness. I would like to thank Stijn for all the fun days and bicycle session in Oosterpark. Another big thank you is for Erik for all the wonderful memories I gathered in and around Netherlands. My great regards to Erik's family for giving me all their warmth.

I would like also to thank my friends and relatives from different parts of the world that were keeping my spirit up all the time with their attention: Aram Ohanyan, Aram Sahakyan, Arsen, Marianna, Goharik, Milena and Jorgen, Mohammad, Mika, Nora, Armen and Siranush(can you appear again in front of the Tropen museum out of nowhere while you suppose to be in Yerevan ;)), Zakar and Hasmik, Emma and Hakob with Stephan and Arpi, Keijo, Jari, Hans, Melanyan and Ruben (I am so proud that you are my friends) and many many more whom I didn't mention but I always keep them in my heart. My great thanks also to Avagyan family for everything and my great aunt Emma's family.

My special and big thanks to Marko for his great support, encouragements and believe in me during last year of my PhD.

This thesis wouldn't have been finished or even started to be written without all the faith, encouragement and support of my family. I will never be able to express my complete gratitude and love to my grandfather and grandmother. **Unfortunately**, I didn't manage to graduate before my grandfather's death, only by several months. I hope he knew how much I loved him and that his dream to see me a doctor of science, as he was himself, came true. Both my beloved grandparents were my best support team ever dur-

ing these years. One phone call with them gave me enough energy to proceed further. And I want to tell my grandmother how blessed I am that I have her.

This thesis is devoted to my mother, the best mother in the world :). A mother, who not only gave a birth to me but also created step-by-step what I am now. With her we passed through war, earthquake and hunger, fell from richness to poorness and recovered again. And during these four years she was the only one who knew all my worries and excitements, my fails and successes, saw me devastated and happy.

Mama, you are the only one who truly understands me and always supports me, and to devote this thesis to you is the least I could do to express my love to you!

Ես ուզում եմ իմ մեծագույն շնորհակալությունը հայտնել իմ բոլոր ընկերներին և ընկերուհիներին աշխարհի տարբեր ծայրերից՝ Արամ Սահակյանին, Արամ Օհանյանին, Արսենին, Մարիաննային, Գոհարիկին, Միլենային և Յորգենին, Միկային, Նորային, Սիրանուշին ու Արմենին (կարող ե՞ք կրկին հայտնվել Տրոպեյն թանքարանի դիմաց, երբ դուք պիտի որ Երևանում լինեք ;)), Ջաքարին և Հասմիկին, Էմմային ու Հակոբին՝ Ստեփանիկի և Արփիի հետ միասին, Մելանիային ու Ռուբենին՝ Էրիչոկի հետ միասին (ես շատ հպարտ եմ, որ ունեմ ձեր նման ընկերներ) և շատ շատերին որոնց ես չեմ նշել, բայց որոնք միշտ իմ սրտում են: Ձեր բոլորի ուշադրությունը չթողեց ինձ ընկճվել նույնիսկ ամենաբարդ թույլներին: Իմ մեծագույն շնորհակալությունն եմ հայտնում նաև Ավագյան ընտանիքին ամեն ինչի համար եւ իմ հրաշալի մորագույր Էմմայի ընտանիքին:

Այս թեզը չէր ավարտվի և նույնիսկ չէր էլ սկսվի առանց իմ ընտանիքի ապավինության, հավատքի իմ ուժերին և ոգևորության: Ես անկարողու եմ արտահայտել իմ անսահման շնորհակալությունն ու սերը իմ պապիկին և բաբուլիկիս: Ավաղ, ես չհասցրեցի ավարտել մինչ պապիկիս մահը՝ ուշացա ընդամենը մի քանի ամսով: Բայց հուսով եմ նա գիտեր, թե ես որքան շատ եմ իրեն սիրում և, որ իր երազանքը՝ տեսնել ինձ գիտությունների դոկտոր, ես իրականացրեցի: Եվ՝ պապիկս և՛ տատիկս իմ ամենալավ ապավեններն էին այս չորս տարիների ընթացքում: Մեկ հեռախոսագրույցը նրանց հետ բավարար էր, որ ես նոր ոգևորությամբ շարունակեմ իմ աշխատանքը: Եվ ես ուզում եմ բաբուլիկիս անպայման ասել, թե որքան երջանիկ եմ ես, որ ունեմ իր նման տատիկ:

Այս թեզը նվիրել եմ մայրիկիս, աշխարհի ամենալավ մայրիկին ☺: Մայր՝ որը ոչ միայն ծնել է ինձ, այլ նաև քայլ առ քայլ կերտել է այն ինչ ես հիմա եմ: Նրա հետ մենք հաղթահարել ենք կոիվը, երկրաշարժը և սովը: Հարուստից դարձել ենք աղքատ և նորից վերականգնվել: Այս տարիների ընթացքում նա միակն էր և է, որ գիտի բոլոր իմ մտատանջությունները և ոքևորությունները, իմ անկումներն ու հաջողությունները, տեսել է ինձ հուսահատված և երջանիկ:

Մամա, դու միակ մարդն ես այս աշխարհում, որ իսկապես հասկանում ես ինձ և միշտ պաշտպանում: Նվիրել այս թեզը քեզ ամենաքիչն է, որ ես կարող եմ անել քո հանդեպ իմ սերն արտահայտելու համար:

