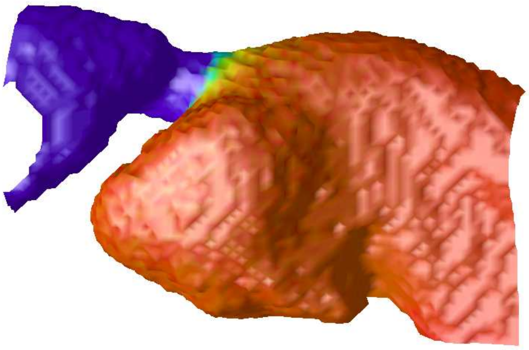
FRIEDRICH-ALEXANDER-UNIVERSITÄT ERLANGEN-NÜRNBERG
INSTITUT FÜR INFORMATIK (MATHEMATISCHE MASCHINEN UND DATENVERARBEITUNG)

Lehrstuhl für Informatik 10 (Systemsimulation)

Numerical Simulation of Bloodflow in Aneurysms using the Lattice Boltzmann Method

Jan Götz



Master Thesis

Numerical Simulation of Bloodflow in Aneurysms using the Lattice Boltzmann Method

Jan Götz

Master Thesis

Aufgabensteller:	Prof. Dr. U. Rüde
Betreuer:	M.Sc. K. Iglberger
Bearbeitungszeitraum:	14.01.2006 – 14.07.2006

Erklärung:

Ich versichere, dass ich die Arbeit ohne fremde Hilfe und ohne Benutzung anderer als der angegebenen Quellen angefertigt habe und dass die Arbeit in gleicher oder ähnlicher Form noch keiner anderen Prüfungsbehörde vorgelegen hat und von dieser als Teil einer Prüfungsleistung angenommen wurde. Alle Ausführungen, die wörtlich oder sinngemäß übernommen wurden, sind als solche gekennzeichnet.

Erlangen, den 12. Juli 2006 ....

Abstract

The largest excess mortality in Germany is due to cardiovascular diseases among both males and females. To diagnose these diseases, modern techniques like computer tomography, digital subtraction angiography, or magnetic resonance imaging play a key role in today's medical care. They allow doctors a detailed insight in the human body up to a resolution of $1mm$ and below. On the other hand, hemodynamic values like shear stress or pressure could hardly be measured or visualized, but might be essential for further treatment of the patient. In case of aneurysms no one can predict the exact risk of rupturing. Particularly the hemodynamics in vessel malfunctions may give the doctor additional information and may influence further treatment. A bloodflow simulation can provide this information.

The Lattice Boltzmann method is known to be a highly efficient method for the simulation of flow problems, especially for complex geometries. Due to this fact it looks like the perfect method for the simulation of bloodflow in the human brain. But the memory requirements for the large domain sizes make these simulations only possible on supercomputers. In this thesis a method is investigated, which drastically reduces the memory requirements for a bloodflow simulation in the brain and allows a simulation on smaller PCs. The procedure from data acquisition over data preparation and simulation to the visualization is discussed in detail. The simulation is based on a standard Lattice Boltzmann approach with an adapted data structure, which also has the flexibility to handle moving boundaries. Additional to velocity and density values, which are the standard output of the Lattice Boltzmann method, the stress tensor is calculated and visualized. Two relevant geometries are simulated and the results are shown and discussed. Furthermore, the costs regarding memory and computational power are investigated.

Zusammenfassung

Sowohl für Männer, als auch für Frauen sind kardiovaskuläre Krankheiten in Deutschland die häufigste Todesursache. Diese Krankheiten können heutzutage mit modernen Diagnoseverfahren wie Computertomographie, Digitaler Subtraktionsangiographie oder Kernspintomographie festgestellt werden. Diese Verfahren ermöglichen den Ärzten auf minimal invasive Weise einen Blick in den menschlichen Körper mit einer millimetergenauen Auflösung. Allerdings können hämodynamische Werte wie der Druck auf die Wand oder die Spannungen im Blut nur schwierig oder gar nicht dargestellt werden. Für die weitere Behandlung des Patienten können diese Größen allerdings eine entscheidende Rolle spielen. Das Risiko eines Aneurysmarisses kann mit heutigen Methoden niemand genau einschätzen. Die hämodynamischen Informationen aus einer Blutflusssimulation könnten den Ärzten Zusatzinformationen liefern um eine genauere Prognose des Risikos treffen zu können. Dadurch könnte auch die Notwendigkeit einer Operation besser eingeschätzt werden.

Die Lattice-Boltzmann-Methode ist für ihre Effizienz bei der Lösung von Strömungsproblemen und besonders bei der Berechnung von Strömungen in komplexen Geometrien bekannt. Daher scheint sie sich hervorragend zur Blutflusssimulation im menschlichen Gehirn zu eignen. Der hohe Speicheraufwand großer Simulationsgebiete macht solche Simulationen heutzutage allerdings nur auf Supercomputern möglich. In dieser Arbeit wird ein Verfahren vorgestellt, das den Speicherbedarf dieser Berechnungen erheblich verringert und somit auch Simulationen dieser Art auf einfachen Rechnern ermöglicht. Alle nötigen Schritte angefangen bei der Datenbeschaffung über die Berechnung bis zur Visualisierung werden ausführlich erläutert. Die Simulation basiert auf einem gewöhnlichen Lattice-Boltzmann-Verfahren, das auf die besondere Geometrie angepasst wurde. Die angepasste Datenstruktur hat zudem die Flexibilität später bewegliche Wände behandeln zu können. Neben Geschwindigkeiten und Dichte – die Standardwerte der Lattice-Boltzmann-Methode – kann auch der Spannungstensor berechnet werden. Die Simulationsergebnisse von zwei medizinisch interessanten Geometrien werden präsentiert und diskutiert. Außerdem werden der Speicherbedarf und der Zeitbedarf der Simulationen behandelt.

Contents

Nomenclature

Abbreviations

α	Womersley Number
ACA	Anterior Cerebral Artery
CCA	Common Carotid Artery
CT	Computer Tomography
CTA	Computer Tomography Angiography
$D2Q9$	2-dimensional LBM model with 9 velocities
$D3Q15$	3-dimensional LBM model with 15 velocities
$D3Q19$	3-dimensional LBM model with 19 velocities
$D3Q27$	3-dimensional LBM model with 27 velocities
DSA	Digital Subtraction Angiography
ECA	External Carotid Artery
FDM	Finite Difference Method
FEM	Finite Element Method
FVM	Finite Volume Method
ICA	Internal Carotid Artery
LBM	Lattice Boltzmann Method
$LGCA$	Lattice Gas Cellular Automata
Ma	Mach Number
MCA	Middle Cerebral Artery
$MFLUPS$	Mega Fluid Lattice Updates per Second
MRI	Magnetic Resonance Imaging
NMR	Nuclear Magnetic Resonance
NS	Navier Stokes
POI	Point of Interest
Re	Reynolds Number
VA	Vertebral Artery

Scalar

$\tilde{f_\alpha}$	post collision particle distribution function
δt	physical time step $[s]$
δt^*	lattice time step (nondimensional)
δx	physical cell size $[m]$
δx^*	lattice cell size (nondimensional)
η	dynamic viscosity $[mPa \cdot s]$
λ	relaxation time $[s]$
ν	kinematic viscosity $[m^2/s]$
ν^*	lattice viscosity (nondimensional)
ω	angular frequency $[1/s]$
ρ	fluid density $[kg/m^3]$
ρ^*	lattice fluid density (nondimensional)
τ	lattice relaxation time (nondimensional)
ξ_α	particle velocity in discrete velocity space
c	lattice speed (nondimensional)
$f_\alpha^{(eq)}$	equilibrium distribution function in discrete velocity space

f_α	particle distribution function in discrete velocity space
P	fluid pressure $[kg/ms^2]$
w_α	weighing factor, depending on the used model

Tensor

$\boldsymbol{\delta}_{ij}$	$i = 1, 2, 3,\ j = 1, 2, 3$	identity tensor
$\mathbf{\Pi}_{ij}$	$i = 1, 2, 3,\ j = 1, 2, 3$	momentum flux tensor
$\boldsymbol{\sigma}_{ij}$	$i = 1, 2, 3,\ j = 1, 2, 3$	stress tensor
$\boldsymbol{\tau}_{ij}$	$i = 1, 2, 3,\ j = 1, 2, 3$	viscous stress tensor
$\mathbf{S}_{ij}$	$i = 1, 2, 3,\ j = 1, 2, 3$	strain rate tensor

Vector

$\boldsymbol{\xi}$	particle velocity vector	
$\mathbf{e}_\alpha$	discrete particle velocity	
$\mathbf{f}$	single fluid particle mass distribution function	
$\mathbf{f}^{(eq)}$	equilibrium distribution function	
$\mathbf{g}_j$	$j = 1, 2, 3$	gravitation forces $[N]$
$\mathbf{J}_j$	$j = 1, 2, 3$	momentum forces $[N]$
$\mathbf{M}_j$	$j = 1, 2, 3$	mass forces $[N]$
$\mathbf{n}$	normal vector	
$\mathbf{O}_j$	$j = 1, 2, 3$	surface forces $[N]$
$\mathbf{u}$	fluid velocity $[m/s]$	
$\mathbf{u}^*$	lattice fluid velocity (nondimensional)	
$\mathbf{U}_j$	$j = 1, 2, 3$	velocity $[m/s]$
$\mathbf{x}_j$	$j = 1, 2, 3$	direction

Note: In this thesis, every value in the parameterized lattice space is marked with *. Only the lattice relaxation time τ and the time t are not specifically marked.

1 Introduction

1.1 Motivation

Noncommunicable diseases are causing a great deal of morbidity and are replacing communicable diseases as the main causes of death in every developed nation. Relative to Eur-A[1] averages, the largest excess mortality in Germany is due to cardiovascular diseases among both males and females [fE06].
Angiography examinations remain the method of choice for vascular diagnostics. During an angiography, a doctor inserts a thin tube (catheter) into the artery through a small nick in the skin about the size of the tip of a pencil. A substance called a contrast agent (X-ray dye) is injected to make the blood vessels visible on the X-ray. Angiography allows to diagnose arteriosclerotic vessel degeneration, for example a stenosis, an aneurysm or a vessel occlusion. However, with angiography only a time dependent flow profile can be visualized, but no additional flow information like the pressure on the wall or the stress can be measured. But this information could be essential for further treatment of the patient. A large aneurysm, for example, could be stable and would not rupture, while a small aneurysm could rupture, due to the great pressure against the wall. Therefore a surgery is necessary, or not. Currently, no one can predict which aneurysms will rupture. Data suggest that aneurysms that rupture tend to be larger than 7–10 mm [Eme06, Cer05]. Some years ago doctors decided the necessity of surgery depending on the diameter of the aneurysm. In today's medical care each aneurysm which is larger than 2.5 mm is treated. However, angiography in conjunction with a simulation of hemodynamics could give the doctor additional information and help to diagnose and correctly treat these diseases.

In the field of numerical fluid dynamics, later called computational fluid dynamics (CFD), several methods have been developed since the beginning of the 50s. The *finite element method*, the *finite volume method* and the *Fourier method* were widely used, just to mention some of them. With the development of computer power, these methods were then ready to solve complex flows in complex geometries.
In the last two decades another method was developed and became an attractive alternative to conventional computational fluid dynamics solvers. This method is called the Lattice Boltzmann method (LBM) and has its origin in the Lattice Gas Automata (LGA). The LBM competes with the traditional solvers in directly obtaining the pressure without solving the Poisson equation and getting the shear stress tensor without using the velocity gradients. Additionally the Lattice Boltzmann method seems to be easy to implement, easy to parallelize and a suitable method for handling complex geometries. Unfortunately a adequate fine standard brain geometry, e.g. with $700 \cdot 700 \cdot 700$ cells, would result in memory requirements of more than $100GB$ in a standard LBM solver. This is the motivation to test the capability of the Lattice Boltzmann method in combination with a modified data structure which reduces the memory requirements in the case of a hemodynamic simulation.

1.2 The Doctoral View

To give a brief introduction in the bloodflow of the brain, some important arteries in the human brain are shown in Figure 1.1 (from [Bee05]). The carotid artery and the vertebral artery are the vessels that supply the circle of Willis with blood. This circle provides the brain with oxygen and nutrients and creates redundancies in the cerebral circulation. There are two vertebral and two carotid arteries, in each case

[1] including Belgium, Denmark, Finland, France, Germany, Ireland, Israel, Italy, Norway, UK

one on the right and one on the left side of the body. The common carotid arteries bifurcate into the internal carotid artery (ICA) and the external carotid artery (ECA). The ICA supplies the brain, whereas the ECA supplies the neck and the face. The vertebral artery connects to the basilar artery, which is the main blood supply to the brain stem and also connects to the circle of Willis.

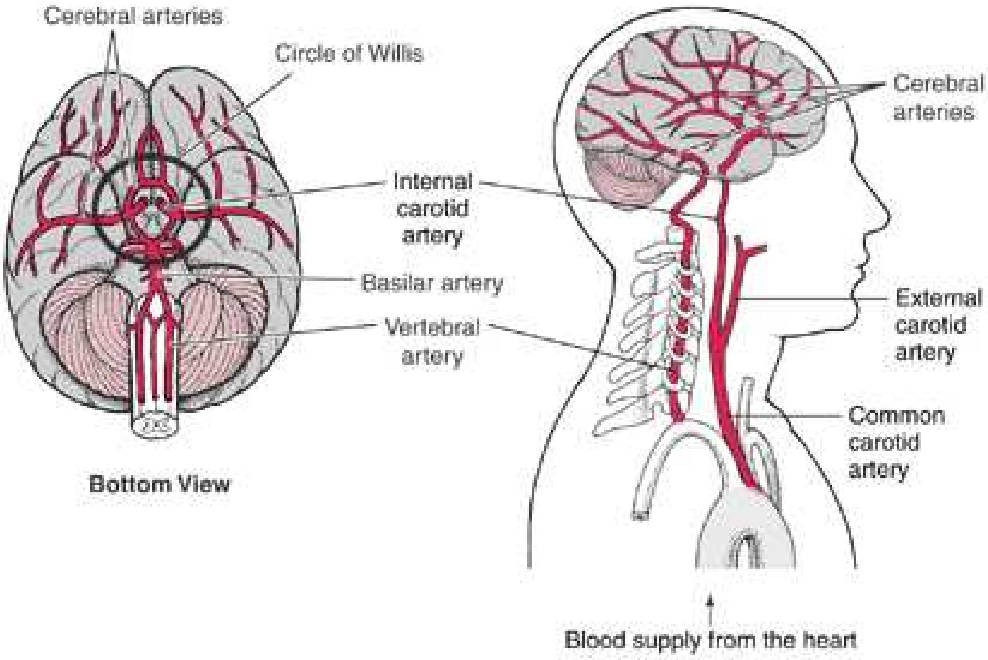


Figure 1.1: The cerebral arteries (from [Bee05])

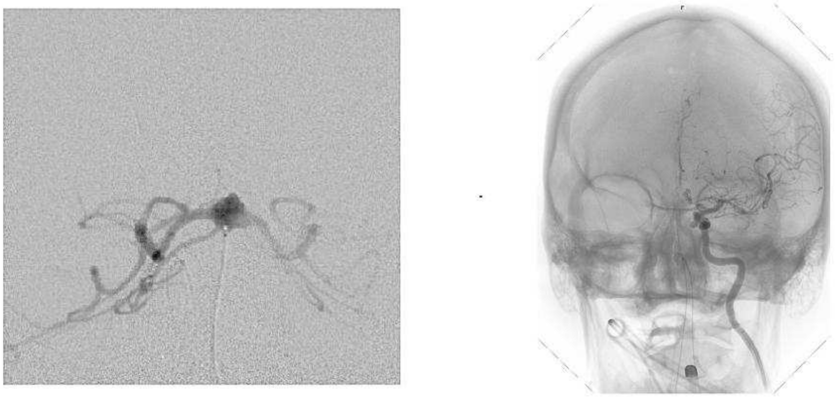
1.2.1 Modern Diagnose Techniques

Some of the latest diagnose techniques which are used in today's medical care are:

- **Computer Tomography** (CT)
  Computer Tomography uses X-rays to display different parts of the body. CT scans are an excellent method for detecting bleeding into the brain and fluid spaces surrounding the brain. A large series of $2D$ X-ray images (slices) are processed by a computer program and result in a $3D$ image of the internal of the human body [Bra06].
- **Magnetic Resonance Imaging** (MRI)
  The MRI technique produces high quality images of insight of the human body without the use of X-rays. It is based on the principles of nuclear magnetic resonance (NMR), a spectroscopic technique used by scientists to obtain microscopic chemical and physical information about molecules. Latest MRI scanners use a field with a strength of 1.5 Tesla. Just to compare, the magnetic field of the earth at the equator is $3.1 \cdot 10^{-5}$ Tesla.
- **Digital Subtraction Angiography** (DSA)
  DSA is an imaging technique which uses X-rays in connection with a contrast agent. In a preprocessing step a standard X-ray series is taken. Then the contrast dye is injected via a catheter and in a second step another series is taken. These images are subtracted from each other, resulting in very accurate pictures of the blood vessel system itself.

- **Computer Tomography Angiography** (CTA)
  Computer Tomography Angiography is a technique combining CT and angiography. Compared to standard catheter angiography, the benefits are the less invasive treatment, the lower costs and the less time needed. This technique produces acceptable pictures, but is not as accurate as standard DSA.

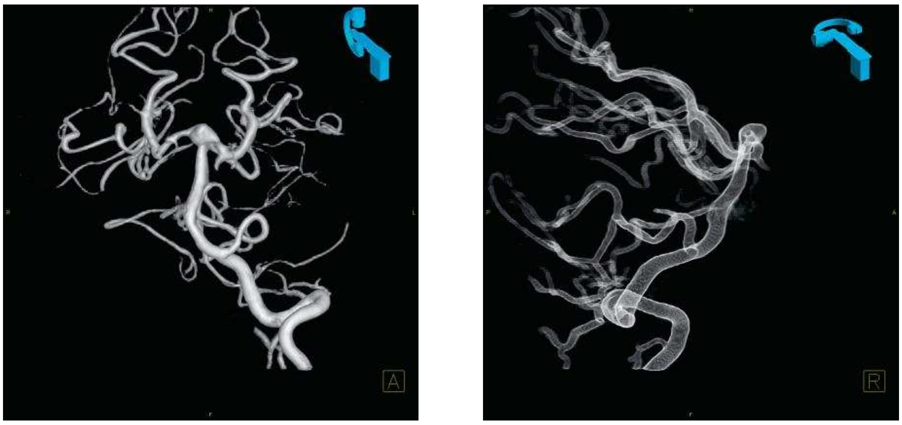
Invasive catheter angiography remains the "gold standard" for detecting an aneurysm causing hemorrhage in the brain. Some pictures of DSA are shown in Figure 1.2 and 1.3 (from [DSA06]). However, recent advances in CT angiography, MR angiography, and computer postprocessing have improved less invasive detection of intracranial aneurysms. These modalities are used for screening in patients with a positive family history for aneurysms or known risk factors [LM98].



(a) Selective DSA: Display of a basilar tip aneurysm

(b) DSA left internal carotid artery: Post-surgery of aneurysm of the anterior communicating artery

Figure 1.2: Examples of DSA (from [DSA06])



(a) $3D$ view of left vertebral artery (anterior): Display of a basilar aneurysm

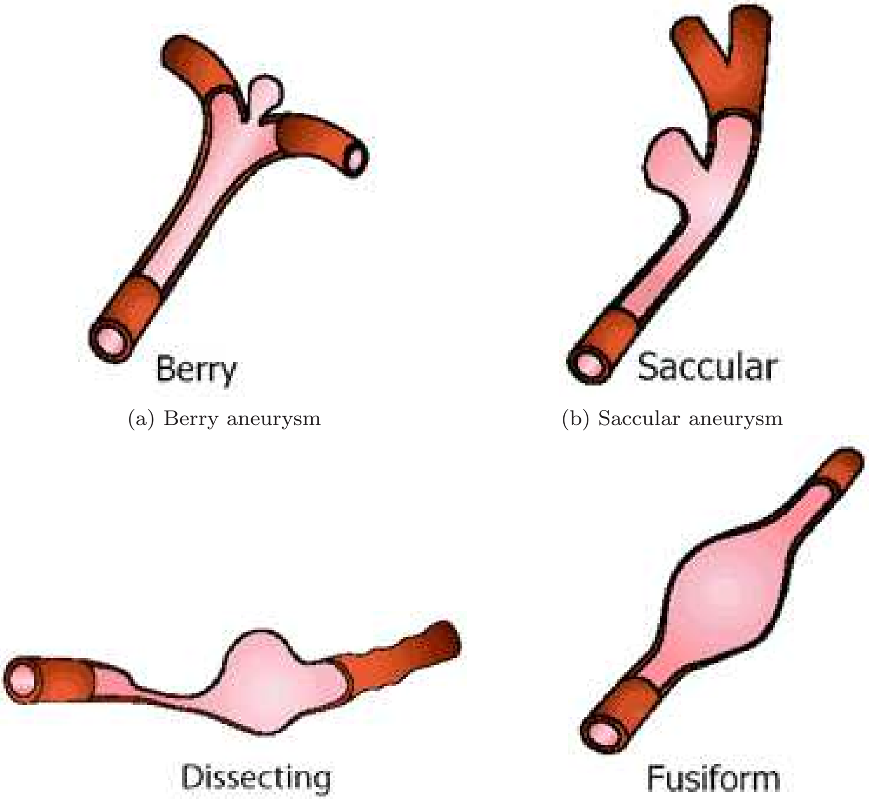
(b) $3D$ view of left vertebral artery (lateral): Display of a basilar aneurysm

Figure 1.3: $3D$ views of left vertebral artery (from [DSA06])

1.2.2 Aneurysms

Aneurysms are local dilatations (baloonings) of the vessel wall and most commonly occur in the base of the brain (cerebral aneurysm), or near the aorta (aortic aneurysm). These aneurysms most often are caused by arteriosclerotic diseases, infections, head injuries, alcohol and smoke abuse, or are born in (e.g. marfan syndrome) [Dio03]. The big danger of an aneurysm is the rupturing and the bleeding into the brain. This most often results in a stroke. Strokes are the number three cause of death in a developed nation after cardiovascular diseases and tumors. The number of deaths however, is low compared to the patients surviving the stroke with a physical or mental handicap. In postmortem examinations of persons not suffering from neurological problems, it is reported, that up to 5% of the tested persons have an unruptured aneurysm [DE06].

