TU

Technische Universität Wien

Dissertation

Hydrodynamic Modelling and Simulation of the Human Arterial Blood Flow

ausgeführt zum Zwecke der Erlangung des akademischen Grades eines
Doktor der technischen Wissenschaften
unter der Leitung von

Univ. Prof. Dipl.-Ing. Dr. Felix Breitenecker
Institut E114
Institut für Analysis und Technische Mathematik

eingereicht an der Technischen Universität Wien
Technisch-naturwissenschaftliche Fakultät

von

Christian Roland Almeder
Matr.Nr.: 9225036
Vogelsanggasse 37/11
A-1050 WIEN

Wien, 8. November 1999

Kurzfassung

In dieser Dissertation wird ein detailliertes Modell der Hydrodynamik des arteriellen Blutflusses im menschlichen Gefäßsystem entwickelt, das die physikalischen Vorgänge im Arteriensystem durch Pulskurven, Fließgeschwindigkeiten und Querschnittsänderungen der Gefäße beschreibt. Dieses Modell ermöglicht die Simulation komplexer Netzwerke im Bezug auf Wellenausbreitung und Wellenreflexion, die ja eine wesentliche Rolle in der Ausformung des arteriellen Druckverlaufes spielen.

Ausgehend von einem vor einigen Jahren von *Martin Suda* an den *Austrian Research Centers Seibersdorf* entwickeltes Modell wird nun ein Modell auf Basis neuer Konzepte, die die Einsatzmöglichkeiten wesentlich erweitern sowie die Qualität der Ergebnisse verbessern, erstellt.

1. Das Modell beruht auf hydrodynamischen Gleichungen, die die Wandelastizität voll berücksichtigen. Dies führt nun zu veränderlichen Gefäßdurchmessern und Schallgeschwindigkeiten in den Arterien, wodurch nun auch Langzeitsimulationen ermöglicht werden und zugleich langfristige an- bzw. absteigende Drucktendenzen des Vorgängermodells eliminiert werden.

2. Das Konzept der fiktiven Arterien (zuvor schon beim statischen Blutflußmodell eingesetzt) wird nun hier um eine dynamische Komponente erweitert, um auf der einen Seite die starken Reflexionen an den ausströmenden Enden des Netzwerkes zu verringern und auf der anderen Seite den peripheren Widerstand und die Mikrozirkulation zu berücksichtigen.

3. Das Modell verwendet mehrere Windkesselersatzmodelle, die eine bessere Beschreibung der Wandelastizität der Aorta ermöglichen. Um auch die Rechenzeit zu verringern, wird ideales Gas in diesen Windkesseln angenommen.

4. Durch eine spezielle Formulierung der Randbedingungen an den Gefäßenden wird die Verbindung mehrerer Arterien an den Verzweigungspunkten wesentlich vereinfacht, indem eine ansonsten notwendige Fallunterscheidung der verschiedenen Fließsituationen vermieden wird.

5. Die Lösung erfolgt erfolgt durch Anwendung der Charakteristikenmethode sowie einer anschließenden Diskretisierung. Eine neue Schrittweitensteuerung im Lösungsalgorithmus verringert die Approximationsfehler und auch

die Effekte der numerischen Dämpfung, denn durch eine dynamische Anpassung der Schrittweite zu jedem Zeitschritt wird immer die maximal mögliche Schrittweite gewählt, ohne dabei die durch die *Courant*-Bedingung vorgegebene Schranke zu überschreiten. Diese Methode erlaubt etwa doppelt so große Schrittweiten als beim Ausgangsmodell.

Diese neuen Konzepte kommen schon teilweise im Modellierungsstadium als auch beim Entwickeln des Lösungsalgorithmus beziehungsweise der Implementierung zum Tragen. Trotz erhöhter Modellkomplexität (fiktive Arterien, variable Schallgeschwindigkeit, ...) kann aufgrund dieser Konzepte eine Verringerung der Rechenzeit erreicht werden.

In dieser Arbeit, unterteilt in acht Kapitel und drei Appendizes, wird zuerst ein kurzer, einleitender Überblick über Modellierungs- und Simulationstechniken sowie auch eine Einführung in die physiologischen Grundlagen bezüglich des menschlichen kardiovaskulären Systems gegeben (Kapitel 1). Es folgt eine Beschreibung des statischen Blutflußmodells (Kapitel 2), das bereits einige wesentliche Merkmale des dynamischen Modells enthält. Es wird auch zur Berechnung von Startwerten für die dynamische Simulation herangezogen. Die grundlegenden Modellgleichungen werden zuerst nur für den Blutfluß in einem einzelnen Gefäß hergeleitet (Kapitel 3), gefolgt von der Entwicklung eines Lösungsalgorithmus (Kapitel 4) durch die Anwendung der Charakteristikenmethode auf die partiellen Differentialgleichungen. Weiters werden die Blutflußmodelle der einzelnen Gefäße miteinander durch gekoppelte Randbedingungen verbunden und das Konzept des Windkessels wird inkludiert, wodurch ein Modell des gesamten Netzes möglich ist (Kapitel 5). Es folgt eine kurze Beschreibung der Implementierung als C++–Programm (Kapitel 6) und weiters die Validierung auf Basis eines 3-Arteriensystems und einige Simulationsexperimente, durchgeführt an realen Daten eines menschlichen Arteriensystems (Kapitel 7). Der Hauptteil wird mit abschließenden Bemerkungen und möglichen zukünftigen Erweiterungen beendet (Kapitel 8). Die drei Appendizes beschreiben die verschiedenen Daten- und Benutzerschnittstellen der Simulationssoftware.

Da die Implementation des Modells in Ansi-C++ vorgenommen wurde, kann das Programm auf unterschiedlichen Plattformen (Microsoft Windows 98/NT, Unix, ...) laufen. Die zusätzliche Matlab-Schnittstelle ist natürlich auch Plattformunabhängig, womit die Software auf den unterschiedlichsten Maschinen eingesezt werden kann. Für die Zukunft ist eine Einbindung dieses dynamischen Modelles in ein schon bestehendes Softwarepaket zur Bypassoptimierung, das auf dem statischen Blutflußmodell beruht, geplant.

Dieses dynamischen Modell des Blutflusses ist durch seine Komplexität an der Grenze der Identifizierbarkeit der Parameter angelangt. In diesem Fall ist eine individuelle Anpassung an Patientendaten nur mehr schwer möglich, da ein patientenspezifisches Modell zur Zeit noch an den Untersuchungs- und Meßmöglichkeiten scheitert. Jedoch können durch die Verwendung standardisierter Daten grundlegende Untersuchungen bezüglich der hydrodynamischen Effekte verschiedenster Herzkreislauferkrankungen, aber auch operativer Eingriffe, durchgeführt werden.

Summary

In this work a detailed model of the hydrodynamics of the arterial blood flow in the human vessel network is developed, which describes the physical processes going on in a network of arteries by means of pressure pulse, flow velocity, and cross-sectional changes. This blood flow model enables the simulation of complex networks concerning wave propagations and the large numbers of reflections, which play a significant role in forming the arterial pulse.

Starting from a model developed by *Martin Suda* from the *Austrian Research Centers Seibersdorf* some years ago, a model based on several new concepts is developed in order to extend the model capabilities and to improve the quality of the results.

1. The model is based on hydrodynamic equations, which take into consideration the elasticity of the arterial walls. This leads to variable diameters and variable wave speeds in the arteries, and in fact gives the opportunity to perform long term simulations avoiding pressure increase or decrease tendencies, which occurred in the former model.

2. The concept of fictitious arteries (used for the static blood flow model) is extended by a dynamical component and was applied to this model in order to reduce the strong reflections at the outstream ends of the network and to take into consideration the peripheral resistance and the microcirculation.

3. This model contains several air chambers as substitute models for the aortic compliance. In order to reduce the computational time ideal gas is assumed in the air chambers instead of normal air.

4. A special formulation of the boundary conditions at the artery ends simplifies the connection of several arteries at the branching points avoiding a distinction of the different flow situations.

5. The solution is developed by using the method of characteristics and a following discretization. A new time step control in the solving algorithm decreases approximation errors and numerical damping effects, because of the dynamical adaption of the step size from time step to time step, the maximal one can be chosen without violating the limitation of the *Courant* condition. So this method allows time steps about twice that in the basic model.

Those new concepts are partly used in the modelling stadium and also during the development of the solving algorithm respectively during the implementation. In spite of the higher complexity of the model (fictitious arteries, variable wave speeds, ...) the computational time could be reduced because of the new concepts.

This dissertation is split up into eight chapters and three appendices. First a short introductory overview about modelling and simulation techniques and a survey of physiological facts concerning the human cardiovascular system are given (chapter 1). This is followed by the description of a static blood flow model (chapter 2) including already some of the main features of the dynamic one. It is also used to calculate initial values necessary to start the dynamic simulation. The basic model equations are derived first for the blood flow in a single artery (chapter 3) and a solution algorithm is obtained (chapter 4) by applying the method of characteristics on the partial differential equations. Furthermore those single artery models are connected using combined boundary conditions and the concept of the air chamber is presented (chapter 5). A brief description of the implementation as a C++ program is given (chapter 6) followed by the validation on the basis of 3-artery models and some simulation experiments on real human arterial network data (chapter 7). It all ends up with some conclusions and remarks and an outlook for future extensions of the model (chapter 8). The three appendices added at the end describe the different data and user interfaces of the simulation software.

Since the implementation of the model was done in standard ANSI-C++, the program can be executed on different operating systems (Microsoft Windows 98/NT, Unix, ...). The additional MATLAB-interface is also platform-independent, so the software can be used on different machines. As a future extension it is planned to embed the model into a software package for bypass optimization, which is based on the static blood flow model.

Due to its complexity this hydrodynamic model of the blood flow has reached the edge concerning the matters of identification of the parameters. In this case the individual adaption on patient data is very difficult, because a patient-specific model fails due to examination and measurement methods. But the usage of standardized data allows general investigations of hydrodynamic effects of different cardiovascular diseases or different operation methods.

Preface

First of all I want to thank all the people who supported me and my work in order to write this dissertation. I reward *Felix Breitenecker* at the Vienna University of Technology for being my tutor and mentor for the last four years and for giving me the opportunity to take part in this project. He provided me always with helping and constructive advises.

Also *Martin Suda* from Austrian Research Centers Seibersdorf (ARCS), who had done the preliminary work on this field, spent a lot of time with me analyzing the equations step for step and evaluating the results in innumerable discussions. But also *Johann Krocza*, the project manager, also from ARCS gave me much support in my work. All in all this paper and the computer programs are the results of a fantastic teamwork.

Special thanks are regarded to my girl friend Julia, who always tried to give me a little bit of distraction, but as well supported me and my work all the time. Furthermore I want to thank my family who made my studies at the TU Vienna possible and my friends who saved me from becoming overwrought and had time for fun and leisure as a compensation for my work.

Christian Almeder

Notation

For all parameters listed below the SI units are used.

A	Sectional area of artery
$\boldsymbol{A}_{\mathbf{a}}$	Admittance matrix for the arteries
$\boldsymbol{A}_{\mathbf{n}}$	Admittance matrix for the nodes
a	Wave speed
a_0	Stationary wave speed
a_x	Acceleration in x-direction
$\mathcal{C}$	Constant of the gas equation
D	Diameter of artery
$\boldsymbol{D}$	Vector of artery diameters
E	Modulus of elasticity
F	Function of dimensionless radius
f	Force
f_g	Gravity force
f_x	Force in x-direction
g	Gravitational acceleration
H	pressure head
H_0	Stationary pressure head
H_{b}	Barometric pressure head in air chamber
H_{stat}	Hydrostatic pressure head in air chamber
h	Thickness of the wall of the artery
h_0	Stationary thickness of the wall
$\boldsymbol{I}_{\mathbf{na}}$	Node-artery-incidence matrix
K	Bulk modulus of elasticity
L	Length of artery
$\boldsymbol{L}$	Vector of artery lengths
m	Mass
N	Number of segments
n	Isentropic exponent
n_{a}	Number of arteries
n_{n}	Number of nodes
p	Pressure
$\boldsymbol{p}_{\mathbf{a}}$	Vector of pressure differences due to friction in the arteries

p_0	Stationary pressure
$\boldsymbol{p}_{\mathbf{drop}}$	Vector of pressure drop in the arteries
p_{ref}	Reference pressure
$\boldsymbol{p}_{\mathbf{hyd}}$	Vector of hydrostatic pressure in the arteries
$\boldsymbol{p}_{\mathbf{n}}$	Vector of pressure at the nodes
Q	Flow
$\boldsymbol{Q}_{\mathbf{a}}$	Vector of blood flows in the arteries
$\boldsymbol{Q}_{\mathbf{n}}$	Vector of blood inflow at the nodes
Q_{ref}	Reference flow
R	Radius of artery
R_0	Stationary radius
Re	Reynolds number
$\boldsymbol{Re}$	Vector of Reynolds numbers in the arteries
t	Time
V	Flow velocity
$\boldsymbol{V}$	Vector of flow velocities in the arteries
V_0	Stationary velocity
$\mathcal{V}$	Gas volume
$\mathcal{V}_{\text{stat}}$	Static gas volume in air chamber
x	Distance along an artery
z	Geodetic height
$\boldsymbol{z}$	Vector of geodetic heights of the nodes
z_0	Static geodetic height of fluid level in air chamber
α	Angle between artery axis and the horizontal
ε_1	Axial strain
ε_2	Lateral strain
ε_{abs}	Absolute wall roughness
ε_{rel}	Relative wall roughness
λ	Friction factor
$\boldsymbol{\lambda}$	Vector of friction factors in the arteries
ν	Kinematic viscosity
μ	Poisson's quotient
ρ	Mass density
σ_1	Axial stress
σ_2	Lateral stress
τ	Shear stress
η	Dynamic viscosity

Contents

List of Figures

List of Tables

Chapter 1

Introduction

The book I was reading was this book I took out of the library by mistake. They gave me the wrong book, and I didn't notice it till I got back to my room.
J. D. Sallinger, The Catcher in the Rye.

1.1 Fundamentals of Modelling and Simulation in Medicine

This section gives a very short survey of modelling and simulation. It contains the basic features of mathematical modelling and computer simulation. That quick overview should only give an introductory motivation for this work.

The purpose of modelling and simulation is the investigation of real world systems. They are utilized in many fields of applications, especially technical sciences, physics, chemistry, economy, medicine, biology and others.

From the view of a scientific-engineering approach the human-world interaction is based on abstraction and formal models. Through observation and experimentation the scientist attempts to create abstract representations and laws that formalize verified hypotheses concerning real world phenomena. The formal models are only useful if they represent the essential features of the real system. After the model-building stage simulation experiments based on the models are performed and the results are interpreted on the real world system (see figure 1.1).

There are two possible motivations for the development of a mathematical model. First the investigation of a real world system and the creation of a formal model increases the insight of the structure of the problem. This provides a better understanding of the system and the causality between the input and the output. The second goal is the development of a control strategy for the real system based on the results of the simulation experiments. In this case it is more important that the model emulates the behaviour of the system, but it is not necessary that it represents the internal structure.

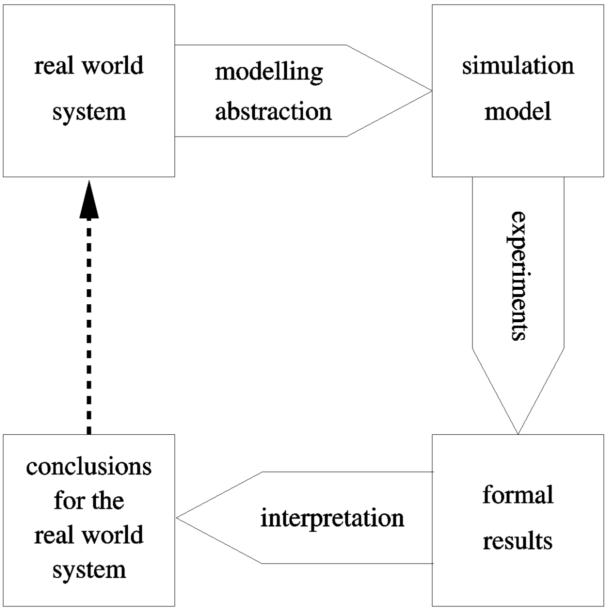


Figure 1.1: Circulation of modelling and simulation

In many cases it is not possible to create a model that fulfils both demands at the same time, because very often a detailed modelling of the structure leads to large and complex models which necessitate large-scale calculations. An example for this aspect is the *Navier-Stokes* equation. This set of partial differential equation describes the relation between the dynamical flow of a fluid and the friction. The structure of this problem is modelled in a very detailed way, but the computation of a solution is very difficult.

Nowadays in most cases simulation means computer simulation. That is the development of a mathematical model and its implementation on the computer. For a computer simulation the following steps are necessary:

1. **Formulation of the problem:** It is necessary to have clear and precise description of the problem and the requested results.

2. **Mathematical (formal) model:** The problem has to be abstracted and formalized by using equations, graphs, etc.

3. **Mathematical transformations:** The model equations are simplified and transformed into a computer equitable form.

4. **Numerical analysis:** The solution algorithms have to be chosen with respect to preciseness and expenditure.

5. **Implementation:** That is the development of a computer program that performs the model calculations.

6. **Verification:** The formal model, the mathematical formulation and the program are checked and compared.

7. **Identification:** The parameters of the model are determined from measurements.

8. **Validation:** It is checked if the computer program represents the real system.

9. **Simulation strategy:** The number and the type of simulation experiments have to be chosen.

10. **Simulation experiments:** The experiments are performed.

11. **Documentation:** A detailed description of the simulation results and the model which they are based on is necessary.

12. **Interpretation:** The results of the simulation experiments are interpreted and conclusions for the real world system are provided.

This 12 steps are not performed in an ordered way, but there are several iterations. For example during the validation stage it is recognized that the computer program does not represent the real world system in the intended way, then maybe the identification of the parameters was wrong or there are mistakes in the formal model. So it is necessary to go back to that stage where the failure was made and do it all again.

In human medicine modelling and simulation is limited by the complexity of the investigated biological process and by the difficulties of gaining data for an individual patient. The first problem — the complexity of the process — complicates the first step of modelling , because very often it is not clear where to draw the line between the parts that must be included in the model and those that are negligible for a certain problem. Therefore in medical modelling a combination of black and white box models are used very often. The second problem — the limited availability of data — demands sometimes a down sizing of too complex models to smaller ones, which can be identified with less data. But the unavailability of data is not the only problem, because some data can only be gained from in-vivo measurements, which are very complicated in most cases, so that in fact only a few measurements can be performed, and those are often imprecise.

So there is a thin line between complex models, which represent the structural information, but need too many parameters for identification, and simplified models, which can be identified using easy available data at the cost of structural information. Therefore the use of complex medical simulation not only for research but also for assisting physicians in their daily work, depends very much on the further development of measuring instruments and diagnostic tools. [Bra97]

1.2 Problem Formulation

This work is part of a large research project performed in a collaboration between the *Austrian Research Centers Seibersdorf* and the *Vienna University of Technology.* This project is concerned with modelling and simulation in medicine, especially with the development of models for the respiratory and the cardiovascular system.

This thesis describes and summarizes the work on the following problems:

- revision and further development of a one-dimensional dynamic blood flow model for the human arterial system.
- adaptations to make long time simulations possible.
- new concepts for the air chamber phenomenon in the aorta with the possibility of including several small air chambers along the vessel.
- new formulation of the boundary conditions at artery connections.
- implementation of the model as C++ program on different platforms.
- building up interfaces to other programs (e.g. MATLAB) for data pre- and post-processing.

The results of this work shall be utilized in software packages for research and clinical use. Another aim is to produce a compilation of different models for training and teaching of medical students.

1.3 Preliminary Studies

In the late 1980's *Martin Suda* from the *Austrian Research Centre Seibersdorf* developed a static model for water supply networks, which was implemented as a FORTRAN program. Further developments of this work leaded to a static model of the blood flow in the human arterial system in the years 1993 to 1995. [Sud93, Sud95] With this program several tests were performed using pre- and postoperative measurements for validation, which showed only a mean difference between calculated and measured data of up to 10%.

As part of the project (see section 1.2) this model was touched up and implemented in a software tool for vascular surgery, which can be used for preoperative tests on different operation methods. This retouching and new implementation of the model was performed as a diploma thesis. A detailed description of this work can be found in chapter 2. [Alm97a]

In parallel to the hydrostatic model *Martin Suda* developed also a hydrodynamic model of the human arterial system based on a dynamic model for water supply networks, which was implemented as FORTRAN program, too. [Ede88, Sud87a, Sud87b] Due to the time consuming calculations the simulation experiments where limited to about 2 heart beats (that is about 1.6 sec.). Therefore only a few experiments where performed with an arterial network of an arm and with a rough network of the whole body (about 140 arteries).

The research work of this doctoral theses is based mainly on those works described above. It is a further development towards a detailed model of the hemodynamics in the human arterial system aiming at a model that is also small enough so that it may be identified individually on different persons, if there are sufficient examination and measurement methods are available. [Anl71, Bal89, McD74, Jun72]

1.4 Physiology of the Human Arterial System

The mammalian cardiovascular system in general and the human system in particular can be described briefly as follows. The heart, the driving force for the blood flow, consists of four chambers arranged in two pairs. On both sides the blood enters first the atriums, the smaller thin-walled chambers. Then it flows through valved orifices into the thick-walled muscular ventricles. The blood leaves the heart through valves either into the aorta (left ventricle) and furthermore into the systemic circulation or into the pulmonary artery (right ventricle) and then into the pulmonary circulation.

