Die approbierte Originalversion dieser Diplom-/Masterarbeit ist an der Hauptbibliothek der Technischen Universität Wien aufgestellt (http://www.ub.tuwien.ac.at).

The approved original version of this diploma or master thesis is available at the main library of the Vienna University of Technology (http://www.ub.tuwien.ac.at/englweb/).

TECHNISCHE UNIVERSITÄT WIEN

VIENNA UNIVERSITY OF TECHNOLOGY

DIPLOMARBEIT

Implementierung eines eindimensionalen Blutflussmodells mittels Finiter Elemente Methode

ausgeführt am Institut für

Analysis und Scientific Computing

der Technischen Universität Wien

unter Anleitung von

Ao.Univ.Prof. Dipl.-Ing. Dr.techn. Felix Breitenecker

durch

Bernhard Hametner

Birkenweg 10

3292 Gaming

Datum

Unterschrift

Kurzfassung

In vielen hochentwickelten Ländern stellen Erkrankungen des Herzkreislaufsystems die häufigste Todesursache dar. Daher ist die Forschungsarbeit auf diesem Gebiet von großer Bedeutung. Die Modellbildung und Simulation des Herzkreislaufsystems nimmt dabei eine wichtige Rolle ein.
Diese Arbeit beschäftigt sich mit der Herleitung und Simulation eines Blutflussmodells für größere Arterien, die einen wesentlichen Bestandteil des menschlichen Blutkreislaufsystems darstellen.

Den Ausgangspunkt der Betrachtungen bilden die Navier-Stokes-Gleichungen für inkompressible Flüssigkeiten. Unter verschiedenen Annahmen, wie die zylindrische Form der Arterien, wird daraus ein Modell in einer Raumdimension abgeleitet. Dieses Blutflussmodell beinhaltet als Zustandsgrößen die Gefäßquerschnittsfläche und den Volumenfluss. Es kann als hyperbolische partielle Differentialgleichung erster Ordnung in Erhaltungsform angeschrieben werden.

Um das Differentialgleichungssystem numerisch zu lösen, wird die Finite Elemente Methode benutzt. Dabei wird ein Taylor-Galerkin Verfahren zweiter Ordnung angewendet. Die Bestimmung der Randwerte geschieht mit Hilfe der charakteristischen Variablen des Systems.

Die Implementierung erfolgt in Matlab. Sie stützt sich auf eine Vorgehensweise für stationäre Probleme, das um die zeitliche Komponente, insbesondere um die Berechnung der Randwerte, erweitert wird.

Zur Verifizierung werden verschiedene Simulationsexperimente durchgeführt. Dabei werden unterschiedliche Druckkurven als Testeingänge verwendet, um die Ergebnisse mit verfügbaren Resultaten in der Literatur zu vergleichen.

Mit diesem Modell kann der Blutfluss in größeren Arterien beschrieben werden, und es soll daher als Teilmodell in ein dynamisches, regulierbares und identifizierbares Herzkreislaufmodell des menschlichen Körpers einfließen, mit dem die Auswirkungen physiologischer Veränderungen des Gefäßsystems auf das globale Verhalten des Blutkreislaufs simuliert und untersucht werden können.

Abstract

In many developed countries diseases of the cardiovascular system are the most common cause of death. Therefore research work in this area is of great importance. In particular, modelling and simulation of the cardiovascular system plays an important role.
In this work a model for blood flow in major arteries, which are a fundamental component of the human blood circuit, will be derived and simulated.

The derivation starts with the Navier-Stokes equations for incompressible fluids. After stating various assumptions like a cylindrical shape of arteries, a one-dimensional model is derived. The state variables in this blood flow model are the cross-sectional area and the averaged volume flux. The system can be written as a hyperbolic partial differential equation of first order in conservation form.

To solve the system of differential equations numerically, the finite element method is used. More precisely a second-order Taylor-Galerkin scheme is applied. The characteristic variables are used to obtain the boundary conditions.

The implementation is done in Matlab, based on a method for stationary problems, which will be extended by the time component, especially by the calculations of the boundary conditions.

For verification purposes, several simulation experiments are performed. Different pressure functions are used as test inputs and the results are compared to those available in literature.

This model can be used to describe the blood flow in the major arteries, and therefore it might be a part of a dynamical, controlled and identifiable model for the cardiovascular system of the human body, which can be used to simulate and analyse impacts of physiological changes of the vascular system to the global characteristics of the blood circuit.