The different types of aneurysms are shown in Figure 1.4 (from [Bra06]). The most important ones are berry, saccular, dissecting and fusiform aneurysms. In the brain, most often the berry aneurysm occurs near a branching of the vessel (called bifurcation).



(a) Berry aneurysm

(b) Saccular aneurysm

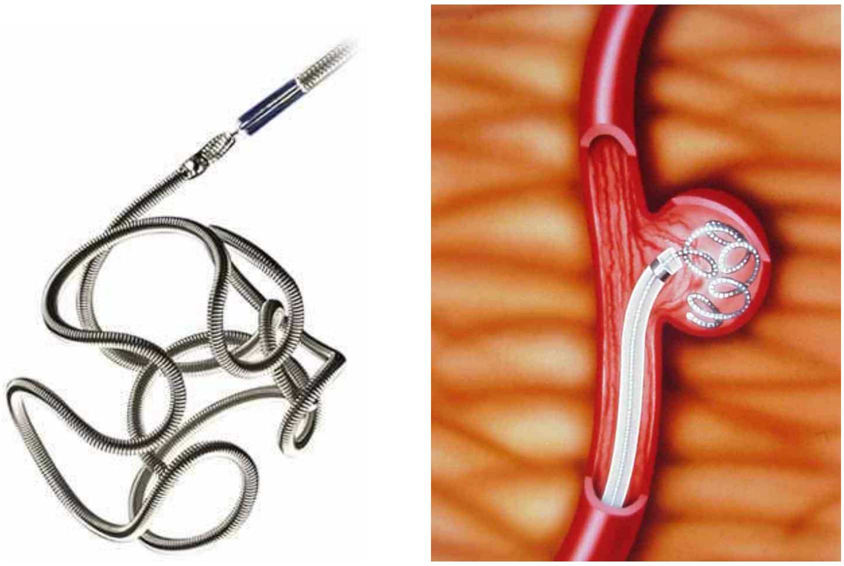
(c) Dissecting aneurysm

(d) Fusiform aneurysm

Figure 1.4: Different types of aneurysms (from [Bra06])

The cerebral aneurysm can be treated by clipping, which means that in an invasive surgery a titanium clip is affixed around the neck of the aneurysm. Another possible treatment is the coiling method. This method was first established by Dr. Guido Guglielmi in the year 1990 [Dio03]. Coiling means inserting a coil into the aneurysm (see Figure 1.5b from [Coi05, Dio03]) with a microcatheter, which is inserted in the groin. The blood is then agglutinating and filling up the hole. This is a minimal invasive treatment. In a recent large prospective study of 2,143 patients with ruptured aneurysms who could equally be treated with clipping or coiling it was shown that the coiled patients suffered significantly less death and dependency as compared to the clipped patients [Dio03]. Thus in modern medical care most aneurysms in the brain are treated with coiling.

Table 1.1 (from [Jan06]) shows where aneurysms occur in the brain. Most of them can be found in the ICA.



(a) $3D$ coil (from [Coi05])

(b) Coiling procedure (from [Dio03])

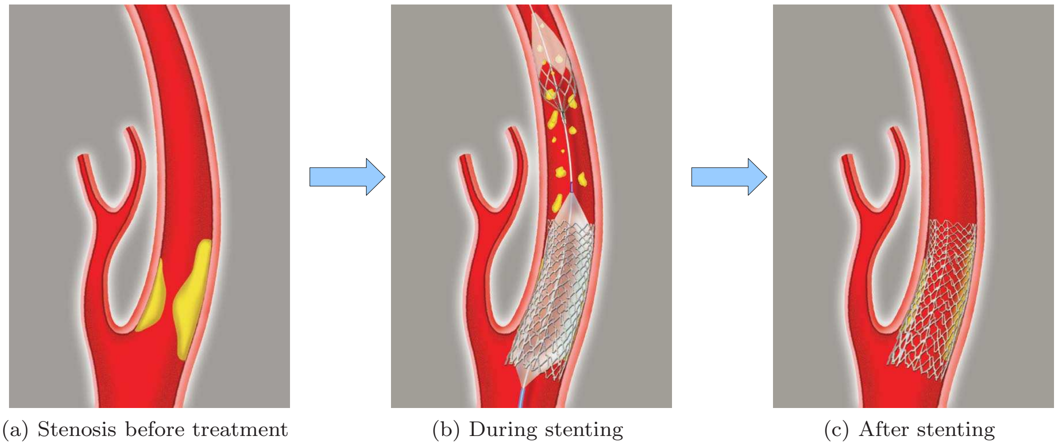
Figure 1.5: Coiling

LOCATION	%
internal carotid artery (ICA)	$35 - 40\%$
anterior cerebral artery (ACA)	*about* 33%
middle cerebral artery (MCA)	$> 20\%$
vertebral artery (VA)	$5 - 10\%$

Table 1.1: Location of aneurysms
[Jan06]

1.2.3 Stenosis

A stenosis is an abnormal narrowing of the vessel wall. Patients with high grade stenosis of the carotid artery suffer a high risk of stroke. In Germany about 15% of the strokes are cause by a stenosis of the carotid artery [Ste06]. About half a million people are living with a stenosis of the carotid, which is constricted more than 50% compared to natural blood vessels. This is most often caused by an arteriosclerotic disease. Calcium and fat particles absorb at the vessel wall resulting in an undersupply of oxygen to the brain. The biggest danger applies, if the agglomerates disengage the wall and close important vessels. Then the part of the vessel behind the occlusion is not being supplied with blood any more - it comes to a stroke. A modern way to treat such a stenosis is a stenting of the narrowed wall. This means a stent is inserted with a catheter to the stenosis and dilates it. A special protection system helps to avoid embolic stroke during the procedure. The treatment is shown in figure 1.6.



(a) Stenosis before treatment

(b) During stenting

(c) After stenting

Figure 1.6: Stenting procedure (from [Ste06])

1.3 Structure of the Thesis

This thesis is structured in six main chapters. After this preliminary chapter some mathematical and physical basics are given. The characteristics of blood are mentioned and the basic equations are derived and discussed. Then a chapter about the Lattice Boltzmann method follows. This chapter is meant for people, who have never dealt with this method. Since only a standard LBM is used, an experienced reader can skip this chapter, but should read the section about parametrization. The fourth chapter deals with the simulation itself. Not only is the section Data Layout essential for the thesis, but also pre- and postprocessing is discussed here in detail. The fifth chapter will present the results in regards to performance and memory issues including pictures of pressure forces, velocities, shear forces and particle tracing. Finally, chapter six concludes the work and gives an outlook on future developments.

2 Mathematical and Physical Background

In this chapter mathematical and physical basics are introduced, involving the special features of blood, the equations which have to be solved and the simplifications. Furthermore, the specific simulation parameters and the stress tensor are discussed.

2.1 Blood

Blood is a complex mixture of cells, proteins, lipoproteins and ions by which nutrients and wastes are transported [Ku97]. In an adult man, the blood is about $1/12th$ of the body weight and this corresponds to $5-6$ liters. Blood consists of 55% plasma and 45% cells called formed elements. In 1 μl there are about $4.5-5.5$ millions of red blood cells, which comprise approximately 89% volume of all blood cells. These red blood cells are small semisolid particles and thus influence the behavior of the fluid drastically. This is discussed in the next paragraphs.

2.1.1 Viscosity

One of the most important mechanical properties for the motion of blood is the dynamic viscosity η. Viscosity describes the tendency of a fluid to resist flow. A well known high viscous fluid is for example honey. The kinematic viscosity ν of a fluid is defined to be the ratio between the dynamic viscosity and the density, so $\nu = \eta/\rho$.
At low flow rates the red blood cells stick together, increasing viscosity, and remain in the center of the vessel. The blood closest to the vessel wall (which supplies side branches) therefore has a lower hematocrit[1]. This process is known as plasma skimming [Rog99]. High flow rates do not allow particles to aggregate, in this velocity range the viscosity is low.

A distinction is drawn between Newtonian (e.g. water) and non-Newtonian fluids. Newtonian fluids comply with Newton's law of viscosity

$$\tau = \eta \frac{\partial v}{\partial x}, \tag{2.1}$$

where τ is called the shear stress and $\partial v/\partial x$ is the velocity gradient perpendicular to the direction of shear. Non-Newtonian fluids can be defined as those for which viscosity is not a constant, but varies based upon the rate of shear or the shear stress at which it is measured. In this respect, blood is a non-Newtonian fluid and approximately 3-4 times more viscous than water.

2.1.2 Viscoelasticity

The first models considered blood as a viscous fluid, assuming that the viscous effects dominate its flow properties. Since blood is a mixture of different cells and plasma it is also known that other effects should be taken into account, depending on the flow parameters (see Figure 2.1). In 1972 its was for the first time measured, that blood has viscoelastic properties [Ela00]. The red blood cells are mainly responsible for the viscoelastic effects. When blood is in rest, these cells tend to stack together to so called agglomerates.

[1] proportion of blood volume that is occupied by red blood cells

As red blood cells aggregate, blood exhibits a more viscoplastic behavior, indicating the existence of a yield stress as the shear rate approaches 0 sec^{-1}.

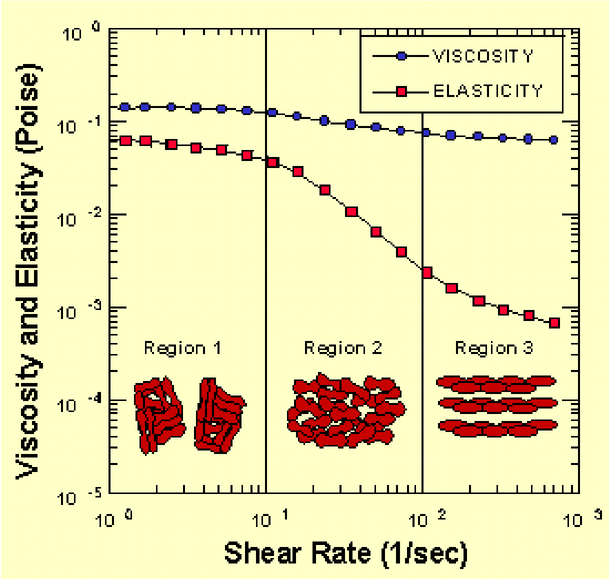


Figure 2.1: The shear rate dependence of normal human blood viscoelasticity at $2Hz$ and $22°C$ (from [Ela00])

2.1.3 Simplifications

Experiments have shown, that in large vessels the shear rate is about 1,000 sec^{-1}. In this range, the elastic behavior of blood becomes insignificant (see Figure 2.1). The viscosity of blood asymptotes to a value of 3-4 $(mPa \cdot s)$ at a temperature of 37 °C. To a fair approximation, the effect of non-Newtonian behavior of blood in arteries in the human brain is small and negligible [Kor03]. Thus in this thesis, blood is considered to be a Newtonian, isotropic, incompressible and homogeneous fluid.

2.2 The Navier Stokes Equations

The macroscopic behavior of a isothermal fluid can be fully described with two equations, known as the conservation of mass and momentum. The law of mass conservation states, that mass can neither be created nor destroyed. Thus, the inflows, outflows and change in storage of mass in a system must be in balance. This leads to the equation of mass continuity:

$$\frac{\partial \rho}{\partial t} + \frac{\partial (\rho \mathbf{U}_i)}{\partial \mathbf{x}_i} = 0 \,, \tag{2.2}$$

where ρ is the density, t is the time and $\mathbf{U}_i$ is the velocity in i-direction ($i = 1, 2, 3$ for x,y,z). This leads to

$$\frac{\partial \mathbf{U}_i}{\partial \mathbf{x}_i} = 0 \tag{2.3}$$

for an incompressible fluid ($\rho = constant$).
The derivation of the momentum equation for the three coordinates $j = 1, 2, 3$ is done in fluid mechanics with use of the second law of Newton in a Lagrangian formulation. The Lagrangian formulation is noted

with $\Re$ and describes in contrast to the Eulerian formulation, the behavior of fluid particles and not the field values in the fluid. The change of momentum of one fluid particle is then equal to the sum of all changes, meaning mass forces, surface forces and molecular change of momentum:

$$\frac{d(\delta \mathbf{J}_i)_\Re}{dt} = \underbrace{\sum (\delta \mathbf{M}_j)_\Re}_{mass\ forces} + \underbrace{\sum (\delta \mathbf{O}_j)_\Re}_{surface\ forces} + \underbrace{\left(\frac{d}{dt}(\delta \mathbf{J}_M)_j\right)_\Re}_{molecular\ change\ of\ momentum} \quad . \tag{2.4}$$

To get the Eulerian formulation every value of Equation 2.4 has to be written in field values. Some reformulations [Dur03] result in:

$$\rho \left(\frac{\partial \mathbf{U}_j}{\partial t} + \mathbf{U}_i \frac{\partial \mathbf{U}_j}{\partial \mathbf{x}_i} \right) = \rho \mathbf{g}_j - \frac{\partial P}{\partial \mathbf{x}_j} - \frac{\partial \boldsymbol{\tau}_{ij}}{\partial \mathbf{x}_i} \, , \tag{2.5}$$

where P is the pressure, $\mathbf{g}_j$ is the gravitational vector and $\boldsymbol{\tau}_{ij}$ is called the viscous stress tensor for a compressible fluid. For a newtonial fluid (i.e. the viscosity is independent of the shear rate) $\boldsymbol{\tau}_{ij}$ is defined as

$$\boldsymbol{\tau}_{ij} = -\eta \left[\frac{\partial \mathbf{U}_j}{\partial \mathbf{x}_i} + \frac{\partial \mathbf{U}_i}{\partial \mathbf{x}_j} \right] + \frac{2}{3} \boldsymbol{\delta}_{ij} \eta \frac{\partial \mathbf{U}_k}{\partial \mathbf{x}_k} \tag{2.6}$$

where $\boldsymbol{\delta}_{ij}$ is the identity tensor. From here one gets for an incompressible fluid and with $\eta = constant$ the well known Navier-Stokes equations

$$\frac{\partial \mathbf{U}_i}{\partial \mathbf{x}_i} = 0 \tag{2.7}$$

$$\rho \left(\frac{\partial \mathbf{U}_j}{\partial t} + \mathbf{U}_i \frac{\partial \mathbf{U}_j}{\partial \mathbf{x}_i} \right) = \rho \mathbf{g}_j - \frac{\partial P}{\partial \mathbf{x}_j} + \eta \frac{\partial^2 \mathbf{U}_j}{\partial \mathbf{x}_i^2} \, . \tag{2.8}$$

Combining pressure forces and forces from exchange of molecular momentum and additionally reformulate Equation 2.8 for an incompressible fluid with unit density gives: [RA01]

$$\frac{\partial \mathbf{U}_j}{\partial t} + \mathbf{U}_i \frac{\partial \mathbf{U}_j}{\partial \mathbf{x}_i} = \mathbf{g}_j - \frac{1}{\rho} \frac{\partial P}{\partial \mathbf{x}_j} + \frac{\eta}{\rho} \frac{\partial \boldsymbol{\sigma}_{ij}}{\partial \mathbf{x}_i} \, , \tag{2.9}$$

where $\boldsymbol{\sigma}_{ij}$ is called the stress tensor.

Since the change in temperature in the human body is minor, the energy equation is not used in this thesis and thus not mentioned here. The derivation of the energy equation can be found in [Dur03].

2.3 The Hydrodynamic Stress Tensor

The forces acting on a fluid in motion can be divided into two categories:

- **Long-range forces**: e.g. gravity or an external pressure gradient
- **Short-range forces**: refer to local dynamic stresses developing within the fluid as it moves

For the second category the stress tensor $\boldsymbol{\sigma}_{ij}$ is defined as:

"The i-component of stress acting on an element of surface with unit normal **n** *in the j-direction"*

and was introduced in the previous section. In Figure 2.2 an example is given for the components of the tensor which are relevant at a unit normal perpendicular to the x-y-plane.

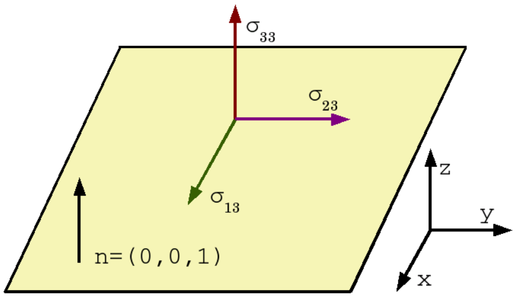


Figure 2.2: Example of three components of the stress tensor

In case of computational hemodynamics it is worth to mention some aspects of the hydrodynamic stress tensor. It is customary to divide the total stress σ_{ij} into an isotropic part and a deviatoric part as in

$$\boldsymbol{\sigma}_{ij} = -P\boldsymbol{\delta}_{ij} + \boldsymbol{\tau}_{ij} \,, \tag{2.10}$$

where $\boldsymbol{\tau}_{ij} = 2\eta\mathbf{S}_{ij}$ is the viscous stress tensor and P is the (mechanical) pressure for an incompressible fluid. The strain rate tensor $\mathbf{S}_{ij}$ is then given by

$$\mathbf{S}_{ij} = \frac{1}{2}\left(\frac{\partial \mathbf{U}_i}{\partial \mathbf{x}_j} + \frac{\partial \mathbf{U}_j}{\partial \mathbf{x}_i}\right) \,. \tag{2.11}$$

In a standard finite element, or finite volume method one would calculate the strain tensor by estimating the derivatives of the velocity gradient from the velocity values. However, the Lattice Boltzmann method obtains the tensor directly, as shown in section 3.1. This is a big advantage of the method, because it reduces the computational amount significantly.

2.4 The Boltzmann Equation

In section 2.2, the Navier-Stokes equations were introduced, which are adequate to describe all macroscopic fluid flow quantities. However, sometimes it is interesting to have a look at smaller scales. The kinetic theory gives an alternative description to model the time evolution of a collection of particles. This equation is known as the Boltzmann equation

$$\frac{\partial \mathbf{f}}{\partial t} + \boldsymbol{\xi}\cdot\frac{\partial \mathbf{f}}{\partial \mathbf{x}} + \mathbf{G}\cdot\frac{\partial \mathbf{f}}{\partial \boldsymbol{\xi}} = Q(\mathbf{f},\mathbf{f}) \,, \tag{2.12}$$

where $\mathbf{f} = \mathbf{f}(\mathbf{x}, \boldsymbol{\xi}, t)$ is a distribution function, $\boldsymbol{\xi}$ is the velocity vector of a molecule, $\mathbf{x}$ is the position of the molecule, $\mathbf{G}$ is the force per unit mass acting on the molecule, t describes the time and the term on the right hand side describes the quadratic collision operator. The first term on the left hand side describes a explicit time variation of the distribution function, the second term gives the spatial variation and the third term describes the effect of any force acting on the particle.

The Boltzmann equation can be solved by different perturbation approaches, for example the Chapman Enskog expansion. With the Chapman Enskog technique the Navier-Stokes equations can be derived from the Boltzmann equation (see [Bui97]). In this thesis the Boltzmann equation is used to solve the fluid flow in the human brain. The discretization of the equation is shown in chapter 3.

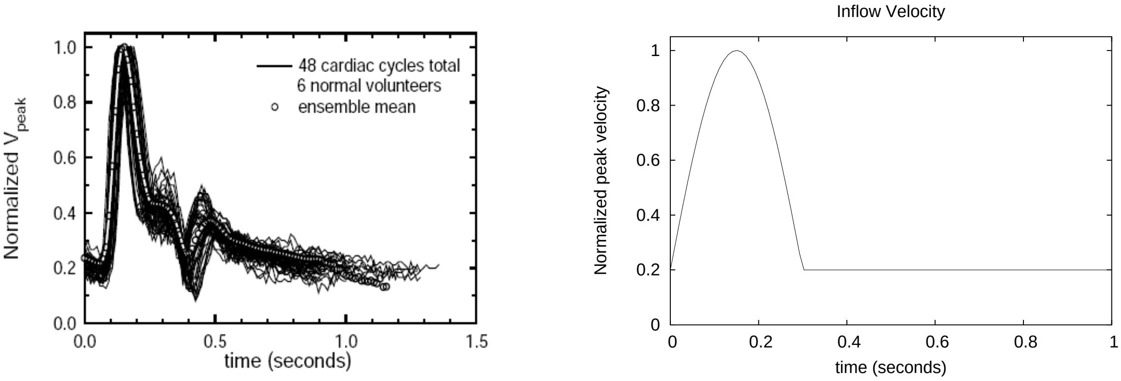
2.5 Specifying Simulation Parameters

2.5.1 Flow Rate

The flow rate is a patient specific variable and thus not easy to include in the simulation. To simplify things, an averaged inflow velocity is used, depending on the artery (see table 2.1, measured with Laser Doppler Anemometry by Ruge [Rug03]). The time dependency of a real patient inflow profile is approximated using a combination of a sin^2 pulse and a fixed velocity at the inflow (see Figure 2.3).

Vessel	Vs	TAV	d
	(cm/s)	(cm/s)	(mm)
common carotid artery (CCA)	96 ± 22	23 ± 5	6.2 ± 0.8
external carotid artery (ECA)	84 ± 22	20 ± 6	4.2 ± 0.7
internal carotid artery (ICA)	64 ± 14	24 ± 5	4.8 ± 0.7
vertebral artery (VA)	48 ± 10	14 ± 4	3.4 ± 0.4

Table 2.1: Velocities and diameters including standard deviation of different arteries: Velocities and diameters averaged from measurements with 156 persons of $20 - 85$ years age taken with Laser-Doppler-anemometry (TAV means time-averaged mean velocity)



(a) Measured bloodflow in carotid artery (from [HNF+99])

(b) Used flow profile

Figure 2.3: Measured and used flow profiles

2.5.2 Viscosity

The viscosity is also patient specific but in a small range of 3-4 $mPa \cdot s$ (see 2.1.3). Furthermore it is quite sensitive to parameters like hematocrit and temperature. At 37 °C and a hematocrit of 45% the value for blood is acceptably taken as 4.0 $mPa \cdot s$ (see [Art03]).