Publications



Journal Publications

- [1] Lilit Axner, Adam Jeays, Alfons G. Hoekstra, Pat Lawford, Rod Hose and Peter M. A. Slood. Simulations of Time Harmonic Blood Flow in the Mesenteric Artery: Comparing Finite Element and Lattice Boltzmann Methods (In preparation).
- [2] Lilit Axner, Henk A. Marquering, Alfons G. Hoekstra, Rob J. van der Geest, Johan H.C. Reiber and Peter M. A. Slood. Decision Support through Computational Hemodynamics in Vascular Surgery In *Physics in Medicine and Biology* (Submitted).
- [3] Lilit Axner, Jörg Bernsdorf, Thomas Zeiser, Peter Lammers, Jan Linxweiler and Alfons G. Hoekstra. Performance Evaluation of a Parallel Sparse Lattice Boltzmann Solver. In *Journal of Computational Physics* (Submitted).
- [4] Lilit Axner, Jonas Latt, Alfons G. Hoekstra, Bastien Chopard and Peter M.A. Slood. Simulating Time Harmonic Flows with the Regularized L-BGK Method. In *International Journal of Modern Physics C*, 18, 4, p.661, 2007.
- [5] Lilit Axner, Alfons G. Hoekstra and Peter M.A. Slood. Simulating Time Harmonic Flows with the Lattice Boltzmann Method. In *Physical Review E*, 75, 3, p. 036709, 2007.

Conference Proceedings

- [6] Lilit Abrahamyan, Jörg Bernsdorf, Thomas Zeiser, Peter Lammers, Alfons G. Hoekstra and Peter M.A. Slood. MPI Parallelization and Performance Aspects of a Graph Based LB Flow Solver. In *Discrete Simulation of Fluid Dynamics DSFD2006*, Geneva, Switzerland, 2006.

- [7] Lilit Abrahamyan, Alfons G. Hoekstra and Peter M.A. Sloot. Choosing Optimal L-BGK Simulation Parameters for Time Harmonic Flows. In *Discrete Simulation of Fluid Dynamics DSFD2006*, Geneva, Switzerland, 2006.
- [8] Lilit Abrahamyan, Jorrit A. Schaap, Alfons G. Hoekstra, Denis Shamonin, Frieke M.A.Box, Rob J. van der Geest, Johan H.C. Reiber, Peter M.A. Sloot. A Problem Solving Environment for Image-Based Computational Hemodynamics. In *LNCS 3514*, p.287, April 2005.
- [9] Abdel Monim Artoli, Lilit Abrahamyan, and Alfons G. Hoekstra. Accuracy versus Performance in Lattice Boltzmann BGK Simulations of Systolic Flows. In *LNCS 3039*, p.548, June 2004.

Miscellaneous Publications

- [10] Lilit Axner and Alfons G. Hoekstra. Virtuele bypassoperatie. In *I/O InformaticaOnderzoek* Journal of Dutch National Foundation, 4, 3, July 2007.
- [11] Lilit Axner. In Uitgelicht stellen ICT-onderzoekers zich aan u voor en vertellen over hun onderzoek en hun fascinatie hiervoor. In *Informaticaonderzoek Platform Nederland online journal*

Bibliography

- [1] <http://www.hpce.nec.com/>.
- [2] <http://www.sara.nl/userinfo/lisa/description/index.html/>.
- [3] <http://www.top500.org/list/2007/06/100>.
- [4] World Health Organization, 2002, The World Health Report 2002, <http://www.who.int/whr/en/>.
- [5] A. Abou-Rjeili and G. Karypis. Multilevel Algorithms for Partitioning Power-Law Graphs. *IEEE International Parallel & Distributed Processing Symposium (IPDPS)*, vol. 10, 2006.
- [6] S. Ansumali and I. V. Karlin. Single Relaxation Time Model for Entropic Lattice Boltzmann Methods. *Physical Review E*, vol. 65, no. 5, pp. 056312 – 9, 2002.
- [7] A. M. Artoli. *Mesoscopic Computational Haemodynamics*. PhD dissertation, University of Amsterdam, Informatic Institute, 2003.
- [8] A. M. Artoli, A. G. Hoekstra, and P. M. Sloot. 3D Pulsatile Flow with the Lattice Boltzmann BGK Method. *International Journal of Modern Physics C*, vol. 13, no. 8, pp. 1119–1134, 2002.
- [9] A. M. Artoli, A. G. Hoekstra, and P. M. Sloot. Simulation of a Systolic Cycle in a Realistic Artery with the Lattice Boltzmann BGK Method. *International Journal of Modern Physics B*, vol. 17, no. 1-2, pp. 95 – 98, 2003.
- [10] A. M. Artoli, A. G. Hoekstra, and P. M. Sloot. Mesoscopic Simulations of Systolic Flow in the Human Abdominal Aorta. *Journal of Biomechanics*, vol. 39, no. 5, pp. 873–884, 2006.
- [11] A. M. Artoli, A. G. Hoekstra, and P. M. Sloot. Optimizing Lattice Boltzmann Simulations for Unsteady Flows. *Computers & Fluids*, vol. 35, no. 2, pp. 227–240, 2006.
- [12] L. Axner, J. Bernsdorf, T. Zeiser, P. Lammers, J. Linxweiler, and A. Hoekstra. Performance Evaluation of a Parallel Sparse Lattice Boltzmann Solver. *Journal of Computational Physics*, vol. (submitted), 2007.