In the systemic circulation the blood flows through the aorta, then through large arteries , which branch off, then through smaller ones, and so on for many subdivisions. The final subdivisions of the arterial network are the arterioles with about 30–100μm of internal diameter followed by the capillaries down to 4–5μm of diameter. Although most parts of the arterial network are tree structured, there are several meshes. Across the walls of the capillaries the exchange of fluids and metabolites between blood and the tissue takes place by the way of diffusion. Afterwards the blood is collected through the smallest veins (venules), and then through a converging system of increasingly large veins. Finally the blood flows through the superior and inferior venae cavae into the right atrium.[1]

The pulmonary circulation starts at the pulmonary artery branching off till the pulmonary capillaries in the walls of the alveoles in the lung. After the gas exchange the reoxygenated blood is returned to the left atrium via the pulmonary veins. [Ped80]

The blood volume in the human body is about 5l (6-8% of the weight). Only about 20% of the blood is in the left heart and the systemic arteries, the rest is in the right heart the veins and the pulmonary circulation. This part functions as a buffer for the blood so that roughly the same amount of blood is in the arterial system all the time. [Sil83, Gan74]

[1] An exception is the coronary circulation, which drains directly into the right atrium.

Wall Structure of the Systemic Arteries

The arterial wall is divided up into three layers, the innermost intima, the media, and the outermost adventitia. Usually the media, which is the thickest part of the artery wall, dominates the elastic behavior. In large arteries (aorta, a. subclavia, a. carotis communis, aa. iliacae) the media consists of multiple layers of elastin and occasional smooth muscle cells. The media of smaller arteries contains mainly spirally wounded muscle cells.

The ratio of wall thickness to internal diameter for large arteries (more than 1mm) varies only between 0.06 and 0.08, but it increases in small muscular arteries and arterioles up to 0.4.

Most of the arteries in the human body are longitudinal constrained. If an artery is removed it shrinks by up to 40% of its in-vivo length. Hence the modulus of elasticity in the longitudinal direction is very high, so that the arteries do not stretch or shrink longitudinally during a pulse wave.

For the radial deformation the dynamic modulus of elasticity increases with the stress, but is constant over a wide range of pulse frequency. So the radial elasticity of the arteries is independent of the heart rate. [Ped80, Bus82]

Blood

The blood is a suspension of formed elements in plasma. The plasma, which makes up to 55% of the volume, is a solution of large molecules. Nevertheless, concerning the scales of motion and the rates of shear normally encountered in the vessels, plasma is a homogeneous Newtonian fluid of kinematic viscosity $1.2–1.6 \cdot 10^{-6}\frac{\mathrm{m}^2}{\mathrm{s}}$ at 37°C.

The other 45% of the blood volume consists mainly of red cells, white cells, and platelets. There are about 5 million red cells per cubic millimeter of blood, so these cells dominate the mechanical properties of blood. They have the form of flexible biconcave discs of about 8 μm diameter and 1 μm thickness at the center and 2-3 μm at the edge. The white cells are slightly larger than the red ones, but equally deformable and far less numerous (5000-10000 per cubic millimeter blood). Therefore they have a negligible influence on the dynamical behaviour. The number of platelets in blood is between red and white cells (half a million per cubic millimeter blood), but they are very small (2-4 μm of diameter) and hence dynamically unimportant as long as they do not react with foreign substances, because they play a significant role in the process of blood clotting.

In large vessels the microstructure of the blood is very small in comparison with the diameter. There the whole blood can be assumed as homogeneous. Under normal flow conditions it is also a good approximation to assume the blood as a Newtonian fluid with density $1052\frac{\mathrm{kg}}{\mathrm{m}^3}$ and kinematic viscosity of about $4 \cdot 10^{-6}\frac{\mathrm{m}^2}{\mathrm{s}}$. [Ped80]

Pulse

The right and left ventricles produce a pulsatile blood flow out of the heart. During the systole, which endures 0.2-0.3s at rest, the blood pressure in the left ventricle is higher than in the aorta, so that the valve is open and the blood flows into the arterial system. Before the valve closes there is a little flow back into the heart for a short time. Then the diastole follows, which endures 0.5-0.7s at rest, during that time no blood flows into the aorta.

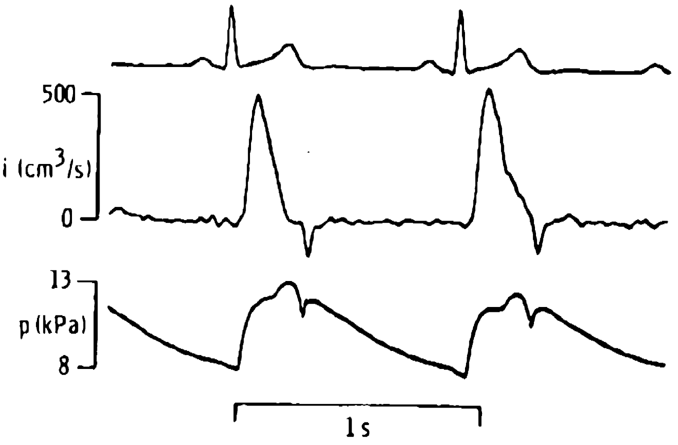


Figure 1.2: Pressure (p) and flow (i) in the aorta ascendens with EKG curve (from *Murgo, J.P. et al.: Circulation 62 (1980), 105*)

Already during the systole the arterial blood pressure differs from the flow curve, because of the positive reflected pulse waves, which increase the pressure to a higher level. So the highest pressure (systolic pressure) is reached at the end of the systole. The kink in the aortic pressure curve (dicrotic notch) is caused by the back-flow into the heart. During the diastole the blood pressure decreases slowly to its minimum (diastolic pressure).

Chapter 2

Hydrostatic Model of the Arterial System

Dimidium facti, qui coepit, habet.

This chapter gives a short survey over a static model of the blood flow in the arterial network, which is also included in the dynamic model for generating initial values and for some precalculations to improve the dynamic computation. A more detailed description of the model can be found in [Alm97a, Alm98b].

The main purpose of this model is to describe the relationship of the morphology and the hydraulics in the human arterial system. It offers methods to calculate mean flow velocity, mean flux, flow direction and blood pressure in the arteries. The results should be accurate enough to make fundamental statements on blood supply in specific parts of the arterial system and on the flow velocity through the arteries.

This model is implemented as C++ program in a software package which can be used as simulation environment for different tests on the static blood flow in different parts of the arterial system. [Alm97b, Alm98a]

2.1 Assumptions

The physiological facts in section 1.4 are the motivation for the identification of the arterial system with a pipe network. Every segment of an artery between two ramification points is represented by a pipe and every ramification point by a hydraulic node. The arteries have a cylindrical fixed shape and consist of a solid material so that no (blood) is lost through the walls. Besides that it is assumed that the arteries are hydraulic smooth, that means that the friction between the blood and the walls of the arteries is low. This fact is important for the calculation of the pressure drop in the arteries.

It is also assumed that blood in larger vessels is a Newtonian fluid. In arteries with a diameter smaller than 0.1mm the blood cells are too large in relation to the diameter so that the characteristics of blood differ too much from that of a Newtonian fluid. Therefore the model must not include such small vessels.

The input into the network is assumed to be a steady state flow of blood at one node, the *reference node*. That means the pulsatile flow produced by the heart is replaced by an average fixed flow, because only mean values for flow and flow velocity are required. Disregarding the pulse of the heart makes it possible to neglect the elasticity of the arteries and to model them as fixed shaped pipes.

But this model is not complete, because it is not possible to determine how much fluid flows off at some node. But these values are necessary for calculating the flow and pressure within the network.

From a physiological point of view the blood that flows out of the nodes supplies other parts of the arterial network that are not taken into account in the model – especially the capillary vessels – or it changes to the venous system. In addition to that another physiological aspect is ignored – the microcirculation – has to be modelled, too. Microcirculation means that if an artery is closed the vessels behind are supplied with blood from other parts of the arterial system. The blood flows through these vessels in the opposite direction as usual. That means that there is a connection of the arteries not only through the reference node, but also through capillary vessels.

If those facts are taken into consideration, the model has to be extended. New (fictitious) arteries are added to the network to collect the blood at an additional (collecting) node (see figure 2.1). This collecting node has the same geodetical height as the reference node and the outflow of fluid is equal to the inflow at the reference node. All fictitious arteries have the same length but variable diameters. Those diameters are used to regulate the resistance of other parts of the arterial network and the resistance of the venous system.

Now the new network is similar to a closed system. At one node blood enters and at another node the same amount of blood leaves the system, within the network no blood is lost or added. The diameters of the fictitious arteries regulate the outflow of the nodes. Therefore they are properties of these nodes – the outflow diameters.

If a part of the human arterial network is modelled by using the described way, it leads to a pipe network with the following properties:

- The network consists of pipes which represent the arteries and hydraulic nodes for connecting those pipes. The nodes represent the ramification points of the arteries.
- For every artery its length and its diameter is known and constant.
- Only at the reference node blood flows into the system independent of time. The flow and the pressure at this node is known.

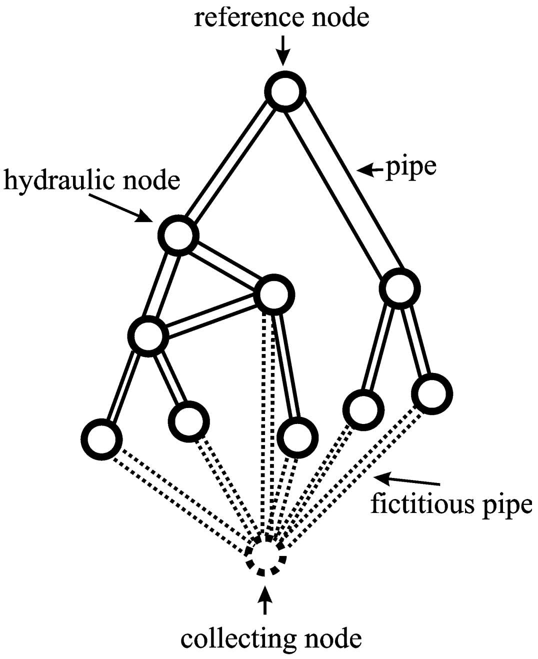


Figure 2.1: Representation of the arterial system as a pipe network including the fictitious arteries

- Blood leaves the network only through the collecting node.
- All fictitious arteries have the same length, but variable diameters.

2.2 Physics of the Hydrostatic Flow in an Artery

In previous sections it was assumed that blood is a Newtonian fluid. Therefore all derivations of the following sections correspond to this assumption.

Newtonian fluids are characterized by *Newton's law of viscosity*:

$$\tau = \eta \frac{dV}{dn} \tag{2.1}$$

where τ is the shear stress, η is the dynamic viscosity and $\frac{dV}{dn}$ is the derivative of the velocity with respect to the orthogonal vector. [Boe93]

There are two different types of flow of a Newtonian fluid.

Laminar Flow: The fluid layers move side-by-side in an ordered way with constant relative velocities.

Turbulent Flow: Turbulence and whirls occur. The fluid seems to flow in an unordered, chaotic way.

The laminar and turbulent flow produce different velocity profiles in the arteries. (see figure 2.2).

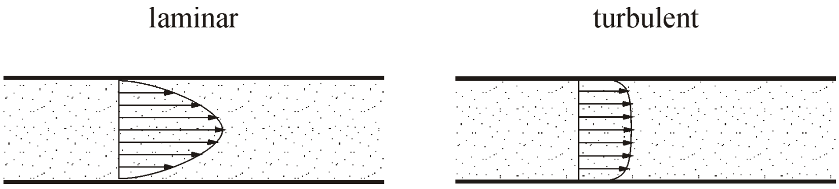


Figure 2.2: Velocity profiles

The laminar flow has a parabolic velocity profile. Near the artery axis the velocity is maximal, near the walls it is very small because of the friction between the blood and the wall. For turbulent flow the internal friction of the blood is much higher because of the turbulence. Therefore the velocity profile is flattened.

The Reynold number

$$Re = \frac{VD}{\nu} \tag{2.2}$$

is used to distinguish between laminar and turbulent cases. It is a combination of the mean flow velocity V, the artery diameter D and the kinematic viscosity ν. If the Reynold number is below 2300, the flow is laminar. If it is above 3000, then the flow is turbulent. The change from laminar to turbulent does not occur abruptly. It depends on the fluid, but usually there is a transition range between 2300 and 3000.

The friction between the blood and the artery walls and the internal friction of the blood causes a pressure drop when the blood flows through an artery. This pressure drop Δp between the start and the end node of a horizontal artery is given by the *Darcy-Weisbach* equation [Boe93, Zoe82, Fox85]

$$\Delta p = \frac{\rho L V^2 \lambda}{2D}. \tag{2.3}$$

If start and end node of the artery are not on the same geodetical height, then the hydrostatic pressure has to be added. So the pressure difference between start node (node 1) and end node (node 2) is

$$p_2 - p_1 = (z_1 - z_2)\rho g - \Delta p, \tag{2.4}$$

where p_1 and p_2 are the pressure values and z_1 and z_2 the geodetical heights of the nodes.

In the laminar case the friction factor is independent of the roughness of the arterial walls, because a layer of blood covers the unevennesses. So the friction is given by

$$\lambda = \frac{64}{Re}. \tag{2.5}$$

In the case of turbulent flow – this is for Reynold numbers greater than 3000 – the roughness has much influence. There are three different kinds of this influence depending on the relative roughness. [Fun84, Zoe82]

Hydraulic Smooth: The relative roughness has no influence, that means it is very small ($\varepsilon_{\text{rel}} < 10^{-6}$). So the wall roughness is covered by a layer of blood. Only the high flow velocity causes turbulences. An approximation formula due to *Colebrook* and *White* is

$$\frac{1}{\sqrt{\lambda}} = -2.0 \lg \Big(\frac{2.51}{\sqrt{\lambda} Re} \Big). \tag{2.6}$$

Transition Range of Roughness: This type occurs if $10^{-6} < \varepsilon_{\text{rel}} < \frac{200}{Re\sqrt{\lambda}}$. The turbulences are caused by high velocities and by the roughness. Now the friction factor depends on the Reynold number and on the relative roughness. *Colebrook* and *White* developed also for this case an approximation formula

$$\frac{1}{\sqrt{\lambda}} = -2.0 \lg \Big(\frac{2.51}{\sqrt{\lambda} Re} + \frac{\varepsilon_{\text{rel}}}{3.72} \Big). \tag{2.7}$$

Hydraulic Rough: In the fully rough case ($\frac{200}{Re\sqrt{\lambda}} < \varepsilon_{\text{rel}}$) the turbulences in the blood are dominated by the roughness of the wall. So the friction factor depends only on the relative roughness

$$\frac{1}{\sqrt{\lambda}} = 2.0 \lg \frac{3.72}{\varepsilon_{\text{rel}}}. \tag{2.8}$$

For the model of the arterial network it was assumed that the arteries are hydraulic smooth, but equation (2.6) cannot be solved with respect to λ. For the implementation of the model it is necessary to interpolate or to use another approximation formula, such as that of *Blasius*

$$\lambda = \frac{0.3164}{\sqrt[4]{Re}}, \tag{2.9}$$

which is valid for $Re < 10^5$.

Very seldom the blood flow in arteries reaches the turbulent range, therefore only an approximation formula for a $3000 < Re < 20000$ is necessary. In this range the *Blasius* formula deviates more than 4 percent from the *Colebrook-White* formula which is the more precise one. Using a similar function and fitting the parameters to the formula (2.6) gives

$$\lambda = \frac{0.3942}{Re^{0.2761}}, \tag{2.10}$$

which has only a 1 percent deviation.

For Reynold numbers between 2300 and 3000 the flow is not laminar, but also not fully turbulent. Measurements have shown a strong increase of the friction in this range from the low laminar friction to the high turbulent friction. In this model it is assumed that the friction is a linear function of the Reynold number, so that the whole function is continuous (see figure 2.3). In this transition range the friction factor is

$$\lambda = 2.2 \cdot 10^{-5} \cdot Re - 0.0228 \tag{2.11}$$

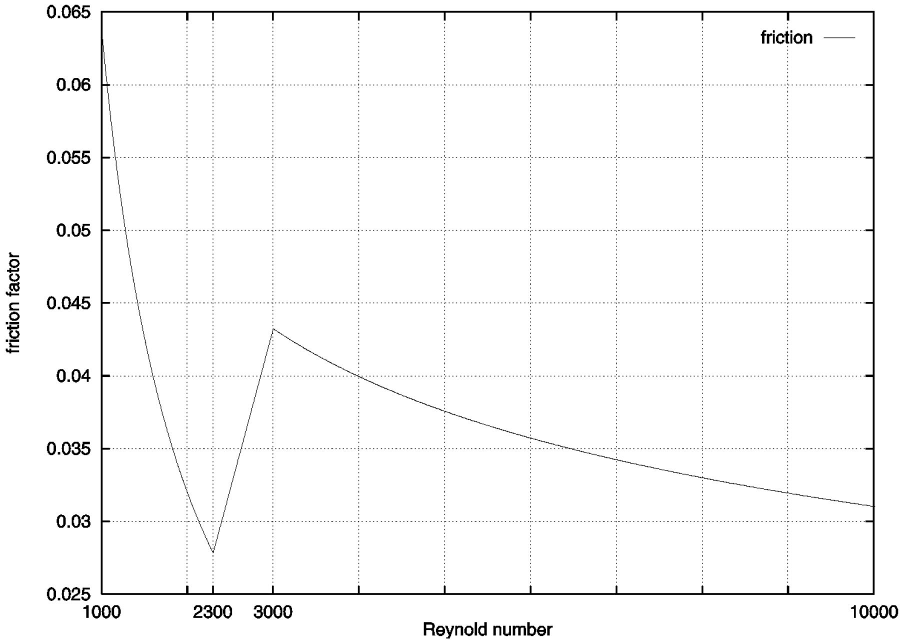


Figure 2.3: The friction factor as a function of the Reynold number

2.3 Mathematical Model

Parameter Description

The arteries and the nodes of the network are numbered starting from 1. The reference node is always the node number 1, the collecting node is always the last node. The fictitious arteries are added at the end.

In the following list the parameters used for the static model are described:

Input parameters: These parameters are used to identify the model and to carry out the calculations.

n_a, n_n: are the number of nodes and the number of arteries in the network.

$\boldsymbol{I}_{\mathbf{na}}$: is the node-artery incidence matrix that holds the information of the network structure, that means that for each artery the start and the end node are specified. The rows of the matrix represent the nodes and the columns the arteries.

$$\boldsymbol{I}_{\mathbf{na}\,i,j} = \begin{cases} 0 & \text{node } i \text{ and artery } j \text{ are not connected,} \\ -1 & \text{artery } j \text{ starts at node } i, \\ 1 & \text{artery } j \text{ ends at node } i. \end{cases} \tag{2.12}$$

$\boldsymbol{L}$: is the vector of the lengths of the arteries measured in meters. The fictitious arteries have all the same length – 0.1m.

$\boldsymbol{D}$: is the vector of the diameters of the arteries.

$\boldsymbol{z}$: is the vector of the geodetical heights of the nodes. Only the height differences between the nodes are important, not the absolute values. The collecting node is set to the same height as the reference node.

$\boldsymbol{Q}_{\mathbf{n}}$: is the vector of the inflows at the nodes, that is the amount of fluid that flows into the network at this node. In this case an outflow of blood is a negative inflow.

$$\boldsymbol{Q}_{\mathbf{n}} = \begin{pmatrix} Q_{\text{ref}} \\ 0 \\ \vdots \\ 0 \\ -Q_{\text{ref}} \end{pmatrix}. \tag{2.13}$$

p_{ref}: is the pressure at the reference node.

Q_{ref}: is the inflow at the reference node.

ρ, ν: are the density and the kinematic viscosity of blood. They have the following values

$$\rho = 1051\,\frac{\text{kg}}{\text{m}^3}, \qquad \nu = 3.8 \cdot 10^{-6}\,\frac{\text{m}^2}{\text{s}} \tag{2.14}$$

Output parameters: These parameters are the results of the simulation or they are by-results of the calculations.

$\boldsymbol{Q}_{\mathbf{a}}$, $\boldsymbol{p}_{\mathbf{a}}$, $\boldsymbol{V}$: are the flow, the pressure difference and the flow velocity of each artery. If the flow and the flow velocity are negative values, then the blood flows in the reverse direction that means from the end to the start node.

$\boldsymbol{p}_{\mathbf{n}}$: is the vector of the pressure values at the nodes.

Working parameters: Working parameters are used for provisional results and to make the model equations more readable.

$\boldsymbol{p}_{\mathbf{hyd}}$ is the hydrostatic pressure for the arteries. Because of the height difference between start and end node the blood in the arteries causes pressure.

$\boldsymbol{p}_{\mathbf{drop}}$: is the pressure drop along the arteries. This pressure drop is caused by the friction and depends on the flow velocity.

$\boldsymbol{Re}$, $\boldsymbol{\lambda}$: are the Reynold numbers and the friction factors for the arteries.

$\boldsymbol{A}_{\mathbf{a}}$, $\boldsymbol{A}_{\mathbf{n}}$: are the admittance matrices for the arteries respectively the nodes. They give a relation between the flow and the pressure in the arteries or at the nodes.

Basic Relations

The model equations are based on 3 different conditions which describe the relation between the flow and the pressure in the network.