2.5.3 Density

The density of blood is depending on the hematocrit and differs from patient to patient. In this thesis a density value of 1,100 $\frac{kg}{m^3}$ is used.

2.5.4 Dimensionless Numbers

In the previous equations all quantities were used with their dedicated standard dimensions. In order to see direct dependency on single values, the following dimensionless numbers are used:

- **Reynolds Number** (Re)
  The Reynolds number describes the ratio between convective inertial and viscous forces and is relevant in every flow situation. It is defined as

  $$Re = \frac{\rho U l}{\eta}\,, \tag{2.13}$$

  where ρ is the density, U is the maximum velocity, l is a characteristic length (e.g. equal to diameter 2r if a cross-section is circular) and η is the dynamic viscosity. For low Reynolds numbers the viscous forces are dominant, where for high Reynolds numbers the inertial forces are more relevant. Thus the Reynolds number is an indicator how far the fluid field is from turbulence.

- **Mach Number** (Ma)
  The Mach number is only relevant in compressible flows and characterizes the rate of the flow velocity and the speed of sound

  $$Ma = \frac{U}{c}\,. \tag{2.14}$$

  As the Lattice Boltzmann method is a "semi compressible method", the Mach number is used to guarantee the validity of the method.

- The third dimensionless number is based on the relationship between transient inertial forces and viscous forces. This relationship is called the **Womersley Number**(α)

  $$\alpha = \frac{D}{2}\sqrt{\frac{\omega}{\nu}}\,. \tag{2.15}$$

  The Womersley number, is derived from the Stokes number

  $$St = \frac{D^2}{4}\frac{\omega}{\nu}\,. \tag{2.16}$$

  Here, D is the diameter of the vessel, ω is the angular frequency of pulsatile flow and ν is the kinematic viscosity term. Typical values in the circulation are $\alpha = 16$ in the abdominal aorta and $\alpha = 9$ in the carotid artery under resting conditions [Art03]. The larger the Womersley number, the greater the importance of the transient inertial force as compared to the viscous force.

2.6 Fluid-Structure Interaction

The wall of a brain artery is complex and comprised of three major layers. The first layer of these arteries near the hollow space (lumen) is the intima, followed by the media and the adventitia (the outermost layer of the wall). Between the intima and the media is a thin layer of elastic tissue. This layer is the only elastic layer occurring in the wall of a brain artery [Khu06]. Because of the elasticity, the diameter of the wall varies with the pulsating pressure. Additionally, the wall propagates pressure and flow waves generated by the heart.

Fluid Structure Interaction (FSI) is a challenge in fluid dynamics, especially in Lattice Boltzmann methods. FSI is for example used, if the walls of the simulation domain are not stiff, but are moving. This field is not very sophisticated and a lot of groups are trying to develop essential models (e.g. [GTK06]). New imaging techniques such as magnetic resonance elastography may allow to gain accurate parameters for elasticity properties of the vessel wall. This motivates more and more complex simulation methods, which include elasticity models in the simulation. However, in large arteries as in the main brain vessels, a maximum change of 10% in diameter is expected, resulting in the same change in the Womersley parameter and the Reynolds number. Therefore for the sake of simplicity, rigid wall assumption is used for the walls.

2.7 Matrix Invariants

The stress tensor in each point of the fluid is a $3 \cdot 3$ matrix and thus difficult to visualize in a $3D$ coordinate system. An alternative is to calculate a scalar value, related to the tensor. A scalar value, which is independent of the coordinate system the matrix is written in, is known as a matrix invariant. Considering the matrix

$$\begin{pmatrix} a_{11} & a_{12} & a_{13} \\ a_{21} & a_{22} & a_{23} \\ a_{31} & a_{32} & a_{33} \end{pmatrix}, \tag{2.17}$$

the three simplest invariants are:

- The first invariant I_1 is the trace of the matrix $tr = a_{11} + a_{22} + a_{33}$
- The second invariant $I_2 = \begin{vmatrix} a_{22} & a_{32} \\ a_{23} & a_{33} \end{vmatrix} + \begin{vmatrix} a_{11} & a_{21} \\ a_{12} & a_{22} \end{vmatrix} + \begin{vmatrix} a_{11} & a_{31} \\ a_{13} & a_{33} \end{vmatrix}$
- The third invariant $I_3 = det(I)$

These three invariants are the simplest possible linear, quadratic, and cubic combinations of the eigenvalues that do not depend on their ordering. In structural mechanics and fluid mechanics other combinations of the eigenvalues are also known and used.

3 The Lattice Boltzmann Method

3.1 Introduction to the Lattice Boltzmann Method

In the last decades a lot of research in the field of *computational fluid dynamics* (CFD) has been conducted. Useful packages for the simulation of fluids have been released. Most of these packages deal with the solution of the Navier Stokes equations, a set of Partial Differential Equations (PDE) of second order (see Chapter 2.2). Usually the set of a momentum equation and continuity equation, which guaranties mass conservation, is used, which is discretized in time and space. This can be done by different methods, for example *finite element methods* (FEM), *finite volume methods* (FVM), or *finite difference methods* (FDM). One drawback of this approach is the discretization of the nonlinear convective term $\mathbf{U}_i \frac{\partial \mathbf{U}_j}{\partial \mathbf{x}_i}$, which makes the numerical treatment difficult and computational expensive.
An alternative approach, called Lattice Boltzmann method (LBM), was introduced in 1988 by McNamara and Zanetti and also Higuera and Jimenez [Art03]. It is based on the original Lattice Gas Cellular Automata (LGCA) . The LGCA divides time and space into steps to form a lattice and discretizes the fluid as particles, which are positioned at certain points in space, called lattice cells. These fluid particles are only allowed to move into discrete directions, depending on the type of the LBM model. The particles collide with each other on lattice cells according to carefully designed collision rules that conserve the number of particles and momentum. The coarse grained fluid variables, such as density and fluid velocity, can be shown to comply with a set of macroscopic equations that approximate the continuity and Navier-Stokes equations in appropriate limits. Instead of dealing with individual particles, the LBM focuses on the averaged macroscopic behaviors. In the original LGCA model the occupations were modeled only boolean. The idea of the LBM was simply to replace the boolean LGCA occupation number with ensemble averaged populations:

$$\frac{\partial \mathbf{f}}{\partial t} + \boldsymbol{\xi} \cdot \nabla \mathbf{f} = Q_\alpha(f_\alpha)\,, \tag{3.1}$$

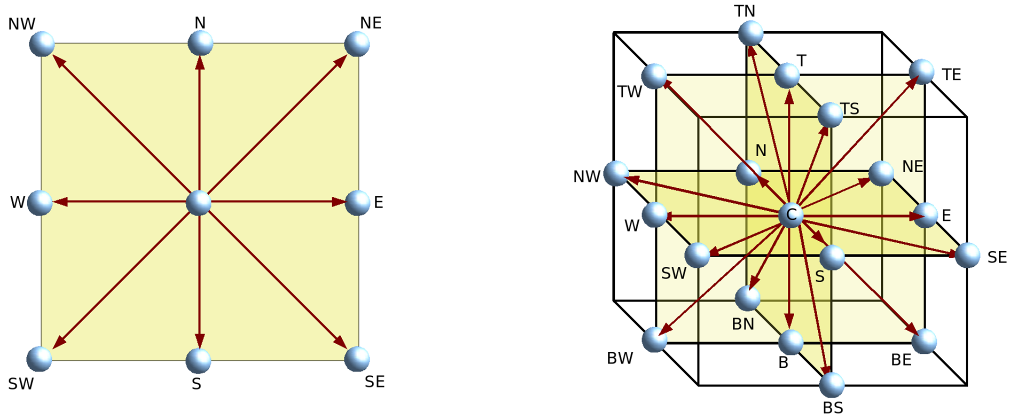
where $\mathbf{f}$ describes the distribution function and $\boldsymbol{\xi}$ the velocity vector of the particles. The term on the right hand side is the collision term. The most popular model for discretizing the collision term with single relaxation time was developed in 1992 by Bhatnagar, Gross and Krook (BGK model) and only uses first derivatives

$$\frac{\partial \mathbf{f}}{\partial t} + \boldsymbol{\xi} \cdot \nabla \mathbf{f} = \frac{1}{\lambda}(\mathbf{f} - \mathbf{f}^{(eq)})\,. \tag{3.2}$$

This model describes the evolution of a single particle distribution function f for particles that move with a microscopic velocity $\boldsymbol{\xi}$ and collide with relaxation time λ till they relax to the Boltzmann-Maxwellian equilibrium distribution function f^0. The relaxation time depends on the fluid viscosity. To obtain a solution from Equation 3.2, the velocity is discretized by a finite set of velocities $\xi_\alpha, \alpha = 0, 1, \cdots N$, resulting in:

$$\frac{\partial f_\alpha}{\partial t} + \xi_\alpha \cdot \nabla f_\alpha = \frac{1}{\lambda}(f_\alpha - f_\alpha^{(eq)})\,. \tag{3.3}$$

The simplest velocity model in 2D is the D2Q9 model shown in Figure 3.1a with 9 discrete velocity directions. For the three-dimensional problem, several velocity models have been proposed, including the fifteen velocity model (D3Q15), the nineteen velocity model (D3Q19), and the twenty-seven velocity model (D3Q27). The D3Q19 model is a good compromise in terms of stability and computational efficiency [MSYL00] and thus used in this thesis.



(a) D2Q9 model for two dimensions

(b) D3Q19 model for three dimensions

Figure 3.1: Models for two and three dimensions

The velocity space in the $D3Q19$ model is discretized in 19 distribution functions. Each cell contains one stationary velocity in the center for the particles at rest, six velocity directions along the Cartesian axes and 12 velocities combining two coordinate directions (see Figure 3.1b). Thus the fluid particles are only allowed to move along these 18 velocity directions, or stay at rest.
The equilibrium distribution for all models is of the form

$$f_\alpha^{(eq)} = \rho^* \cdot w_\alpha \cdot [1 + \frac{3}{c^2}\mathbf{e}_\alpha \cdot \mathbf{u}^* + \frac{9}{2c^4}(\mathbf{e}_\alpha \cdot \mathbf{u}^*)^2 - \frac{3}{2c^2} \cdot \mathbf{u}^* \cdot \mathbf{u}^*] \quad , a = 1, \cdots N \, , \tag{3.4}$$

where w_α is a weighting factor, N is the number of directions and $\mathbf{e}_\alpha$ is the given velocity vector and can be obtained by Taylor expansion of the Maxwellian distribution. w_α and $\mathbf{e}_\alpha$ are dependent on the given model. For the D3Q19 model the weights are

$$w_\alpha = \begin{cases} 1/3 & \alpha = C \\ 1/18 & \alpha = N, S, E, W, T, B \\ 1/36 & \alpha = NW, NE, SW, SE, TN, TS, TE, TW, BN, BS, BE, BW \end{cases} \quad ,$$

where $\alpha = 0, \cdots 18$ denotes the direction of the discrete velocity vector. To improve the readability, the numbers $0, \cdots 18$ are replaced by directional abbreviations, which consist of E (east), W (west), S (south), N (north), T (top) and B (bottom) and all relevant combinations of these abbreviations. The velocity vectors themselves are given by

$$\mathbf{e}_C = c(0, 0, 0)$$

$$\mathbf{e}_{T,B}, \mathbf{e}_{N,S}, \mathbf{e}_{E,W} = c(\pm 1, 0, 0), c(0, \pm 1, 0), c(0, 0, \pm 1)$$

$$\mathbf{e}_{TS,TN,BS,BN}, \mathbf{e}_{NE,NW,SE,SW}, \mathbf{e}_{TE,TW,BE,BW} = c(\pm 1, \pm 1, 0), c(0, \pm 1, \pm 1), c(\pm 1, 0, \pm 1) \, ,$$

where $c = \Delta x^* / \Delta t^*$ is the lattice speed with the lattice cell size Δx^* and the lattice time step Δt^*.
The hydrodynamic density value ρ^* and the velocity values $\mathbf{u}^*$ from Equation 3.4 are determined in terms of the particle distribution functions from:

$$\rho^* = \sum_\alpha f_\alpha \tag{3.5}$$

and

$$\rho^* \mathbf{u}^* = \sum_\alpha \mathbf{e}_\alpha f_\alpha \,. \tag{3.6}$$

Equation 3.3 is then discretized in space and time into the well-known lattice BGK equation [Art03]

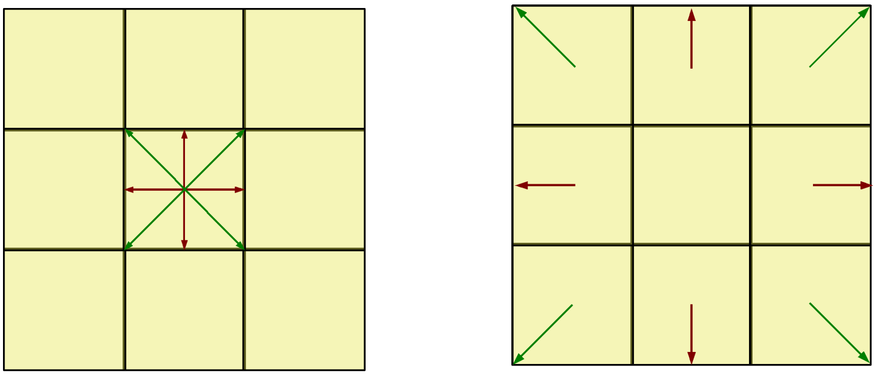
$$f_\alpha(\mathbf{x}_i + \mathbf{e}_\alpha \Delta t^*, t + \Delta t^*) - f_\alpha(\mathbf{x}_i, t) = -\frac{1}{\tau}[f_\alpha(\mathbf{x}_i, t) - f_\alpha^{(eq)}(\mathbf{x}_i, t)] \,, \tag{3.7}$$

where τ is the lattice relaxation time. Equation 3.7 is typically solved in two steps:
Collide:

$$\tilde{f}_\alpha(\mathbf{x}_i, t + \Delta t^*) = f_\alpha(\mathbf{x}_i, t) - \frac{1}{\tau}[f_\alpha(\mathbf{x}_i, t) - f_\alpha^{(eq)}(\mathbf{x}_i, t)] \tag{3.8}$$

Streaming:

$$f_\alpha(\mathbf{x}_i + \mathbf{e}_\alpha \Delta t^*, t + \Delta t^*) = \tilde{f}_\alpha(\mathbf{x}_i, t + \Delta t^*) \tag{3.9}$$



(a) Particle distribution before stream step

(b) Particle distribution after stream step

Figure 3.2: Particle distributions before and after stream step

In the collide step , which can be interpreted as many particle collisions, new distribution functions are calculated according to the old time step. The equilibrium distribution functions, which are necessary to do the collide step, are calculated with Equation 3.4. After the collide step, all particles are streamed to their neighboring cells according to their velocity directions. This step, also called stream step, is shown in Figure 3.2. The stream step can either be performed by a pushing of the distribution functions from one cell to their neighbors, or by a pull step in each cell, which pulls every distribution function, which points to the cell. The first option improves the computational speed [Igl05], but is in our case difficult to implement, because pushing in all cells means also pushing into the ghost layer (see section 4.2.1). In this case the exchange of the ghost layer after a stream step would be complicated, because the distribution functions of two cells would have to be merged. This is why the pull method is used in this thesis.
An error analysis of the Lattice Boltzmann method can be found in [Art03].

With the use of a Taylor expansion of Equation 3.7 up to the order $\mathcal{O}(\Delta t^2)$ and application of multiscale Chapman Enskog techniques by expanding f_α about $f_\alpha^{(eq)}$ and introducing two timescales $t_0 = t$ and $t_1 = t\Delta t$ with the assumption $\lambda \sim \Delta t$ one can write the evolution equation as

$$(\partial t_0 + \mathbf{e}_\alpha \cdot \boldsymbol{\nabla}) f_\alpha^{(eq)} = -\frac{1}{\tau} f_\alpha^{(1)} \tag{3.10}$$

to the first order and

$$\partial t_1 f_\alpha^{(0)} + (\partial t_0 + \mathbf{e}_\alpha \cdot \boldsymbol{\nabla})(1 - \frac{1}{2\tau}) f_\alpha^{(1)} = -\frac{1}{\tau} f_\alpha^{(2)} \tag{3.11}$$

to the second order. Equation 3.10 gives the continuity and momentum conservation equation to the first order

$$\partial t_0 \rho + \boldsymbol{\nabla} \cdot (\rho \mathbf{u}) = 0 \tag{3.12}$$

$$\partial t_0 (\rho \mathbf{u}) + \boldsymbol{\nabla} \cdot \boldsymbol{\Pi}^{(0)} = 0\,, \tag{3.13}$$

where $\boldsymbol{\Pi}$ is the momentum flux tensor. The tensor has the form

$$\boldsymbol{\Pi}^{(0)}_{ij} = \rho c_s^2 \boldsymbol{\delta}_{ij} + \rho \mathbf{U}_i \mathbf{U}_j\,, \tag{3.14}$$

where c_s is the speed of sound[1]. For the $D3Q19$ model the speed of sound is given by $c_s = 1/\sqrt{3}$. Following a similar way, the conservation equations can be derived from Equation 3.11 to the second order

$$\partial t_1 \rho = 0 \tag{3.15}$$

$$\partial t_1 (\rho \mathbf{u}) + \boldsymbol{\nabla} \cdot (1 - \frac{1}{2\tau}) \Pi^{(1)} = 0 \tag{3.16}$$

with

$$\boldsymbol{\Pi}^{(1)}_{ij} = \sum_\alpha f^{(1)}_\alpha \mathbf{e}_{\alpha i} \mathbf{e}_{\alpha j} = \Delta t c^2 \tau \left[(\frac{c_s^2}{c^2} - \frac{1}{3}) \delta_{ij} \boldsymbol{\nabla} \cdot (\rho \mathbf{u}) - \frac{1}{3} (\partial_i \rho \mathbf{U}_j + \partial_j \rho \mathbf{U}_i) \right]\,. \tag{3.17}$$

For incompressible fluids ($\boldsymbol{\nabla} \cdot (\rho \mathbf{u}) = 0$) Equation 3.17 simplifies to

$$\boldsymbol{\Pi}^{(1)}_{ij} = \Delta t c^2 \tau \rho \left[-\frac{1}{3} (\partial_i \rho \mathbf{U}_j + \partial_j \rho \mathbf{U}_i) \right] \tag{3.18}$$

or equivalently

$$\boldsymbol{\Pi}^{(1)}_{ij} = -\frac{2}{3} \Delta t c^2 \tau \rho \mathbf{S}_{ij}\,. \tag{3.19}$$

The strain tensor is therefore

$$\mathbf{S}_{ij} = -\frac{3}{2 \Delta t c^2 \tau \rho} \boldsymbol{\Pi}^{(1)}_{ij}\,. \tag{3.20}$$

Substituting these formulas into Equation 2.10 and using the viscosity

$$\nu = \frac{1}{3} c^2 \Delta t (\tau - \frac{1}{2}) \tag{3.21}$$

gives

$$\boldsymbol{\sigma}_{ij} = -\rho c_s^2 \boldsymbol{\delta}_{ij} - (1 - \frac{1}{2\tau}) \sum_\alpha f^{(1)}_\alpha \mathbf{e}_{\alpha i} \mathbf{e}_{\alpha j}\,, \tag{3.22}$$

which is the formula to calculate the stress tensor $\boldsymbol{\sigma}_{ij}$.
As mentioned in the previous chapter, the stress tensor in the Lattice Boltzmann method can be easily calculated without the use of the velocity gradients.

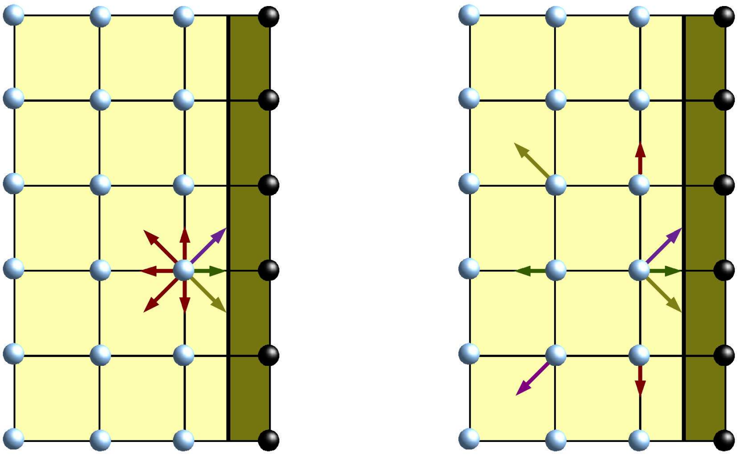
[1] Note, that the speed of sound mentioned here does not correspond to the physical speed of sound of the simulated fluid. It is the speed with which the information is transported through the grid

3.2 Boundary Treatment

The boundary conditions play an important role for the stability and the convergence of the Lattice Boltzmann method. Three different types of boundary cells are used in this thesis:

- no-slip
- inflow
- outflow

The *no-slip* cells represent rigid walls and are modeled with a standard bounce-back schema . This means all distribution functions are reflected at the walls in the opposite direction. An example of one cell near the wall is given in Figure 3.3.



(a) Particle distribution of one cell before bounce-back step

(b) Particle distribution of one cell after bounce-back step

Figure 3.3: Particle distributions of one cell before and after bounce-back at the wall

At the *inflow* cells the velocities are given. Then the distribution functions are calculated based on the equilibrium distribution.

The *outflow* distribution functions $f_I(t, \mathbf{x})$ of the cells can either be calculated by setting the velocities, or by using a boundary condition, which reacts on the flow

$$f_I(t, \mathbf{x}) = -f_\alpha(t, \mathbf{x}) + f_{inv}^{(eq)}(\rho, \mathbf{u}) + f_\alpha^{(eq)}(\rho, \mathbf{u}) \,, \tag{3.23}$$

where f_{inv} is the distribution function opposite to f_α, $\mathbf{u}$ and ρ are the velocity and the density values which would be valid in the cell, if it was not an outflow cell.

3.3 LBM Parametrization

While in the first section of this chapter, the Lattice Boltzmann method itself was fully described, it must be mentioned, that the parametrization needs further explanation. In the following, all relevant parameters for the LBM are mentioned and the calculation of the lattice values is shown.