- [13] L. Axner, A. Hoekstra, and P. Sloot. Simulating Time Harmonic Flows with the Lattice Boltzmann Method. *Physical Review E*, vol. 75, no. 3, pp. 036709+7, 2007.
- [14] L. Axner, J. Latt, A. G. Hoekstra, B. Chopard, and P. M. Sloot. Simulating Time Harmonic Flows with the Regularized L-BGK Method. *International Journal of Modern Physics C*, vol. (accepted), 2007.
- [15] A. Basermann, J. Clinckemallie, T. Coupez, J. Fingberg, H. Digonnet, R. Ducloux, J.-M. Gratien, U. Hartmann, G. Lonsdale, B. Maerten, D. Roose, and C. Walshaw. Dynamic Load Balancing of Finite Element Applications with the DRAMA Library. *Applied Mathematical Modelling*, vol. 25, pp. 83 – 98, 2000.
- [16] G. Bell, J. Gray, and A. Szalay. Petascale Computational Systems. *IEEE Computer*, vol. 39, no. 1, pp. 110 – 112, 2006.
- [17] R. Benzi, S. Succi, and M. Vergassola. The Lattice Boltzmann Equation: Theory and Applications. *Physics Reports*, vol. 222, no. 3, pp. 145 – 197, 1992.
- [18] J. Bernsdorf, S. E. Harrison, S. M. Smith, P. V. Lawford, and D. R. Hose. High Performance Computing on Vector Systems 2006. In *Proceedings of the High Performance Computing Center Stuttgart*, vol. March, 2006.
- [19] M. Bouzidi, M. Firdaouss, and P. Lallemand. Finite Element Modeling of Blood Flow in Arteries. *Physics of Fluids*, vol. 13, no. 11, pp. 3452–3459, November 2001.
- [20] G. A. Buxton, R. Verberg, D. Jasnow, and A. C. Balazs. Newtonian Fluid Meets an Elastic Solid: Coupling Lattice Boltzmann and Lattice-Spring Models. *Physical Review E*, vol. 71, p. 056707, 2005.
- [21] J. Cebra, R. Löhner, P. Choyke, and P. Yim. Parallel Patient-Specific Computational Haemodynamics. In *Lecture Notes in Computer Science*, vol. 2367, pp. 18 – 34, 2002.
- [22] H. Chen. Volumetric Formulation of the Lattice Boltzmann Method for Fluid Dynamics: Basic Concept. *Physical Review E*, vol. 58, no. 3, pp. 3955 – 3963, September 1998.
- [23] S. Chen and G. D. Doolen. Lattice Boltzmann Method for Fluid Flows. *Annual Review Fluid Mechanics*, vol. 30, pp. 329 – 364, 1998.
- [24] S. Chen, Z. Wang, X. Shan, and G. D. Doolen. Lattice Boltzmann Computational Fluid Dynamics in Three Dimensions. *Journal of Statistical Physics*, vol. 68, no. 3/4, pp. 379 – 400, 1992.
- [25] B. Chopard and M. Droz. *Cellular Automata Modeling of Physical Systems*. Cambridge University Press, 1998.
- [26] D. Clague, D. Kandhai, R. Zhang, and P. M. Sloot. On the Hydraulic Permeability of (Un)Bounded Fibrous Media Using the Lattice-Boltzmann Method. *Physical Review E*, vol. 61, no. 1, pp. 616–625, 2000.
- [27] J. Cosgrove, J. Buick, S. Tonge, C. Munro, C. Greated, and D. Campbell. Application of the Lattice Boltzmann Method to Transition in Oscillatory Channel Flow. *Journal of Physics A: Mathematical and General*, vol. 36, pp. 2609 – 2620, 2003.

- [28] C. Pan, J. Prins, and C. T. Miller. A high-performance lattice boltzmann implementation to model flow in porous media. *Comp. Phys. Comm.*, vol. 158, no. 2, p. 89, 2004.
- [29] O. Cuisenaire. *Distance Transformations: Fast Algorithms and Applications to Medical Image Processing*. PhD dissertation, Universite catholique de Louvain, Laboratoire de Telecommunications et Teledetection, 1999.
- [30] R. R. A. Dampney. Functional Organization of Central Pathways Regulating the Cardiovascular System. *Physiological Reviews*, vol. 74, no. 2, pp. 323 – 364, April 1994.
- [31] P. de Koning, J. Schaap, J. Janssen, J. Westenberg, R. van der Geest, and J. Reiber. Automated Segmentation and Analysis of Vascular Structures in Magnetic Resonance Angiographic Images. *Magnetic Resonance in Medicine*, vol. 50, no. 6, pp. 1189 – 1198, 2003.
- [32] J.-C. Desplat, I. Pagonabarraga, and P. Bladon. LUDWIG: A parallel Lattice-Boltzmann code for complex fluids. *Computer Physics Communications*, vol. 134, no. 3, p. 273, 2002.
- [33] A. Dupuis and B. Chopard. An Object Oriented Approach to Lattice Gas Modeling. *Future Generation Computer Systems*, vol. 16, no. 5, pp. 523 – 532, 2000.
- [34] H. Fang, Z. Wang, Z. Lin, and M. Liu. Lattice Boltzmann Method for Simulating the Viscous Flow in Large Distensible Blood Vessels. *Physical Review E*, vol. 65, pp. 051925–11, 2002.
- [35] O. Filippova and D. Heanel. Grid Refinement for Lattice-BGK Models. *Journal of Computational Physics*, vol. 147, no. 1, pp. 219 – 228, 1998.
- [36] A. Finkelstein, J. Hetherington, L. Li, O. Margoninski, P. Saffrey, R. Seymour, and A. Warner. Computational Challenges of Systems Biology. *IEEE Computer*, vol. 37, no. 5, pp. 26 – 33, 2004.
- [37] I. Foster. *Designing and Building Parallel Programs: Concepts and Tools for Parallel Software Engineering*. Addison Wesley, 1995.
- [38] G. Fox, M. Johnson, G. Lyzenga, S. Otto, J. Solman, and D. Walker. *Solving Problems on Concurrent Processors*. Prentice-Hall Int. Inc., 1988.
- [39] E. Gallopoulos, E. N. Houstis, and J. R. Rice. Computer as Thinker/Doer: Problem-Solving Environments for Computational Science. *Computational Science and Engineering, IEEE*, vol. 1, no. 2, pp. 11 – 23, 1994.
- [40] S. Geller, M. Krafczyk, J. Toelke, S. Turek, and J. Hron. Benchmark Computations Based on Lattice-Boltzmann, Finite Element and Finite Volume Methods for Laminar Flows. *Computers and Fluids*, vol. 35, pp. 888–897, 2006.
- [41] R. Gin, A. G. Straatman, and D. A. Steinman. Numerical Modelling of the Carotid Artery Bifurcation Using a Physiologically Relevant Geometric Model. *7th Annual Meeting of the CFD Society of Canada*, vol. Halifax NS, 1999.