Danksagung

Zum Gelingen dieser Diplomarbeit haben viele Menschen ihren Teil beigetragen. Zu allererst gilt mein Dank Prof. Felix Breitenecker. In seiner Forschungsgruppe konnte ich viele wertvolle Erfahrungen sammeln, die weit über das übliche Ausmaß einer Diplomarbeitsbetreuung hinausgehen und Lust auf eine weitere Zusammenarbeit machen.

Weiters möchte ich mich bei Prof. Dirk Praetorius für seine Unterstützung bedanken. Auch Herrn Edwin Oswald vom Institut für Strömungsmechanik und Wärmeübertragung sei für seine Hilfsbereitschaft Dank gesagt. Ein Gespräch mit ihm verhalf mir zu einem Erkenntnisgewinn, der für den weiteren Fortschritt der Arbeiten von wesentlicher Bedeutung war.

Diese Arbeit entstand in Zusammenarbeit mit der Abteilung Biomedical Engineering der Austrian Research Centers. Ich möchte mich bei Sigi Wassertheurer für das in mich gesetzte Vertrauen und für die Abwicklung, die von der Themenfindung bis zu Plänen weiterer Zusammenarbeit reicht, bedanken.
Mein besonderer Dank gilt Johannes Kropf, der mich durch diese Diplomarbeit begleitet hat. Das hier vorliegende Ergebnis wäre ohne seinen Einsatz und die unzähligen Treffen und E-mails nicht möglich gewesen.

Die Fertigstellung meiner Diplomarbeit kennzeichnet das Ende meines Studiums der Technischen Mathematik, und es ist kaum möglich, an dieser Stelle all jenen zu danken, die mir bei der Bewältigung des Studiums hilfreich zur Seite gestanden sind.

Abseits des universitären Umfelds gilt der Dank vor allem meiner Familie, auf deren Unterstützung ich nicht zuletzt in finanzieller Hinsicht immer bauen konnte.

Gliedert sich das Studium formal in drei Abschnitte, so lässt sich mein persönliches Studium ebenso in drei Abschnitte teilen. Zu Beginn steht die Periode der ersten Semester, die durch Pflichtvorlesungen und die dazugehörigen Übungen geprägt ist. Ohne die Zusammenarbeit mit vielen Kolleginnen und Kollegen in dieser Zeit hätte mein Studium sicherlich nie die Phase der Diplomarbeit erreicht. Einige von ihnen werden mir hoffentlich noch lange als gute Freunde erhalten

bleiben.

Den zweiten Abschnitt stellt mein einjähriger Aufenthalt in Dänemark an der DTU in Lyngby dar. Die dort gemachten Erfahrungen und Erlebnisse sind für mein Leben von großer Bedeutung.
Die letzte Periode meines Studiums ist vor allem durch die Mitarbeit in der Gruppe rund um Prof. Felix Breitenecker gekennzeichnet, bei der ich mich für die freundliche Aufnahme bedanken möchte.

Abschließend möchte ich noch einer Person besonders danken. Schon am ersten Vorlesungstag saß sie neben mir in den dicht gefüllten Reihen des Hörsaals. Im Laufe des Studiums wurden die Reihen lichter und die Säle kleiner, sie aber begleitete mich durch fast alle Lehrveranstaltungen meines Studiums. Neben den vielen Stunden der gemeinsamen Übungs- und Prüfungsvorbereitung ist vor allem die gemeinsame Arbeit an Projekten im Bereich der Modellbildung und Simulation in bester Erinnerung, die in einer gegenseitigen Hilfe bei der Bewältigung unserer Diplomarbeiten ihren Höhepunkt gefunden hat. Neben ihrer fachlichen Kompetenz sind es vor allem ihre menschlichen Qualitäten, die sie zu einer hervorragenden Arbeits- und Gesprächspartnerin machen. Ich möchte mich bei Xenia Descovich sehr herzlich für die vielen gemeinsamen Stunden und die erfolgreiche Zusammenarbeit bedanken.