3.3.1 Parameter Calculation

The standard parameters describing a given fluid flow problem are:

- the size of a fluid cell $\Delta x[m]$
- the fluid density $\rho[\frac{kg}{m^3}]$
- the fluid viscosity $\nu[\frac{m^2}{s}]$
- the fluid velocity $u[\frac{m}{s}]$.

The lattice time step Δt^*, the lattice density ρ^* and the lattice cell size Δx^* are kept constant during the whole simulation:

$$\Delta t^* = \frac{\Delta t}{\Delta t} = 1 \tag{3.24}$$

$$\rho^* = \frac{\rho}{\rho} = 1 \tag{3.25}$$

$$\Delta x^* = \frac{\Delta x}{\Delta x} = 1\,. \tag{3.26}$$

Due to this fact, no multiplications with real world values of the time step, the density or the lattice size are necessary. This is a big advantage compared to other methods like FVM or FEM. Unfortunately this also results in some drawbacks. The step size control gets more complicated compared to other methods, since the time step is fixed to a value of 1. Also the scaling of the computation domain and the calculation of real world values during the simulation are more complicated.

With the physical parameters and using Equations 3.24– 3.26, the corresponding lattice values can be calculated as

$$\nu^* = \nu\frac{\Delta t}{\Delta x^2} \tag{3.27}$$

$$\mathbf{u}^* = \mathbf{u}\frac{\Delta t}{\Delta x} \tag{3.28}$$

$$\mathbf{g}^* = \mathbf{g}\frac{\Delta t^2}{\Delta x}\,. \tag{3.29}$$

As the lattice time step, the lattice density and the lattice cell size, the lattice velocity $\mathbf{u}^*$, the lattice viscosity ν^* and the lattice gravity $\mathbf{g}^*$ are dimensionless. A lattice velocity of 0.1 for example means that the fluid moves 0.1 lattice cells per lattice time step.
To relate all lattice values to the physical ones, the physical time step has to be calculated. This is either done by Equation 3.28 or Equation 3.29, depending on which one gives the smaller time step. In this thesis the gravity is not taken into account and thus only Equation 3.28 is used to calculate the time step. From Equation 3.28, assuming a maximal lattice velocity of 0.1, the time step is restricted to

$$\Delta t = \frac{0.1\Delta x}{|\mathbf{u}|}\,. \tag{3.30}$$

Then Equation 3.27 can be used to calculate the lattice viscosity ν^*.
Using

$$\nu^* = c_s^2 \cdot (\tau - \frac{1}{2}) = \frac{2\tau - 1}{6}\,, \tag{3.31}$$

where $c_s = 1/\sqrt{3}$ is the speed of sound, the relaxation time τ can be calculated. Following these formulas it is obvious that the lattice viscosity and thus the τ and $\omega = 1/\tau$ factors are limited in a given range. For this thesis ω values in the range $0.4 < \omega < 1.96$ are used. Otherwise the simulation would become instable.

3.3.2 An Example Problem

For simulating bloodflow in the brain the size of the lattice cells is directly taken from the resolution of CT or MRI scanners. Unfortunately the rest of the parameter set can only be averaged from physical and clinical data (see section 2.5). In brief, the parameters for the common carotid artery (see section 1.2) as an example are:

- the size of a LBM cell Δx =0.12 mm
- the fluid density ρ =1,100$\frac{kg}{m^3}$
- the kinematic fluid viscosity ν =4.0 $\cdot 10^{-6}\frac{m^2}{s}$
- the maximum inflow fluid velocity u =0.96 $\frac{m}{s}$.

With these physical parameters, the lattice parameters can be calculated following section 3.3.1.

4 Simulation

In the previous chapters all necessary information for solving the hemodynamic situation in the brain was introduced. The method, which is used is the Lattice Boltzmann method. However, in order to conduct the simulation, a lot of additional work has to be done. These tasks, from pre- to postprocessing, are discussed in the following sections.

4.1 Preprocessing

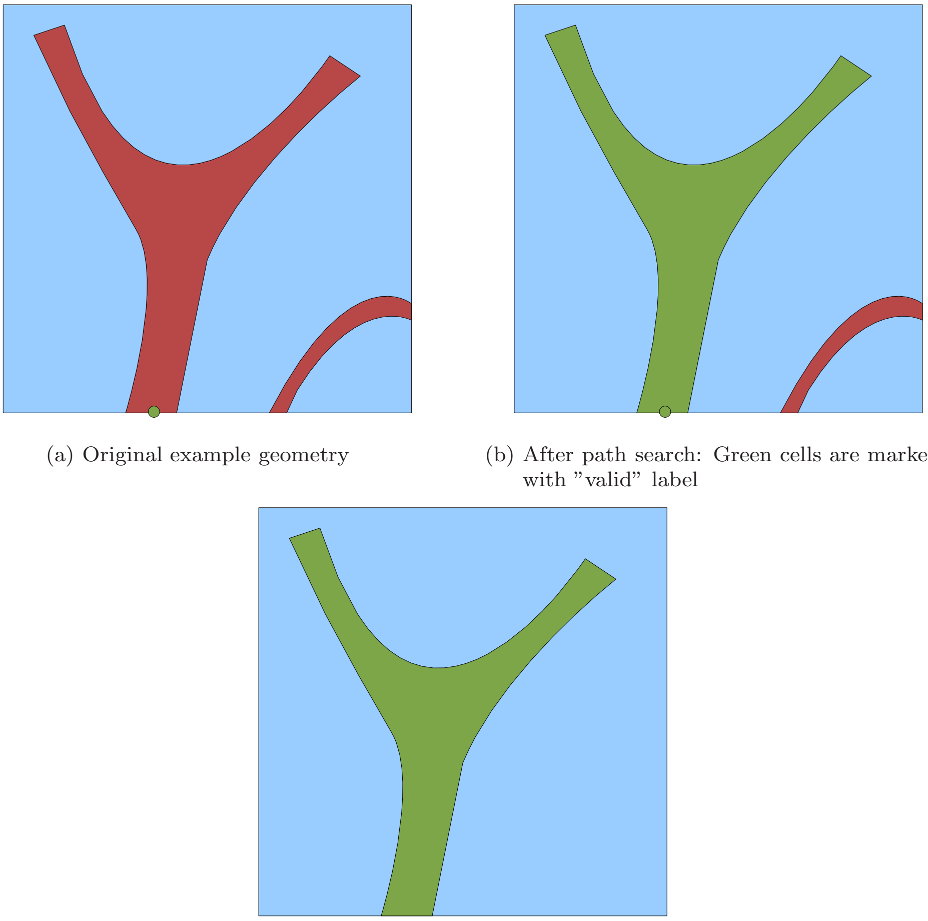
In a preprocessing step, the data from clinical scanners is prepared for the simulation. Several methods were implemented to support the user of the program. The algorithms are discussed in the next chapter. Unfortunately, not all the work can be done automatically and thus has to be done by hand.

4.1.1 Data Cleaning

In a given data set, not all vessels are relevant for the simulation. At most times, one is only interested in the simulation of one inflow vessel and all connected vessels. In order to achieve this, a data cleaning algorithm is used, which extracts only the vessels connected to a given starting point. A simple 2D example of the procedure can be seen in Figure 4.1. First, a point at the inflow is marked (circle in Figure 4.1). Then, the algorithm follows the fluid cells, interconnected with the starting point and marks them with a "valid" label. After that, all remaining fluid cells are set to obstacle cells and thus not needed any more. These cells need not to be saved, which reduces the memory requirements and the computational costs. The pseudo code of the data cleaning is shown in Algorithm 4.1 and is started with one inflow point.

Algorithm 4.1 ProcessCell(Cell p)

```
for all neighbors nb do
    $nb = getNeighbor(p)$
    if $nb = FLUID$ then
        mark(nb)
        ProcessCell(nb)
    end if
end for
```



(a) Original example geometry

(b) After path search: Green cells are marked with ”valid” label

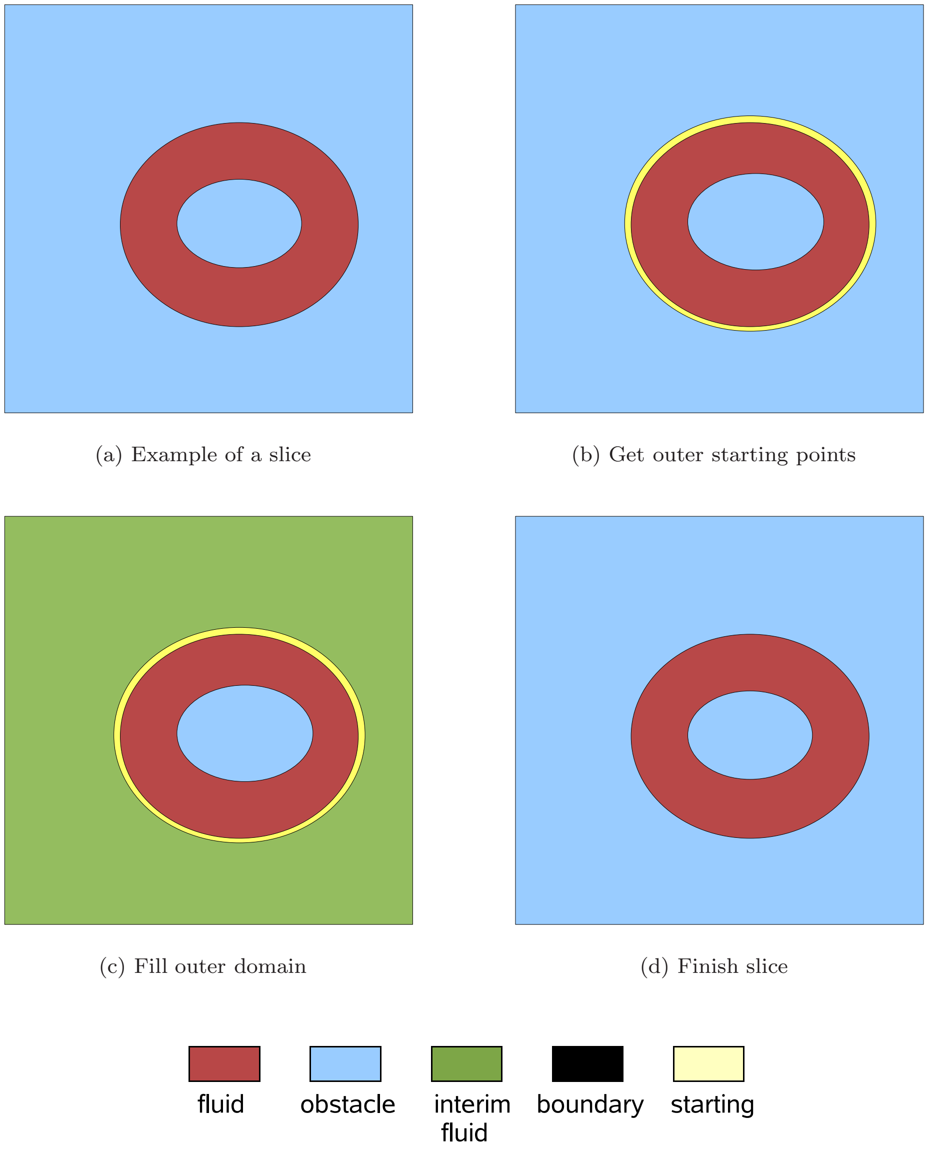
(c) After erasing unneeded cells

Figure 4.1: Example of path search algorithm: All cells, which are not connected to a given starting point (see circle in figure 4.1a and 4.1b) are erased

4.1.2 Hole Filling

A problem arising early in the data processing is the inconsistency of the angiography data sets with the simulation needs. In the original sets, the vessels have inclusions, which means that blood is not recognized as blood, but as a rigid body in the vessel during DSA. This circumstance is due to the contrast agent, which sometimes is not distributed equally through the vessel. This fact does not influence doctors, because only the outer vessel wall is visualized for them. However, the inclusions are disastrous for the simulation, thus they have to be filled.

Hole filling is done for each slice in three Cartesian coordinate directions. If the domain is $100 \cdot 200 \cdot 300$ cells, then 100 y-z-slices, 200 x-z-slices and 300 y-z-slices are processed. In Figure 4.2 and 4.3 it is illustrated, how the hole filling works. First, small holes in the vessel wall of the size of one cell are located and filled. Then the boundary points between fluid and obstacle points are identified (green area in Figure 4.2b and 4.3b). Starting from one of these points a floodfill algorithm is used, which sets all fluid points connected to the starting point to interim fluid points. At the end it is checked, if the domain boundary was reached by the floodfill algorithm. If this has occurred, then outer points of the vessel wall were chosen as starting points or the vessel wall has larger holes, which can not be filled. In this case the original slice has to be restored and all values have to be reset to their initial ones (see Figure 4.2d). If the boundary was not reached, the algorithm has filled the hole. Then the interim fluid points are set to real fluid points (see Figure 4.3d).



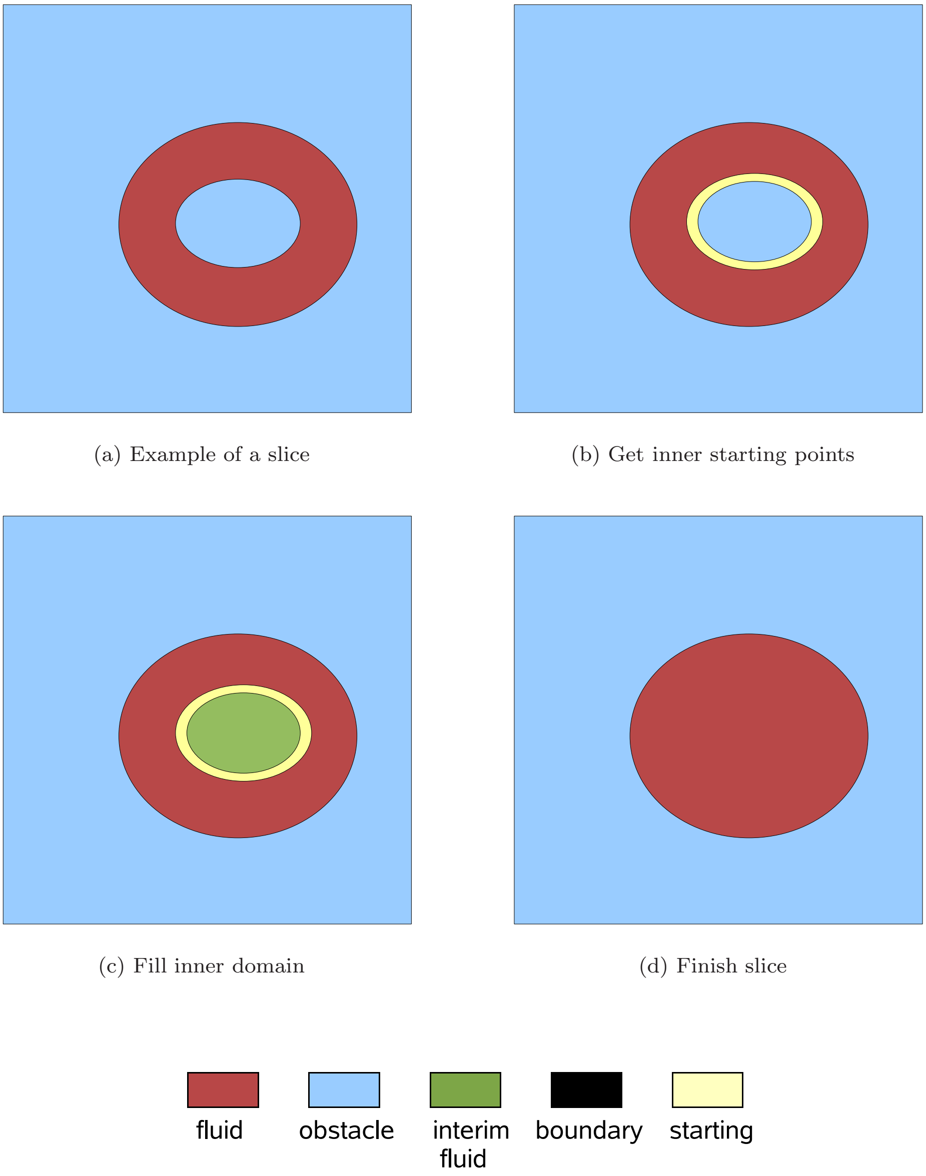
(a) Example of a slice

(b) Get outer starting points

(c) Fill outer domain

(d) Finish slice

Figure 4.2: Example of hole filling algorithm where outer cells are filled



(a) Example of a slice

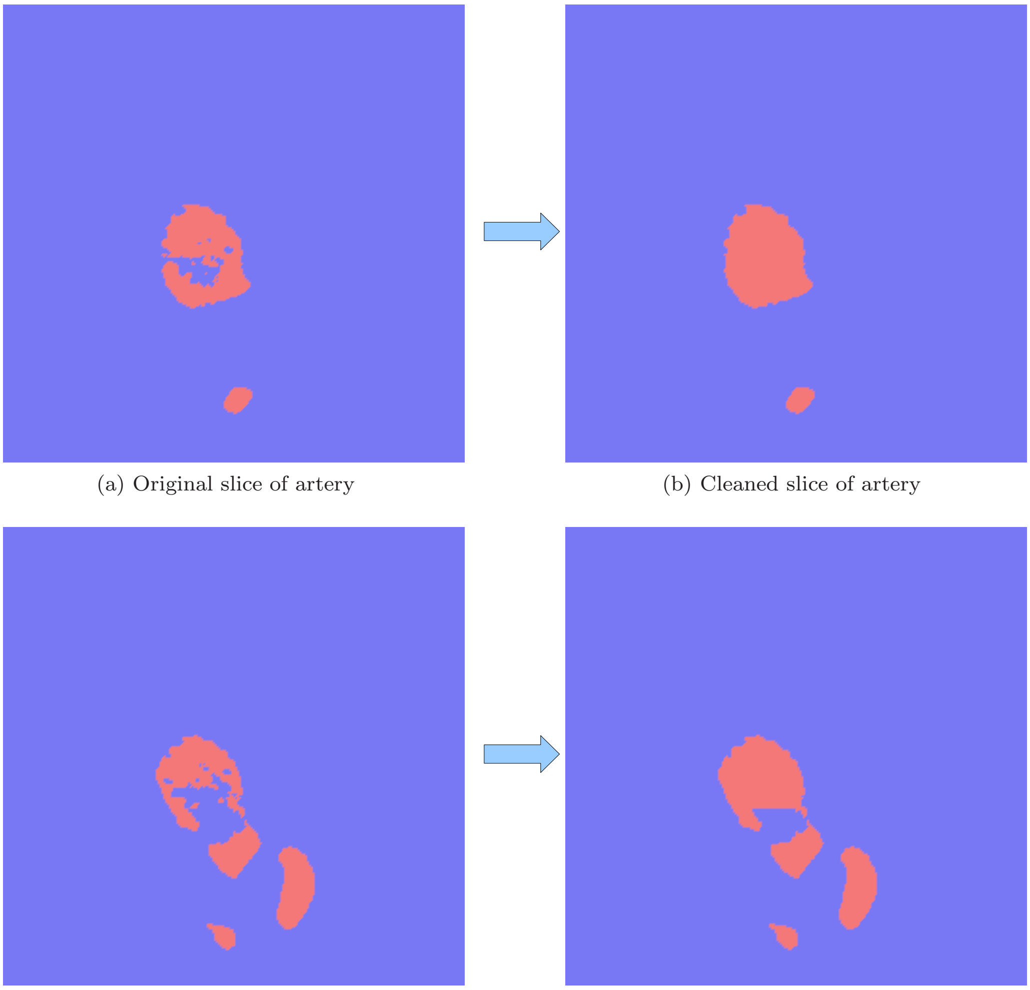
(b) Get inner starting points

(c) Fill inner domain

(d) Finish slice

Figure 4.3: Example of hole filling algorithm where inner cells are filled

Two examples of the hole filling algorithm with real patient geometry are shown in Figure 4.4. In the first case, the algorithm is working properly and has filled the hole completely. The second example is located near the bifurcation. In this example, the algorithm is able to fill each part of the outgoing vessel itself, but it is not able to predict where the inner vessel wall is lying.



(a) Original slice of artery

(b) Cleaned slice of artery

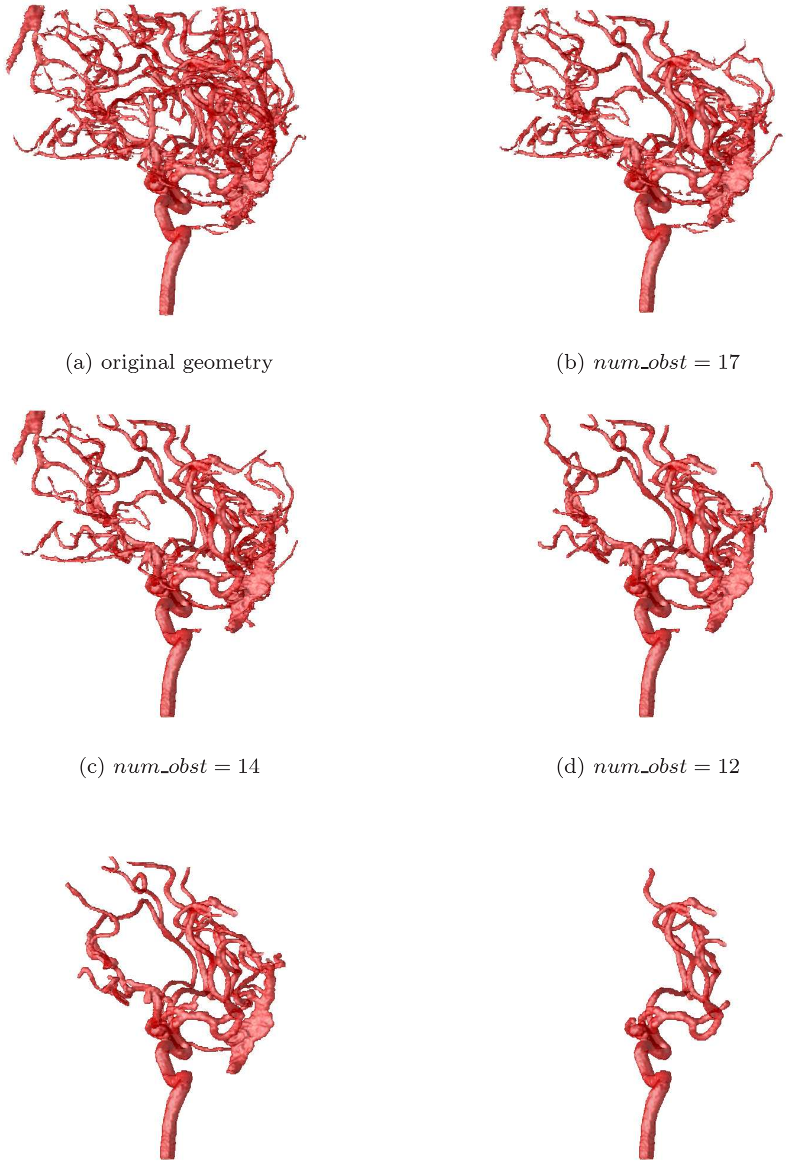
(c) Original slice of bifurcation area

(d) Cleaned slice of bifurcation area

Figure 4.4: Example of hole filling algorithm with patient data: Left is the original data, right is the processed data

4.1.3 Data Reduction

To control the size of the vessels, which are simulated, a parameter *num_obst* can be set. This parameter represents the maximum number of surrounding obstacle cells. If the number of a cell is higher than the given one, the cell itself is set to an obstacle cell. This means, that fluid cells which have many obstacles in their direct neighborhood are erased. This closes small vessels. After this step, the data filling algorithm is used to erase the part of the vessels which is not anymore connected to the inflow vessel. This leads to a reduction of the simulated vessels, which is shown in Figure 4.5. For small parameters, the vessel is slightly reduced, but using higher parameters results in a geometry, which only contains the biggest vessels (see Figure 4.5f for *num_obst* = 10).