- [42] Z. Guo, T. S. Zhao, and Y. Shi. Preconditioned lattice-Boltzmann Method for Steady Flows. *Physical Review E*, vol. 70, pp. 066706 – 066714, 2004.
- [43] Z. Guo, C. Zheng, and T. Zhao. A Lattice BGK Scheme with General Propagation. *Journal of Scientific Computing*, vol. 16, no. 4, pp. 569–585, 2001.
- [44] X. He and L.-S. Luo. Lattice Boltzmann Model for the Incompressible Navier-Stokes Equation. *Journal of Statistical Physics*, vol. 88, no. 3-4, pp. 927 – 944, 1997.
- [45] X. He and L.-S. Luo. A priori derivation of the lattice Boltzmann equation. *Physical Review E*, vol. 55, p. R6333, 1997.
- [46] X. He, L.-S. Luo, and M. Dembo. Some Progress in Lattice Boltzmann Method. Part1: Nonuniform Mesh Grids. *Journal of Computational Physics*, vol. 129, pp. 357–363, July 1996.
- [47] B. Hendrickson and K. Devine. Dynamic Load Balancing in Computational Mechanics. *Computer Methods in Applied Mechanics and Engineering*, vol. 184, pp. 485 – 500, 2000.
- [48] F. J. Higuera. Lattice Gas Method Based on the Chapman-Enskog Expansion. *Physics of Fluids A*, vol. 2, no. 6, pp. 1049–1051, 1990.
- [49] R. W. Hockney and C. R. Jesshope. *Parallel Computers 2: Architecture, Programming and Algorithms*. IOP Publishing Ltd., Bristol, UK, 1988.
- [50] E. N. Houstis and J. R. Rice. Future Problem Solving Environments for Computational Science. *Mathematics Computer Simulation*, vol. 54, pp. 243–257, 2000.
- [51] G. HÁzi. Accuracy of the Lattice Boltzmann Method based on Analytical Solutions. *Physical Review E*, vol. 67, no. 5, pp. 056705–5, 2003.
- [52] J. Janssen, G. Koning, P. de Koning, J. Tuinenburg, and J. Reiber. A Novel Approach for the Detection of Pathlines in X-Ray Angiograms: the Wavefront Propagation Algorithm. *International Journal of Cardiovascular Imaging*, vol. 18, no. 5, pp. 317 – 324, 2002.
- [53] A. Jeays, P. Lawford, R. Gillott, P. Spencer, D. Barber, K. Bardhan, and D. Hose. Characterisation of the Haemodynamics of the Superior Mesenteric Artery. *Journal of Biomechanics*, vol. doi:10.1016/j.jbiomech.2006.09.009, 2006.
- [54] B. Kandhai, D.-E. Vidal, A. Hoekstra, H. Hoefsloot, P. Iedema, and P. Slood. Lattice-Boltzmann and Finite Element Simulations of Fluid Flow in a SMRX Mixer. *International Journal of Numerical Methods in Fluids*, vol. 31, pp. 1019 – 1033, 1999.
- [55] D. Kandhai, D. Dubbeldam, A. G. Hoekstra, J. Timonen, and P. M. A. Slood. Parallel Lattice-Boltzmann Simulation of Fluid Flow in Centrifugal Elutriation Chambers. In *Lecture Notes in Computer Science*, vol. 1401, pp. 173 – 182, 1998.
- [56] D. Kandhai, D. Hlushkou, A. G. Hoekstra, P. M. Slood, H. van As, and U. Tallarek. Influence of Stagnant Zones on Transient and Asymptotic Dispersion in Macroscopically Homogeneous Porous Media. *Physical Review letters*, vol. 88, no. 23, pp. 234501:1–4, 2002.