Node condition: For every node in the network there must be a balance of the flows. This means that the same amount of blood that flows into the node either from arteries or from the outside of the system must flow out of the node through arteries or out of the network. Hence, the inflow of a node is equal to the difference of the flow of the arteries to and from this node. This can be written as

$$\boldsymbol{Q}_{\mathbf{n}} = -\boldsymbol{I}_{\mathbf{na}}\,\boldsymbol{Q}_{\mathbf{a}}. \tag{2.15}$$

Mesh condition: By analogy to the first condition the pressure loss of a pipe results from the difference of the pressure at the start and the end node.

$$\boldsymbol{p}_{\mathbf{a}} = -\boldsymbol{I}_{\mathbf{na}}^{\top}\boldsymbol{p}_{\mathbf{n}}. \tag{2.16}$$

Hydraulic condition: This condition gives a relation between the pressure and the flow. It is given by the admittance matrix with the equation

$$\boldsymbol{Q}_{\mathbf{a}} = \boldsymbol{A}_{\mathbf{a}}(\boldsymbol{p}_{\mathbf{a}} - \boldsymbol{p}_{\mathbf{hyd}}). \tag{2.17}$$

The admittance matrix is a diagonal matrix of the form

$$\boldsymbol{A}_{\mathbf{a}} = \begin{pmatrix} \frac{\boldsymbol{Q}_{\mathbf{a}1}}{\boldsymbol{p}_{\mathbf{drop}_1}} & 0 & & \\ 0 & \frac{\boldsymbol{Q}_{\mathbf{a}2}}{\boldsymbol{p}_{\mathbf{drop}_2}} & \ddots & \\ & \ddots & \ddots & 0 \\ & & 0 & \frac{\boldsymbol{Q}_{\mathbf{a}n_{\mathrm{a}}}}{\boldsymbol{p}_{\mathbf{drop}_{n_{\mathrm{a}}}}} \end{pmatrix}, \tag{2.18}$$

The pressure drop in an artery is given by equation (2.3), which relates the pressure drop with the flow velocity and therefore with the flow. This implicit equation can be solved with respect to the flow and hence, the admittance matrix $\boldsymbol{A}_{\mathbf{a}}$ can be written as a function of the pressure difference in the arteries and therefore also as a function of the pressure values at the nodes.

Model Equations

Now taking the equations (2.15), (2.16) and (2.17) a relation between the inflow and the node pressures can be derived

$$\boldsymbol{Q}_{\mathbf{n}} = \boldsymbol{I}_{\mathbf{na}}\boldsymbol{A}_{\mathrm{a}}(\boldsymbol{I}_{\mathbf{na}}^{\top}\boldsymbol{p}_{\mathbf{n}} + \boldsymbol{p}_{\mathbf{hyd}}). \tag{2.19}$$

In this equation $\boldsymbol{Q}_{\mathbf{n}}$, $\boldsymbol{I}_{\mathbf{na}}$ and $\boldsymbol{p}_{\mathbf{hyd}}$ are known parameters.

Rewriting (2.19) in function style

$$\boldsymbol{Q}_{\mathbf{n}} = \boldsymbol{I}_{\mathbf{na}}\boldsymbol{A}_{\mathrm{a}}(\boldsymbol{p}_{\mathbf{n}})\Big(\boldsymbol{I}_{\mathbf{na}}^{\top}\boldsymbol{p}_{\mathbf{n}} + \boldsymbol{p}_{\mathbf{hyd}}\Big). \tag{2.20}$$

To make the equation more readable a new matrix is introduced – the admittance matrix of the nodes

$$\boldsymbol{A}_{\mathrm{n}} = \boldsymbol{I}_{\mathbf{na}}\boldsymbol{A}_{\mathrm{a}}\boldsymbol{I}_{\mathbf{na}}^{\top}. \tag{2.21}$$

This matrix is also a function of $\boldsymbol{p}_{\mathbf{n}}$ and it is no longer a diagonal matrix, but a symmetric one. Its structure is determined by the network topology.

$$\boldsymbol{A}_{\mathrm{n}i,j} = \begin{cases} \sum_k \boldsymbol{A}_{\mathrm{a}k,k} & i = j \text{ (artery } k \text{ is connected with node } i\text{)}, \\ -\boldsymbol{A}_{\mathrm{a}k,k} & i \neq j \text{ (artery } k \text{ connects node } i \text{ and node } j\text{)}. \end{cases} \tag{2.22}$$

Now (2.19) can be transformed to

$$\boldsymbol{Q}_{\mathbf{n}} = \boldsymbol{A}_{\mathrm{n}}\boldsymbol{p}_{\mathbf{n}} + \boldsymbol{I}_{\mathbf{na}}\boldsymbol{A}_{\mathrm{a}}\boldsymbol{p}_{\mathbf{hyd}}. \tag{2.23}$$

If the model does not include node heights – that means that the network lies horizontally in a plane – then the hydrostatic pressure $\boldsymbol{p}_{\mathbf{hyd}}$ is zero and the last term disappears

$$\boldsymbol{Q}_{\mathbf{n}} = \boldsymbol{A}_{\mathrm{n}}(\boldsymbol{p}_{\mathbf{n}})\boldsymbol{p}_{\mathbf{n}}. \tag{2.24}$$

The matrix $\boldsymbol{A}_{\mathbf{n}}$ is singular, it has rank $n_{\mathrm{n}} - 1$. This fact occurs because the flow in the network depends only on the pressure differences at the nodes, but not on the absolute values. Therefore the pressure at one node must be known (p_{ref} at the reference node).

Remark: If only laminar flows occur in the whole network, then the admittance matrix is constant and the model equations are a linear system. This case cannot be determined before the calculation, but it is good way to assume a laminar network and solve the linear equations. Afterwards the results can be tested, if all flows in the network are laminar.

2.4 Results of Sensitivity Analysis

Before the model was implemented in a larger software package, several sensitivity tests were performed. For these tests different arterial models (coronary arteries, cerebral arteries, ...) and also small example networks were used.

The question about the influence of the geodetical height of the nodes can be answered by analyzing the model equations. The change of the heights results in changes of the hydrostatic pressure and the node pressure, but it has no influence on the flow through the arteries and therefore the flow velocity keeps unchanged.

The pressure drop along an artery is a linear function of the length. Therefore only small changes of flow and flow velocity are recognized when the lengths are varied. The diameters of the arteries have more influence, because the flow is proportional to the 4th power of the diameter in the laminar case and nearly to the 5th power in the turbulent case. But the changes of the results are only slightly higher than for the length tests.

A special case are the diameters of the fictitious arteries, because they are all connected by one node. Therefore changing those parameters has a significant influence on the flow and flow velocity within the network, but the changes are within acceptable ranges.

In summary the sensitivity of the results is small enough so that a precision of 5% to 10% can be reached. But for detailed information on a sensitivity analysis of the static model see [Alm97a].

2.5 Summary

Martin Suda from ARCS validated this static arterial model using preoperative and postoperative measurements of several patients [Sud95]. These tests showed that it is possible to adapt the networks to the patient data with a deviation of only 5% to 10%. The adapted models could predict the postoperative results with a deviation of 5% to 20%. The precision of the results depended strongly on the quality of the patient data available. But all in all this validation showed that the model can be used in the intended way – as a tool that makes predictions on global postoperative results.

The static model of the human arterial system is designed to meet two expectations. The model includes all necessary physiology details so that the results can be used for general statements about the condition of someone's blood circulation. On the other hand the model is simple enough so that it can be used for different patients. The experimental use of the model has shown that it is appropriate to fulfill both aspect.

Till now only smaller parts of the human arterial system are modelled. But in future if better methods for the measurements of the flow velocity and the structure of the arterial network are available, larger parts or the whole system at once will be modelled. The computer program is capable of such large models.

The numerical tests have shown that there is a strong connection between the quality of the patient data and the quality of the results. So it can be expected that with better measurements also the preciseness will be increased.

Chapter 3

Dynamic Arterial Blood Flow Model

There lay the Count, but looking as if his youth had been half-renewed, for the white hair and the moustache were changed to dark grey; the cheeks were fuller, and the white skin seemed ruby-red underneath; the mouth was redder than ever, for on the lips were gouts of fresh blood, which trickled from the corners of the mouth and ran over the chin and neck.

B. Stoker, Dracula.

3.1 Assumptions

Many of the assumptions for the static model (see section 2.1) are also used for the dynamic model. First of all an artery is assumed to have a cylindrical shape in the case of no pressure difference along the walls. That means only pressure changes can produce a change of the sectional area, but independent of the place along the artery axis. As before no blood is lost through the walls and the arteries are hydraulic smooth.

To get an usable model for the wall elasticity of the arteries the following assumptions are made:

1. The arteries are fixed against longitudinal stretching. The elasticity in the longitudinal direction is very high, especially because of the strong constraint. Therefore the longitudinal movements are negligible in comparison with the radial changes.

2. *Poisson's* ratio is very small, i.e. the axial stress has only small influence on the lateral strain.

3. The elasticity modulus is constant, i.e. it is independent of pressure, stress, or strain. [Hwa77]

4. The wall material of the arteries keeps always constant independent of the strain.

As the blood is modelled as a Newtonian fluid, the same restrictions concerning the artery diameters as for the static model are effective. [Gri76]

3.2 Basics

The blood flow in arteries can be described in terms of the flow velocity V and the pressure p. In this case only the average velocity of the sectional area and the average pressure are necessary, because an one-dimensional model is used. Therefore V and p are functions of position x along the axis of the artery and of time t. [Hor89]

Usually the pressure p is replaced by the pressure head H which is related to the pressure by

$$p(x,t) = \rho g\bigl(H(x,t) - z(x)\bigr), \tag{3.1}$$

where $z(x)$ is the elevation of the centerline of the artery at position x. Hence, the derivative of z is

$$z'(x) = \sin\alpha, \tag{3.2}$$

if α is the angle between the axis of the artery segment and the horizontal (positive if artery is ascending).

In the equations that are derived in the following sections, the total derivative with respect to time can be expressed as

$$\frac{d}{dt} = \frac{\partial}{\partial t} + V\frac{\partial}{\partial t}.$$

3.3 Equation of Continuity

To obtain the one-dimensional continuity equation the conservation of mass is applied to a control volume (see figure 3.1).

Conservation of mass means, that the difference of input and output of a control volume is equal to the change of the mass. That fact can be expressed by

$$\rho AV - \Bigl(\rho AV + \frac{\partial(\rho AV)}{\partial x}\Delta x\Bigr) = \frac{\partial(\rho A\Delta x)}{\partial t}, \tag{3.3}$$

or after simplification

$$-(\rho AV)_x\Delta x = (\rho A\Delta x)_t. \tag{3.4}$$

Expanding the parentheses, regrouping and dividing by $\rho A\Delta x$ gives

$$\frac{1}{\rho}(\rho_t + V\rho_x) + \frac{1}{A}(A_t + VA_x) + V_x + \frac{(\Delta x)_t}{\Delta x} = 0, \tag{3.5}$$

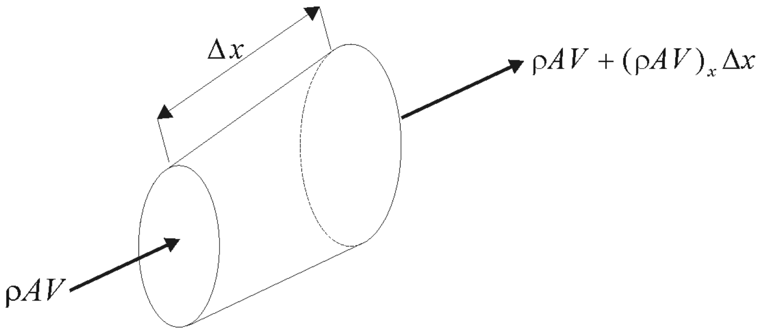


Figure 3.1: Definition sketch for continuity equation

or using the definition of the total derivative

$$\frac{1}{\rho}\frac{d\rho}{dt} + \frac{1}{A}\frac{dA}{dt} + V_x + \frac{1}{\Delta x}\frac{d(\Delta x)}{dt} = 0, \tag{3.6}$$

because Δx does not vary in the spatial dimension.

Since the main part of blood is water, it is a good approximation to assume blood as incompressible fluid – especially in the normal pressure range. Therefore the derivative of the density ρ is 0.

The artery walls are very elastic, but longitudinal constrained, so the lateral strain is much more larger than the axial strain. Therefore it is assumed that the artery is fixed against longitudinal stretching, and the last term of equation (3.6) vanishes, too. The result is

$$\frac{1}{A}\frac{dA}{dt} + V_x = 0. \tag{3.7}$$

If the radius R is introduced instead of the sectional area A, it gives

$$\frac{2}{R}\frac{dR}{dt} + V_x = 0. \tag{3.8}$$

The change of the radius can be expressed by the change of the lateral strain

$$\frac{d\varepsilon_2}{dt} = \frac{1}{R}\frac{dR}{dt}. \tag{3.9}$$

Both the axial and lateral strains are a combination of the axial and lateral stresses (σ_1, σ_2) connected by *Poisson's* ratio

$$\varepsilon_1 = \frac{1}{E}\left(\sigma_1 - \mu\sigma_2\right), \tag{3.10}$$

$$\varepsilon_2 = \frac{1}{E}\left(\sigma_2 - \mu\sigma_1\right). \tag{3.11}$$

As assumed before, the axial strain ε_1 is zero. Therefore equation (3.10) leads to the following relation for the lateral and axial stress

$$\sigma_1 = \mu\sigma_2. \tag{3.12}$$

Substituting the results of equations (3.11) and (3.12) into equation (3.9) gives

$$\frac{1}{R}\frac{dR}{dt} = \frac{1-\mu^2}{E}\frac{d\sigma_2}{dt}. \tag{3.13}$$

The lateral forces caused by the pressure must be equal to the forces induced by the stress.

$$2\sigma_2 h\Delta x = 2R\Delta x p \tag{3.14}$$

Hence, the lateral stress depends on the pressure, the radius and the wall thickness

$$\sigma_2 = \frac{pR}{h}. \tag{3.15}$$

Integrating equation (3.13) and substituting (3.15) results in the following relation between radius and pressure

$$\frac{pR}{h} = \frac{E}{1-\mu^2}\ln R + c. \tag{3.16}$$

The integration constant can be evaluated from the initial conditions with a static pressure p_0

$$\frac{pR}{h} = \frac{E}{1-\mu^2}\ln\left(\frac{R}{R_0}\right) + \frac{p_0R_0}{h_0}. \tag{3.17}$$

If it is assumed that the arterial wall material is constant, then the wall thickness changes with the radius, but the product stays constant

$$Rh = R_0h_0 = \text{const.} \tag{3.18}$$

Hence, after some transformations equation (3.17) becomes

$$p\frac{R_0(1-\mu^2)}{h_0E} = \left(\frac{R_0}{R}\right)^2\left(\ln\frac{R}{R_0} + \frac{p_0R_0(1-\mu^2)}{h_0E}\right). \tag{3.19}$$

By taking the total derivative of equation (3.19), one can obtain the following

$$\frac{dp}{dt}\frac{R_0(1-\mu^2)}{h_0E} = \frac{1}{R}\frac{dR}{dt}\left(\frac{R_0}{R}\right)^2\left(1 - 2\ln\frac{R}{R_0} - \frac{2p_0R_0(1-\mu^2)}{h_0E}\right), \tag{3.20}$$

or after regrouping and using abbreviations

$$\frac{1}{R}\frac{dR}{dt} = \frac{dp}{dt}\frac{1}{2\rho {a_0}^2F(\frac{R}{R_0})}, \tag{3.21}$$

with

$$a_0{}^2 = \frac{Eh_0}{2\rho R_0(1-\mu^2)}, \tag{3.22}$$

$$F\left(\frac{R}{R_0}\right) = \left(\frac{R_0}{R}\right)^2 \left(1 - 2\ln\frac{R}{R_0} - \frac{p_0}{\rho a_0{}^2}\right). \tag{3.23}$$

By setting

$$a = a_0\sqrt{F\left(\frac{R}{R_0}\right)} \tag{3.24}$$

and by substituting equation (3.21) into equation (3.8) it results in the continuity equation

$$\frac{1}{\rho a^2}\frac{dp}{dt} + V_x = 0. \tag{3.25}$$

Substituting the pressure head gives

$$H_t + \frac{a^2}{g}V_x + VH_x - V\sin\alpha = 0. \tag{3.26}$$

This is the *continuity equation* for the blood flow in an artery segment. [Str67]

Rewriting equation (3.23) in terms of the pressure head leads to

$$F\left(\frac{R}{R_0}\right) = \left(\frac{R_0}{R}\right)^2 \left(1 - 2\ln\frac{R}{R_0} - \frac{g}{a_0{}^2}\left(H_0 - z\right)\right). \tag{3.27}$$

Remark. The continuity equation for arteries (flexible tubes) is equal to the continuity equation for non-flexible pipes, except the wave speed varies with the radius (and pressure) in this case.

3.4 Equation of Motion

The equation of motion is derived for blood flow through a short segment of an artery. The segment remains always full of blood with mass density ρ. The equation for the unsteady flow is derived by applying Newton's Second Law to a control volume of blood in the artery segment considering only the streamline direction which is parallel to the axis of the segment. For this special case *Newton's Second Law* is

$$\sum f_x = ma_x = m\frac{dV}{dt}. \tag{3.28}$$

Figure 3.2 shows a control volume in an artery segment and the forces which have an effect on that element. Those forces are (in order): pressure force on

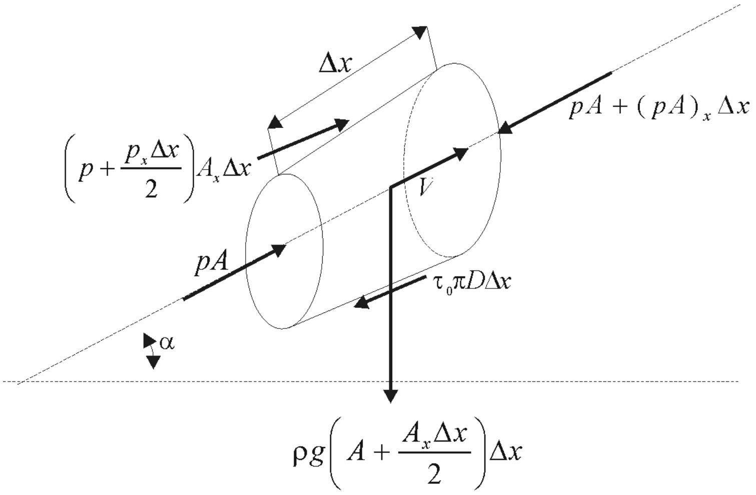

Δx
$\left(p+\frac{p_x\Delta x}{2}\right)A_x\Delta x$
$pA+(pA)_x\Delta x$
V
pA
$\tau_0\pi D\Delta x$
α
$\rho g\left(A+\frac{A_x\Delta x}{2}\right)\Delta x$

Figure 3.2: Definition sketch for equation of motion

left-hand area, pressure force on right-hand area, pressure force due to increase of area, frictional wall shear force, component of weight in x-direction. Substituting these terms into equation (3.28) results in

$$\begin{aligned} pA-\left(pA+\frac{\partial(pA)}{\partial x}\Delta x\right)+\left(p+\frac{\partial p}{\partial x}\frac{\Delta x}{2}\right)\frac{\partial A}{\partial x}\Delta x-\tau_0\pi D\Delta x- \\ -\rho g\Delta x\left(A+\frac{\partial A}{\partial x}\frac{\Delta x}{2}\right)\sin\alpha=\rho\Delta x\left(A+\frac{\partial A}{\partial x}\frac{\Delta x}{2}\right)\frac{dV}{dt}. \end{aligned} \tag{3.29}$$

By simplifying the above equation and cancelling the terms of second order (Δx^2 terms) the following equation turns out:

$$p_xA\Delta x+\tau_0\pi D\Delta x+\rho g\Delta xA\sin\alpha+\rho\Delta xA\frac{dV}{dt}=0. \tag{3.30}$$

After dividing the whole equation by $\rho A\Delta x$ the result is

$$\frac{dV}{dt}+\frac{1}{\rho}\frac{\partial p}{\partial x}+g\frac{\partial z}{\partial x}+\frac{4\tau_0}{D\rho}=0. \tag{3.31}$$

The form with the shear stress τ_0 is not very useful, therefore this term is substituted using the *Darcy-Weisbach* equation

$$\Delta p=\frac{\rho\lambda LV^2}{2D}. \tag{3.32}$$

The force balance gives

$$\Delta p\frac{\pi D^2}{4}=\tau_0\pi DL. \tag{3.33}$$

Hence the shear stress can be written as

$$\tau_0 = \frac{\rho\lambda V|V|}{8}. \tag{3.34}$$

The splitting of the term V^2 using the absolute value is necessary to direct the shear stress in the same direction as the velocity. Now equation (3.31) can be transformed to

$$V_t + VV_x + \frac{1}{\rho}p_x + gz_x + \frac{\lambda V|V|}{2D} = 0. \tag{3.35}$$

Substituting the pressure head and the radius leads to

$$gH_x + V_t + VV_x + \frac{\lambda V|V|}{4R} = 0. \tag{3.36}$$

This is the *equation of motion* for the blood flow in a cylindrical artery segment. [Str67, Wat79, Wyl93]

Remark. It is important to notice that λ and R are functions of x and t.