Inhaltsverzeichnis

1 Einleitung **1**

1.1 Motivation 1

1.2 Übersicht 3

1.3 Modellbildung und Simulation 4

1.4 Physiologische Grundlagen 6

2 Finite Elemente Methode **12**

2.1 Bilinearformen und das Lex-Milgram Lemma 12

2.2 Schwache Formulierung eines Modellproblems 13

2.3 Galerkin-Verfahren 15

2.4 Fehlerabschätzung – Lemma von Céa 18

2.5 Basisfunktionen 19

3 FEM mit Matlab **21**

3.1 Das Modellproblem 21

3.2 Zerlegung in finite Elemente 22

3.3 Aufstellen der Matrizen 23

3.4 Aufstellen des Lastvektors 26

3.5 Randbedingungen 27

3.6 Lösen des Gleichungssystems 29

4 Blutflussmodell **32**

4.1 Hyperbolische partielle Differentialgleichungen erster Ordnung . . 32

4.2 Modellherleitung 35

4.3 Diskretisierung 41

4.3.1 Taylorentwicklung 41

4.3.2 Galerkin-Verfahren 42

4.3.3 Berechnung der Matrizen 44

4.3.4 Berechnung des Lastvektors 45

5 Implementierung **56**

5.1 Aufbau des Programms 56

5.2 Aufstellen der Matrizen 57

5.3 Aufstellen der rechten Seite 59

5.4 Randbedingungen 63

5.5 Lösen des Gleichungssystems 67

5.6 Bifurkationen 68

6 Simulationsergebnisse **73**

7 Ausblick **82**

Abbildungsverzeichnis **84**

Literaturverzeichnis **86**

Index **88**

Kapitel 1

Einleitung

1.1 Motivation

Erkrankungen des Herzkreislaufsystems stellen in vielen hochentwickelten Ländern die häufigste Todesursache dar. Den Angaben im Österreichischen Todesursachenatlas 1998/2004 der Statistik Austria [16] zufolge haben 49,9 % aller Todesfälle in Österreich eine Erkrankung des Herzkreislaufsystems als Ursache. In den meisten osteuropäischen Ländern liegt der Anteil dieser Todesursachengruppe noch höher.
Die zahlenmäßig bedeutsamsten Todesursachen innerhalb dieser Gruppe liegen im Bereich der Herzkrankheiten, Hirngefäßkrankheiten und den Auswirkungen von Bluthochdruck.

Daher ist die Forschungsarbeit auf diesem Gebiet von großer Bedeutung. Die Modellbildung und Simulation des Herzkreislaufsystems nimmt dabei eine wichtige Rolle ein. Mit Hilfe von Simulationen können die Auswirkungen von Operationen bereits vor dem eigentlichen Eingriff untersucht werden. Eine Simulation bietet weiters die Möglichkeit, Auswirkungen von Medikamenten zu untersuchen, indirekte Messverfahren zu entwickeln oder Messdaten virtuell für weitere Berechnungen zu generieren.

Die Modellbildung und Simulation des menschlichen Herzkreislaufsystems befindet sich in einem Spannungsfeld. Einerseits existieren relativ leicht handhabbare Modelle des ganzen Blutkreislaufs, die allerdings nicht alle in der Wirklichkeit auftretenden Phänomene abbilden. Andererseits gibt es sehr komplexe Modelle, die jedoch nur einen Teilbereich des Kreislaufsystems betrachten.

Ein geregeltes und identifizierbares dynamisches Modell des menschlichen Kreislaufsystems wurde von Johannes Kropf in [8] entwickelt. Dazu werden verschiedene Teilmodelle geeignet gekoppelt. Eine schematische Übersicht bietet die Ab-

bildung 1.1.

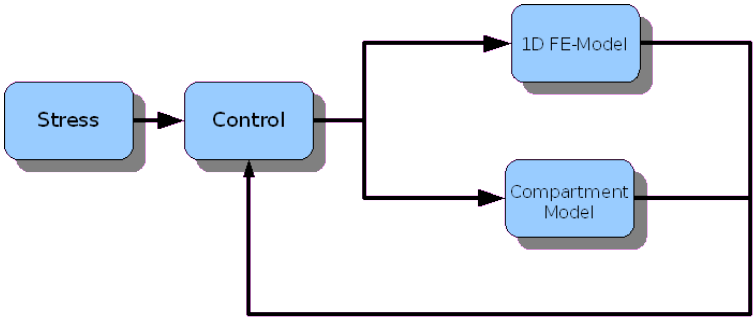

Abbildung 1.1: Schema einer Herzkreislaufsimulation

Einen Teil des Schemas in Abbildung 1.1 nimmt ein eindimensionales Modell für den Blutfluss in Arterien ein, das mit Hilfe der Finiten Elemente Methode gelöst wird. Das Hauptziel der hier vorliegenden Arbeit ist es, ein solches Modell zu entwickeln und vor allem die numerische Lösung mittels Finiter Elemente Methode zu implementieren. Mit Hilfe dieses Modells kann dann die Simulation des Blutflusses in den größeren Arterien des menschlichen Körpers als Teilbereich des vollständigen Herzkreislaufmodells realisiert werden.