(a) original geometry

(b) $num\_obst = 17$

(c) $num\_obst = 14$

(d) $num\_obst = 12$

(e) $num\_obst = 11$

(f) $num\_obst = 10$

Figure 4.5: Example of data reduction with patient data for different *num_obst* values: *num_obst* is a parameter to control the simulated vessels

4.2 Data Layout

In the general case, the Lattice Boltzmann method uses a rectangular lattice with fluid and obstacle cells. The data structure is often saved as a 1D-array in memory and is completely filled.
For the simulation of blood vessels in the human brain, most of the cells are obstacles and except for the boundary cells are not taken into account. Measurements have shown, that in a standard brain geometry about 95% of all cells are neither used in the simulation, nor updated. Thus, at best, one could reduce the memory requirements to a value of 5% of the original geometry. An efficient data layout, which uses only related cells, is described in detail in [HKR05]. This layout is unfortunately not very flexible in the sense of moving boundaries, because cells that form in case of moving walls will destroy the data alignment in memory. To leave the door open for a later implementation of moving boundaries, another approach is used, which is discussed in the following section.

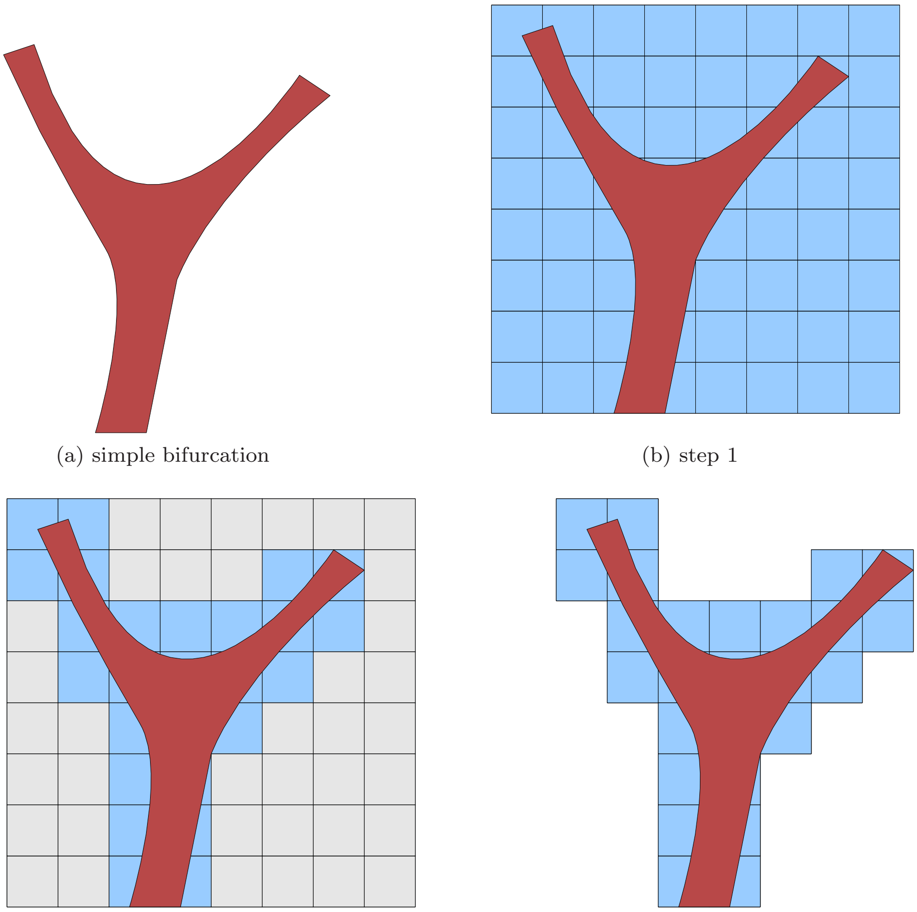
4.2.1 Grid Decomposition

In order to save memory and computation time, the lattice is divided into equal boxes and only boxes that contain fluid cells are saved in a vector. All other boxes are not taken into account. Each box represents a domain, which can be solved independently of other boxes in one time step. Another benefit of the grid decomposition is the possibility to parallelize the program easily. The layout of the boxes is shown in Algorithm 4.2. All necessary information for the simulation of the complete domain is saved in this structure. For the standard Lattice Boltzmann method, two fields called source (src) and destination (dst) are needed for saving the distribution functions. It is also possible to do the simulation with just one grid, which is one cell larger in each direction. This is called a grid compression algorithm (see [Igl05]), but not used here. Additionally, a field with information about the cells is needed. In that field the cell type is stored, which is for example *fluid*, *obstacle*, *inflow*, *outflow* or *no-slip*. Just to do a comfortable programming, arrays with the velocity and the density of each cell is added to the box structure. These values are not needed for the simulation itself, but later for the visualization. Every field value of a lattice cell can be reached by a local i,j,k-index in each box. The rest of the data structure involves the grid decomposition. The number of points in each direction and all surrounding box-neighbors are stored in order to get a global i,j,k-index. To reduce the calculation effort, the starting indices of the boxes are also saved. The allocation of the box structure is done in a preprocessing step.

Algorithm 4.2 Struct box

```
$int\ nPointsX, nPointsY, nPointsZ$; {number of points in one direction}
$int\ i1, i2, j1, j2, k1, k2$; {start and end of box in direction z,y,x}
$int\ neighbor[27]$ {array of neighbors}
$unsigned\ int * \ flag$; {flag values}
$real * \ src$; {source field for the LBM}
$real * \ dst$; {destination field for the LBM}
$real * \ vel$; {supporting velocity field}
$real * \ dens$; {supporting density field}
```

A 2D-example of the grid decomposition and the box reduction is given in Figure 4.6. Here only 26 boxes need to be saved instead of 64 boxes, which means a reduction of more than 59%. In the first step, the geometry is divided into equal boxes in Cartesian coordinates. If there are fluid points in one of these boxes, this box is marked and saved to the box vector. All other boxes (gray boxes in Figure 4.6c) are not necessary and thus not saved. This results in a reduced number of boxes (see Figure 4.6d).
The reduction factor regarding 3D-cases is even higher and in the range of $86 - 92\%$. For two exemplary data sets with a number of $256 \cdot 256 \cdot 400$ points and 16 boxes in each direction only $11 - 19\%$ of the memory requirements compared to the standard layout were needed.



(a) simple bifurcation

(b) step 1

(c) step 2

(d) final boxes

Figure 4.6: Grid decomposition and reduction of a simple 2D-model of a bifurcation

To recall the data layout of the boxes, a 2D example is shown in Figure 4.7. Each box contains a fix number of lattice cells.

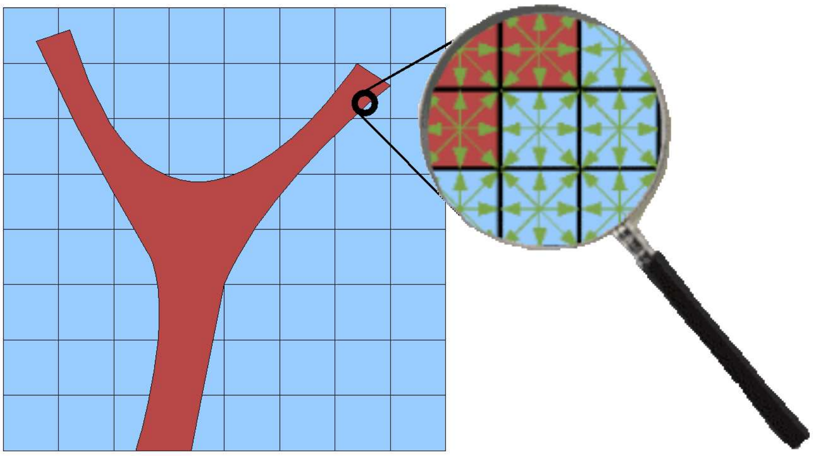


Figure 4.7: Example of fluid and obstacle cells in a box: Red are fluid cells and blue are obstacle cells

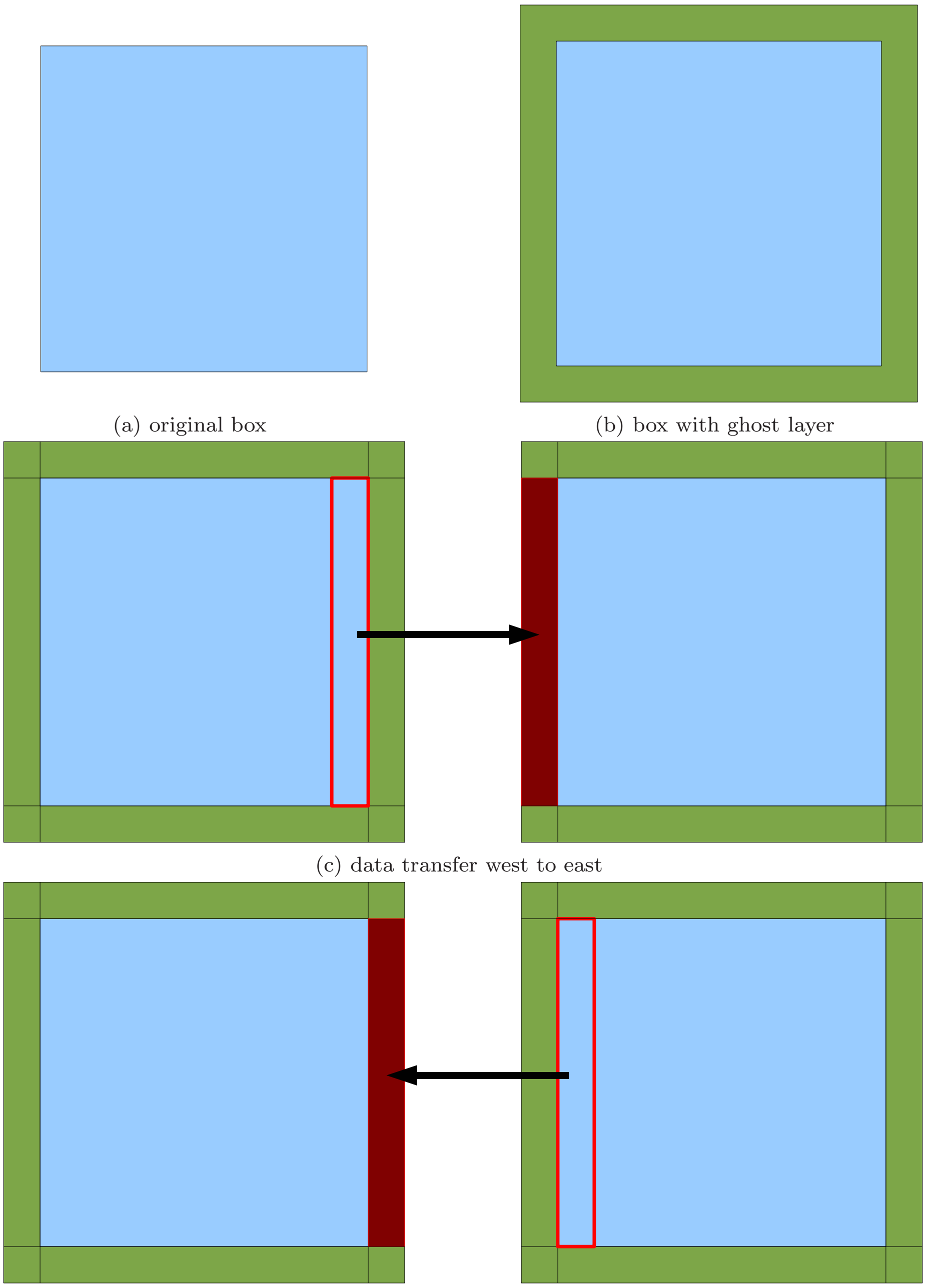
4.2.2 Data Transfer

To treat each box independently in the stream and collide step and to reduce the data transfer, every box has an additional ghost layer (see Figure 4.8b) which surrounds the box. The data of the ghost layers of the boxes has to be updated after each time step. In 2D, only edges or vertices are updated. In 3D, additional updates of adjacent surfaces are necessary. An example of the ghost layer transfer with a neighboring connection in east west direction is shown in Figure 4.8. In the first step (Figure 4.8c), the left box copies its east data cells of the domain into the west ghost layer of the right box. Then the west domain cells of the right box are copied into the east ghost layer of the left box.

The pseudo code of the data transfer of the ghost layer is shown in Algorithm 4.3. Here, the different connections are selected by the direction of the neighbor, which is saved in the data structure. Based on the direction of the neighbor, the corner, the edge, or the adjacent area of the two boxes is exchanged. For the D3Q19 model, only the adjacent area and the edge of the box have to be exchanged.

Algorithm 4.3 ExchangeGhostlayer

```
for all boxes do
  if box has neighbor $N, S, E, W, T, B$ then
    exchange area
  else if box has neighbor $NE, NW, SE, SW, TE, TW, TN, TS, BE, BW, BN, BS$ then
    exchange edge
  else if box has neighbor $TNW, TNE, TSW, TSE, BNW, BNE, BSW, BSE$ then
    exchange corner
  end if
end for
```



(a) original box

(b) box with ghost layer

(c) data transfer west to east

(d) data transfer east to west

Figure 4.8: Example of ghost layer and data transfer between two neighboring boxes

4.3 The Solver

The basis of the solving algorithm is a standard Lattice Boltzmann solver with no-slip bounce-back schema at the wall. In all implemented methods, the most significant change to the standard solver is an additional outer loop over all existing boxes.
The following methods were implemented for the solving:

- Boundary()
  This method is setting the boundary cells, so it updates the no-slip, the inflow and the outflow cells. For the inflow and the outflow one can choose if a time dependent field or a steady field is used. Additionally, a flat or a parabolic profile can be chosen.
- StreamCollideStep()
  In this method the stream step is done, followed by the collide step.
- MoveParticles()
  Here a particle tracing is implemented. The velocities are interpolated with a trilinear interpolation schema.
- ExchangeGhostlayer()
  As mentioned in section 4.2.2, the ghost cells have to be updated after every time step. To accelerate this method, all corresponding boxes are only updated once. This means that if box A is processed and exchanges the ghost layer with box B, box B will not be touched any more.
- GridSwapping()
  Since two grids are used (source grid and destination grid), they have to be swapped after each time step.

These methods are used in each time step as shown in Algorithm 4.4.

Algorithm 4.4 Solver

```
for $timestep < max\_time$ do
    Boundary()
    StreamCollideStep()
    MoveParticles()
    ExchangeGhostlayer()
    GridSwapping()
end for
```

4.4 Postprocessing

For the postprocessing, two different output formats are supported:

- PovRay
- OpenDX

The results, e.g. pressure, velocity and stress, are exported in these formats. Both programs used for the visualization are free software and can easily be downloaded and installed on Linux systems.
PovRay is a raytracer, which is in this thesis mainly used for the particle tracing. Velocities can also be visualized with this program. After writing a PovRay file from simulation data, this file has to be rendered

in a second postprocessing step.
For the velocity, the forces on (pressure) and near (shear) the wall and also for the particle tracing OpenDX can be used. First experiments showed, that using glyphs for the forces makes the picture difficult to recognize and interpret. The visualization also depends on the glyph density. Thus the forces were color-coded and mapped on the wall. This results in a picture, which can easily be interpreted. However, only scalars can be visualized with this method. Thus, the stress tensor had to be transfered to a scalar value. This is done with a matrix invariant (see section 2.7).

5 Results

The flow field and the shear stress play important roles in understanding, diagnosis and treatment of vessel malfunctions. In this chapter, two relevant geometries are presented and the simulation results for the velocity, the pressure and the shear stress are discussed. Furthermore, the performance and the memory requirements are investigated. All simulations were done on one dual node of the LSS[1] cluster with an AMD Opteron CPU (2.2 GHz, 1 MB L2 cache, 4 GB DDR 333 RAM).

5.1 Memory requirements

The memory requirements for the simulation of a fixed domain size depend on the number of boxes that are used for the domain decomposition. When the overall number of boxes increases, keeping the size of the domain constant, the size of the boxes is getting smaller, so the number of lattice cells in each box decreases. This means that for a rising number of boxes more and more boxes are not necessary, because they do not include fluidal points. Since these boxes are not saved, the percentage of used boxes is getting smaller for a larger number of boxes (see Figure 5.1). This effect is generally true, but it has a certain limiting value, because the boxes are getting smaller with increasing number of boxes. In the used geometry about 3.1% of the nodes are fluidal. The lowest percentage value for used boxes would mean that all boxes are completely filled with fluid cells. For a high number of boxes the graph in Figure 5.1 asymptotes to a value of approximately 4%. This is quite near the lowest possible limit. However, for the memory requirements not only the number of boxes is important. Also the requirements for the ghost layer have to be taken into account. With increasing number of boxes the number of ghost cells surrounding the boxes also increases, which means higher memory requirements. Thus these two effects influence the overall requirements which is shown in Figure 5.2. Up to a critical value of approximately 51,000 boxes, the memory requirements are decreasing, because up to this limit the memory requirements depend strongly on the percentage of used boxes. This is shown in Figure 5.1, where the graph is decreasing rapidly for values from 1 to 51,000 boxes. With more than 51,000 boxes, the increasing number of ghost cells is becoming dominant, resulting in rising memory requirements. The minimum for this geometry ($\approx 1.16GB$) is reached with 51,200 boxes.

Different memory requirements for the same number of boxes come from different domain decompositions. If for example 1000 boxes are used, the domain can be divided by $10 \cdot 10 \cdot 10$ boxes, or by $4 \cdot 50 \cdot 5$ boxes. Depending on the geometry, there are more and less efficient decomposition possibilities.

[1] Lehrstuhl für Systemsimulation der Universität Erlangen-Nürnberg

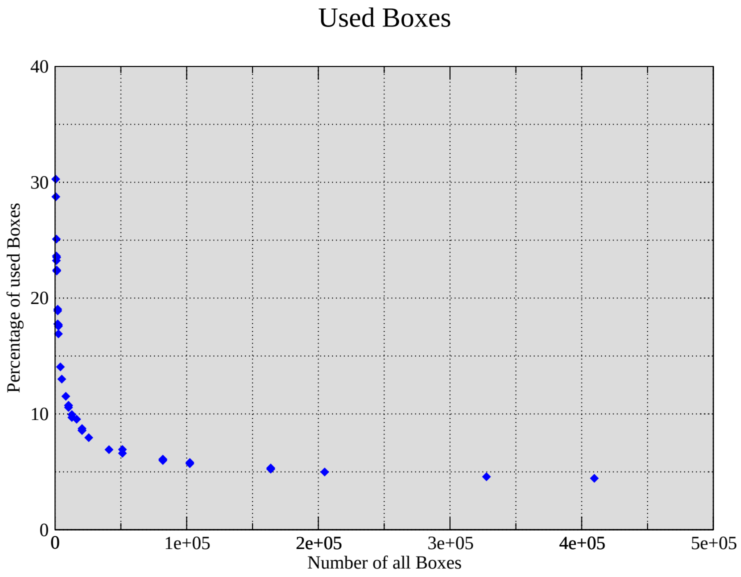


Figure 5.1: Measurement of the number of used boxes per number of all boxes for a given geometry

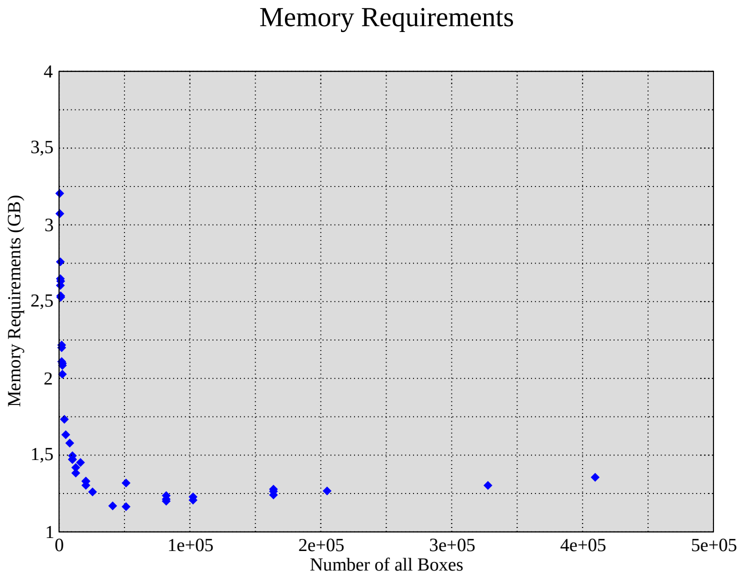


Figure 5.2: Measurement of the memory requirement for the simulation per number of boxes for a given geometry

5.2 Performance

For performance issues, the MFLUPS[2] value and the runtime of the modules were measured for a given geometry. In the runtime graph (Figure 5.3) it is shown, that for a rising number of used boxes the time for the stream collide step and the boundary treatment is reduced. This effect results from the fact, that every box is completely processed, even if it is nearly empty. With increasing number of boxes, every box gets densely filled with fluid cells, which means that mostly fluid cells are processed and no other cells, which are unnecessary for the simulation result. Thus the time for the simulation steps (boundary treatment and stream collide) decreases continuously. If that would be the only effect, one would use a very large number of boxes to get short simulation durations. Unfortunately there exists an antithetic effect regarding time costs. Increasing the number of boxes leads to a higher number of interchanged ghost cells, which makes the ghost exchange computational expensive (see triangles in Figure 5.3). These antithetic effects cause a maximum for the MFLUPS value of about 0.6 (see Figure 5.4). This value is reached with 3,382 used boxes and a number of 51,200 overall boxes, which is also the most efficient splitting regarding memory costs. Compared to a standard LBM solver, only 10% of the memory requirements are needed. As mentioned before, the efficiency regarding runtime depends on the domain splitting.