3.5 Radius – Pressure Relation

The formulation of the continuity equation must be extended by another relation. Because for the evaluation of the function F, it is necessary to know the actual radius R of the artery. Equation (3.19) gives a relation between the radius and the pressure, which can be used to express the radius. For simplification it is assumed that Poisson's ratio μ is very small ($\mu \ll 1$), so that $\mu^2 \approx 0$. (A very small Poisson's ratio means that the axial stress has only very small influence on the lateral strain.) By substituting the pressure head the result is a basic relation between radius and the pressure head

$$\frac{R^2}{{R_0}^2}\left(H - z\right) = 2\frac{{a_0}^2}{g}\ln\frac{R}{R_0} + \left(H_0 - z\right). \tag{3.37}$$

Solving this equation with respect to the radius R is only possible in terms of the *LambertW function*[1]

$$R = R_0 e^{-\frac{1}{2}\left(\mathrm{LamW}\left(-\frac{g(H-z)e^{-\frac{g(H_0-z)}{{a_0}^2}}}{{a_0}^2}\right)+\frac{g(H_0-z)}{{a_0}^2}\right)}. \tag{3.38}$$

Unfortunately the values of the LambertW function can only be approximated numerically. In the case of blood flow through an artery the pressure variation

[1] The LambertW function is the inverse function of $f(x) = xe^x$. In the complex plane this function has several branches. The main branch, which is used here, is the only one that is analytic at the origin.

is within a very small range around the static pressure head H_0. Therefore it seems better to approximate the radius (as a function of the pressure head) by a Taylor sum. The derivatives can be calculated from equation (3.37) by implicit differentiation.

$$\begin{aligned} R &= R_0 + R'(H_0)\,(H - H_0) + \frac{R''(H_0)}{2}\,(H - H_0)^2 + \ldots \\ R'(H_0) &= -\frac{1}{2}\frac{R_0}{H_0 - z - \frac{{a_0}^2}{g}} \\ R''(H_0) &= \frac{1}{4}\frac{R_0(3H_0 - 3z - 5\frac{{a_0}^2}{g})}{(H_0 - z - \frac{{a_0}^2}{g})^3} \end{aligned} \tag{3.39}$$

3.6 Summary

In section 3 all necessary equations have been derived to describe the blood flow in arteries. The model consists of two partial differential equations of first order. The unknown are the flow velocity V and the pressure head H (see table 3.1).

Equation of Motion
$gH_x + V_t + VV_x + \frac{\lambda V\vert V\vert}{2D} = 0$
Equation of Continuity
$H_t + \frac{a^2}{g}V_x + VH_x - V\sin\alpha = 0$

Table 3.1: System of partial differential equations for the blood flow

Besides that, a number of additional equations are necessary to obtain a solvable system (see table 3.2).

A few remarks should be made about those equations:

- The stationary wave speed a_0 holds only if there is a pressure difference between the inside and the outside of the artery. Otherwise the correction term F is not equal to one in the initial state and the stationary wave speed is $a_0\sqrt{F(1)}$.
- Theoretically it is possible that the correction term F becomes negative if the radius is increased too much. This point is far from the normal range, but it indicates that the elasticity model used here, is limited to that range and can not be used for extreme situations.

- The radius R as a function of the pressure head H as shown in the table is only a Taylor approximation (see section 3.5).
- That form of the friction factor was used in the static network calculations of the arterial system [Alm97a, Fun84]. The turbulent case ($Re > 3000$) is an approximation of the formula due to *Coolbrook* and *White*. The range between laminar and turbulent case is a linear interpolation, such that the whole function is continuous.

Wave Speed

$$a = a_0 \sqrt{F\left(\frac{R}{R_0}\right)}$$

Stationary Wave Speed (with small Poisson's ratio)

$$a_0 = \sqrt{\frac{Eh_0}{2\rho R_0}}$$

Function of the Dimensionless Radius

$$F\left(\frac{R}{R_0}\right) = \left(\frac{R_0}{R}\right)^2 \left(1 - 2\ln\frac{R}{R_0} - \frac{g(H_0 - z)}{{a_0}^2}\right)$$

Radius (as Function of Pressure Head)

$$\begin{aligned} R \approx & R_0 + c_1(H - H_0) + c_2(H - H_0)^2 \\ c_1 = & -\frac{R_0}{2(H_0 - z - \frac{{a_0}^2}{g})} \\ c_2 = & \frac{R_0(3H_0 - 3z - 5\frac{{a_0}^2}{g})}{8(H_0 - z - \frac{{a_0}^2}{g})^3} \end{aligned}$$

Friction factor

$$\lambda = \begin{cases} \dfrac{64}{Re} & 0 < Re \leq 2300 \\ 2.2 \cdot 10^{-5} \cdot Re - 0.0228 & 2300 < Re < 3000 \\ \dfrac{0.3942}{Re^{0.2761}} & 3000 \leq Re \end{cases}$$

Reynold number

$$Re = \frac{VD}{\nu}$$

Table 3.2: Additional equations necessary for solving the PDE-system

Chapter 4

Solution Method for the Flow Model

His Majesty sent for three great scholars who were then in their weekly waiting (according to the custom in that country). These gentlemen, after they had a while examined my shape with much nicety, were different opinions concerning me. They all agreed that I could not be produced according to the regular Laws of Nature, because I was not framed with a capacity of preserving my life, either by swiftness, or climbing of trees, or digging holes in the earth.

J. Swift, Gulliver's Travels (A Voyage to Brobdingnag).

4.1 Method of Characteristics

The method of characteristics transforms the two partial differential equations into two sets of ordinary differential equations, which are called the characteristic equations. These equations describe curves in the x-t-plane and relations between V and H valid only along those curves.

To transform the PDEs into the ODEs, a linear combination of the two equations (3.36) and (3.26) is taken and after regrouping the terms, it gives

$$(\varepsilon g + V)H_x + H_t + (\varepsilon V + \frac{a^2}{g})V_x + \varepsilon V_t + \varepsilon\frac{V|V|\lambda}{2D} - V\sin\alpha = 0, \qquad (4.1)$$

where ε is the free parameter. Substituting two different values for the parameter ε results in two independent equations describing the same system as the original ones.

To get two ordinary differential equations the spatial parameter x is assumed to be a function of time t. So that the total derivatives of H and V with respect to time t can be written as

$$\frac{dH}{dt} = H_x \frac{dx}{dt} + H_t \quad \text{and} \quad \frac{dV}{dt} = V_x \frac{dx}{dt} + V_t. \tag{4.2}$$

To transform the terms of (4.1) into total derivatives, the following equation must be solved with respect to ε

$$\frac{dx}{dt} = \varepsilon g + V = V + \frac{a^2}{g\varepsilon}. \tag{4.3}$$

Hence, the two solutions of ε are

$$\varepsilon = \pm \frac{a}{g}. \tag{4.4}$$

So the results are two different characteristics – both are systems of two ordinary differential equations

$$\begin{aligned} C^+ &: \begin{cases} \frac{g}{a}\frac{dH}{dt} + \frac{dV}{dt} + \frac{V|V|\lambda}{2D} - \frac{g}{a} V \sin\alpha = 0 \\ \frac{dx}{dt} = V + a \end{cases} \\ C^- &: \begin{cases} -\frac{g}{a}\frac{dH}{dt} + \frac{dV}{dt} + \frac{V|V|\lambda}{2D} + \frac{g}{a} V \sin\alpha = 0 \\ \frac{dx}{dt} = V - a \end{cases} \end{aligned} \tag{4.5}$$

The equations $\frac{dx}{dt} = V \pm a$ describe the curves in the x-t-plane – they are called *characteristics*. Along these curves the ordinary differential equations of V and H are valid. At the intersection points of the characteristics both can be solved for V and H.

4.2 x-t-Discretization

The solution of the characteristic equations can be approximated by discretization of the problem and solving it on a x-t-grid. For that purpose the artery segment is divided into N identically long parts. Each part is Δx long and the points are numbered from x_0 to x_N. The second dimension of the grid – the time axis – is also divided into sections, but different intersections are possible. The time step from the t_j-line to the t_{j+1}-line is Δt_j wide.

For solving the equations on the grid of figure 4.1, it is assumed first that V and H is known along the whole t_j-line, especially at the points A and B. To calculate the velocity $V_{i,j+1}$ and the pressure head $H_{i,j+1}$ at the upper middle point the finite differences have to be substituted into the characteristic equations,

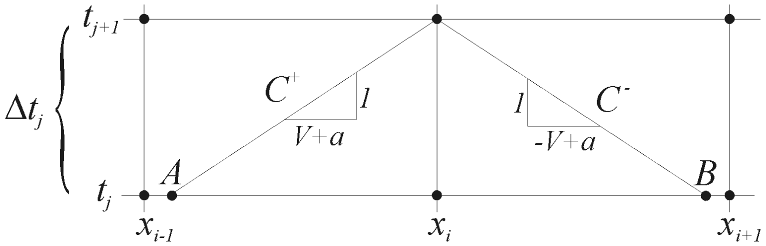


Figure 4.1: Part of the x-t-grid for the approximated solution

which results in

$$\begin{aligned}
\frac{g}{a_A}(H_{i,j+1} - H_A) + (V_{i,j+1} - V_A) + \Delta t_j\Big(\frac{V_A|V_A|\lambda_A}{2D_A} - \frac{g}{a_A}V_A \sin\alpha\Big) &= 0,\\
-\frac{g}{a_B}(H_{i,j+1} - H_B) + (V_{i,j+1} - V_B) + \Delta t_j\Big(\frac{V_B|V_B|\lambda_B}{2D_B} + \frac{g}{a_B}V_B \sin\alpha\Big) &= 0.
\end{aligned} \tag{4.6}$$

Hence – solving these equations – it gives

$$\begin{aligned}
V_{i,j+1} =& \frac{1}{a_A + a_B}\left(a_A V_A + a_B V_B\right) + \frac{g}{a_A + a_B}\left(H_A - H_B\right) - \\
&- \frac{\Delta t_j}{a_A + a_B}\left(a_A\frac{V_A|V_A|\lambda_A}{2D_A} + a_B\frac{V_B|V_B|\lambda_B}{2D_B}\right) + \\
&+ \frac{\Delta t_j g}{a_A + a_B}\left(V_A \sin\alpha - V_B \sin\alpha\right), \\
H_{i,j+1} =& \frac{a_A a_B}{g(a_A + a_B)}\left(V_A - V_B\right) + \frac{1}{a_A + a_B}\left(a_B H_A + a_A H_B\right) - \\
&- \frac{\Delta t_j a_A a_B}{g(a_A + a_B)}\left(\frac{V_A|V_A|\lambda_A}{2D_A} - \frac{V_B|V_B|\lambda_B}{2D_B}\right) + \\
&+ \frac{\Delta t_j}{a_A + a_B}\left(a_B V_A \sin\alpha + a_A V_B \sin\alpha\right).
\end{aligned} \tag{4.7}$$

As mentioned before, it is necessary to know V and H at the points A and B at time t_j to calculate the values for time t_{j+1} at the point x_i. But usually V and H are only known at the grid points on the t_j-line. So to get the values at A and B, interpolation must be used.

If $\overline{V}_j(x)$ and $\overline{a}_j(x)$ are the interpolation functions of the flow velocity and the wave speed in the interval $[x_0, x_n]$, then the point A is the solution of

$$\frac{x_i - A}{\Delta t_j} = \overline{a}_j(A) + \overline{V}_j(A). \tag{4.8}$$

The interpolation function for the wave speed can be calculated from the interpolation function for the pressure head by using the equations of table 3.2.

Similar to the above equation the point B can be calculated from

$$\frac{B - x_i}{\Delta t_j} = \overline{a}_j(B) - \overline{V}_j(B). \tag{4.9}$$

4.3 Courant Condition

It is important to recognize the limitation in the selection of the grid-mesh ratio. If the time step Δt_j is increased too much, the points A and B will be outside of of the range from x_{i-1} to x_{i+1}, especially at the boundaries the points would lie outside the artery. So for a fixed segmentation the time increment is limited to

$$\Delta t \leq \frac{\Delta x}{\max(a + |V|)}. \tag{4.10}$$

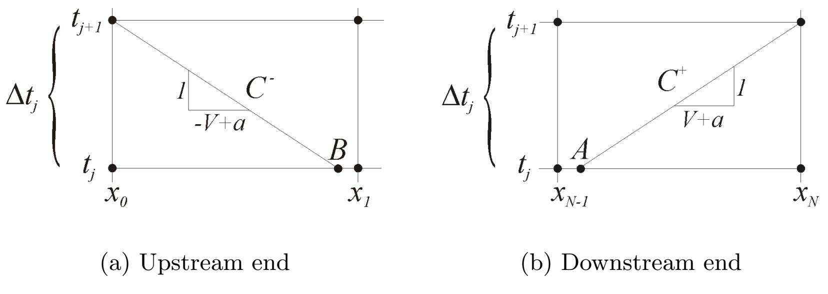
This limitation is called the *Courant* condition.

A simply way to fulfill this inequality is to choose only very small time steps. Although there is no instability, the interpolation error at the points A and B increases if these points move together. This fact causes a numerical damping of the shock waves and therefore inaccurate results. So its necessary to choose the time increment as high as possible within the limitation of (4.10).

4.4 Initial Values and Boundary Conditions

To perform the solution technique described in section 4.2 it is necessary to know the values of the flow velocity V and the pressure head H at all grid points on an initial time line $t = 0$. Then it is possible to calculate the new values for each time step within the boundaries of the pipe.

At the boundaries of the grid (x_0 and x_N) some conditions must be added, because only one equation can be derived at these points using the characteristics (see figure 4.2).



(a) Upstream end

(b) Downstream end

Figure 4.2: Situation at the boundaries

Those relations derived from the characteristics are shown below, (4.11) for the upstream end and (4.12) for the downstream end.

$$V_{0,j+1} = \, {}^{\mathrm{u}}_{1}C_j - \, {}^{\mathrm{u}}_{2}C_j H_{0,j+1} \tag{4.11}$$

$$V_{N,j+1} = - \, {}^{\mathrm{d}}_{1}C_j + \, {}^{\mathrm{d}}_{2}C_j H_{N,j+1} \tag{4.12}$$

The constants can be calculated from the characteristic equations (4.6).

$${}^{\mathrm{u}}_{1}C_j = V_B - \frac{g}{a_B} H_B - \Delta t_j \big(\frac{V_{V_B} |V_{V_B}| \lambda_{V_B}}{2 D_{V_B}} + \frac{g}{a_{V_B}} V_{V_B} \sin\alpha \big) \tag{4.13}$$

$${}^{\mathrm{u}}_{2}C_j = -\frac{g}{a_B} \tag{4.14}$$

$${}^{\mathrm{d}}_{1}C_j = -V_A - \frac{g}{a_A} H_A + \Delta t_j \big(\frac{V_{V_A} |V_{V_A}| \lambda_{V_A}}{2 D_{V_A}} - \frac{g}{a_{V_A}} V_{V_A} \sin\alpha \big) \tag{4.15}$$

$${}^{\mathrm{d}}_{2}C_j = -\frac{g}{a_A} \tag{4.16}$$

Hence, to get a unique solution at the boundaries, it is necessary to give an additional relation for the flow velocity V and the pressure head H.

Depending on the type of relation the boundary conditions can be classified in the following form:

	non-dynamical BC	dynamical BC
passive BC	$f(H, V) = 0$	$f(H, V, \frac{dV}{dt}, \frac{dH}{dt}) = 0$
active BC	$f(H, V, t) = 0$	$f(H, V, \frac{dV}{dt}, \frac{dH}{dt}, t) = 0$

Examples for non-dynamic boundary conditions are artery junctions (passive) and variable inflow (active). An air chamber is a dynamical passive boundary condition.

4.5 Summary and Remarks

The method of characteristics in fluid dynamics was developed some years ago. Nevertheless it enables to solve the partial differential equations with a minimum of calculation effort and with a minimum on numerical failures at the same time. The only disadvantage in the case of flows, which do not change very much for longer period, is the Courant condition, which limits the maximal time step, depending on the spacial discretization.

There exist already new time-implicit methods for a stable algorithm without time step limitation [Seu82]. But in the case of pulse simulation the maximal time step is about 2 milliseconds, and in comparison the back-flow into the heart before the valve closes endures about 20 milliseconds. Therefore a larger time step would neglect such short but characteristic features of the flow from the heart.

The method of characteristics makes it also possible to include the boundary conditions in a very simple way. The constants for these boundary conditions, introduced in section 4.4, differ from those in literature. The reason for choosing other forms is the simpler form of equations for describing the connection of several arteries (see section 5.2).

Chapter 5

Artery Network Model

The modern masters promise very little; they know that metals cannot be transmuted and that the elixir of life is a chimera. But these philosophers, whose hands seem only made to dabble in dirt, and their eyes to pore over the microscope or crucible, have indeed performed miracles. They penetrate into the recesses of nature and show how she works in her hiding places. They ascend into the heavens; they have discovered how the blood circulates, and the nature of air we breathe.
M. Shelley, Frankenstein.

5.1 Assumptions

The network concept used for the static model (see section 2.1) is included in the dynamic model in a similar way. The arterial system is represented as a network of vessels and connecting nodes. The dynamics of the blood flow within the arteries is described by the equations of the previous two chapters. At the connecting nodes a combined boundary condition gives the relation between flow and pressure. This results in a model with the following features:

1. The arterial system is represented by a pipe network consisting of highly flexibly pipes. (Only radial flexibility is taken into consideration.)

2. No blood loss occurs through the walls of the arteries. The only points where blood can leave the system are connecting and end nodes.

3. There is no pressure loss at the ramification points, regardless of the absolute and relative positions of the arteries to each other.

4. Several air chambers included in the aortic arch are used to model the compliance of the aorta.

5. All blood, that leaves the system, is collected through fictitious arteries with fixed length and wave speed, but variable diameters (cf. static model).

The reasons for using the concept of fictitious arteries are likely the same as for the static model. First of all there is the problem of determining how much blood flows out at an certain node. One way to solve that is by using the results of the static calculation. But setting an constant outflow at these nodes is doubtful, because this causes reflections of the pulse wave at the outflow nodes of the networks. Therefore the fictitious arteries are used as a buffer which reduces this effect a little bit. As mentioned in chapter 2 microcirculation is another reason for using the fictitious arteries concept.

5.2 Connection of the Artery Models

In this case n arteries are connected at one hydraulic node. At this node there is a variable outflow of blood given by the function $Q_{\text{out}}(t)$. (see figure 5.1)

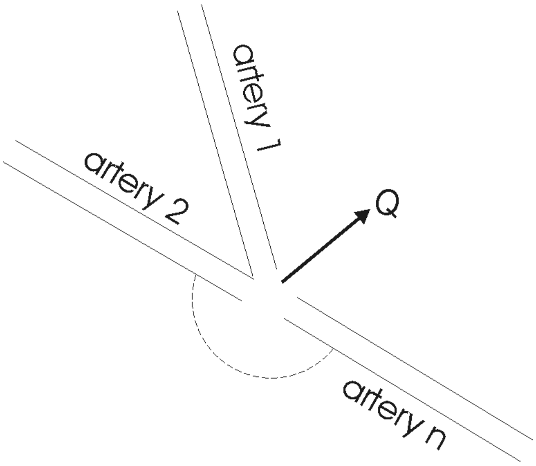


Figure 5.1: Multiple artery junction

For deriving a general relation between the flow velocities and the pressure heads the vessels are numbered from 1 to n. Superscripts like (j) are used to indicate a parameter of the j-th artery. Furthermore it is assumed that the head loss at the junction is negligible. So if $U \subseteq \{1, \dots, n\}$ is the set of arteries, which are connected to the hydraulic node with their upstream ends, and D the complementary set, then the pressure head at the hydraulic node at time t_{j+1} can be written as

$$H_{j+1} = \frac{\sum_{k \in U} {}^{\mathrm{u}}_{1}C^{(k)}_j A^{(k)} + \sum_{k \in D} {}^{\mathrm{d}}_{1}C^{(k)}_j A^{(k)} + \frac{1}{\Delta t_j} \int_{t_j}^{t_{j+1}} Q_{\text{out}}(t)dt}{\sum_{k \in U} {}^{\mathrm{u}}_{2}C^{(k)}_j A^{(k)} + \sum_{k \in D} {}^{\mathrm{d}}_{2}C^{(k)}_j A^{(k)}}. \tag{5.1}$$

Proof The proof for this equation is very simple. It is the solution of the following equation system, that describes the hydraulic node.

$$V^{(k)}_{0,j+1} = {}^{\mathrm{u}}_{1}C^{(k)}_{j} - {}^{\mathrm{u}}_{2}C^{(k)}_{j} H^{(k)}_{0,j+1} \qquad \forall k \in U \tag{5.2}$$

$$V^{(k)}_{N,j+1} = -\,{}^{\mathrm{d}}_{1}C^{(k)}_{j} + {}^{\mathrm{d}}_{2}C^{(k)}_{j} H^{(k)}_{N,j+1} \qquad \forall k \in D \tag{5.3}$$

$$H^{(k)}_{0,j+1} = H^{(l)}_{0,j+1} = H^{(m)}_{N,j+1} \qquad \forall k, l \in U, \forall m \in D \tag{5.4}$$

$$\sum_{k\in D} V^{(k)}_{N,j+1} A^{(k)} = \sum_{k\in U} V^{(k)}_{0,j+1} A^{(k)} + \frac{1}{\Delta t_j} \int_{t_j}^{t_{j+1}} Q_{\mathrm{out}}(t)dt \tag{5.5}$$

Since all pressure heads are equal, they are substituted with a general pressure head of the hydraulic node. Then only equations (5.2) and (5.3) have to be substituted into equation (5.5) and solved for the pressure head H at the node. □

5.3 Air Chamber

An air chamber is a hydraulic tool that smooths pressure waves. The structure of an air chamber is shown in figure 5.2. If a pressure wave moves through artery 1 and reaches the air chamber, the gas in the upper part will be compressed, because the blood is in comparison to the gas incompressible. So a smoothed pressure wave moves through artery 2.

Air chambers are used to model the compliance of the aorta, because the elasticity model, subsumed in the variability of the wave speed, can not describe the high flexibility of that vessel alone.