Diese Arbeit beschäftigt sich wie zuvor erwähnt mit der Modellbildung und Simulation des Blutflusses in Arterien. Als Grundlage dient dafür ein Blutflussmodell für Arterien in einer Raumdimension. Das Hauptaugenmerk liegt auf der Lösung mittels der Finiten Elemente Methode. Dies beginnt mit einer geeigneten Diskretisierung des Zustandsraums bezüglich Zeit und Ort. Ein weiterer wichtiger Punkt stellt die Berechnung von entsprechenden Randbedingungen dar, darunter fällt auch die Kopplung mehrerer Arterienstücke zu einer Bifurkation. Sämtliche Berechnungen werden in der Entwicklungsumgebung Matlab implementiert, und in weiterer Folge werden damit verschiedene Simulationsexperimente durchgeführt.

Mit Hilfe dieses Blutflussmodells können die Auswirkungen von Arterienveränderungen untersucht werden. Darunter fallen vom Körper selbst ausgebildete Veränderungen wie die Versteifung der Arterienwände oder die Verengung von Gefäßen. Aber auch chirurgische Maßnahmen, wie die Einsetzung eines Stents, können simuliert werden. Neben diesen eher lokalen Betrachtungen dient das Modell aber vor allem der bereits erwähnten Simulation der großen Arterien im Rahmen eines vollständigen Herzkreislaufmodells.

1.2 Übersicht

Diese Arbeit ist in sieben Kapitel unterteilt. Den Kern der Arbeit bilden dabei die Abschnitte drei bis sechs. Ist der Leser mit den physiologischen und mathematischen Grundlagen der Arbeit vertraut, so kann er die ersten Teile überspringen. Fehlt hingegen das Wissen in bestimmten Bereichen, so kann es in den ersten Kapiteln nachgelesen werden, für noch tiefergehende Erläuterungen sei auf die verwendete Literatur verwiesen.

Im ersten Kapitel findet sich neben einer kurzen Motivation ein kleiner Überblick über das Gebiet der Modellbildung und Simulation, gefolgt von einer Einführung in die physiologischen Grundlagen des menschlichen Herzkreislaufsystems, wobei besonders auf die für diese Arbeit relevanten Phänomene Rücksicht genommen wird.

Das zweite Kapitel soll die Ideen hinter dem Finiten Elemente Verfahren erklären. Dazu wird ein einfaches Modellproblem eingeführt. Dieses Modellproblem wird auch im dritten Kapitel wieder verwendet, es dient dort zur Veranschaulichung einer Implementierungsmethode der Finiten Elemente Methode für stationäre Probleme in Matlab. Treten bei der Lektüre der folgenden Kapiteln Unklarheiten zur verwendeten Lösungsmethode oder deren Implementierung auf, so sollte in diesen beiden Abschnitten die passende Antwort zu finden sein.

Im vierten Kapitel wird das Blutflussmodell eingeführt. Zuerst werden die Systemgleichungen hergeleitet. Der eigentliche Schwerpunkt liegt aber in der Diskretisierung, die beim Galerkin Verfahren auftretenden Skalarprodukte werden ausführlich beschrieben.

Das fünfte Kapitel beschreibt die Implementierung des Blutflussmodells in Matlab. Dazu gehören auch die Berechnungen von passenden Randbedingungen über die charakteristischen Variablen oder die Modellierung einer Bifurkation im Gefäßsystem.

Im sechsten Kapitel werden die Ergebnisse mehrerer Simulationsexperimente erläutert. Zur Veranschaulichung dienen Grafiken mit den Verläufen verschiedener Systemgrößen.

Am Schluss folgt noch ein kurzer Ausblick, wie das implementierte Blutflussmodell in Zukunft verwendet werden kann.

1.3 Modellbildung und Simulation

Das Gebiet der Modellbildung und Simulation ist breit gefächert, und eine genaue Definition der Begriffe ist schwierig. Hier soll nur ein kurzer Überblick über diesen Themenbereich unter Zuhilfenahme der Unterlagen zur Vorlesung „Modellbildung und Simulation“ an der Technischen Universität Wien [15] gegeben werden.