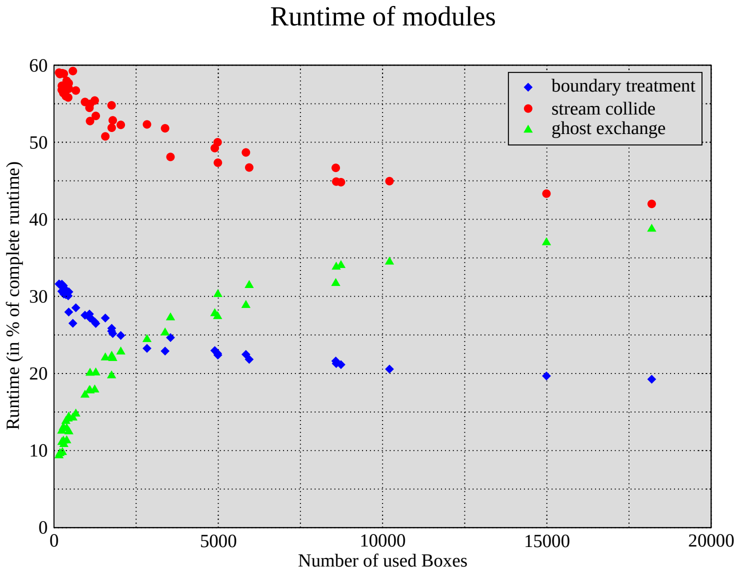


Figure 5.3: Time measurement for the different simulation methods on one dual-node of the LSS cluster per number of used boxes for a given geometry

[2] Mega Fluid Updates per Second

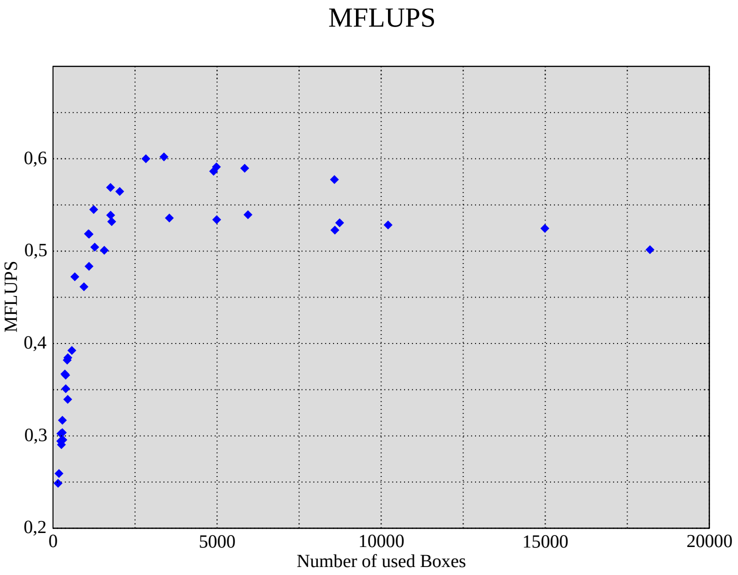


Figure 5.4: Measurement of the mega fluid updates per second for the simulation on one dual node of the LSS cluster per number of used boxes for a given geometry

5.3 Stress

Stress is an important quantity regarding diagnosis and treatment of vessel malfunctions. As mentioned in section 3.1, with the Lattice Boltzmann method the nine Cartesian components of the stress tensor can be calculated directly without the need to estimate the velocity gradients. This reduces the computational costs significantly compared to standard FEM or FVM solvers. Furthermore, a calculation of the stress using velocity gradients would introduce an approximation error due to the nonlinear velocity gradients at the wall.
In this thesis, the von Mises effective stress is visualized, defined as

$$\sigma = \sqrt{\frac{(\sigma_{xx}-\sigma_{yy})^2+(\sigma_{yy}-\sigma_{zz})^2+(\sigma_{zz}-\sigma_{xx})^2+6(\sigma_{xy}^2+\sigma_{yz}^2+\sigma_{zx}^2)}{2}} \tag{5.1}$$

which is a second invariant of the stress tensor (see section 2.7).

5.4 Common Carotid Artery

In this section, the results for the common carotid artery (CCA) as inflow vessel are presented. The CCA is one of the largest vessels in the human head in diameter. It bifurcates into the ICA and the ECA. For more details see section 1.2. The geometry is given in Figure 5.5 and shows a stenosis and a small saccular aneurysm in front of the stenosis in the internal carotid artery behind the bifurcation. This stenosis would be treated with stenting. The flow enters the domain through the CCA and leaves it either through the ECA or the ICA.

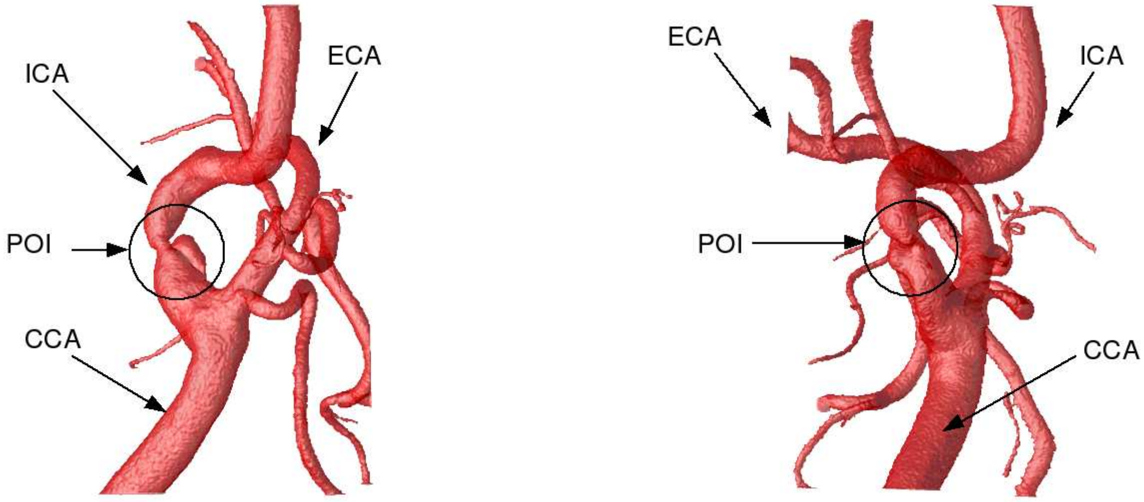


Figure 5.5: Different perspectives of the common carotid artery: Display of a stenosis in internal carotid artery

The simulation is done with a time dependent inflow pulse (see subsection 2.5.1) and the following parameter set:

- the size of a LBM cell $\Delta x = 0.12\ mm$
- the fluid density $\rho =$1,100$\frac{kg}{m^3}$
- the fluid viscosity $\nu = 4.0 * 10^{-6}\frac{m^2}{s}$
- the maximum inflow fluid velocity $u = 0.96\frac{m}{s}$
- the number of cells in x-direction $n_x = 256$
- the number of cells in y-direction $n_y = 256$
- the number of cells in z-direction $n_z = 400$
- the number of boxes in x-direction $n\_Boxes_x = 32$
- the number of boxes in y-direction $n\_Boxes_y = 32$
- the number of boxes in z-direction $n\_Boxes_z = 50$
- the number of time steps $timesteps$ =80,000.

This results in a physical time step of $1.25 \cdot 10^{-5}s$ and an omega value of 1.95918. The Reynolds Number, approximated for the diameter of the inflow vessel as the relevant length scale, is 1,033.22. The simulation needs 3,382 of 51,200 boxes and has a memory requirement of $1.16GB$. With a standard LBM solver the geometry would need about $8.6GB$.

The simulation is done on the complete geometry shown in Figure 5.5, the results are only visualized in the point of interest (POI), here near the stenosis, which is shown in Figure 5.6.
In the next paragraphs the velocity, the pressure and the shear stress are presented for the time steps of the pulse shown in Figure 5.7. The complete pulse had a duration of 1,000 ms with a sin^2 dependent peak of 300 ms.

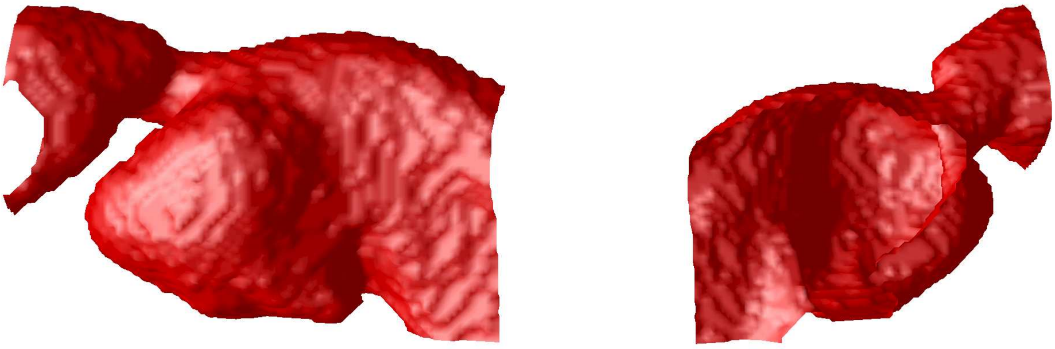


Figure 5.6: Different perspectives of the stenosis in the internal carotid artery

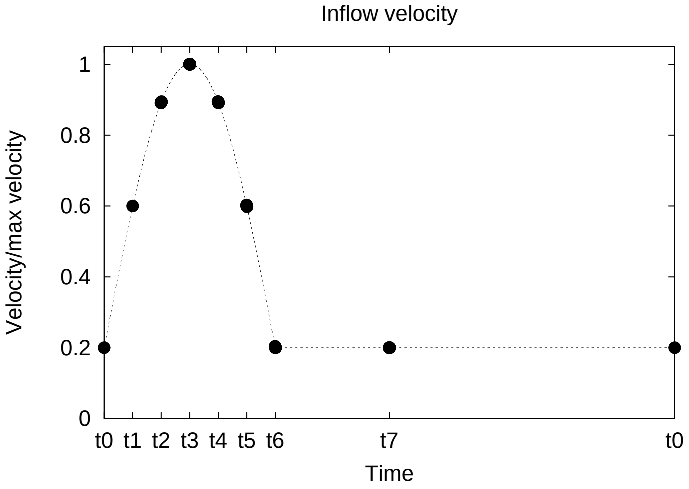


Figure 5.7: Relevant points at the inflow pulse

5.4.1 Velocity

In Figure 5.8 the velocity values are shown and discussed for the perspective of the POI illustrated in the left figure of 5.6. To keep an overview, the isosurface of the vessel wall is also shown in the figures. The flow enters the domain from the right and leaves it to the left side. The velocities are visualized with glyphs where long and red arrows characterize high velocities, whereas short and blue arrows mean low velocities. Additional to the glyph representation, two streamline examples at time steps $t = t_3$ and $t = t_4$, calculated with OpenDX, are shown in Figure 5.9. Red areas are representing high velocity and blue low velocity values.

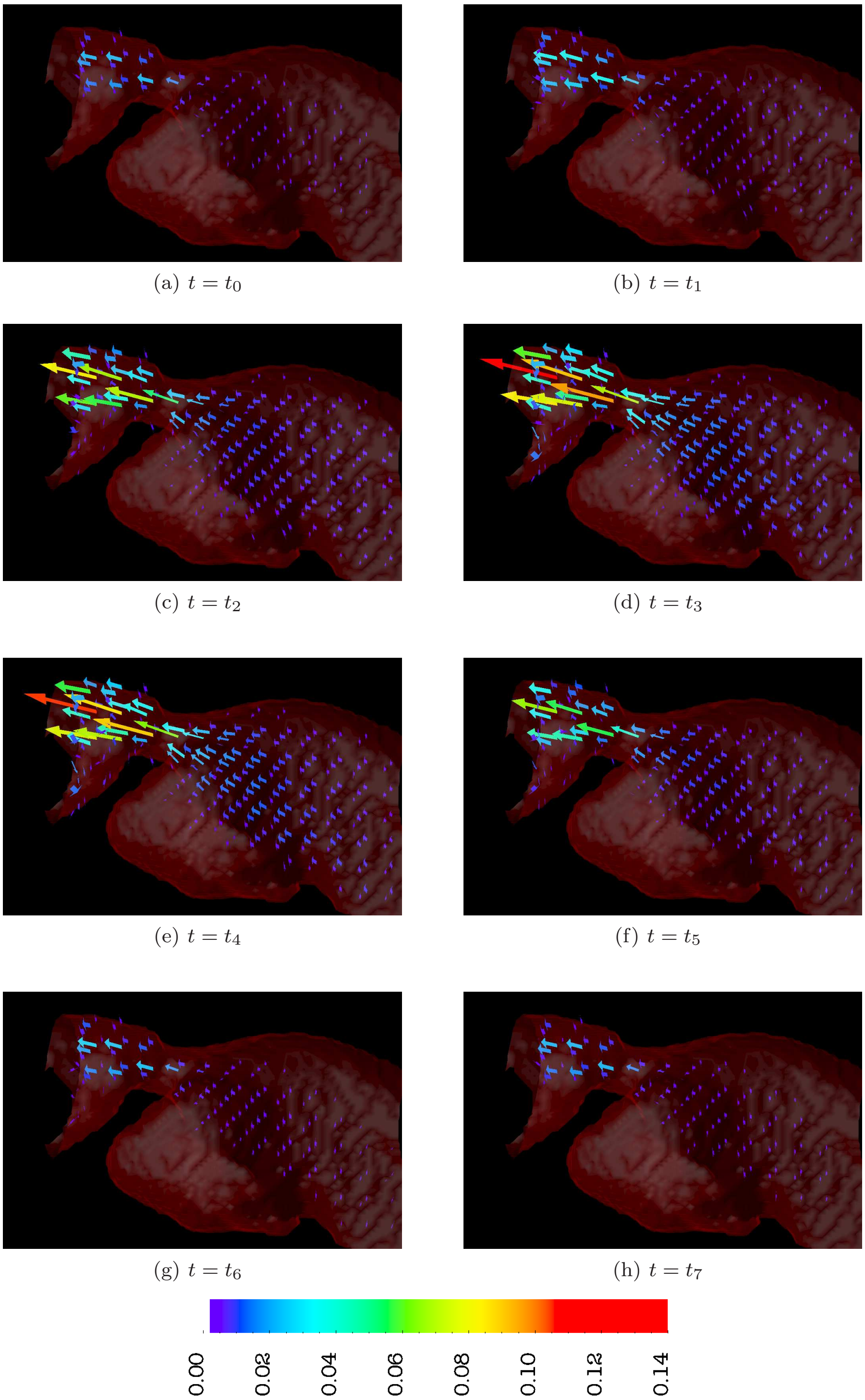
Even at the beginning of the systole ($t = t_0$), the flow in the stenosis is quite high. With rising inflow ($t = t_1 - t_3$) the velocity increases and reaches its maximum value of 0.14 in lattice units (equals 1.344 m/s in physical units) at $t = t_3$. In the stenosis the flow is much faster than in the surrounding area. This is due to the fact, that the vessel diameter in the stenosis is much smaller than the diameter in front of the stenosis. After the stenosis a back-flow region occurs, which can be seen better in Figure 5.9. After the peak of the pulse, the velocity decreases again and reaches its minimum before the next pulse arises ($t = t_0$).

5.4.2 Pressure

First experiments showed that a glyph representation of the pressure is difficult to understand and thus not useful. Hence, in Figure 5.10 the pressure is mapped to the arterial wall, where a high pressure load is color-coded in red and a low pressure load in blue. This procedure is done with OpenDX and gives an intuitive picture of the load on the wall. The used perspective is shown in the left figure of 5.6.
Increasing the inflow velocities ($t = t_0 - t_3$), the pressure reaches its maximum in the saccular aneurysm in front of the stenosis. This means that the maximal pressure load occurs in the aneurysm. In the stenosis and after it, the pressure remains nearly constant and has no significant peak. When the inflow decreases again, the pressure is also going down and relaxes to its initial value at $t = t_0$ before the next pulse arises.

5.4.3 Shear Stress

Analogous to the pressure, the von Mises effective stress is visualized by color-coding and mapping the values to the arterial wall. For the stress, a different perspective is used, which is shown in the right figure of 5.6. This is necessary, because the stress in the outflow region can be seen clearer from that point of view. Here the flow streams from the lower left side to the upper right side through the artery.
The von Mises effective stress reaches its maximum in the stenosis on the front wall at the peak of the inflow pulse ($t = t_3$). Here the highest velocity gradients occur, because the velocity values in the stenosis are also the highest ones. Even in the outlet region the stress is very high. When the flow drops ($t = t_4 - t_6$) the stresses near the wall decrease and relax to their minimal value. At the inflow region and in the aneurysm the shear stress keeps very low during the whole pulse.



(a) $t = t_0$

(b) $t = t_1$

(c) $t = t_2$

(d) $t = t_3$

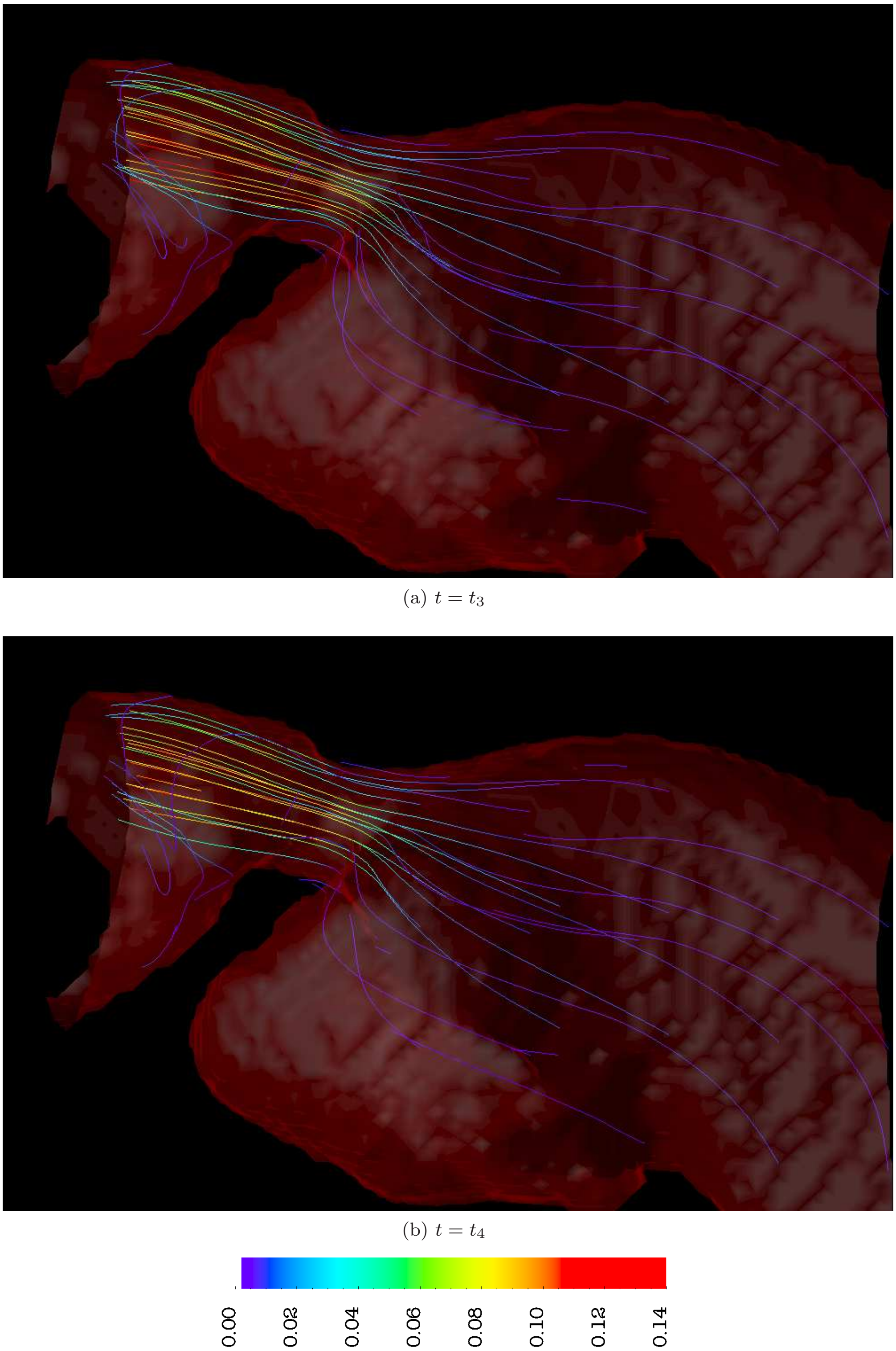
(e) $t = t_4$

(f) $t = t_5$

(g) $t = t_6$

(h) $t = t_7$

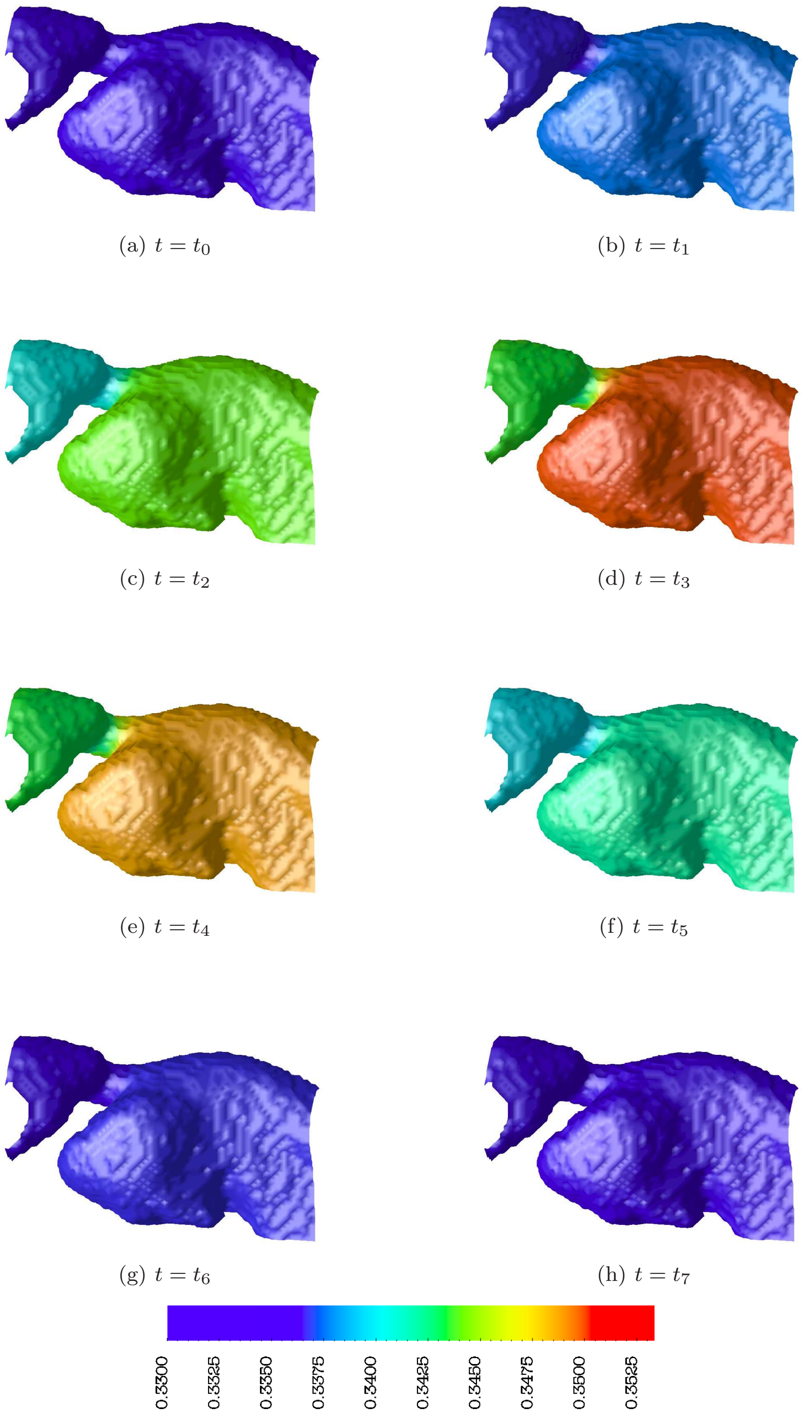
Figure 5.8: Glyph representations of the velocity in lattice units at the point of interest in the CCA during pulse