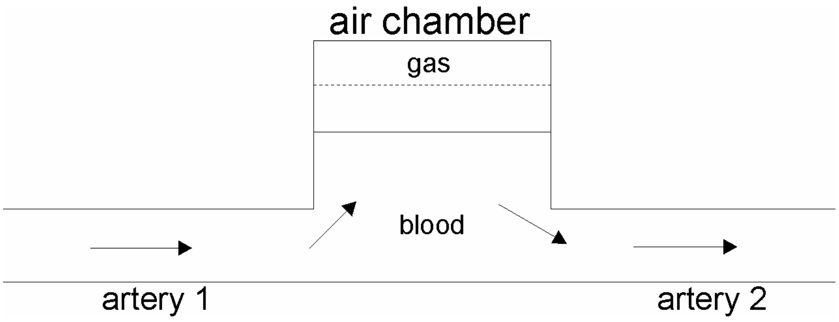


Figure 5.2: Scheme of an air chamber

If the influence of the air chamber is assumed to be concentrated at one point (hydraulic node), where the vessels are connected, the following equations can be derived for this point. [Sud87b]

The positive characteristic from artery 1 (multiplied by the sectional area):

$$Q^{(1)}_{N,j+1} = (-\,{}^{\mathrm{d}}_{1}C^{(1)}_{j} + {}^{\mathrm{d}}_{2}C^{(1)}_{j} H^{(1)}_{N,j+1}) A^{(1)} \tag{5.6}$$

The negative characteristic from artery 2 (multiplied by the sectional area):

$$Q^{(2)}_{0,j+1} = ({}^{\mathrm{u}}_{1}C^{(2)}_{j} - {}^{\mathrm{u}}_{2}C^{(2)}_{j} H^{(2)}_{0,j+1}) A^{(2)} \tag{5.7}$$

The node condition (pressure loss is negligible):

$$H^{(1)}_{N,j+1} = H^{(2)}_{0,j+1} \tag{5.8}$$

The flow balance equation (the flow out of artery 1 is equal to the flow into artery 2 and the flow into the air chamber):

$$Q^{(1)}_{N,j+1} = Q^{(2)}_{0,j+1} + Q^{\mathrm{ac}}_{j+1} \tag{5.9}$$

The polytropic relation:

$$(H_{\mathrm{stat}} + H_{\mathrm{b}} - z_{\mathrm{stat}})\mathcal{V}^{n}_{\mathrm{stat}} = (H^{(1)}_{N,j+1} + H_{\mathrm{b}} - z_{j+1})\mathcal{V}^{n}_{j+1} = \mathcal{C} \tag{5.10}$$

The gas volume change in the air chamber:

$$\mathcal{V}_{j+1} = \mathcal{V}_{\mathrm{stat}} - A^{\mathrm{ac}}(z_{j+1} - z_{\mathrm{stat}}) \tag{5.11}$$

The change of the fluid level in the air chamber is given by:

$$z_{j+1} = z_0 + (Q^{\mathrm{ac}}_{j+1} + Q^{\mathrm{ac}}_{j}) \frac{\Delta t_j}{2A^{\mathrm{ac}}} \tag{5.12}$$

The equations (5.6) - (5.12) form a system for the unknown parameters $V^{(1)}_{N,j+1}$, $V^{(2)}_{0,j+1}$, $H^{(1)}_{N,j+1}$, $H^{(2)}_{0,j+1}$, Q^{ac}_{j+1}, $\mathcal{V}_{j+1}$, and z_{j+1}.

To solve this equation system, the relation (5.8) is substituted into (5.6) and the result is put into (5.7). That gives

$$Q^{(2)}_{0,j+1} = \Big({}^{\mathrm{u}}_{1}C^{(2)}_{j} - \frac{{}^{\mathrm{u}}_{2}C^{(2)}_{j}}{{}^{\mathrm{d}}_{2}C^{(1)}_{j}} \big({}^{\mathrm{d}}_{1}C^{(1)}_{j} + \frac{Q^{(1)}_{N,j+1}}{A^{(1)}} \big) \Big) A^{(2)} \tag{5.13}$$

Then substituting (5.13) into (5.9) and solving for $Q^{(1)}_{N,j+1}$ results into

$$\begin{aligned} Q^{(1)}_{N,j+1} =& \frac{A^{(1)}}{{}^{\mathrm{d}}_{2}C^{(1)}_{j} A^{(1)} + {}^{\mathrm{u}}_{2}C^{(2)}_{j} A^{(2)}} \times \\ & \times \Big({}^{\mathrm{d}}_{2}C^{(1)}_{j} Q^{\mathrm{ac}}_{j+1} + {}^{\mathrm{u}}_{2}C^{(1)}_{j} \, {}^{\mathrm{u}}_{1}C^{(2)}_{j} A^{(2)} - {}^{\mathrm{u}}_{2}C^{(2)}_{j} \, {}^{\mathrm{d}}_{1}C^{(1)}_{j} A^{(2)} \Big). \end{aligned} \tag{5.14}$$

Hence, this equation can be substituted into (5.6) and solved for $H^{(1)}_{N,j+1}$

$$H^{(1)}_{N,j+1} = \frac{Q^{\mathrm{ac}}_{j+1} + {}^{\mathrm{u}}_{1}C^{(2)}_{j} A^{(2)} + {}^{\mathrm{d}}_{1}C^{(1)}_{j} A^{(1)}}{{}^{\mathrm{d}}_{2}C^{(1)}_{j} A^{(1)} + {}^{\mathrm{u}}_{2}C^{(2)}_{j} A^{(2)}}. \tag{5.15}$$

Now taking (5.12) and substituting it into (5.11) gives

$$\mathcal{V}_{j+1} = \mathcal{V}_{\mathrm{stat}} - \frac{\Delta t_j}{2}(Q^{\mathrm{ac}}_{j+1} + Q^{\mathrm{ac}}_j). \tag{5.16}$$

If the equations (5.15), (5.16), and (5.12) are substituted into equation (5.10), the result is

$$(c_1 Q^{\mathrm{ac}}_{j+1} + c_2)(c_3 Q^{\mathrm{ac}}_{j+1} + c_4)^n - \mathcal{C} = 0, \tag{5.17}$$

with

$$\begin{aligned}
c_1 &:= \frac{1}{{}^{\mathrm{d}}_{2}C^{(1)}_j A^{(1)} + {}^{\mathrm{u}}_{2}C^{(2)}_j A^{(2)}} - \frac{\Delta t_j}{2A^{\mathrm{ac}}} \\
c_2 &:= \frac{{}^{\mathrm{d}}_{1}C^{(1)}_j A^{(1)} + {}^{\mathrm{u}}_{1}C^{(2)}_j A^{(2)}}{{}^{\mathrm{d}}_{2}C^{(1)}_j A^{(1)} + {}^{\mathrm{u}}_{2}C^{(2)}_j A^{(2)}} + H_{\mathrm{b}} - z_0 - \frac{Q^{\mathrm{ac}}_j \Delta t_j}{2A^{\mathrm{ac}}} \\
c_3 &:= -\frac{\Delta t_j}{2} \\
c_4 &:= \mathcal{V}_{\mathrm{stat}} - \frac{Q^{\mathrm{ac}}_j \Delta t_j}{2}
\end{aligned} \tag{5.18}$$

Equation (5.17) can be solved for Q^{ac}_{j+1} and then all other parameters can be calculated. The isentropic exponent n is in the range between 1 and 1.4. It would be convenient, if $n = 1$, which is true in the case of an ideal gas in the air chamber. As it has been described before the air chambers are only used to model the compliance of the aorta. Therefore they have no real meaning and it can be assumed that all air chambers in the model are filled with an ideal gas. In this case (5.17) results into a quadratic equation which can be solved analytically.

5.4 Courant Condition for the Arterial Network

When combining several artery models, as it is done for a network, it is necessary to choose an uniform time step for the whole network, so that the equations at the connections can be evaluated simultaneously. Hence an overall *Courant* condition can be derived from the *Courant* condition for a single artery (see equation (4.10)

$$\Delta t \leq \min_k \left(\frac{\Delta x^{(k)}}{\max(a^{(k)} + |V^{(k)}|)} \right), \tag{5.19}$$

where k runs in the set of arteries.

Because all segments of one artery should have the same length, another more practical form can be used

$$\Delta t \leq \min_k \left(\frac{L^{(k)}}{N^{(k)}} \frac{1}{\max(a^{(k)} + |V^{(k)}|)} \right), \tag{5.20}$$

where N is the number of segments and L the length of the artery.

As the flow velocities V and the wave speeds a are not constant, this limitation varies during a simulation run. Therefore it is necessary to calculate these changes and adapt the time step, so that the calculation is as accurate and as fast as possible.

5.5 Static Precalculation

The reasons for performing a precalculation using the hydrostatic model are the initial values needed for the dynamic calculations and average values of flow velocity and blood pressure for setting up a time step control (see section 5.6).

According to the method described in section 2.3 the mean flow and pressure values are calculated by assuming that all flows are laminar. This leads to the problem of solving a linear equation system. Afterwards the results are checked for non-laminar flows in the network and in that case an iteration is started. Each iteration step updates the admittance matrices according to the previous results and the linear equation system is solved again [Alm97a].

Hence, if the static model is used for precalculation instead of a standalone simulation, the flow and pressure values of the fictitious arteries are needed as well as those of the other vessels.

5.6 Segmentation and Time Step Control

There are two different aspects which have to be taken into consideration when setting the segmentation of the arteries in the network. Those are the calculation time and detailed results on the one side and the *Courant* condition on the other side.

If a segmentation is optimized with respect to the *Courant* condition (i.e. the time limit is nearly the same in all arteries.), it is easy to get to a refined optimized segmentation by simply bisecting the old segments. But otherwise if the segmentation is reduced, it may not be optimal anymore. Therefore a good procedure for segmentation is to find a rough optimal one and refine it, if it is necessary.

This rough segmentation can be found by choosing the artery with the smallest maximal time step by setting the whole artery as one segment. This maximal time step can be calculated using the results of the hydrostatic precalculations

$$\Delta t_{\max} = \frac{L}{\max(a + |V|)}. \tag{5.21}$$

The segmentation of the other arteries is set so that their time limit is slightly higher than $\Delta t_{\max}$. Using equation (5.20) this means

$$\Delta t_{\max} \leq \frac{L^{(k)}}{N^{(k)}} \frac{1}{\max(a^{(k)} + |V^{(k)}|)}, \tag{5.22}$$

or, solved for $N^{(k)}$

$$N^{(k)} = \left\lfloor \frac{1}{\Delta t_{\max}} \frac{L^{(k)}}{\max(a^{(k)} + |V^{(k)}|)}. \right\rfloor \tag{5.23}$$

During the dynamic calculations the flow velocities and the wave speeds in the arteries change and it is possible that in one artery the time limit gets smaller than the maximal time step, because only mean values, results from the hydrostatic precalculations, are used to determine the maximal time step. Therefore the *Courant* condition must be checked always during the dynamic calculation. This is very simple by testing if the interpolation points A and B are within the next segmentation points or not. Hence the maximal time step has to be decreased, if one of the points lays outside.

5.7 Summary

With the features developed in this chapter, it is now possible to build up a whole network of artery model components developed in the chapters 3 and 4. To construct a model of an arterial network the structure is mapped directly on a network of those artery models.

Several important facts were developed in this chapter:

1. A general equation for the connection nodes was presented, which describes the relation between the flow velocities and the pressure independent of the number of arteries connected and independent of the flow directions. This formulation avoids the handling of different flow situations.

2. An air chamber with ideal gas is used, which means a major simplification for the calculation.

3. A special algorithm is introduced to find an optimal segmentation based on hydrostatic precalculations. In addition a method for the time step control is described.

So in the last three chapters all necessary things are included to design a model which describes the pulsatile flow in an arterial system.

Chapter 6

Implementation of the Flow Model in C++

How full of briars is this working-day world!

W. Shakespeare, As You Like It.

6.1 Mapping the Network Structure on an Object Hierarchy

In the last three chapters the model was developed in a bottom-up approach starting from equations describing the blood flow in one artery, formulation of boundary conditions and connection of those small models to develop a network model. This hierarchical structure should be preserved in the implementation step, because a modular construction makes it easier to replace parts of the model with new developed features. By transferring the network structure into an object oriented implementation will lead to a more readable and understandable code.

To represent the network structure also in the program three classes are defined:

Net: It represents the whole network including both arteries and nodes. The calculation controlling mechanism and the data I/O are the main parts of this object.

Artery: This object is the implementation of the artery blood flow model representing one artery.

Node: This is a general interface, which subsumes the different kinds of boundary conditions (artery connection, air chamber, variable inflow).

To store the structural information of the network a data structure similar to a double chained list of Artery and Node objects is used. This means every

Artery is connected to its start and its end Node using pointers. The Node objects store all Artery objetcs with which they are connected (see figure 6.1).

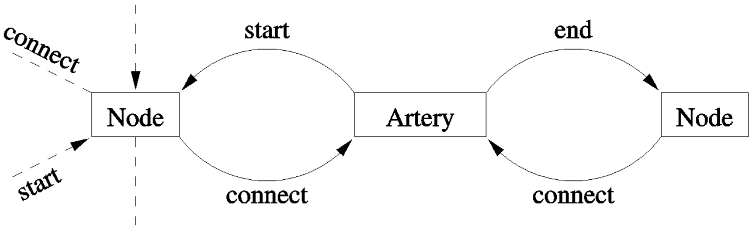


Figure 6.1: Scheme of the chained list used to store the network structure

In addition to these three classes several other ones are necessary for the static precalculations, the data output, and other features. In the following chapters those classes and their functionality are described in detail using a top-down approach.

6.2 Object NET

The Net object represents the whole model. It includes methods for

1. loading data
2. building up the network structure
3. managing the static precalculation
4. initializing the network
5. perform a whole simulation run including time step control
6. returning results

The Net class includes a list of all Artery and Node objects (line 20 – 21) representing the arterial network. The blood flow from the heart is stored in a Function object (line 13). The other two objects mentioned in figure 6.2, the StatCalc and the DataColl objects, are used only temporarily within the methods for the static precalculations and the data recording of the dynamic simulation.

There are three main methods in class Net, which are necessary to perform a whole simulation experiment. The function `staticcalc` (line 40) loads the network structure and all other necessary data into object StatCalc, which calculates the hydrostatic flow situation. The returned results are stored again in each Artery and Node object.

The method `initnet` initializes and prepares the network for the dynamic calculation. The main task of this function is to calculate the segmentation of the

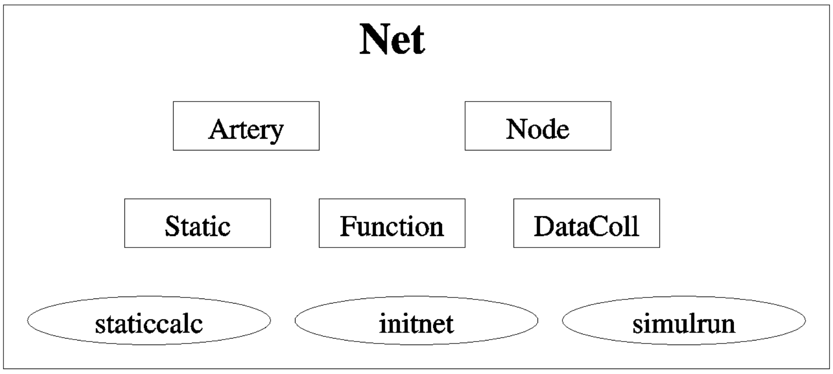


Figure 6.2: Main objects and methods used in class Net

arteries and the maximal time step. Depending on the parameter value it does an automatic segmentation (`seglen=0`) or tries to fit the segmentation length to the given value.

The most important method is `simulrun` (line 43 – 47), which executes the whole hydrodynamic calculation. The time step control and the communication with the data recording object DataColl are included here. The method `calc_1tstp` (line 42) is called for calculating one time step for the whole network. If the *Courant* condition fails in one of the arteries during a time step calculation, this function returns an error to `simulrun`, so that the time step control can be activated.

As all other classes the >>– and the <<–operators are overloaded to customize the data input and output.

Definition of class NET

```
1   class TNet {
```

The operators for loading and writing the network data are overloaded.

```
2     friend istream& operator>> (istream&, TNet&);
3     friend ostream& operator<< (ostream&, TNet&);
```

The number of arteries and nodes are only for internal use. The simulation status indicates the actual status of the network (see table B.1).

```
4   private:
5     int     numofart;         // number of arteries
6     int     numoffictart;     // number of fictitious arteries
7     int     numofnode;        // number of nodes
8     int     simstatus:        // simulation status
```

The network parameters and the file name of the heart flow function file are used also by other objects, therefore they have to be defined global.

```
public:
  char    title[120];      // model title
  int     closed;          // is there a fictitious part ?
  double  bloodvisc;       // kinematic visc. of blood
  double  blooddens;       // specific density of blood
  TFunction mainf;         // main inflow function
  char    mainffile[40];   // name of main function file
```

Some parameters are necessary for the static calculation respectively variables to store the results. The static reference flow can be calculated from the inflow function, but the reference pressure is necessary to calculate the absolute pressure values during the static calculation.

```
private:
  double  s_refpres;       // static reference pressure
  double  s_colpres;       // static pressure at coll. node
  double  s_refflow;       // static reference flow
  TStatCalc *scalc;        // object for static calculation
```

The following variables store the actual simulation time and the maximal time step.

```
  double  simtime;         // actual time of simulation
  double  maxtstp;         // maximal time step possible
```

The Artery and Node objects are stored in chained lists.

```
  struct art_list   *alist;
  struct node_list  *nlist;
```

There are constructor and destructor methods and some functions to read out the internal parameters.

```
public:
// --- constructor and destructor
  TNet (void);
  ~TNet (void);
// --- variable manipulating functions
  int    getnumofart (void);
  int    getnumofnode (void);
  double getsimtime (void);
  double getmaxtstp (void);
  int    getsimstatus (void);
```

There are two methods to search for Artery or Node objects with a given identification number. The following two private methods remove the chained lists used for storing the network structure.

```
  TNode* getnode (int nonum);          // find node
  TArtery* getartery (int arnum);      // find artery
private:
  void rmalist (struct art_list *al);  // remove artery list
  void rmnlist (struct node_list *nl); // remove node list
```

The following four methods are the main interface of this class. The method staticcalc manages the static calculation. It has no input parameters, but the return value indicates different types of errors (see table 6.1). The method initnet initializes the net for the dynamic calculation (see table 6.2), and the function calc_1tstp calculates one time step for the whole network (see table 6.3). For managing a whole simulation run, the method simulrun is used (see table 6.4).

```
public:
  int    staticcalc (void);
  int    initnet (double seglen);
  int    calc_1tstp (double tstp);
  int    simulrun (double tstart, double tend, double tstp,
               double cint, istream& DataOS, ostream& f_OS,
               ostream& v_OS, ostream& p_OS, ostream& h_OS,
               ostream& d_OS, ostream& w_OS, int cont,
               int verb, ostream& log_OS);
};
```

return value	description
0	no error
1	Warning! Calculation was already finished.
-9	no network loaded
-1xx	error from add_node of StatCalc
-2xx	error from add_edge of StatCalc
-3xx	error from complete_net StatCalc
-5xx	error from calculate of StatCalc
-11xx	error from get_node of StatCalc
-12xx	error from get_edge of StatCalc

Table 6.1: Return values of staticcalc

parameter	**description**
`seglen`	segmentation length (0 — automatic calculation)
return value	**description**
0	no error
-9	no hydrostatic calculation

Table 6.2: Parameters and return values of `initnet`

parameter	**description**
`tstp`	time step
return value	**description**
0	no error
-1	time step too great
-9	network not initialized
-1xxxxxx	*Courant* condition failed in artery xxxxxx

Table 6.3: Parameters and return values of `calc_1tstp`

parameter	**description**
`tstart`	start time of simulation
`tend`	end time of simulation
`tstp`	time step (0 — automatic, -1 — maximal fixed)
`cint`	communication interval
`DataOS`	input stream for DataColl
`f_OS`	output stream for flow data
`v_OS`	output stream for velocity data
`p_OS`	output stream for pressure data
`h_OS`	output stream for pressure head data
`d_OS`	output stream for diameter data)
`w_OS`	output stream for wavespeed data
`cont`	0 — new run, 1 — continue an old run
`verb`	0 — no output, 1 — information output
`log_OS`	output stream for log information
return value	**description**
0	no error
-9	network not initialized
-1xxxxxx	*Courant* condition failed in artery xxxxxx

Table 6.4: Parameters and return values of `simulrun`

6.3 Object ARTERY

This object represents one artery, including all its properties and also the integration algorithm for solving the artery blood flow model. Again operator overloading is used for data input and output.

The ARTERY objects as well as the NODE objects have a pointer to that NET they are part of (line 5). This is necessary to get access to general model data (e.g. viscosity, density, . . .).

The properties of the artery (line 6 – 15) are loaded using the operator function. The results of the static calculations are stored for the later use of initializing the dynamic calculation (line 16 – 18). Furthermore during the dynamic calculation the values of flow velocity and pressure head at the segmentation points are stored in arrays (line 21 – 22). A general interpolation function (line 47) is used to interpolate the different parameters (line 49 – 54) between the segmentation points.

Before starting the dynamic calculation the methods `setnseg` (line 55) and `setinitial` (line 56) are used to set the number of segmentation points and initialize the arrays.

But the main method for the calculation is `calc_tstp` (line 60) which calculates one time step for the artery, except the boundary points, because the boundary conditions have to evaluated by the NODE objects first using the function `getndconsts` (line 62 – 63) to get the necessary parameters. Then the boundary values can be calculated by the method `calc_bound` (line 64 – 65). The functions `calcA` and `calcB` (line 57 – 58) find the interpolation points A and B using a bisection algorithm. If the points lay outside the segmentation points — that means the *Courant* condition is not fulfilled — they return an error code and the method `rm_calc_tstp` (line 61) is used to take back the last time step.