Eine erste Charakterisierung des Begriffes Simulation bietet die VDI-Richtline 3633: Simulation ist die Nachbildung eines dynamischen Prozesses in einem Modell, um zu Erkenntnissen zu gelangen, die auf die Wirklichkeit übertragbar sind.

Anders ausgedrückt ist eine Simulation ein Experimentieren mit einem (Computer-) Modell. Die Qualität der Ergebnisse hängt stark von der Qualität des Modells und dessen Gültigkeitsgrenzen ab. Zwischen der Modellbildung und der Simulation besteht ein enger Zusammenhang. Simulationen haben als Basis ein Modell, sie können aber auch Hilfsmittel für die Modellbildung sein.

Eine modellbasierte Simulation ist ein Instrument zur Problemlösung. Eine Simulation kann eine Entscheidungshilfe darstellen, sie ist aber kein Ersatz für Entscheidungen des Menschen.

Eine schematische Übersicht über die Vorgangsweise bei der Durchführung einer Simulation bietet der Simulationskreislauf in Abbildung 1.2 (nach [15], S. 2).

[15] bietet eine Vielzahl von möglichen Definitionen für den Begriff „Modell“, hier eine Auswahl:

- Ein Modell ist eine vereinfachte Darstellung der Funktion eines Gegenstands oder des Ablaufs eines Sachverhalts, die eine Untersuchung oder Erforschung erleichtert oder erst möglich macht.
- Modelle sind Darstellungen von Objekten, Prozessen, von irgendetwas, das wir zu beschreiben oder dessen Verhalten wir zu analysieren wünschen.
- Ein mathematisches Modell ist ein abstraktes, vereinfachtes, mathematisches Gebilde, das zu einem Teil der Realität in Bezug steht und für einen bestimmten Zweck erzeugt wurde.
- Modellierung bezeichnet den Prozess des Organisierens von Wissen über ein gegebenes System.

Wesentlich für das Bilden eines Modells sind die Abgrenzung des Systems gegen die Umwelt sowie die Auswahl genau jener Merkmale aus den vielen vorhandenen, die notwendig und ausreichend sind, um den Vorgang so genau zu beschreiben, dass die Ziele erreicht werden. Die Welt lässt sich daher aus Sicht der

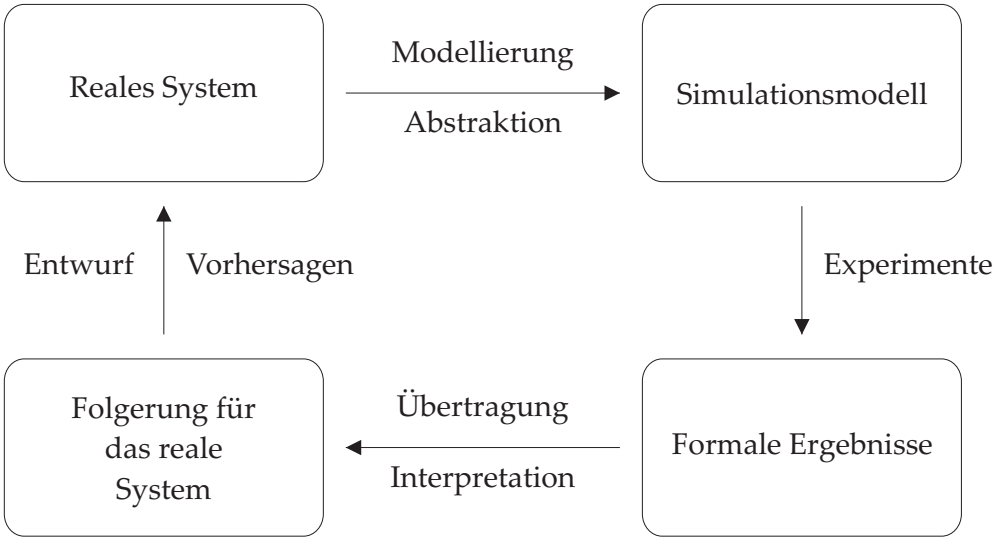

Abbildung 1.2: Simulationskreislauf (nach [15], S. 2)

Modellbildung in drei Gruppen einteilen: In Dinge, die gänzlich vernachlässigt werden, in Dinge die das Modell beeinflussen, aber nicht mittels des Modells untersucht werden sollen sowie in Dinge, derentwegen das Modell erstellt wird.