- [57] D. Kandhai, A. Koponen, A. G. Hoekstra, M. Kataja, J. Timonen, and P. M. Slood. Lattice Boltzmann Hydrodynamics on Parallel Systems. *Computer Physics Communications*, vol. 111, pp. 14 – 26, 1998.
- [58] D. Kandhai, A. Koponen, A. G. Hoekstra, M. Kataja, J. Timonen, and P. M. A. Slood. Implementation aspects of 3D Lattice-BGK: Boundaries, Accuracy and a new Fast Relaxation Method. *J. Comp. Phys.*, vol. 150, pp. 482–501, 1999.
- [59] D. Kandhai, A. Koponen, A. G. Hoekstra, J. Timonen, and P. M. A. Slood. Implementation Aspects of 3D Lattice-BGK: Boundaries Accuracy, and a New Fast Relaxation Method. *Journal of Computational Physics*, vol. 150, pp. 482–501, 1999.
- [60] D. Kandhai, W. Soll, S. Chen, A. G. Hoekstra, and P. M. Slood. Finite-Difference Lattice-BGK Methods on Nested Grids. *Computer Physics Communication*, vol. 129, pp. 100 – 109, 2000.
- [61] G. Karypis and V. Kumar. Multilevel Graph Partitioning Schemes. In *International Conference on Parallel Processing*, vol. 3, pp. 113–122, 1995.
- [62] G. Karypis and V. Kumar. Multilevel k-way Partitioning Scheme for Irregular Graphs. *Journal of Parallel and Distributed Computing*, vol. 481, no. 1, pp. 96 – 129, 1998.
- [63] A. Koponen, M. Kataja, J. Timonen, and D. Kandhai. Simulation of Single-Fluid Flow in Porous Media. *International Journal of Modern Physics C*, vol. 9, pp. 1505–1522, 1998.
- [64] D. N. Ku. Blood Flow in Arteries. *Annual Review of Fluid Mechanics*, vol. 29, pp. 399 – 434, 1997.
- [65] J. P. Ku, M. T. Draney, F. R. Arko, W. A. Lee, F. Chan, N. J. Pelc, C. K. Zarins, and C. A. Taylor. In Vivo Validation of Numerical Predictions of Blood Flow in Arterial Bypass Grafts. *Annals of Biomedical Engineering*, vol. 30, no. 6, pp. 743 – 752, 2002.
- [66] P. Lallemand and L.-S. Luo. Theory of the Lattice Boltzmann Method: Dispersion, Dissipation, Isotropy, Galilean Invariance, and Stability. *Physical Review E*, vol. 61, no. 6, pp. 6546 – 6562, 2000.
- [67] P. Lammers, G. Wellein, T. Zeiser, and M. Breuer. Have the vectors the continuing ability to parry the attack of the killer micros? In M. Resch, T. Bönisch, K. Benkert, T. Furui, Y. Seo, and W. Bez, editors, *High Performance Computing on Vector Systems. Proceedings of the High Performance Computing Center Stuttgart, March 2005*, pp. 25–39, Berlin, 2006. Springer.
- [68] J. Latt and B. Chopard. Lattice Boltzmann Method with Regularized Nonequilibrium Distribution Functions. *Mathematics Comp. Sim.*, vol. 72, pp. 165 – 168, 2006.
- [69] W. Li, X. Wei, and A. E. Kaufman. Implementing Lattice Boltzmann Computation on Graphics Hardware. *The Visual Computer*, vol. 19, no. 7-8, pp. 444–456, 2003.
- [70] J.-M. Luursema, W. B. Verwey, P. A. Kommers, R. H. Geelkerken, and H. J. Vos. Optimizing conditions for computer-assisted anatomical learning. *Interact. Comp.*, vol. 18, pp. 1123 – 1138, 2006.

- [71] H. Marquering, J. Dijkstra, P. de Koning, B. Stoel, and J. Reiber. Towards Quantitative Analysis of Coronary. *International Journal of Cardiovascular Imaging*, vol. 21, pp. 73–84, 2005.
- [72] D. Martinez, W. Matthaeus, and S. Chen. Comparison of Spectral Method and Lattice Boltzmann Simulations of Two-Dimensional Hydrodynamics. *Physics of Fluids*, vol. 6, no. 3, pp. 1285 – 1298, 1993.
- [73] F. Mazzocco, C. Arrighetti, G. Bella, L. Spagnoli, and S. Succi. Multiscale Lattice Boltzmann Schemes: A Preliminary Application to Axial Turbomachine Flow Simulations. *International Journal of Modern Physics C*, vol. 11, no. 2, pp. 233 – 245, 2000.
- [74] H. X. nad Gongwen Peng and S.-H. Chou. Finite-Volume Lattice Boltzmann Method. *Physical Review E*, vol. 59, no. 5, pp. 6202–6205, 1998.
- [75] F. Nannelli and S. Succi. The Lattice Boltzmann Equation on Irregular Lattices. *Journal Statistical Physics*, vol. 68, no. 3-4, pp. 401 – 407, 1992.
- [76] X. Niu, C. Shu, Y. Chew, and T. Wang. Investigation of Stability and Hydrodynamics of Different Lattice Boltzmann Models. *Journal of Statistical Physics*, vol. 117, no. 3-4, pp. 665 – 680, 2004.
- [77] G. Peng, H. Xi, C. Duncan, and S.-H. Chou. Lattice Boltzmann Method on Irregular Meshes. *Physical Review E*, vol. 58, no. 4, pp. R4124 – R4127, October 1998.
- [78] T. Pohl, F. Deserno, N. Thurey, U. Rude, P. Lammers, G. Wellein, and T. Zeiser. Performance Evaluation of Parallel Large-Scale Lattice Boltzmann Applications on Three Supercomputing Architectures. In *SC '04: Proceedings of the 2004 ACM/IEEE conference on Supercomputing*, p. 21, Washington, DC, USA, 2004. IEEE Computer Society.
- [79] B. R. Rau and J. A. Fisher. Instruction-Level Parallel Processing: History, Overview and Perspective. *Journal of Supercomputing*, vol. 7, pp. 9–50, May 1993.
- [80] C. Redaelli, M. Schilling, and M. Buchler. Intraoperative laser Doppler Flowmetry: a Predictor of Ischemic Injury in Acute Mesenteric Infarction. *Digestive Surgery*, vol. 15, no. 1, pp. 55 – 9, 1998.
- [81] J. Rivet and J. Boon. *Lattice Gas Hydrodynamics*. Cambridge University Press, 2001.
- [82] M. Rohde. *Extending the Lattice Boltzmann Method: Novel Techniques for Local Grid Refinement and Boundary Conditions*. PhD dissertation, Technical University of Delft, 2004.
- [83] T. Saito and J. Toriwaki. New Algorithms for Euclidean Distance Transformation of an N-dimensional Digitized Picture with Applications. *Pattern Recognition*, vol. 27, no. 11, pp. 1551 – 1565, 1994.
- [84] N. Satofuka and T. Nishioka. Parallelization of lattice Boltzmann method for incompressible flow computations. *J. Comput. Mech.*, vol. 23, pp. 164–171, 1999.
- [85] M. Schulz, M. Krafczyk, J. Toelke, and E. Rank. Parallelization Strategies and Efficiency of CFD Computations in Complex Geometries using Lattice-Blotzmann