Definition of class ARTERY

```
 1   class TArtery {
```

The operators for loading and saving the data are overloaded.

```
 2     friend istream& operator>> (istream&, TArtery&);
 3     friend ostream& operator<< (ostream&, TArtery&);
```

The parent NET object and the artery properties.

```
 4   private:
 5     TNet   *net;               // net structure (parent)
 6     int    number;             // number of arteries
 7     double sinang;             // sinus of ascending angle
 8     double length;             // length of artery
 9     double s_diameter;         // static diameter of artery
10     double i_wavespeed;        // init wave speed
11     double s_wavespeed;        // static wave speed
12     TNode  *node_start;        // start node
```

```
  TNode  *node_end;          // end node
  double height_start;       // height of start node
  double height_end;         // height of end node
```

Variables used to store the results of the static calculation.

```
  double s_flow;             // static flow
  double s_velo;             // static flow velocity
  double s_hpres;            // static pressure head
```

Variables and arrays for the dynamic calculation.

```
  int    nseg;               // number of segments
  double Deltax;             // segmentation length
  double *velo, *oldvelo;    // velocities at segm. points
  double *hpres, *oldhpres;  // press. head at segm. points
  double cu1, cu2, cd1, cd2; // constants for boundary cond.
```

Constructor and destructor method and functions for setting or reading artery properties.

```
public:
  TArtery (TNet *n);
  TArtery (TNet *n, int num, int nst, int nen, double len,
           double s_dia, double s_wave, double sf, int ns);
  ~TArtery (void);
  int    getnumber (void);
  double getangle (void);
  void   setangle (void);
  double getlength (void);
  double gets_diameter (void);
  double gets_wavespeed (void);
  void   seti_wavespeed (void);
  void   sets_flow (double f);
  double gets_flow (void);
  void   sets_velo (double v);
  double gets_velo (void);
  void   sets_hpres (void);
  void   setnode_start (TNode *nst);
  TNode* getnode_start (void);
  void   setnode_end (TNode *nen);
  TNode* getnode_end (void);
  int    getnseg (void);
```

For calculating the different parameters between the segmentation points the interpolation method `interp` is used together with the public interface methods, one for each parameter.

```
privat:
  double interp (double *val, double x);
public:
  double radius_ip (double x);
  double velo_ip (double x);
  double hpres_ip (double x);
  double flow_ip (double x);
  double pres_ip (double x);
  double wavespeed_ip (double x);
```

The methods for setting the segmentation and initializing the variables for the dynamic calculation.

```
  void   setnseg (int n)
  void   setinitial (void);
```

The methods for the dynamic calculation

```
  double calcA (int ind, double tstp);
  double calcB (int ind, double tstp);
  double fricterm (double x);
  int    calc_tstp (double tstp);
  void   rm_calc_tstp (void);
```

The method getndconsts is called by the connected Node objects to get the parameters of the artery end for calculating the boundary condition. The other function takes those results from the Node object and completes the time step.

```
  void   getndconsts (TNode *nd, double *c1, double *c2,
                      double *area);
  void   calc_bound (TNode *node, double hp,
                     double v, int h_v);
};
```

6.4 Object NODE

According to the different types of boundary conditions and different types of nodes in the network there are also different types of Node classes:

1. Node is the general connecting node which represents exactly the boundary condition described in section 5.2. The number of arteries connected with this node is limited.

2. Node_Ref is derived from Node (as all following classes) and is used for the reference node with its variable flow curve as input to the net. Only one artery can be connected to this node.

3. NODE_COL is very similar to the NODE class, but can be used for other than constant outflows. It is used to represent the collecting node.
4. NODE_AC is the implementation of the air chamber (see section 5.3). Only two arteries can be connected to this node.

Similar to the ARTERY class an overloaded operator is used for data input and output. The main method is `calc_bound` (line 39, 83) which solves the equations for the boundary conditions.

Definition of class NODE and the derived types NODE_REF, NODE_COL and NODE_AC

```
 1  class TNode {
```

The operators for loading and saving the data are overloaded.

```
 2    friend istream& operator>> (istream&, TNode&);
 3    friend ostream& operator<< (ostream&, TNode&);
```

The parent NET object and the node properties.

```
 4  protected:
 5    TNet*    net;                 // net structure (parent)
 6    int      number;              // number of node
 7    double   height;              // geodetic height
 8    int      nodetype;            // type of node
 9    int      numofart;            // number of connected art.
10    TArtery* artlist[MAXARTCON];  // list of connected art.
```

Variables used for the hydrostatic calculation.

```
11    double   s_pressure;          // static pressure
12    double   s_outflow;           // static outflow
13    double   o_diameter;          // ouflow diameter
```

Except of the constructor and destructor methods all others are declared virtual, because some of them are redefined in derived classes. The functions `read_op` and `write_op` are called by the >>– and <<–operator so that they also can be redefined. Some of the methods are only useful in derived classes, but they have to be defined in the basic class in order to get a general interface.

```
14  public:
15    TNode (void);
16    TNode (TNet *n);
17    ~TNode (void);
18    virtual istream& read_op (istream&);
19    virtual ostream& write_op (ostream&);
```

```
  virtual int    getnumber (void);
  virtual double getheight (void);
  virtual void   setnodetype (int t);
  virtual int    getnodetype (void);
  virtual int    getnumofart (void);
  virtual int    addartery (TArtery* artery);
  virtual void   sets_pressure (double p);
  virtual double gets_pressure (void);
  virtual void   sets_outflow (double o);
  virtual double gets_outflow (void);
  virtual double geto_diameter (void);
  virtual double flow (double t);
  virtual double getarea (void);
  virtual double getfluid_stat (void);
  virtual double getgas_stat (void);
  virtual void   sets_inflow (double i);
  virtual double getinflow (void);
  virtual double gets_inflow (void);
  virtual double getgasconst (void);
```

The main function for calculating the boundary conditions.

```
  virtual void   calc_bound (double t, double tstp);
};
```

In the definition of the derived class Node_Ref only the method `flow` is redefined and adapted for variable inflow.

```
class TNode_ref : public TNode {
public:
  TNode_ref (void);
  TNode_ref (TNet *n);
  ˜TNode_ref (void);
  double flow (double t);
};
```

The class Node_Col is very similar to the previous one, but has an additional constructor, because the collecting node is not included in the network data files and therefore has to be added separatly.

```
class TNode_col : public TNode {
public:
  TNode_col (void);
  TNode_col (TNet *n);
  TNode_col (TNet *n, int nonr, double hei,
             double spres, double sflow);
  ~TNode_col (void);
  double flow (double t);
};
```

In class NODE_AC some new parameters are defined, therefore several methods have to be included for manipulating those parameters.

```
class TNode_ac : public TNode {
private:
  double area;            // sectional area of ac
  double fluid_stat;      // static fluid height
  double gas_stat;        // static gas volume
  double gas_small;       // smallest gas volume
  double s_inflow;        // static flow into ac
  double inflow;          // flow into ac at last time step
  double fluid_h;         // dynamic fluid height
  double gasconst;        // gas constant
public:
  TNode_ac (TNet *n);
  ~TNode_ac (void);
  istream& read_op (istream&);
  ostream& write_op (ostream&);
  void   sets_pressure (double p);
  int    addartery (TArtery *artery);
  double getarea (void);
  double getfluid_stat (void);
  double getgas_stat (void);
  double getgas_small (void);
  void   sets_inflow (double i);
  double getinflow (void);
  double gets_inflow (void);
  double getgasconst (void);
  void   setgasconst (void);
  void   calc_bound (double t, double tstp);
};
```

6.5 Subsidiary Objects

As mentioned in section 6.2 there are three more classes which are used for the calculation, but not for representing a part of the network structure. The most important one is the STATCALC class used for the static calculation. A detailed description of the implementation of the static model can be found in [Alm97a]. Only little changes were done to include this part into the new program. Because this package always adds the fictitious part of the network automatically, the data must be prepared before loaded into this module and the results have to be adapted before used for the initialization of the network.

The FUNCTION class is a very simple one, that reads and writes data and calculates a periodic linear interpolation and the mean value of the function described by the data. It is used for the heart flow function at the reference node. There are also features for scaling the time and the flow axis.

The remaining class is DATACOLL which is a simple interface to write out the different calculated quantities during the simulation. The positions for this data recording are loaded and saved using the overloaded >>– and <<–operators. The interpolation functions of the ARTERY class are used to get the results at the defined positions.

6.6 Summary

The modular construction of this implementation makes it easy to adapt the code to further developments of the model. For example if the elasticity model of the arteries is replaced by another one, only small changes in two methods of the ARTERY class are necessary.

Another advantage is that this object structure can be easily included in other program packages and with only a few function calls whole simulation runs can be executed. A small interface program was designed, which offers a simple simulation environment (see chapter A).

The way, the static calculations are implemented, seems a little bit complicated and confusing, because the network data are loaded into another object, which executes the calculations. It would be easier to include the calculations directly into the NET class. But this type of static model program was available and well tested, therefore it was decided to include this old model instead of a reimplementation.

Chapter 7

Validation and Results

Fulmen in clausula

7.1 3-Artery System

In order to validate and verify the model and its implementation, a simple network consisting of three arteries is used. These vessels represent the aorta and both leg arteries, because those are the most important parts of the cardiovascular system with the greatest influence on the pulse curves. Hence, such a simple model should be able to reproduce the main qualitative features of the pulse, although the numerical results are not representative for a real arterial system.

The parameters used for this network are set as for a standing human body (see figure 7.1), so the hydrostatic pressure induced by the natural gravity leads to greater pressure levels in the legs.

For this simulation experiment and all but one described in the following sections (except the insufficiency test) the global parameters are chosen as:

- heart rate: 75 bpm
- initial mean arterial pressure: 85 mmHg
- mean cardiac output: 5 l/min

This choice for the parameters represents a normal rest situation. The mean arterial pressure seems pretty low, but during the simulation the mean pressure at the heart is a slightly higher due to reflected pulse waves.

The dynamic flow out of the heart according to those parameters is shown in figure 7.2. The small negative flow at the end of the systole is caused by the closing valve.

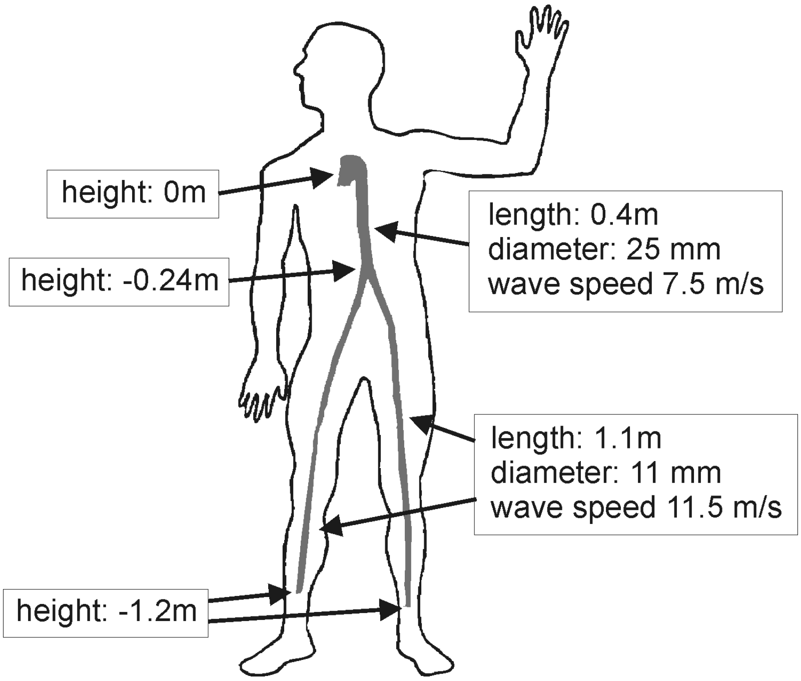


Figure 7.1: Properties of the 3-artery system

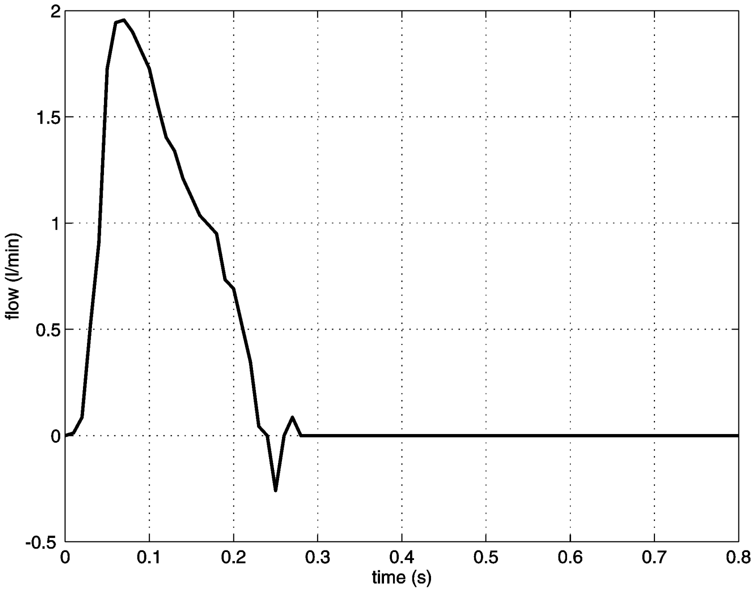


Figure 7.2: Flow at the heart (one beat)

In all simulation experiments it is necessary to do a 1-2 beat long precalculation in order to reach a quasi steady-state situation, that means the pulse curves do not change anymore.

The results in figure 7.3 show the pulse forms at three different positions. At 1.6 sec. the third beat starts with the opening of the valve at the heart. Therefore the pressure reaches its first maximum at about 1.68 sec. Then the flow out of the heart decreases but the reflection of the first peek at the bifurcation gives another peek (1.8 sec.) before the valve is closed. Then there is a strong decrease when the valve is closed and the diastole begins till the reflection of the first peek returns form the end of the leg arteries. Then the pressure level comes down to its minimum just before the next systole starts.

At the end of the aorta the form of the pulse curve is similar to that at the beginning, only a little bit delayed and the first peek is increased due to reflections. The pulse in the leg has only two peeks, the first is the increased systolic pressure, the second one is caused by the reflection at the closed valve during the diastole.

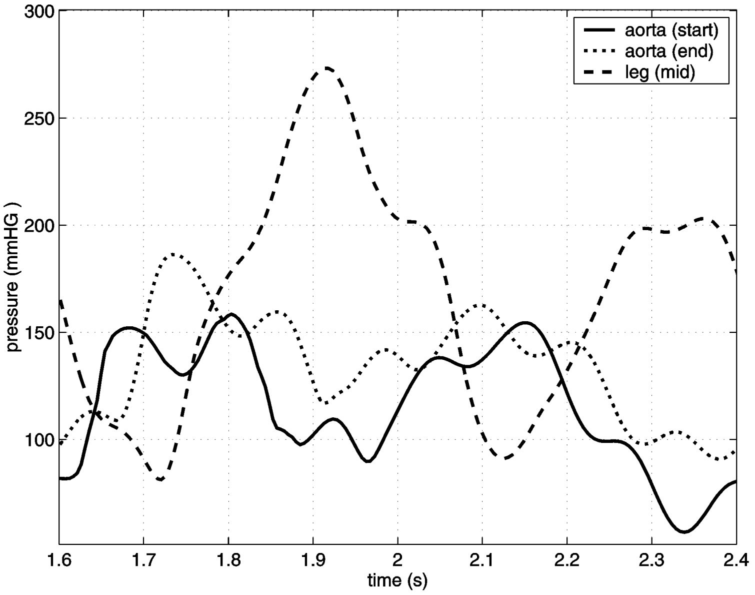


Figure 7.3: 3-artery system — pressure curves at the beginning and the end of the aorta and at the middle of the leg artery

The flow velocities (see figure 7.4) at the end of the aorta and in the leg have similar characteristics as the pressure curves. Only at the beginning of the aorta the flow velocity is constrained by the boundary condition.

The higher frequencies occurring in the pulse waves within the aorta causing about six peeks during one pulse are the results of the reflections at the bifurcation of the aorta and at the end of the leg arteries. Because of the simplicity of the network those are the only points where pressure waves can be reflected. In the

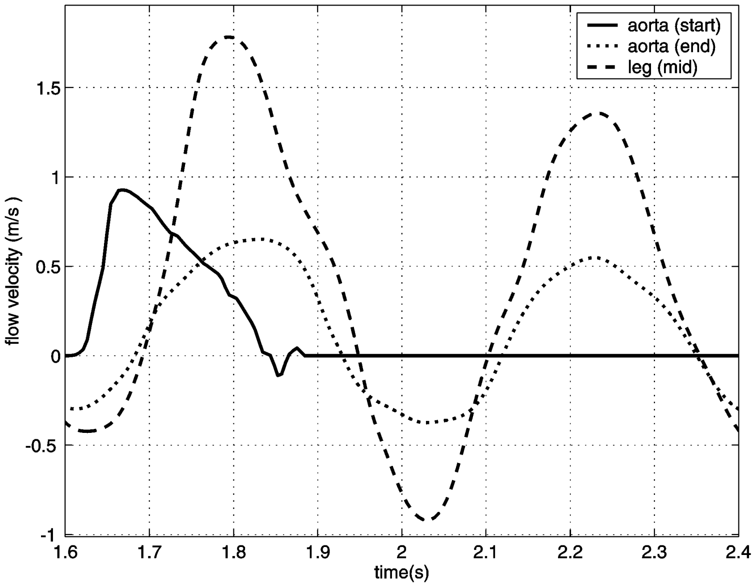


Figure 7.4: 3-artery system — flow velocities at the beginning and the end of the aorta and at the middle of the leg artery

real arterial system many bifurcations and turns of the vessels cause permanent small reflections, those result in a slow decrease of the arterial pressure during the diastole.

Another important factor is the air chamber effect of the aorta which smooths the pulse waves and lowers the peeks of the peripheral pressure (see next section).

7.2 3-Artery System with Air Chamber

The 3-artery system with air chamber differs only slightly from the previous model. After a fourth of the aorta an air chamber is included, the segment before the air chamber is a little bit wider and more elastic (lower wave speed), the segment behind is smaller and stiffer (higher wave speed), but the main characteristics of the network are the same (see figure 7.5). This system was chosen to validate the air chamber concept for the aorta, although the parameters of the air chamber differ from those in more complex networks.

The results of the pressure curves show similar features as those of the previous section but the peeks are smoothed and the arterial blood pressure in the aorta comes down from 160 mmHg (sys.) – 60 mmHg (dia.) to 125 mmHg (sys.) – 75 mmHg (dia.).

The dicrotic notch is clearly visible at the heart at about 1.85 sec., at the end of the aorta it is still observable at 1.9 sec., but smoothed. The pulse amplitude in the leg is about double that in the aorta which results in an arterial pressure of

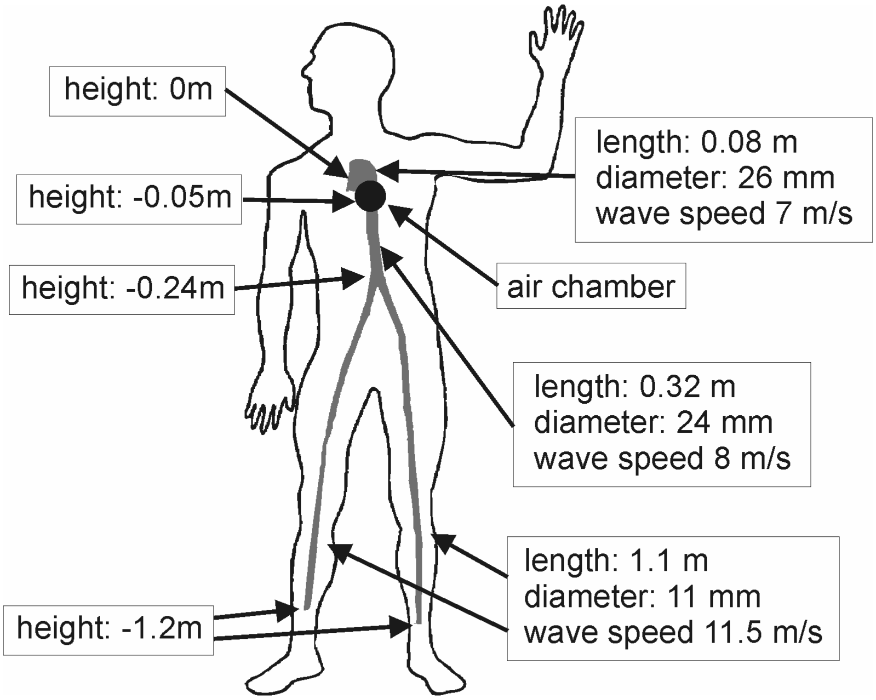


Figure 7.5: Properties of the 3-artery system with air chamber

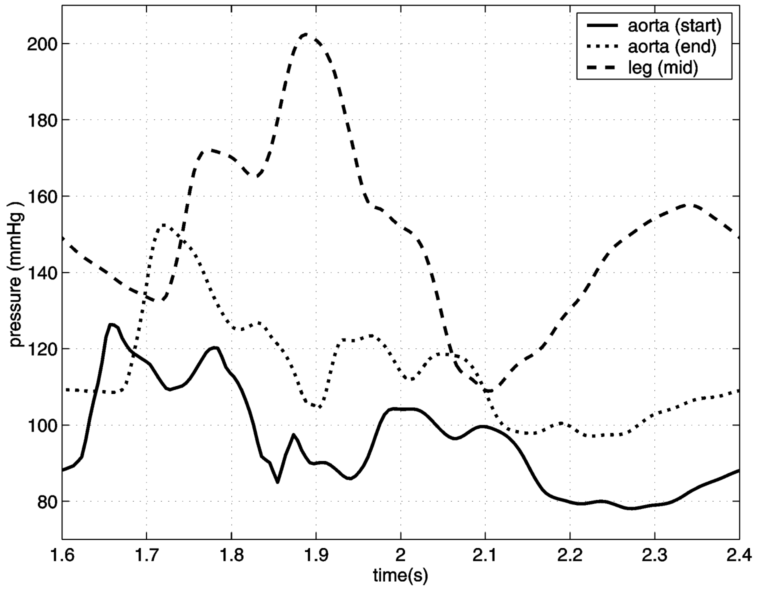


Figure 7.6: 3-artery system with air chamber — pressure curves at the beginning and the end of the aorta and at the middle of the leg artery

205 mmHg (sys.) – 110 mmHg (dia.). Those values include also the hydrostatic pressure which is about 55 mmHg at that position.