(a) $t = t_3$

(b) $t = t_4$

Figure 5.9: Streamlines at the POI in the CCA at different points of the pulse calculated with OpenDX based on the velocity values


(a) $t = t_0$

(b) $t = t_1$

(c) $t = t_2$

(d) $t = t_3$

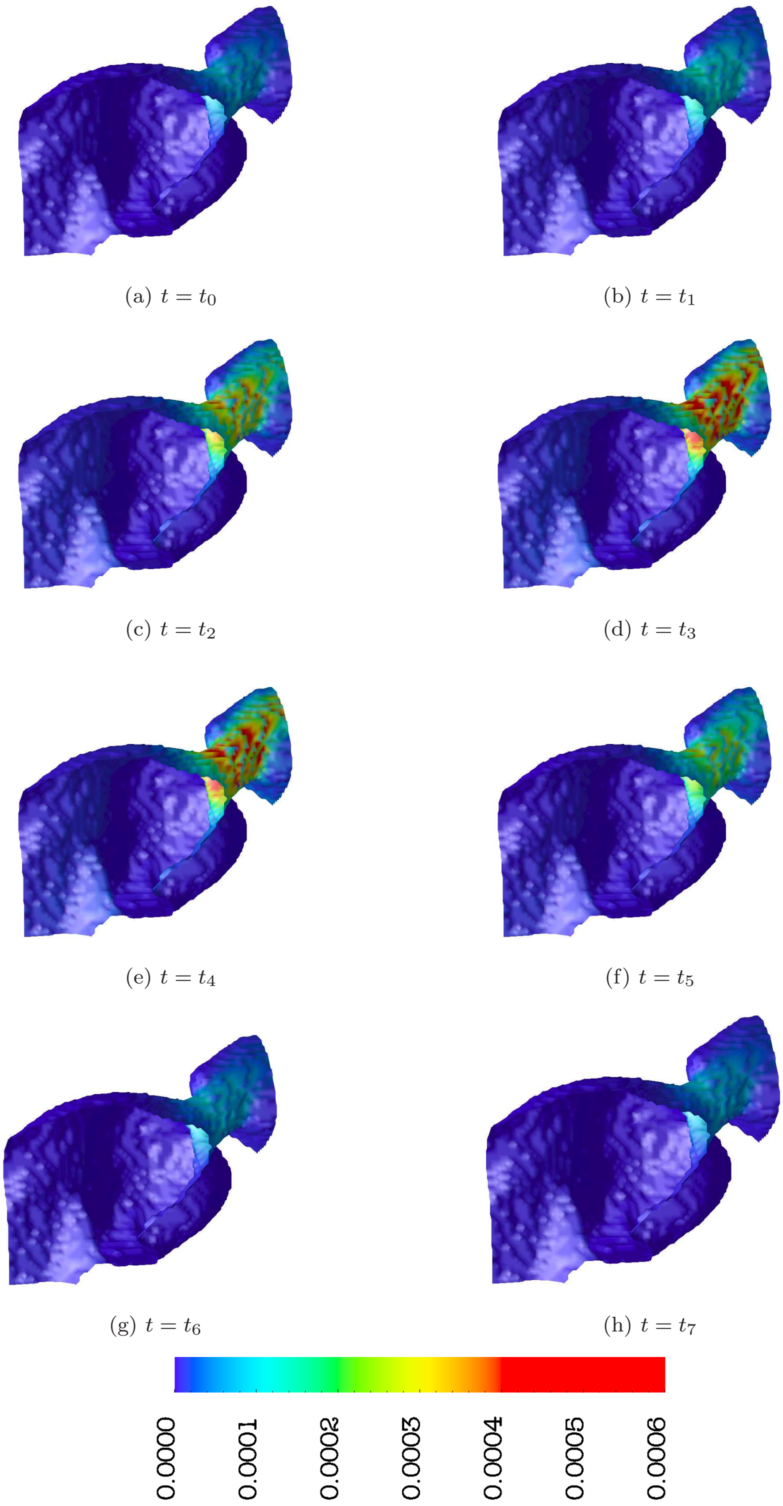
(e) $t = t_4$

(f) $t = t_5$

(g) $t = t_6$

(h) $t = t_7$

Figure 5.10: Pressure on the wall in lattice units at the point of interest in the CCA during pulse



(a) $t = t_0$

(b) $t = t_1$

(c) $t = t_2$

(d) $t = t_3$

(e) $t = t_4$

(f) $t = t_5$

(g) $t = t_6$

(h) $t = t_7$

Figure 5.11: Von Mises stress near wall in lattice units at the point of interest in the CCA during pulse

5.5 Internal Carotid Artery

A patient geometry of the internal carotid artery is shown in Figure 5.12. This patient suffers from an aneurysm near the posterior communicating artery which is the point of interest in this geometry. The flow enters the domain through the ICA and leaves it through several smaller vessels.

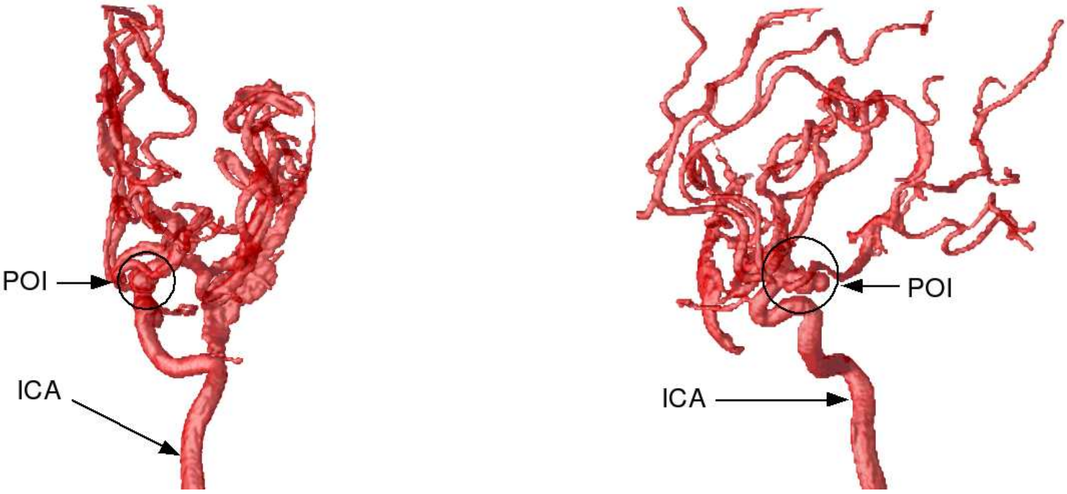


Figure 5.12: Different perspectives of the internal carotid artery: Display of an aneurysm

As in the simulation of chapter 5.4, a time dependent inflow pulse (see Figure 5.7) with a duration of 1,000 ms was used. The parameter set was:

- the size of a LBM cell $\Delta x = 0.17333\ mm$
- the fluid density ρ =1,100$\frac{kg}{m^3}$
- the fluid viscosity $\nu = 4.0 * 10^{-6} \frac{m^2}{s}$
- the maximum inflow fluid velocity $u = 0.64\frac{m}{s}$
- the number of cells in x-direction $n_x = 660$
- the number of cells in y-direction $n_y = 720$
- the number of cells in z-direction $n_z = 672$
- the number of boxes in x-direction $n\_Boxes_x = 132$
- the number of boxes in y-direction $n\_Boxes_y = 144$
- the number of boxes in z-direction $n\_Boxes_z = 96$
- the number of time steps $timesteps$ =37,000.

The original data set consists of $220 \cdot 240 \cdot 224$ cells, but had to be refined by the factor 3 in each direction to reduce the omega value below the critical value of 1.96. This refined set results in a physical time step of $2.7 \cdot 10^{-5} s$ and an omega value of 1.95765. The Reynolds Number is 705.975, again approximated for the inflow diameter as the relevant length scale. The simulation uses 19,458 of 1,824,768 boxes and has memory requirements of $2.95 GB$, which is less than 2.9% compared to a standard LBM solver.

Figure 5.13 shows the interesting area of the patient geometry of the internal carotid artery. To get a better view on the relevant values, the results are only shown for this area. The aneurysm, called berry aneurysm, occurs near the trifurcation where the ICA divides into the posterior communicating artery, the anterior cerebral artery and the middle cerebral artery. Generally, the patient would be treated by coiling.

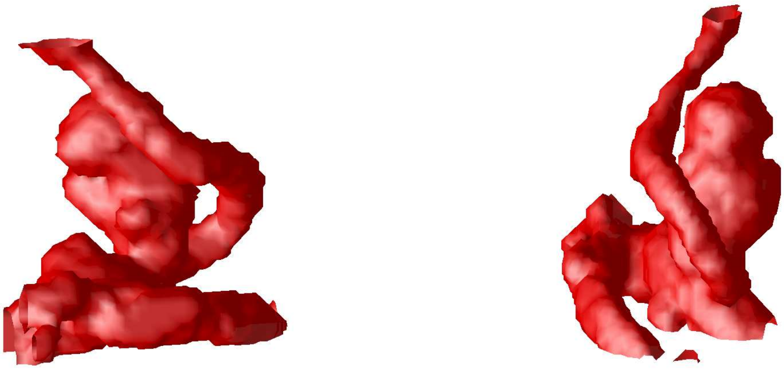


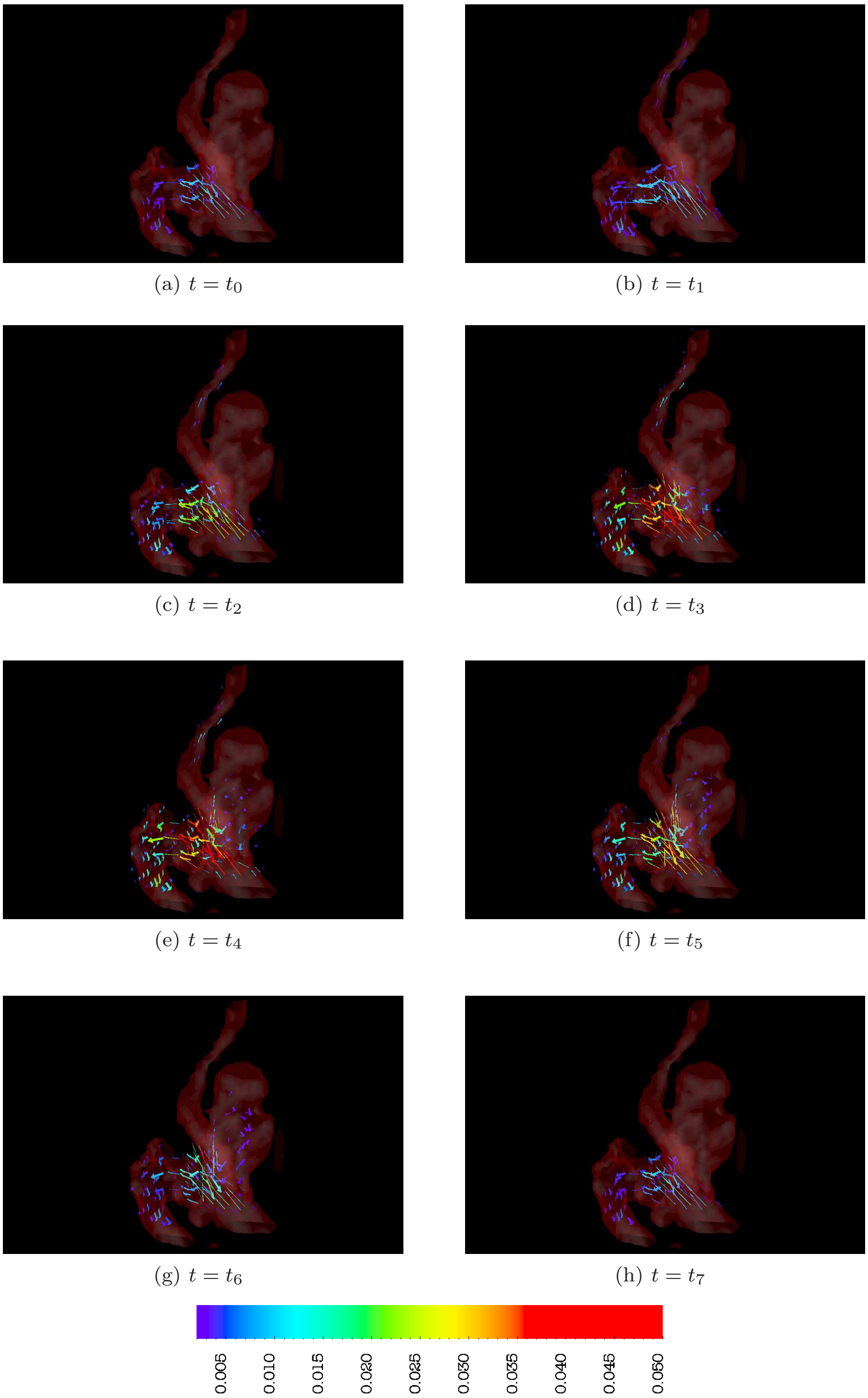
Figure 5.13: Different perspectives of the aneurysm of the internal carotid artery

5.5.1 Velocity

For the visualization of the velocity, the perspective shown in the right figure of 5.13 is used. To keep an overview, the glyph representation of the velocities is overlaid with a picture of the vessel wall. The flow enters the geometry from the lower right and leaves the domain on top, on the lower left side and to the back. The aneurysm occurs above the inflow region and looks like a balloon.
The plots in Figure 5.14 show that during systole ($t = t_0 - t_3$) the velocities increase rapidly up to values of 0.06 in lattice units (equals 0.384 m/s in physical units) at peak ($t = t_3$). The highest velocities occur near the trifurcation. At time steps $t = t_3$ and $t = t_4$ one can see a flow through the aneurysm in clockwise direction, which gets clearer from the streamline representation shown in Figure 5.15. After the pulse, the velocities follow the inflow profile and decrease to a constant level.

5.5.2 Pressure

The pressure values are again color-coded and mapped to the arterial wall (see Figure 5.16) using the point of view shown in the left figure of 5.13. The flow enters the geometry from the lower back and leaves it either to the top or to the left.
When the inflow pulse arrives at the inlet ($t = t_0$) the pressure increases. The highest pressure values occur at the neck of the aneurysm. After the pulse, the pressure relaxes again. To summarize, the overall pressure change in this geometry is quite low.



(a) $t = t_0$

(b) $t = t_1$

(c) $t = t_2$

(d) $t = t_3$

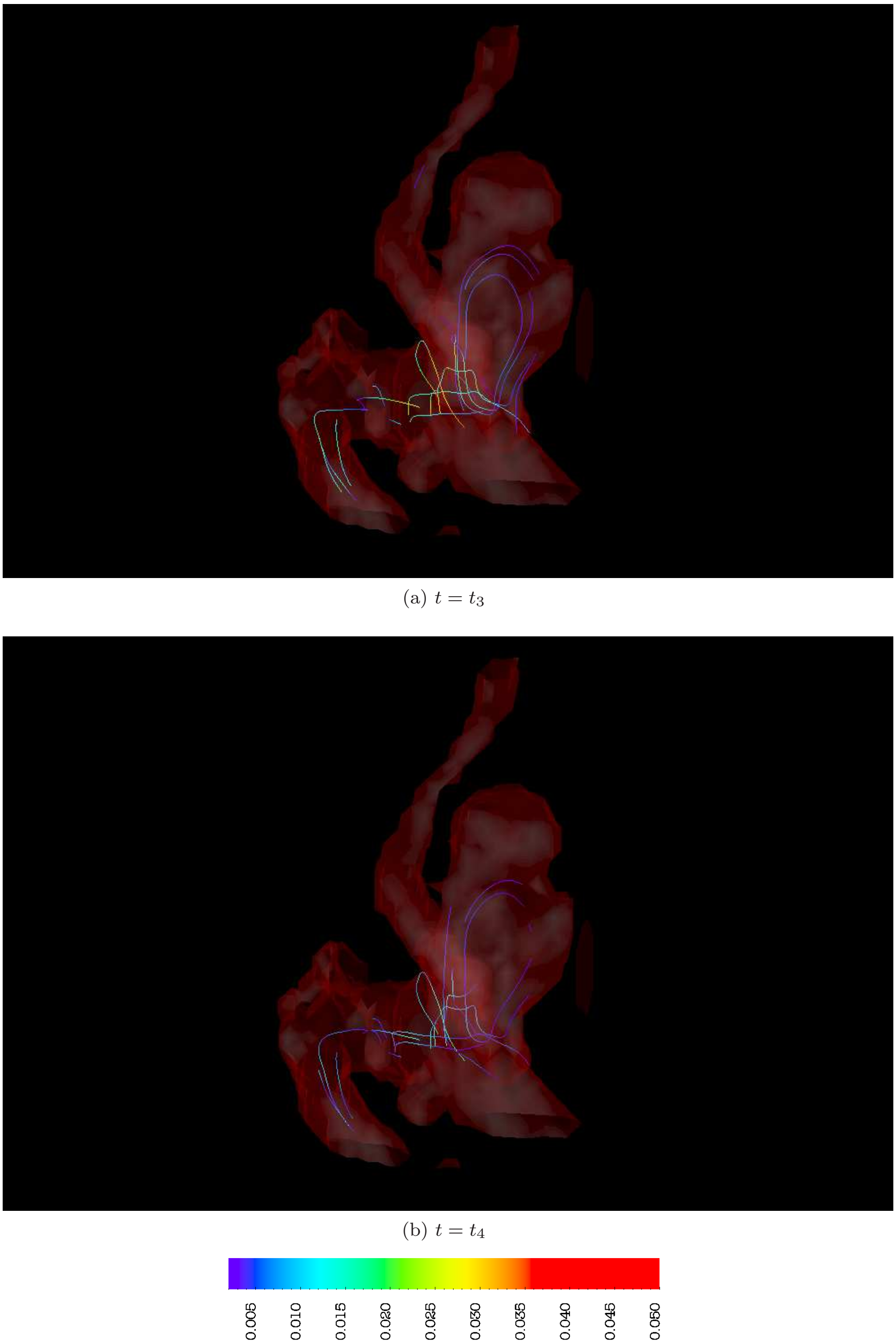
(e) $t = t_4$

(f) $t = t_5$

(g) $t = t_6$

(h) $t = t_7$

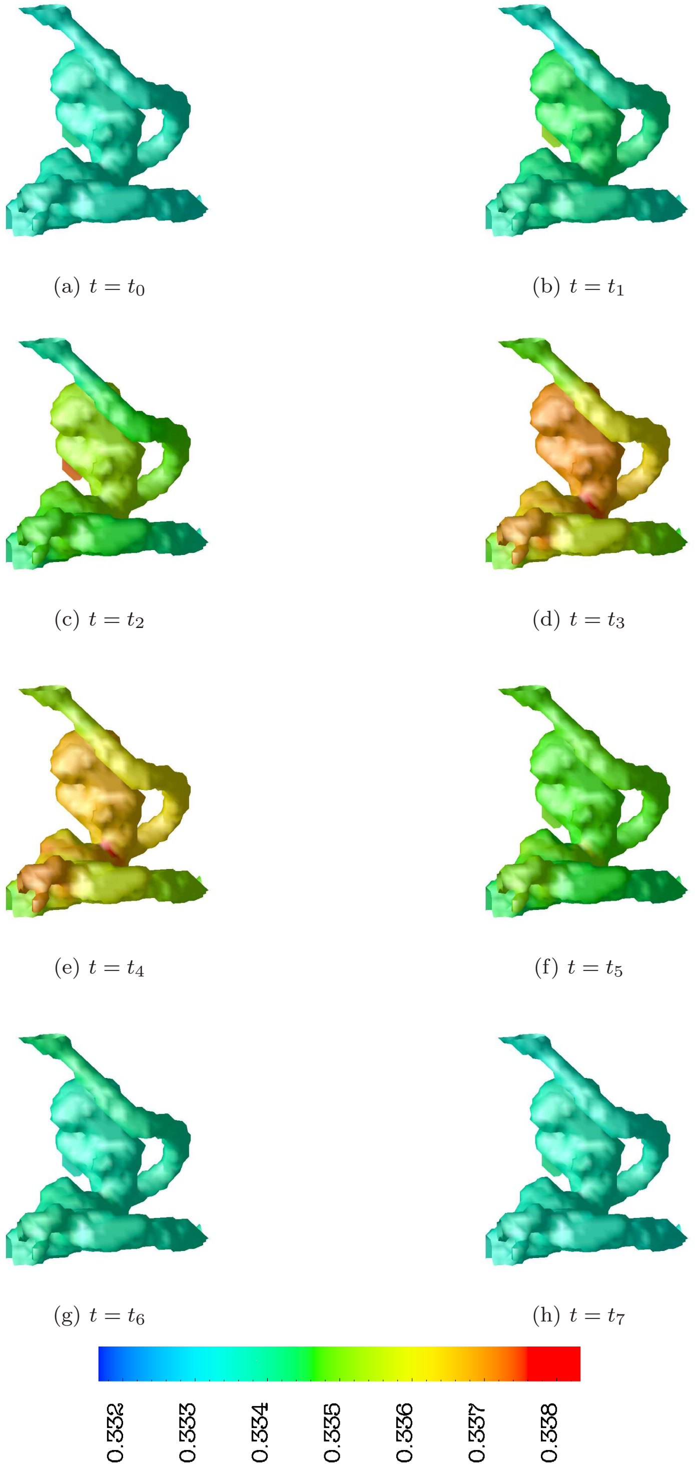
Figure 5.14: Glyph representations of the velocity in lattice units at the point of interest in the ICA during pulse