- Methods on High-Performance Computers. *3rd International FORTWIHR Conference on HPSEC, Erlangen,,* vol. March 12-14, 2001, p. 115, 2002.
- [86] J. Sethian. A Fast Marching Level Set Method for Monotonically Advancing Fronts. *National Academy of Science of USA*, vol. 93, no. 4, pp. 1591 – 1595, 1996.
- [87] L. A. Shemer. Laminar-Turbulent Transition in a Slowly Pulsating Pipe Flow. *Physics of Fluids*, vol. 28, no. 12, pp. 3506 – 3509, 1985.
- [88] Y. Shi, T. Zhao, and Z. Guo. Lattice Boltzmann Method for Incompressible Flows with Large Pressure Gradients. *Physical Review E*, vol. 73, no. 2, pp. 026704 – 11, 2006.
- [89] P. Sloot and A. Hoekstra. *Handbook of Dynamic System Modeling*, vol. Modeling Dynamic Systems with Cellular Automata. Series: Cpaman & Hall/CRC Computer & Information Science Series, 2007.
- [90] P. M. A. Sloot, J. A. Kaandorp, A. G. Hoekstra, and B. J. Overeinder. Distributed Cellular Automata: Large-Scale Simulation of Natural Phenomena. *Solutions to Parallel and Distributed Computing Problems*, vol. 1, pp. 1 – 46, 2001.
- [91] M. Sonka, M. Winniford, and S.M.Collins. Robust Simultaneous Detection of Coronary Borders in Complex Images. *IEEE Trans Medical Imaging*, vol. 14, pp. 151–161, 1995.
- [92] D. Steinman. Simulated Pathline Visualization of Computed Periodic Blood Flow Patterns. *Journal of Biomechanics*, vol. 33, no. 5, pp. 623 – 628, 2000.
- [93] D. Steinman. Image-Based Computational Fluid Dynamics Modeling in Realistic Arterial Geometries. *Annals of Biomedical Engineering*, vol. 30, no. 4, pp. 483 – 482, 2002.
- [94] J. Sterling and S. Chen. Stability Analysis of Lattice Boltzmann Methods. *Journal of Computational Physics*, vol. 123, no. 1, pp. 196–206, 1996.
- [95] M. Stiebler, J. Tölke, and M. Krafczyk. An Upwind Discretization Scheme for the Finite Volume Lattice Boltzmann Method. *Computers & Fluids*, vol. 35, no. 8-9, pp. 814 – 819, 2006.
- [96] E. Strohmaier, J. J. Dongarra, H. W. Meuer, and H. D. Simon. Recent Trends in the Marketplace of High Performance Computing. *Parallel Computing*, vol. 31, pp. 3–4, March/April 2005.
- [97] S. Succi. *The Lattice Boltzmann Equation for Fluid Dynamics and Beyond*. New York:Oxford, 2001.
- [98] C. A. Taylor and M. T. Draney. Experimental and Computational Methods in Cardiovascular Fluid Mechanics. *Annual Review Fluid Mechanics*, vol. 36, pp. 197–231, 2004.
- [99] C. A. Taylor, M. T. Draney, J. P. Ku, D. Parker, B. N. Steele, K. Wang, and C. K. Zarins. Predictive Medicine: Computational Techniques in Therapeutic Decision-Making. *Computer Aided Surgery*, vol. 30, pp. 231 – 247, 1999.