The qualitative features of the flow velocity curves of the 3-artery system with air chamber can be compared with the results of the model without an air chamber, but a quantitative analysis shows that the velocities are lowered significantly. So also the backwards flow is decreased in intenseness and duration.

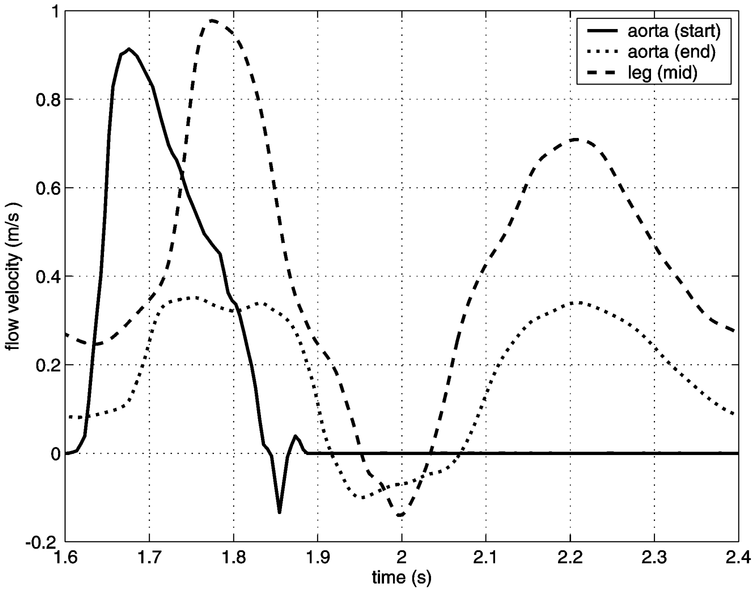


Figure 7.7: 3-artery system with air chamber — flow velocities at the beginning and the end of the aorta and at the middle of the leg artery

7.3 Human Arterial System — Normal Condition

The following three sections describe different experiments on a rough overall model of the human arterial system consisting of about 150 artery segments (see figure 7.8). Again the body is in an upright position as it is shown in the figure, which causes the hydrostatic pressure to affect the pulse.

There are two air chambers added to the aorta, one near the heart, the other near the bifurcation. The wave speed is set between 6 m/s at the heart and 14 m/s in the peripheral arteries in the feet and the hands, which means a high elasticity of the vessels — normal values for persons at young ages. All other parameters are the same as in section 7.1.

During the simulation runs the results are recorded at four different positions (see figure 7.8):

- aortic arch (Arcus Ao.)

- aorta before the bifurcation (Ao. abdominalis)
- thigh (A. femoralis)
- lower leg (A. peronaea)

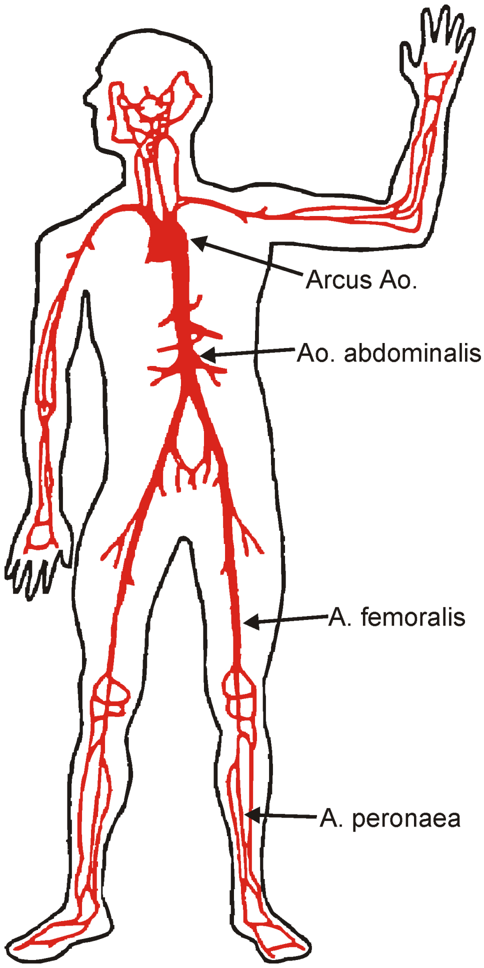


Figure 7.8: Human arterial system with positions for data recording

The pulse curves (see figure 7.9) show all important qualitative and quantitative characteristics as measured arterial pressure curves described in relevant literature. Both pulse curves recorded in the aorta have a strong pressure increase at the begin of the systole, a little dicrotic notch , and a slow decrease of the pressure during the diastole. The pressure curves in the leg have a second peek due to the reflection at the closed valve and the amplitude is about double that in the aorta. Measuring the pulse in the upper arm shows a blood pressure of about 128 mmHg (sys.) – 70 mmHg (dia.), which seem a very realistic value.

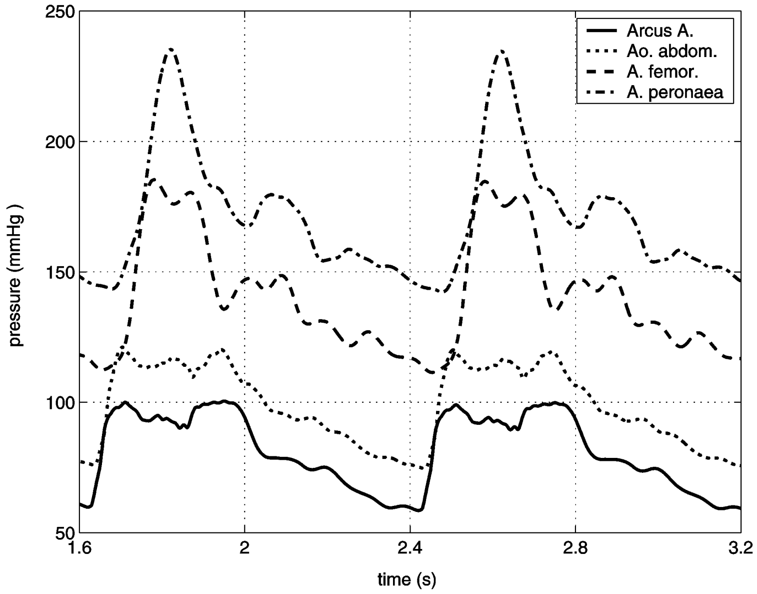


Figure 7.9: Human arterial system — normal condition — pressure curves (2 beats)

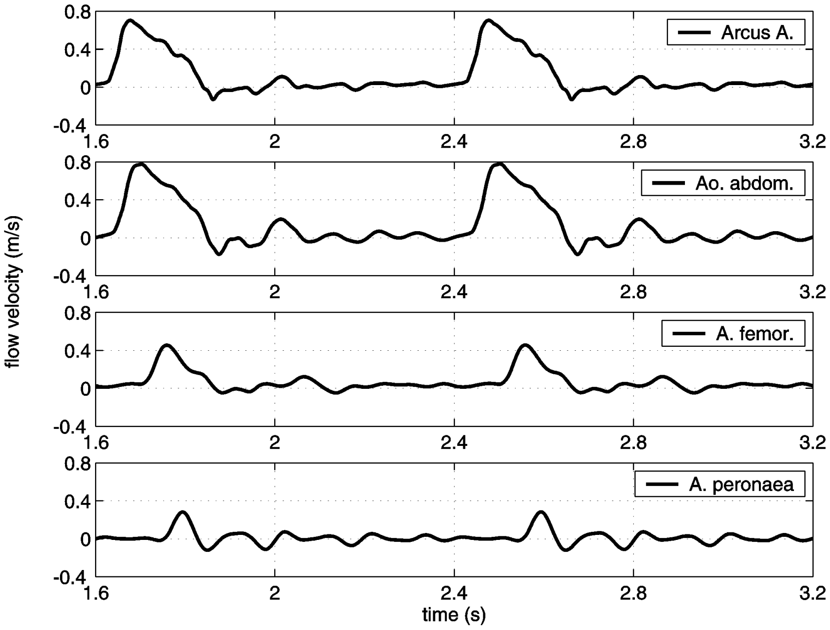


Figure 7.10: Human arterial system — normal condition — flow velocities (2 beats)

The flow velocity curves (see figure 7.10) in the aorta stay nearly the same all along the vessel. In the leg arteries the velocity is reduced to the half, although the backwards flow is still clearly noticeable.

The figure 7.11 shows the relationship between the pressure, the diameter, and the wave speed in the a. femoralis. The diameter is directly proportional to the pressure and the wave speed indirect proportional to the diameter. Due to the hydrostatic pressure the diameter is enlarged from its original 13 mm to about 13.5 mm and the wave speed is lowered from 10.4 m/s to 9-10 m/s.

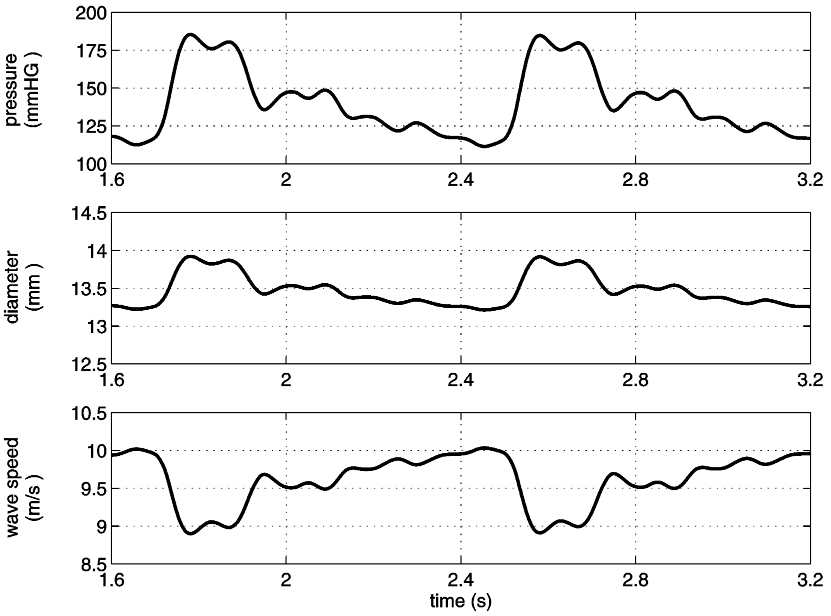


Figure 7.11: Human arterial system — normal condition — relationship between pressure, radius, and wave speed in the a. femoralis (2 beats)

7.4 Human Arterial System — Stiff Vessels

This test should show the effects of the reduced elasticity of the arteries. Therefore the wave speed of all vessels are increased by 25%, which would be appropriate for elder, but still healthy persons. All other model parameters remaining the same as in the previous section.

As presumed figure 7.12 shows that the pressure amplitude is increased significantly. On account of the higher wave speeds in the arteries there are more and greater reflections which have a noticeable impact on the pulse in the peripheral arteries. Also the blood pressure value in the upper arm is raised to about 138 mmHg (sys.) – 60 mmHg (dia.).

The flow velocities are a little bit lower than in the previous case, but the

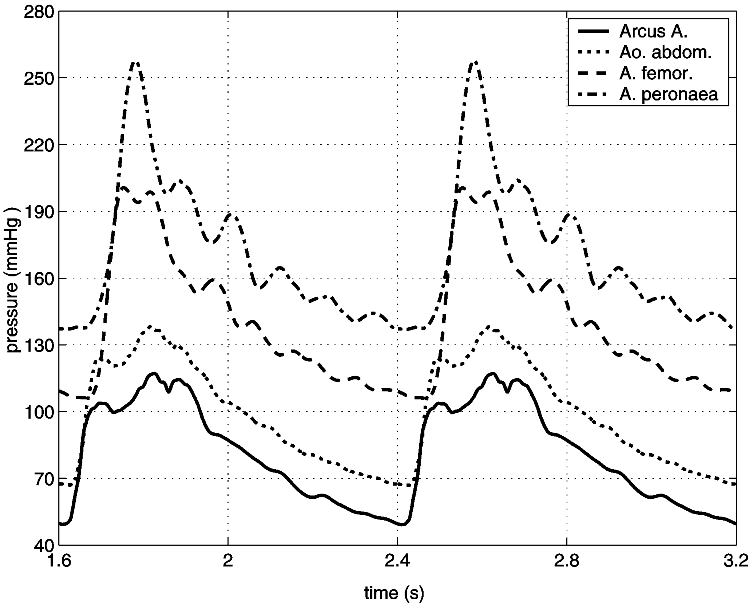


Figure 7.12: Human arterial system — stiff vessels — pressure curves (2 beats)

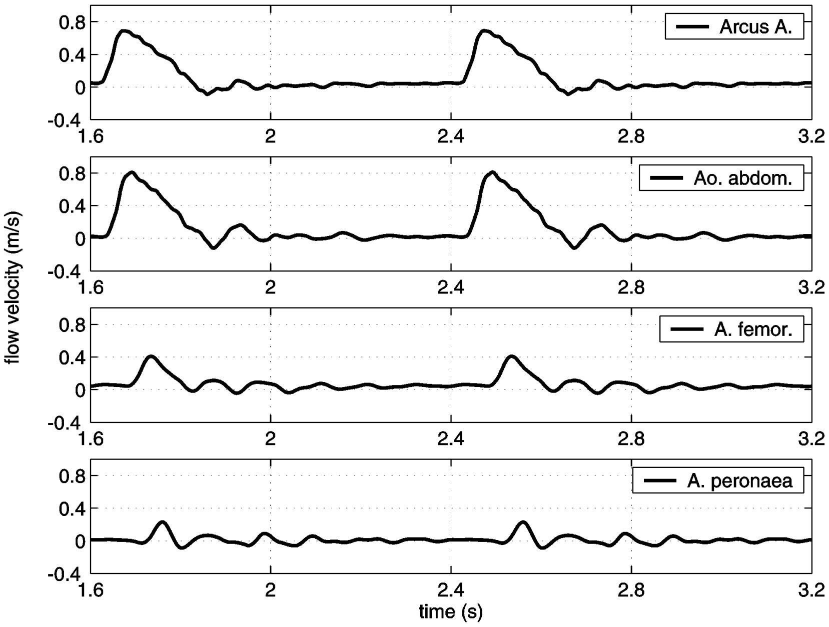


Figure 7.13: Human arterial system — stiff vessels — flow velocities (2 beats)

patterns are very similar, although the higher wave speeds produce greater fluctuations in the leg.

7.5 Human Arterial System — Insufficiency

Now the parameters are again the same as in section 7.3, but a valvular insufficiency is assumed by using a different flow function at the heart (see figure 7.14). Because of the valvular defect blood flows back into the heart during the diastole. The outflow during the systole is increased so that the mean cardiac output is again about 5 l/min.

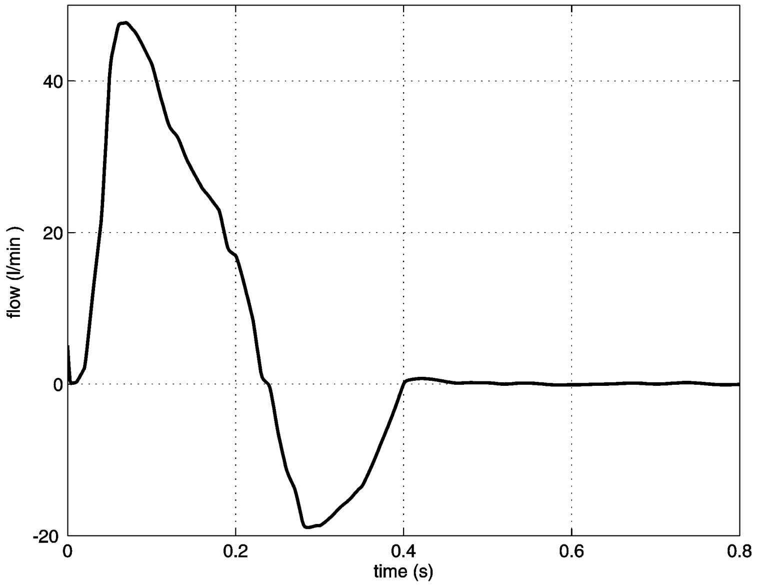


Figure 7.14: Flow at the heart - insufficiency

Comparing figure 7.15 with figure 7.9 the pressure amplitude increases by about 50% due to the strong back-flow into the heart. So the blood pressure in the upper arm is 151 mmHg (sys.) – 64 mmHg (dia.). The second peek of the pulse in the leg caused by reflections is increased significantly, so that the dicrotic notch comes out clearly.

The flow velocities (see figure 7.16) are greater than under normal conditions, but there are also longer periods of strong backwards flow. In the thigh a second peek of positive flow occurs after the backwards flow, and in the lower leg there are three longer periods of substantial back-flow. So the valvular insufficiency generates high fluctuations of the flow velocities.

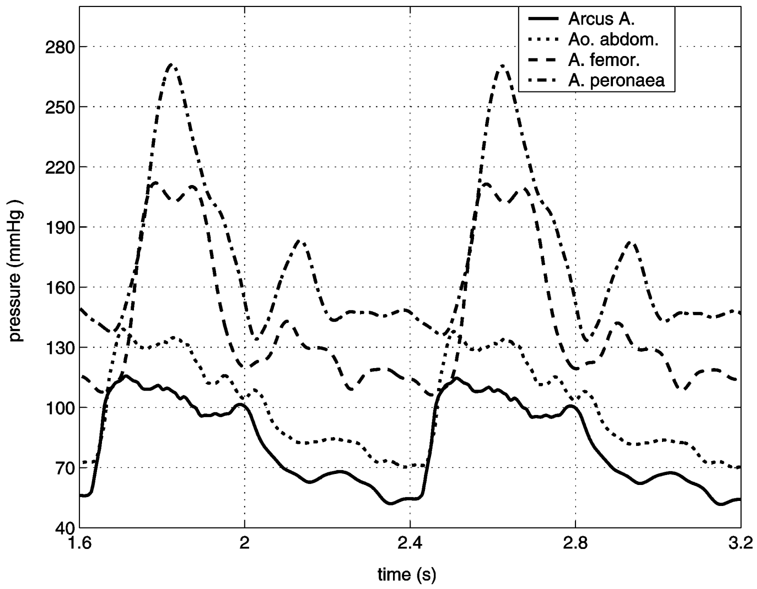


Figure 7.15: Human arterial system — insufficiency — pressure curves (2 beats)

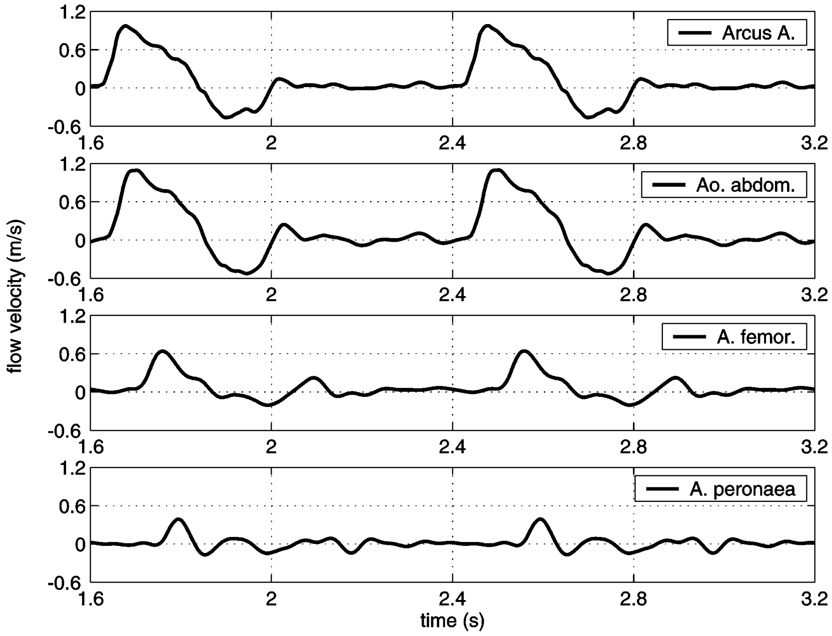


Figure 7.16: Human arterial system — insufficiency — flow velocities (2 beats)

7.6 Summary

The tests with the simple 3-artery systems were performed in order to validate and verify the model assumptions and the implementation. As expected the results reproduced the qualitative features of a real system, but they were far away from representing the highly dynamical complex processes going on in the cardiovascular system. On the other hand those tests showed also that such a simple network contains the most important facts which dominate the behavior of the pulse in the arterial system.

Extending the simple 3-artery systems to a full arterial model showed that with the increased complexity of the network, also the complex structure of the pulse can be modelled in a more detailed way. The results show a very good accordance with measured pressure and flow velocity curves of real systems. A further improvement would be the extension of the networks used in here.

The computational time for the whole system is directly proportional to the mean flow velocities in the vessels and the wave speeds. The full networks took about 5-10 min. to simulate 4 beats (3.2 sec.) on a PC (400MHz, Linux).

Chapter 8

Conclusion and Outlook

Intentions for the next millennium.