(a) $t = t_3$

(b) $t = t_4$

Figure 5.15: Streamlines at the POI in the ICA at different points of the pulse calculated with OpenDX based on the velocity values



(a) $t = t_0$

(b) $t = t_1$

(c) $t = t_2$

(d) $t = t_3$

(e) $t = t_4$

(f) $t = t_5$

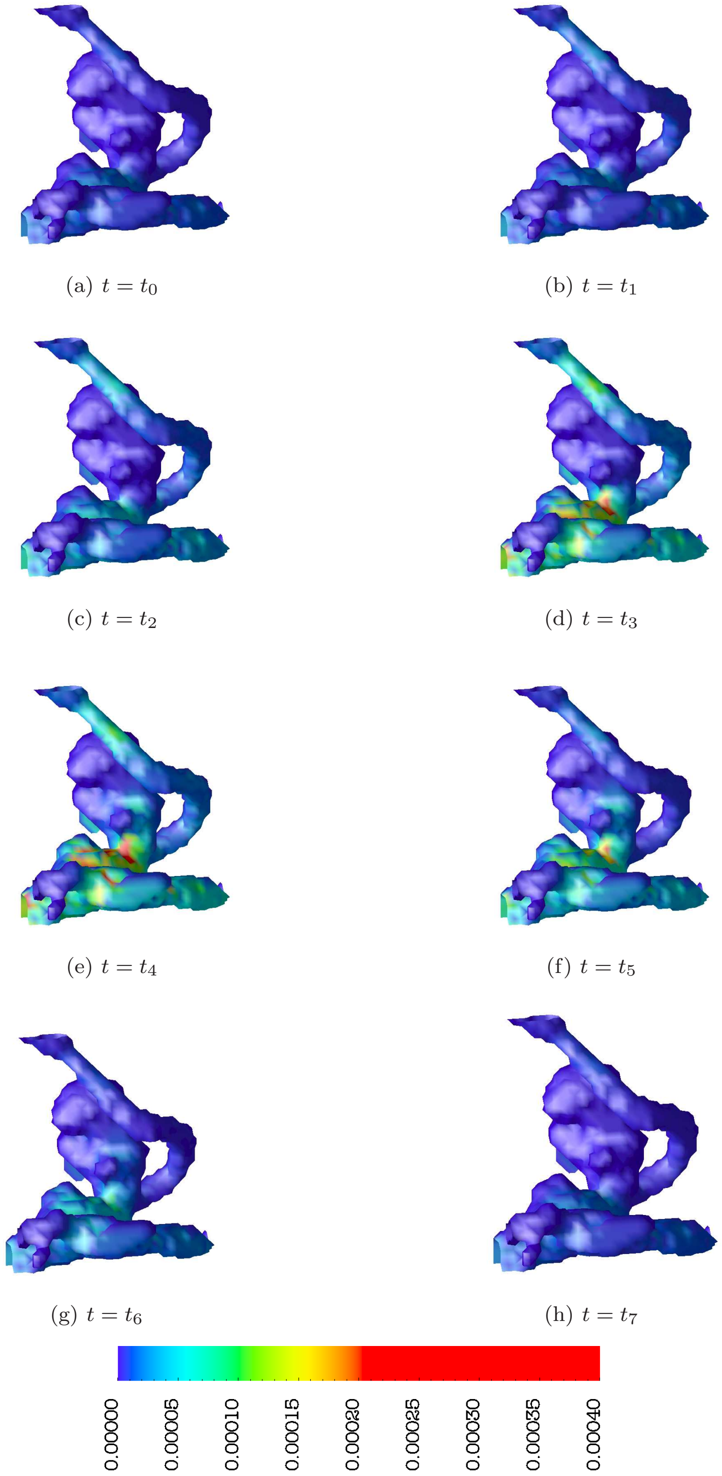
(g) $t = t_6$

(h) $t = t_7$

Figure 5.16: Pressure on the wall in lattice units at the point of interest in the ICA during pulse

5.5.3 Stress

For the shear stress, the same point of view as for pressure is used, which is shown in the left figure of 5.13. The values are again color-coded and mapped to the wall (see Figure 5.17).
With rising inflow velocity ($t = t_0 - t_3$) the stress on some parts of the wall increase. The largest stress values occur at the neck of the aneurysm and near the aneurysm (left buckle in figures). Here are the largest forces at the arterial wall regarding the von Mises effective stress. After the sin^2 pulse ($t = t_6$), the stress values relax again, but stay slightly higher in the two regions than in the rest of the geometry. In the communicating artery, which runs to the top, the stress is also quite high. This is caused by a narrowing of the vessel wall in this artery.


(a) $t = t_0$

(b) $t = t_1$

(c) $t = t_2$

(d) $t = t_3$

(e) $t = t_4$

(f) $t = t_5$

(g) $t = t_6$

(h) $t = t_7$

Figure 5.17: Von Mises stress near wall in lattice units at the point of interest in the ICA during pulse

5.6 Particle Tracing

An interesting question for aneurysms is, how long blood particles stay in the aneurysm. If these particles have long residence times, they may be clotting. This leads to the necessity of a particle tracing algorithm, which visualizes the duration of particles in a fixed region of interest.
The velocities of the particles are calculated in each time step with a trilinear interpolation schema. In Figure 5.19 the particles in the area of interest, here the aneurysm and the surrounding region, are shown in two different time steps which are marked in Figure 5.18. In the graph the duration of the red particles was longer compared to the duration of green and blue particles. The geometry can be seen in Figure 5.13. The simulation was done with 32,000 time steps where every 400 time steps new particles were injected at the inflow region.
Figure 5.19 shows that most particles flow through the main vessel and do not enter the aneurysm. The movements of the particles in the aneurysm are quite low during cardiac cycle, which means that the velocities in the aneurysm are low.

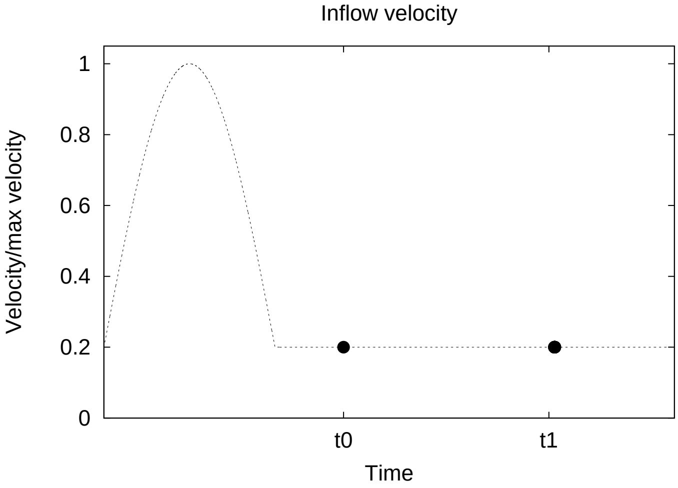
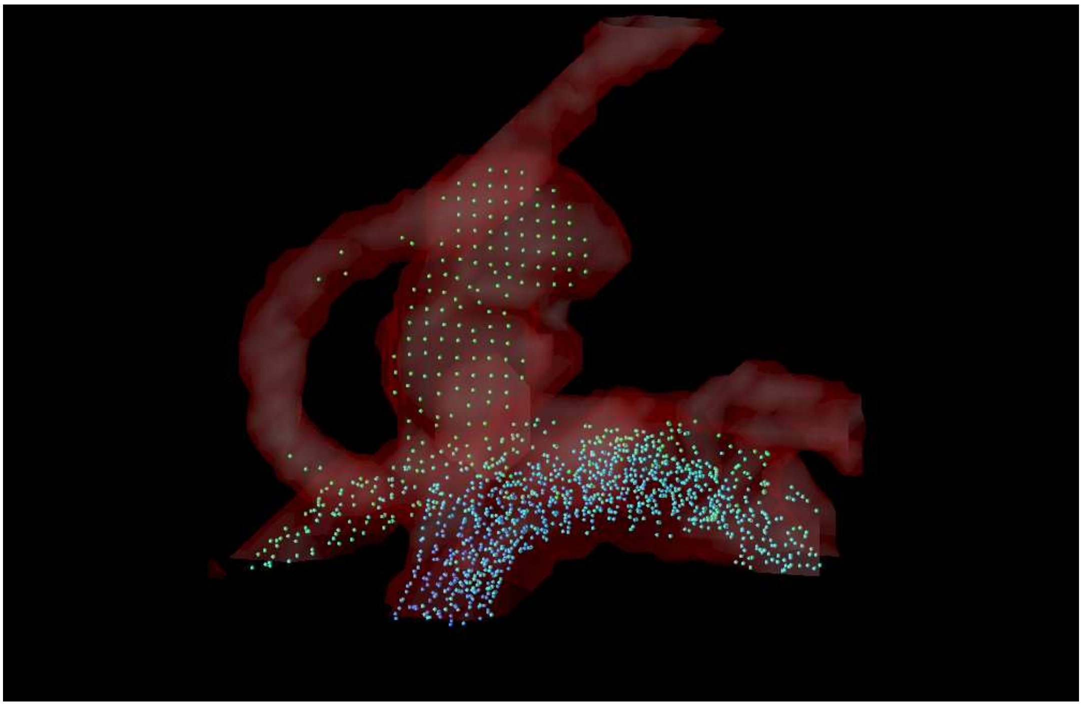
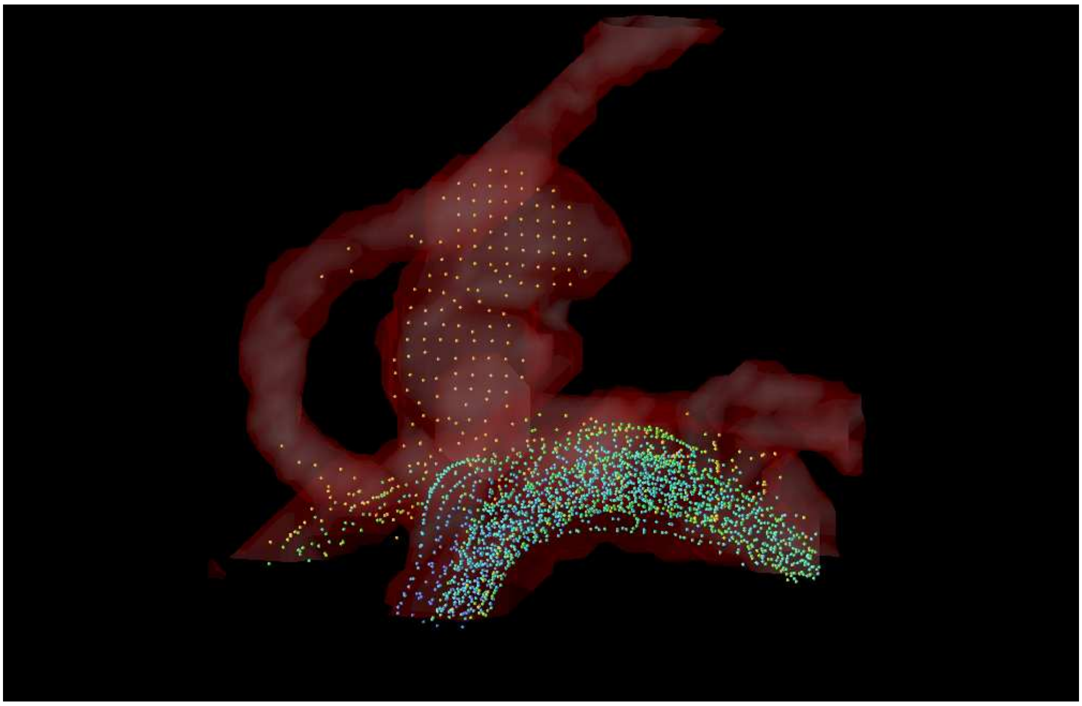


Figure 5.18: Relevant time steps for particles tracing at the inflow pulse



(a) $t = t_0$



(b) $t = t_1$

Figure 5.19: Example of particle tracing at different points of the pulse: Red colors show particles with a long time in the geometry, blue with a short duration

6 Conclusion

The simulation of the hemodynamic situation in the human brain is possible with techniques presented in this thesis. A Lattice Boltzmann method was implemented and adapted for the special geometry of the vessels in the brain. Due to the improved data structure, the requirements regarding memory and computational costs could be reduced significantly. Important geometries based on real patient data were simulated and the results were presented. The hemodynamic situation, like velocity, pressure or stress was visualized with free software tools and discussed. However, the data produced by modern diagnose techniques has to be processed and cleaned for a fluid simulation. For this reason algorithms like a data cleaning, or a hole filling algorithm were implemented and used. Even with these improvements, some adaption could not be done automatically and would have to be done by hand.

With the given parameters, especially high inflow velocities, the Lattice Boltzmann method works at its limits, meaning the simulation is near an unstable region. The parameters result in large ω values, which can only be reduced by decreasing the lattice cell size. For this reason, a refining was implemented. Unfortunately, a refining with a factor of 2 enlarges the memory requirements with a factor of 8 in $3D$. This strictly limits the simulation possibilities. Furthermore, a reduction of the cell size also reduces the time step, which leads to long simulation times. To decrease the omega value, it would be possible to implement a turbulence model (see [TR05]), which increases the viscosity to smooth the swirls. However, in the human brain, no turbulence should occur, thus a turbulence model would give a more stable method, but also disturbs the quality of the results. Another possibility to minimize memory and computational costs is just to reduce the simulation domain, so simulate the points of interest only. This sounds quite easy, but cutting out the region of interest in $3D$ and setting the correct inflow values is a tricky task. But this seems necessary, since the simulation time for a relevant physical time is too long for clinical use during surgery.

Other possibilities for improvement are to use more accurate boundary conditions, or to implement moving walls, for example with the immersed boundary method (see for example [Pes02]). This would add a relevant feature to the simulation and result in more accurate values for pressure, velocity and stress. Furthermore, an implementation of the simulation in clinical environment like InSpace seems possible. Here, doctors would also be able to do surgery planning on the PC by setting stents or coils into the artery and try to optimize the hemodynamic situation.

In summary, a simulation of bloodflow in the human brain is possible with methods presented in this thesis. To do a pre-surgery planning, some improvements have to be done, but the vision of giving the doctors relevant additional information seems possible in the near future.

7 Acknowledgments

I would like to thank Professor Dr. Ulrich Rüde and M.Sc. Klaus Iglberger for the support and ideas. Furthermore, I give special thanks to Professor Dr. Arnd Dörfler and Dr. Gregor Richter for the support with clinical data and information and Marcus Prümmer and Eva Kollorz for preparing the data. Thanks also to Christian Feichtinger, Carolin Körner, Harald Köstler, Christoph Freundl, Nils Thürey and Dr. Kamen Beronov for their help with several problems and questions and Marianne Raschke for the spell check. Additionally, thanks to Dany for her patience.

List of Figures

List of Tables

Index

Bibliography

[Art03] A. Artoli. *Mesoscopic Computational Haemodynamics*. PhD thesis, University of Amsterdam, 2003.

[Bee05] M. H. Beers. The Merck manual of medical information – Second home edition, online version, www.merck.com/mmhe/index.html, 2005.

[Bra06] Cerebral (Brain) Aneurysms, www.brainavm.uhnres.utoronto.ca/malformations/cerebral_aneurysms_index.htm, 2006.

[Bui97] J. M. Buick. *Lattice Boltzmann Methods in Interfacial Wave Modelling*. PhD thesis, University of Edinburgh, 1997.

[Cer05] Cerebral Aneurysm Fact Sheet, www.ninds.nih.gov/disorders/cerebral_aneurysm/detail_cerebral_aneurysm.htm, 2005.

[Coi05] Image Gallery – Radiological Society of North America, www.radiologyinfo.org/photocat/gallery1.cfm, 2005.

[DE06] A. Dörfler and T. Engelhorn. MR- und CT-Angiographie bei zerebrovaskulären Erkrankungen: Technik, Indikation und klinische Anwendung. *Radiologie up2date*, 6(101055S2006925018):49–64, 2006.

[Dio03] J. E. Dion. Brain Aneurysms Successfully Treated Without Open Surgery, www.rsna.org/Media/briefings/2003/upload/Dion.ppt, 2003.

[DSA06] Leonardo Workstation Images – Siemens AG, www.medical.siemens.com/webapp/wcs/stores/servlet/psproductimagedisplay?productid=16732&storeid=10001&langid=-1&catalogid=-1&cattree=100001,12764,12751, 2006.

[Dur03] F. Durst. Grundlagen der Strömungsmechanik - Vorlesungsskript, 2003.

[Ela00] Plasma viscosity and blood viscoelasticity, www.vilastic.com/tech10.html, 2000.

[Eme06] Aneurysms, Brain, www.emedicinehealth.com/aneurysm_brain/article_em.htm, 2006.

[fE06] Copenhagen: WHO Regional Office for Europe. Highlights on health in Germany 2004, www.euro.who.int/document/e88527.pdf. Technical report, World Health Organization, 2006.

[GTK06] S. Geller, J. Toelke, and M. Krafczyek. Lattice-Boltzmann Method on quadtree type grids for Fluid-Structure-Interaction. In H.-J. Bungartz and M. Schäfer, editors, Fluid-Structure Interaction: Modelling, Simulation, Optimisation Springer Verlag, 2006, 2006.

[HKR05] F. Hülsemann, M. Kowarschik, and U. Rüde, editors. *Optimizing Performance of the Lattice Boltzmann Method for Complex Structures on Cache-based Architectures*. ASIM, SCS Publishing House, 2005.

[HNF⁺99] D. Holdsworth, C. Norley, R. Fraynek, D. Steinman, and B. Rutt. Characterization of common carotid artery blood-flow waveforms in normal human subjects. *Physiological Measurement*, 20:219–240, 1999.

[Igl05] K. Iglberger. Lattice Boltzmann Simulation of Flow around moving Particles. Master's thesis, University of Erlangen, 2005.

[Jan06] J. Janus. Subarachnoidalblutungen – Diagnostik und Therapie der Hirnbasisarterien - Aneurysmen, `www.jjanus.de/J__Janus/Neurochirurgie/Aneurysma/aneurysma.html`, 2006.

[Khu06] Dr. V.G. Khurana. Brain Arteries, `www.brain-aneurysm.com/ba1.html`, 2006.

[Kor03] Korea Advanced Institute of Science and Technology. *Blood Flow Simulation toward Actual Application at Hospital*, 2003.

[Ku97] D. Ku. Blood Flow in Arteries. *Annual Review of Fluid Mechanics*, 29:399–434, 1997.

[LM98] S. Laughlin and W. Montanera. Central nervous system imaging – When is CT more appropriate than MRI. *Postgraduate Medicine*, 104(5):73ff, 1998.

[MSYL00] Renwei Mei, Wei Shyy, Dazhi Yu, and Li-Shi Luo. Lattice Boltzmann Method for 3-D Flows with Curved Boundary. *Journal of Computational Physics*, 161:680–699, 2000.

[Pes02] C. Peskin. The immersed boundary method. *Acta Numerica*, 11:479–517, 2002.

[RA01] H. Rubin and J. Atkinson. *Environmental Fluid Mechanics.* Marcel Dexter Inc., 2001.

[Rog99] J. Rogers. Cardiovascular Physiology, `www.nda.ox.ac.uk/wfsa/html/u10/u1002_01.htm`, 1999.

[Rug03] C. Ruge. *Altersabhängigkeit der Hirndurchblutung im Erwachsenenalter: Eine farbduplexsonographische Studie.* PhD thesis, Eberhard-Karls-Universität zu Tübingen, 2003.

[Ste06] Carotis-stenting - Kleiner Eingriff schützt vor Schlaganfall,
`www.aktion-meditech.de/techno_gefaesse.php?id=techno_gefaesse_carotis`, 2006.

[TR05] N. Thürey and U. Rüde. Turbulent Free Surface Flows with the Lattice Boltzmann Method in adaptively coarsened grids. Technical report, University of Erlangen, 2005.