- [100] C. A. Taylor, T. J. Hughes, and C. K. Zarins. Finite Element Modeling of Blood Flow in Arteries. *Computer Methods in Applied Mechanics and Engineering*, vol. 158, pp. 155 – 196, 1998.
- [101] M. J. Thubrikar. *Vascular Mechanics and Pathology*. Springer Science+Business Media, LLC, 2007.
- [102] A. Tirado-Ramos, D. Groen, and P. Sloot. On-line Application Performance Monitoring of Blood Flow Simulation in Computational Grid Architectures. In *18th IEEE International Symposium on Computer-Based Medical Systems*, vol. CBMS'05, pp. 511–516, 2005.
- [103] A. Tirado-Ramos and P. Sloot. A Conceptual Grid Architecture for Interactive Biomedical Applications. In *19th IEEE International Symposium on Computer-Based Medical Systems*, vol. CBMS'06, pp. 762–767, 2006.
- [104] A. Tirado-Ramos, P. M. Sloot, A. G. Hoekstra, and M. T. Bubak. An Integrative Approach to High-Performance Biomedical Problem Solving Environments on the Grid. *Parallel Computing*, vol. 30, no. 9-10, pp. 1037 – 1055, 2004.
- [105] J. Tölke, S. Freudiger, and M. Krafczyk. An adaptive scheme for LBE Multiphase Flow Simulations on Hierarchical Grids. *Computers and Fluids*, vol. 35, no. 8-9, p. 820, 2006.
- [106] J. Tölke, M. Krafczyk, and E. Rank. A Multigrid Solver for the Discrete Boltzmann Equation. *Journal of Statistical Physics*, vol. 107, no. 1/2, pp. 573 – 591, 2002.
- [107] J. Tölke, M. Krafczyk, M. Schulz, E. Rank, and R. Berrios. Implicit Discretization and Non-uniform Mesh Refinement Approaches for FD Discretizations of LBGK Models. *International Journal of Modern Physics C*, vol. 9, no. 8, pp. 1143–1157, 1998.
- [108] A. J. van der Steen. Overview of Recent Supercomputers. *Netherlands National Computing Facilities Foundation*, vol. The Haag, January 2004.
- [109] N. J. Wald and M. R. Law. A Strategy to Reduce Cardiovascular Disease by more than 80 BMJ Publishing Group, vol. 326, pp. 1419 – 1423, June 2003.
- [110] C. Walshaw and M. Everett. Parallel Dynamic Graph Partitioning for Adaptive Unstructured Meshes. *Journal of Parallel and Distributed Computing*, vol. 47, pp. 102 – 108, 1997.
- [111] J. Wang, X. Zhang, A. G. Bengough, and J. W. Crawford. Domain-decomposition method for parallel lattice Boltzmann simulation of incompressible flow in porous media. *Phys. Rev. E*, vol. 72, pp. 016706+11, 2005.
- [112] G. Wellein, G. H. T. Zeiser, and S. Donath. On the single processor performance of simple lattice Boltzmann kernels. *Computers and Fluids*, vol. 35, no. 8-9, pp. 910–919, 2006.
- [113] G. Wellein, T. Zeiser, P. Lammers, and U. Küster. Towards Optimal Performance for Lattice Boltzmann Applications on Terascale Computers. In A. Deane, G. Brenner, A. Ecer, et al., editors, *Parallel Computational Fluid Dynamics: Theory and Applications, Proceedings of the 2005 International Conference on Parallel Computational Fluid Dynamics, May 24-27, College Park, MD, USA*, pp. 31–40, Amsterdam, 2006. Elsevier.

- [114] N. Westerhof, F. Bosman, C. de Vries, and A. Noordergraaf. Analog Studies of the Human Systemic Arterial Tree. *Journal of Biomechanics*, vol. 2, pp. 121 – 143, 1969.
- [115] J. White, III, and S. Bova. Where's the Overlap? - An Analysis of Popular MPI . . . , 1999.
- [116] N. Wilson, F.R.Arko, and C. Taylor. Predicting Changes in Blood Flow in Patient-Specific Operative Plans for Treating Aortoiliac Occlusive Disease. *Computer Aided Surgery*, vol. 10, no. 4, pp. 257–277, 2005.
- [117] J. Womersley. Method for the Calculation of Velocity, Rate of Flow and Viscous Drag in Arteries when the Pressure Gradient is Known. *Journal of Physiology*, vol. 127, no. 3, pp. 553 – 563, 1955.
- [118] N. Wood. Aspects of Fluid Dynamics Applied to the Large Arteries. *J. Theor. Biol.*, vol. 199, pp. 137 – 161, 1999.
- [119] P. R. Woodward. Interactive Scientific Visualization of Fluid Flow. *IEEE Computer*, vol. 26, no. 10, pp. 13 – 25, 1993.
- [120] R. A. Worthing, J. Mozer, and G. Seeley. Stability of Lattice Boltzmann Methods in Hydrodynamic Regimes. *Physical Review E*, vol. 56, no. 2, pp. 2243 – 2253, 1997.
- [121] X. Wu, V. Taylor, S. Garricka, D. Yu, and J. Richard. Performance Analysis, Modeling and Prediction of a Parallel Multiblock Lattice Boltzmann Application Using Prophecy System. In *IEEE International Conference on Cluster Computing*, vol. 8, pp. 1–8, 2006.
- [122] K. Xu and K. Prendergast. Numerical Navier-Stokes solutions from gas kinetic theory. *J. Comput. Phys.*, vol. 114, pp. 9–17, 1994.
- [123] Y.H.Qian, D. d'Humieres, and P. Lallemand. Lattice BGK Models for Navier-Stokes Equation. *Europhys Letters*, vol. 17, pp. 479 – 484, 1992.
- [124] D. Yu and S. Girimaji. Multi-Block Lattice Boltzmann Method: Extension to 3D and Validation in Turbulence. *Physica A*, vol. 362, pp. 118–124, 2006.
- [125] D. Yua, R. Mei, and W. Shyy. A Multi-Block Lattice Boltzmann Method for Fluid Flows. *International Journal for Numerical Methods in Fluids*, vol. 39, pp. 99 – 120, 2002.
- [126] E. V. Zudilova and P. M. Sloot. Multi-Modal Interaction in Biomedicine, in Y. Cai, editor, *Ambient Intelligence for Scientific Discovery. Lecture Notes in Artificial Intelligence*, vol. 3345, pp. 184 – 201, 2005.