The model of the human arterial blood flow developed in this dissertation is a new extension of the model introduced by *Martin Suda* at the *Austrian Research Centre Seibersdorf* [Sud95]. It is based more on physiological and physical facts rather than using heuristic assumptions or substitute models (i.e. compartment models, spring models, ...).

In general this model uses a 1-1 mapping to represent the arterial network, which makes it easier to interprete and understand the meaning of certain parameters. It gives also the possibility to gain detailed results in any part of the network, and to have the opportunity to extend the model by adding more vessels.

Several new features were developed in this work, which enhances the capability of the model in many different ways:

Fictitious arteries: The transfer of the fictitious arteries concept from the static model to the dynamic one allows to include the peripheral resistance and also the microcirculation. So the parts of cardiovascular system, which lay outside the model boundaries are taken into consideration.

Wall elasticity: This concept of flexible tubes, which means variable diameters and wave speed, is not a new one, but for the first time it was used in such an arterial network model. This enables the performing of long term simulation experiments with a better preciseness by avoiding pressure increase or decrease tendencies. The fact, that ideal gas is used in the air chambers, reduces the calculation time significantly, but still is useful for modelling the aortic elasticity.

Artery connections: A special formulation of the boundary conditions in the method of characteristics concept simplifies the connection rules, such that no more fall differentiation according to the different flow situations is necessary.

So this model allows to investigate the hydraulic and hydrodynamic effects of a variety of diseases. So not only valvular defects, as presented in section 7.5, but also other things like local or global arteriosclerosis, different kinds of arrhythmia, stenoses and occlusions of vessels, and other disease of the cardiovascular system can be modelled and simulated.

In fact of the high complexity of the human arterial system the parameter identification for an individual patient is very difficult, but this model will profit from the improvement of the medical measuring and imaging facilities. Nevertheless a clinical study will be necessary for a better validation and further improvements of the system. Other possible extensions are

- a better elasticity model of the arterial walls, especial the aorta, so that the air chamber concept can be simplified or removed completely.
- an additional resistance term at the bifurcations due to the angle between the connected arteries.
- an improvement of the solving algorithm with regard to preciseness and computation time, in order to use larger models with more artery segments.
- an adaption of the model in order to use it for the low-pressure part of the cardiovascular system (venous system, pulmonary system), too.

The hydrodynamic model of the human arterial blood flow can be used as one starting point for developing overall cardiovascular models, which are based on a biophysical and biochemical description of the human physiology. In contrast to other models of the blood flow it allows a macroscopic view and a systemic assessment by simulating a whole network in stead of only small parts. [Alm99a, Alm99b, Iid89a, Iid89b]

Appendix A

Simulation Interface Program PULSE

In order to automatize the simulation experiments a command line program was designed, which uses an configuration file for general settings and a command file for controling the simulation run. The data formats for the data and result files are described in appendix B.

As the whole implementation was done in Ansi-C++, the code is highly portable. Therefore there are a Unix and a Micrsoft Windows version of Pulse available.

The call of the program Pulse includes some optional parameters:

```
pulse [-h] [-v] [-c configfile] commandfile
```

`-h`	display help information.
`-v`	verbose mode (write information on standard output)
`-c configfile`	use `configfile` as configuration file (default: pulse.cfg)
`commandfile`	command file

In the command file several commands are available to control the simulation experiments. All blank lines and comment lines (starting with //) are neglected, the others are executed successively. The commands are case insensitive, but not the parameters of the commands (e.g. file names). The commands must be on one line and cannot be split over several ones. An example configuration file can look like this (a list of all commands is described in table A.1)

```
// load the model from a file
LoadModel(ncstruct.dat)

// set data collecting file
SetDataCollFile(dc.dat)

// set data output files
```

```
SetDataOutput(f.dat,v.dat,p.dat,h.dat,d.dat,w.dat)

// perform static calculation
StaticCalc()

// initialize network for dynamic calculation
// with automatic segmentation
InitNet(0)

// perform simulation with time step control
// in verbose mode
SimulationRun(0,5,0,0,1,simrun.log)
```

The configuration file is used to set global constants for the calculation and the units for data input and output. The typical command in the configuration file is

keyword=value(or string)

Comments can be included at any point of the file by starting with //. The rest of the line will be ignored. Also blank lines are neglected. All commands and strings are case insensitive. For data input and output different units can be used (see table A.2 for available units).

Before starting Pulse there must exist a configuration file even if it is empty. If no configuration file is specfied in the command line, then the program tries to open automatically the standard configuration file pulse.cfg. If this file does not exist, the program stops with an error message. There are some more keywords than listed in table A.3, which are only for internal use or debugging.

<table>
<tr><th>command</th><th>description</th></tr>
<tr><td>InitNet(<i>seglen</i>)</td><td>initialize the network using <i>seglen</i> as segmentation length (0 — automatic segmentation)</td></tr>
<tr><td>LoadModel(<i>netfile</i>)</td><td>load the network data from <i>netfile</i></td></tr>
<tr><td>SaveModel(<i>netfile</i>)</td><td>save network data (including results of hydrostatic calculation) to <i>netfile</i></td></tr>
<tr><td>SetDataCollFile(<i>dcfile</i>)</td><td><i>dcfile</i> will be used as data collecting file which includes the data recording positions</td></tr>
<tr><td>SetDataOutput(<i>flowfile</i>, <i>velofile</i>, <i>presfile</i>, <i>hpresfile</i>, <i>diafile</i>, <i>wvspfile</i>)</td><td>sets the file names for storing the results
<table>
<tr><td><i>flowfile</i></td><td>flow data</td></tr>
<tr><td><i>velofile</i></td><td>velocity data</td></tr>
<tr><td><i>presfile</i></td><td>pressure data</td></tr>
<tr><td><i>hpresfile</i></td><td>pressure head data</td></tr>
<tr><td><i>diafile</i></td><td>diameter data</td></tr>
<tr><td><i>wvspfile</i></td><td>wave speed data</td></tr>
</table></td></tr>
<tr><td>SimulationRun(<i>timestart</i>, <i>timeend</i>, <i>timestep</i>, <i>comint</i>, <i>verbose</i>, <i>logfile</i>)</td><td>executes a dynamic simulation experiment
<table>
<tr><td><i>timestart</i></td><td>start time of simulation</td></tr>
<tr><td><i>timeend</i></td><td>end time of simulation</td></tr>
<tr><td><i>timestep</i></td><td>time step
0 — time step control
-1 — maximal time step
others — fixed time step</td></tr>
<tr><td><i>comint</i></td><td>communication interval
0 — every time step</td></tr>
<tr><td><i>verbose</i></td><td>0 — non verbose mode
1 — verbose mode</td></tr>
<tr><td><i>logfile</i></td><td>name of log file</td></tr>
</table></td></tr>
<tr><td>StaticCalc()</td><td>execute static calculation</td></tr>
</table>

Table A.1: Command description for the PULSE command file

length **pressure head**	**velocity** **wave speed**	**flow**	**pressure**
mm	mm/s	l/s	Pa
cm	cm/s	mˆ3/s	kPa
dm	dm/s	l/min	bar
m	m/s	mˆ3/min	mmHg
km	km/s	l/h	mWS
	mm/min	mˆ3/h	
	cm/min		
	dm/min		
	m/min		
	km/min		
	mm/h		
	cm/h		
	dm/h		
	m/h		
	km/h		

Table A.2: Units for the configuration file

keyword	**default**	**description**
`baroheight`	10.1972	barometric height (m)
`bisect_limit`	$1e-7$	break limit for bisection algorithm
`delimiter`	$\|\|$	delimiter used in data files
`fict_art_base`	10000	start number for fictitious arteries
`fict_art_len`	0.1	length of fictitious arteries (m)
`fict_art_nseg`	-1	number of segments (-1 — variable)
`fict_art_wsp`	7	wave speed of fictitious arteries (m/s)
`gravity`	9.80665	natural gravity (m/s)
`node_col_num`	10000	identification number of collecting node
`outprec_data`	5	output precision of dynamic results
`outprec_net`	4	output precision of static results
`tstp_change`	500	changes of time step (maxtstp/tstp_change)
`unit_flow_i`	mˆ3/h	unit for flow input data
`unit_flow_o`	mˆ3/h	unit for flow output data
`unit_head_i`	m	unit for pressure head input data
`unit_head_o`	m	unit for pressure head output data
`unit_length_i`	mˆ3/h	unit for length input data
`unit_length_o`	mˆ3/h	unit for length output data
`unit_pressure_i`	Pa	unit for pressure input data
`unit_pressure_o`	Pa	unit for pressure output data
`unit_velocity_i`	m/s	unit for velocity input data
`unit_velocity_o`	m/s	unit for velocity output data
`unit_wavespeed_i`	m/s	unit for wave speed input data
`unit_wavespeed_o`	m/s	unit for wave speed output data

Table A.3: Keywords for the configuration file

Appendix B

Data Formats

All data files are coded in Ascii, so that they can be read and changed by hand in a very simple way. In all files delimiters are used to seperate the data entries, which are on one line. This delimiter character can be changed in the configuration file of the program. In this description DL indicates such a delimiter. The units written beside the parameters are the default ones. They can also be changed in the configuration file.

The input data is split up into three different files:

Network data file: It stores the network structure.

Heart flow file: It gives the blood flow from the heart into the first node.

Data collecting file: This file specifys the positions in the network where data should be recorded during the simulation.

The output data is also split up into several files each for one kind of data. All these files have the same format and store the particular data for all positions which are given by the data collecting file. In the first two lines of those data output files the positions are stored where the data were recorded. Then the data follows columnwise, in the first column there is always the simulation time, the others show the data recorded at that time. The results of the hydrostatic calculations and the initialization of the network are stored again in the network data file.

line	**description**
1	model title
2	status 0 only network data 10 finished static calculation 20 initialized DL closed 0 add no fictitious part 1 add fictitious part 2 add fictitious part and save it 3 (status>0) outflow for collecting node DL number of nodes DL number of arteries DL kinematic viscosity of blood (m^2/s) DL specific density of blood (kg/m^3s) DL static reference pressure (Pa) DL static pressure at collecting node (Pa) (output) DL static reference flow (m^3/h) DL maximum time step (output)
3	file name for the flow function (heart flow file)
4 — $n_\mathrm{n}+3$	type of node 0 normal connecting node 10 reference node 11 collecting node 20 air chamber DL identification number DL geodesical height (m) DL static pressure (Pa) (output) DL type=0,10,11 outflow diameter (m) DL static outflow (m^3/h) (output) type=20 sectional area of the air chamber (m^2) DL static fluid height (m) DL static gas volume (m^3) DL smallest ags volume (m^3) DL inflow into the air chamber (m^3/h) (output)
$n_\mathrm{n}+4$ — $n_\mathrm{a}+n_\mathrm{n}+3$	identification number DL id number of start node DL id number of end node DL length of artery (m) DL static diameter of artery (m) DL static wavespeed of artery (m/s) DL static flow through artery (m^3/s) (output) DL static velocity in artery (m/s) (output) DL number of segments (output)

Table B.1: Format of the network data file

line	**description**
1	number of data points DL length of period (0 — no period)
2	time value scale DL function value scale
3 — end	time value DL function value

Table B.2: Format of the heart flow file

line	**description**
1	*positions for flow data recording* artery DL position DL artery DL position . . . -1 DL -1
2	*positions for velocity data recording* artery DL position DL artery DL position . . . -1 DL -1
3	*positions for pressure data recording* artery DL position DL artery DL position . . . -1 DL -1
4	*positions for pressure head data recording* artery DL position DL artery DL position . . . -1 DL -1
5	*positions for diatemer data recording* artery DL position DL artery DL position . . . -1 DL -1
6	*positions for wave speed data recording* artery DL position DL artery DL position . . . -1 DL -1

Table B.3: Format of the data collection file (Positions are between 0 (start) and 1 (end).)

Appendix C

MATLAB Interface

The program Pulse was extended by a simple Matlab interface, so that Matlab can be used as simulation environment and its tools can be used for data pre- and postprocessing. The interface consists of three different M-functions, for loading network data into the Matlab workspace, saving network data to a file and executing a simulation experiment.

For handling the network data and the results, a certain `net` structure is used. The information is stored in several fields and substructures.

`net.title` — title of the model

`net.node` is matrix holding the node information (see table C.1). This matrix has one row for each node and 11 columns for the different properties (the columns 7 – 11 are only used for nodes with air chambers).

column	**contents**
1	identification number
2	node type
3	geodesical height
4	static pressure
5	outflow diameter
6	outflow
7	sectional area of air chamber
8	static fluid height in air chamber
9	static gas volume in air chamber
10	minimal gas volume in air chamber
11	initial inflow into in air chamber

Table C.1: Structure of the `net.node` matrix

`net.artery` is also matrix storing the artery properties, each artery in one row (see table C.2).

column	**contents**
1	identification number
2	number of start node
3	number of end node
4	length of artery
5	diameter of artery
6	static wave speed
7	static flow
8	static flow velocity
9	number of segments

Table C.2: Structure of the `net.artery` matrix

`net.inflow` describes the inflow function into the network. It is a $n \times 2$–Matrix, in the first column are the time values, in the second the flow values.

`net.period` — the period of the inflow function.

`net.bloodvisc` — kinematic viscosity of blood

`net.blooddens` — specific density of blood

`net.refpres` — static pressure at reference node

`net.colpres` — static pressure at collecting node

`net.refflow` — static flow at reference node

`net.status` — simulation status

`net.closed` — close network with fictitious part (see table B.1)

`net.maxtstp` — maximal time step

`net.results` — data structure for the results of a simulation experiment

`net.results.time` — vector of simulation time steps

`net.results.flow` — flow data, one column for each position

`net.results.velo` — flow velocity data

`net.results.pres` — pressure data

`net.results.hpres` — pressure head data

`net.results.dia` — diameter data

`net.results.wvsp` — wave speed data

`net.results.pos` — data recording positions where the first column gives the identification numbers of the arteries, the second column holds the positions within the artery (a number between 0 and 1).

`net.results.unitflow` — unit for the flow data

`net.results.unitvelo` — unit for the flow velocity data

`net.results.unitpres` — unit for the pressure data

`net.results.unithpres` — unit for the pressure head data

`net.results.unitdia` — unit for the diameter data

`net.results.unitwvsp` — unit for the wave speed data

The function for loading the network data into the MATLAB workspace has only the file name of the data file as argument. It returns the `net` structure as described before.

```
net = loadnetfile(filename)
```

Although only the network data file is given, also the heart flow file is loaded. Therefore it must be available in the same directory.

The function for saving the network using the data formats of the PULSE program is

```
savenetfile(netfile, flowfile, net)
```

It is necessary to specify a name for the network data file and also a name for the heart flow file.

The main function for executing a simulation run is

```
netres = pulsesimul(net, seglen, tstart, tend, tstp, pos, cint)
```

It returns the whole `net` structure including the results. The calling arguments are the network data in `net`, the segmentation length `seglen` (0 — automatic segmentation), the start time `tstart` and end time `tend` of the simulation run, the time step `tstp` (0 — automatic time step control), the positions for data recording `pos` (using the same format as for `net.results.pos`), and the communication interval `cint` (0 — data recording after every time step).

So the MATLAB environment can be used to manipulate the network data, control the dynamic pulse simulation, and analyze the results.

Bibliography

[Alm97a] Ch. Almeder. *Simulation of the Human Arterial System.* Diploma theses, Technical University of Vienna, 1997.

[Alm97b] Ch. Almeder, F. Breitenecker, S. Wassertheurer, K. Kaser, J. Krocza, M. Suda. *Modelling of the human arterial network for an expert system for preoperative predictions.* In: Proceedings of the 2nd Mathmod, p. 1110, Vienna, 1997.

[Alm98a] Ch. Almeder, F. Breitenecker, S. Wassertheurer, K. Kaser, J. Krocza, M. Suda. *Interaktive Beratungssoftware für Gefäßoperationen.* In: Proceedings of Simulation und Visualisierung 98, Magdeburg, 1998.

[Alm98b] Ch. Almeder, F. Breitenecker, S. Wassertheurer, K. Kaser, J. Krocza, M. Suda. *Modelling of the Human Arterial Network for Preoperative Predictions.* In: Proceedings of the 12th European Simulation Multiconference, pp. 313–315, Manchester, 1998.

[Alm99a] Ch. Almeder, F. Breitenecker, J. Krocza, M. Suda. *Simulation of the Humand Arterial System - Static and Dynamic.* In: Proceedings of the 1st BMES-EMBS Conference, Atlanta, 1999.

[Alm99b] Ch. Almeder, F. Breitenecker, J. Krocza, N. Popper, M. Suda. *Simulation of the Transient Flow of Blood through the Human Arterial System.* In: Proceedings of EMBEC, pp. 1152–1153, Vienna, 1999.

[Anl71] M. Anliker, R.L. Rockwell. *Nonlinear Analysis of Flow Pulses and Shock Waves in Arteries.* Journal of Applied Mathematics and Physics, 22:217–246, 563–581, 1971.

[Bal89] S.D. Balar, T.R. Rogge, D.F. Young. *Computer Simulation of Blood Flow in the Human Arm.* Journal of Biomechanics, 22(6/7):691–697, 1989.

[Boe93] L. Boeswirth. *Technische Strömungslehre.* Vieweg, 1993.

[Bra97] B.R. Bracio, W. Horn, D.P.F. Moeller. *Sensor Fusion in Biomedical Systems.* In: Proceedings of the 19th EMBS Conference, Chicago, 1997.

[Bus82] R. Busse, Hrsg. *Kreislaufphysiologie.* Georg Thieme Verlag Stuttgart – New York, Stuttgart, Germany, 1982.

[Ede88] O.J. Eder, M. Suda. *Network Simulation of an Arterial System using a Hydrodynamic Model.* Proceedings of BME-Austria, pp. 174–178, 1988.

[Fox85] R.W. Fox,A.T. McDonald. *Introduction to Fluid Mechanics.* John Wiley & Sons, 1985.

[Fun84] Y.C. Fung. *Biodynamics (Circulation).* Springer-Verlag, New York, 1984.

[Gan74] W.F. Ganong. *Lehrbuch der Medizinischen Physiologie.* Springer Verlag Berlin – Heidelberg – New York, Berlin, Germany, 3nd ed.. Auflage, 1974.

[Gri76] B. Gribbin, A. Steptoe, P. Sleight. *Pulse Wave Velocity as a Measure of Blood Pressure Change.* Physiology, 13(1):86–90, 1976.

[Hor89] J.B.A.M. Horsten, A.A. Van Steenhoven, M.E.H. Van Dongen. *Linear Propagation of Pulsatile Waves in Viscoelastic Tubes.* Journal of Biomechanics, 22(5):447–484, 1989.

[Hwa77] N.H.C. Hwang, N.A.Normann, Hrsg. *Cardiovascular Flow Dynamics and Measurements.* University Park Press, Baltimore, USA, 1977.

[Iid89a] N. Iida. *Physical Properties of Resistance Vessel Wall in Peripheral Blood Flow Regulation - I. Mathematical Model.* Journal of Biomechanics, 22(2):109–117, 1989.

[Iid89b] N. Iida. *Physical Properties of Resistance Vessel Wall in Peripheral Blood Flow Regulation - II. Structural and Mechanical Behavior.* Journal of Biomechanics, 22(2):119–127, 1989.

[Jun72] H. Jungmann, E. Gadermann. *Diagnostische Möglichkeiten der unblutigen Arterienpulsregistrierung.* Medizinische Klinik, 67(9):281–285, 1972.

[McD74] D.A. McDonald. *Blood Flow in Arteries.* Edward Arnold, London, Great Britain, 2nd ed.. Auflage, 1974.

[Ped80] T.J. Pedley. *The Fluid Mechanics of Large Blood Vessels.* Cambridge University Press, Cambridge, Great Britain, 1980.

[Seu82] G.J. Seus, G. Schmitz. *Calculation of unstable pipe flow by the Imoc method - A new solution to an old problem.* 3R International, 21(1/2):32–38, 1982.

[Sil83] S. Silbernagl, A. Despopoulos. *Taschenatlas der Physiologie.* Georg Thieme Verlag Stuttgart – New York, Stuttgart, Germany, 2nd ed.. Auflage, 1983.

[Str67] V.L. Streeter, E.B. Wylie. *Hydraulic Transients.* McGraw-Hill Book Company, USA, 1967.

[Sud87a] M. Suda. *Ein Computerprogramm zur Berechnung hydraulischer Transienten in komplexen Leitungsnetzen.* GWF Wasser – Abwasser, 128(7):371–378, 1987.

[Sud87b] M. Suda, H. Hick, H. Willer. *Computersimulation dynamischer Druckschwankungen in Wasserleitungsnetzen.* OEFZS-A-0982, Austrian Research Centre Seibersdorf, 1987.

[Sud93] M. Suda, O. Eder, B. Kunsch, D. Magometschnigg, H. Magometschnigg. *Preoperative assessment and prediciton of postoperative results in an artificial arterial network using computer simulation.* Computer Methods and Programs in Biomedicine, 41:77–87, 1993.

[Sud95] M. Suda. *Die Computersimulation des arteriellen Kreislaufs als diagnostisches Instrument der Gefäßchirurgie: Endbericht.* OEFZS-A-3579, Austrian Research Centre Seibersdorf, 1995.

[Wat79] G.Z. Watters. *Modern Analysis and Control of Unsteady Flow in Pipelines.* Ann Arbor Science, Ann Arbor, Michigan, 1979.

[Wyl93] E.B. Wylie, V.L. Streeter. *Fluid Transients in Systems.* Prentice - Hall, Englewood Cliffs, New Jersey, 1993.

[Zoe82] H. Zoebl, J. Kruschik. *Strömungen durch Rohre und Ventile.* Springer-Verlag, Wien, 1982.

Index