Index

- 3D editing, 16, 18–20, 24, 64–67
- abdominal aorta (AA), 1, 3, 20, 22–24, 36, 40, 41, 49, 50, 52, 54, 56, 66
- accuracy, 11, 12, 27–29, 38, 39, 41, 55
- analysis, 16, 27, 28, 30, 35, 40, 43, 49, 64
- analytical, 9, 27, 28, 30, 31, 33–35, 38, 41
- aneurysm, 2, 3, 17, 19, 23
- artery, 1–3, 15, 16, 19, 20, 23, 65, 67
- asymptotic error, 30, 35
- atherosclerosis, 1–3, 15, 67
- Bessel function, 28
- Bhatnagar-Gross-Krook (BGK), 7
- bifurcation, 1, 19, 22, 24, 36, 37
- blood flow, 1, 3, 6, 15, 16, 19, 23–25, 36, 41, 63, 64, 66, 70, 71
- Bounce back on links (BBL), 28, 36, 38, 40, 66
- Bouzidi boundary condition (BBC), 28, 38, 40
- bypass, 15, 19, 24, 64, 67, 69
- cardiovascular disease, 1, 13, 15
- carotid artery, 67–70
- Centerline extraction, 18
- CFD, 4, 7, 13, 16, 24, 43, 64
- Chapman-Enskog analysis, 7, 38
- collision, 7, 11, 27
- complex geometries, 11, 27, 53, 54
- compressibility error, 7, 30, 33–35, 41
- computational mesh, 11, 19
- computer tomography angiography (CTA), 18
- constraint optimization scheme, 13, 20, 27, 28, 35, 36, 40, 41, 67
- contour detection, 18, 64, 65
- convergence, 7, 13, 27, 28, 67
- Courant-Friedrich-Levy condition, 7, 27
- Curved Multi Planar Reformatted (CMPR), 65
- decision support, 63, 71
- decomposition, 9, 10, 43, 44, 50
- density distribution, 7, 8, 11, 44
- edge-cuts, 43, 45, 49, 50, 53, 54
- efficiency, 11, 12, 25, 45, 46, 49, 50, 55, 58
- equilibrium, 8, 37
- execution time, 27–31, 39, 41, 46, 47, 49, 51, 52, 54, 55, 59, 60
- Finite Element Method (FEM), 13, 19, 20
- FLOTRAN (ANSYS Inc.), 20
- flow behavior, 1, 20
- fluid dynamics, 1, 4, 15, 27, 64
- FV-LBM, 11
- Galilean invariance, 7
- geometric schemes, 5
- graph partitioning library, 10, 44
- Grid, 15, 25, 63, 71
- hemodynamics, 1, 4, 6, 15, 19, 25, 63, 68, 71

- HemoSolve, 13, 15, 16, 19, 23–25, 63–66, 68, 70, 71
- high performance computing, 13, 43
- human vascular system, 11, 19, 23, 25, 63, 64, 71
- inlet/outlet boundary conditions, 20, 28, 38
- interaction, 24, 70
- iso-error, 30, 31
- L-BGK, 27, 37–40
- lattice gas automata (LGA), 7
- Lattice Updates per second, 55
- LBM, 7, 9–13, 19, 20, 24, 27, 28, 30, 34, 35, 40
- load balance, 9, 45, 55
- lumen contour detection, 65
- Mach number (Ma), 7, 8, 27–31, 34, 39, 67
- Mach number(Ma), 29
- Medical data segmentation, 16, 64
- mesh generation, 16, 19, 20, 64, 66
- METIS, 10, 43, 45, 48, 53, 57
- modeling, 3, 7
- MRI, 20
- multi-block, 12
- multilevel graph partitioning, 5
- multilevel K-way, 45, 49, 52–54, 57
- multilevel RB, 45, 49, 52–54, 57
- multilevel schemes, 5
- Navier-Stokes, 6, 7, 13, 20, 23, 27, 38
- NEC SX-8, 5, 50–55, 57–60, 62
- Newtonian flow, 7, 19
- non-equilibrium, 11, 12, 37, 38
- ORB, 10
- overhead, 44, 46–49, 53–55, 57
- partitioning, 5, 44, 45, 47–49, 52–55, 57, 58
- pathline, 16, 66
- PC cluster, 50, 51, 53–55, 57
- performance prediction model, 13, 44, 45, 50, 55
- personal space station (PSS), 24
- Poisson equation, 7
- porous media (PM), 27, 49, 50, 54
- pre-operational planing, 15, 63
- problem solving environment (PSE), 12, 13, 15, 24, 25, 63, 64, 68, 70
- propagation, 7, 11, 12
- quasi-incompressible, 28
- regularization scheme, 27
- relaxation parameter, 8, 11, 28, 36
- Reynolds number (Re), 8, 24, 27–29, 35, 38, 41, 66, 67
- RL-BGK, 27, 38, 39, 41
- scalability, 44, 45, 58
- segmentation, 19, 67
- SIMD architecture, 5
- simulation error, 28, 29, 40, 41
- single processor speed, 51, 55
- single-relaxation scheme, 12
- sparse-LBM, 43, 44, 66
- Spectral methods, 5
- SSE2, 5
- stability, 7, 27, 28, 30, 35, 39–41
- stenosis, 3, 67
- stent placement, 15
- straight square channel (SSC), 49, 52–54
- stress tensor, 38
- supercomputer, 4, 5, 57
- superior mesenteric artery (SMA), 20–22
- systole, 16, 20, 24
- time-harmonic, 1, 6, 9, 13, 19, 23, 25, 27, 28, 31, 36, 40, 41, 64, 70, 71
- transversal contour, 18, 64
- unstructured, 10, 11, 67
- user-friendly, 15, 19, 65, 71
- vector machine, 4, 50, 51, 55, 57

velocity profiles, 20, 64, 67
vertices, 45
viscosity, 8, 11, 30, 41
visualization, 19, 23, 24, 66, 70

WaveProp, 18
Westerhof model, 20
Womersley number (α), 8, 28, 38, 66
Womersley solution, 9, 20, 27, 28, 34, 38,
41