Für die Modellbildung sind neben Messdaten die Einsicht in die Prozesse von grundlegender Bedeutung. Das Ausmaß der Einsicht in einen Prozess bestimmt die Vorgehensweise bei der Modellbildung wesentlich, man kann daher zwischen zwei Hauptmethoden unterscheiden:

Deduktion: Die Deduktion beruht auf der Kenntnis von Grundgesetzen über den Prozess und führt zu Strukturmodellen.

Induktion: Die Induktion führt zu Eingangs/Ausgangsmodellen (Verhaltensmodelle, Black-Box Modelle).

In der Praxis treten häufig Mischformen dieser beiden Methoden auf. Versucht man verschiedene Anwendungsgebiete von transparent (vollständig bekannte Struktur) bis schwarz einzuteilen, so wäre eine mögliche Reihenfolge: Elektrotechnik - Mechanik - Chemie - Biologie - Ökonomie - Soziologie - Psychologie. Dabei ist vor allem im dunkleren Bereich die Abfolge nicht eindeutig.

Abschließend sei noch darauf hingewiesen, dass ein Modell bzw. ein Simulationsergebnis nicht mit der durch das Modell abgebildeten Realität verwechselt

werden darf. Dieses Phänomen kann in verschiedenen Situationen immer wieder beobachtet werden und die Gefahr ist nicht zu unterschätzen.

1.4 Physiologische Grundlagen

Dieser Abschnitt bietet eine kurze Einführung in die Physiologie des menschlichen Blutkreislaufs. Die Informationen wurden den Büchern „Anatomie, Physiologie, Pathophysiologie des Menschen“ von Gerhard Thews et al. [14], „Physiologie des Menschen mit Pathophysiologie“ von Robert Schmitt et al. [11] sowie „Color Atlas of Physiology“ von Stefan Silbernagl et al. [3] entnommen.

Der Blutkreislauf ist das wichtigste Transportsystem des menschlichen Körpers. Es dient u.a. zum Transport von Atemgasen, Nährstoffen, Wasser, Elektrolyten, Säuren, Basen, Wärme, hormonalen Informationsträgern sowie Teilen des zellulären und humoralen Abwehrsystems. Die Wichtigkeit der Kreislauffunktion wird ersichtlich, wenn es zu einem Kreislaufstillstand kommt. Dabei treten bereits nach wenigen Sekunden die ersten Funktionsstörungen auf, und nach ca. 3 bis 5 Minuten kommt es zu ersten irreparablen Schäden im Gehirn. Mit Hilfe des Blutkreislaufs kann Sauerstoff innerhalb von 20 Sekunden von der Lunge bis zu den entferntesten Körperregionen transportiert werden.

Der menschliche Blutkreislauf besteht aus einem System von parallel und seriell geschalteten Blutgefäßen. Um eine gerichtete Blutströmung aufrecht zu erhalten, sind zwei Pumpen in das System eingefügt, nämlich das rechte und das linke Ventrikel (rechtes und linkes Herz). Vom linken Ventrikel strömt das Blut in den so genannten Körperkreislauf. Das Blut kommt zuerst in die Aorta und strömt weiter in die großen Arterien (eine Übersicht über die Arterienstämme des Körpers bietet Abbildung 1.3, aus [14], S. 214). Diese teilen sich immer weiter auf, bis das Blut zu den kleinsten arteriellen Gefäßen, den so genannten Arteriolen, vordringt. Es folgen die Kapillaren, die ein dichtes Gefäßnetz bilden, dort finden die Austauschvorgänge zwischen dem Blut und den umgebenden Zellen statt. Danach gelangt das Blut in die Venolen, die sich in weiterer Folge zu Venen vereinigen. Aus den Venen kommt das Blut schließlich in das rechte Ventrikel, das den Ausgangspunkt für den Lungenkreislauf bildet.

Eine schematische Darstellung des Blutkreislaufes ist in Abbildung 1.4 (aus [11], S. 604) zu sehen. Dabei geben die Prozentzahlen die Anteile des Herzzeitvolumens an, die unter Ruhebedingungen durch die verschiedenen Organgebiete fließen. Die gezackten Linien symbolisieren die Größen der regionalen Strömungswiderstände.

Wandaufbau: Die Wände der großen herznahen Arterien sind durch elastische Netze in der mittleren Wandschicht gekennzeichnet. Diese Arterien besitzen daher eine stärkere Dehnbarkeit und werden deshalb als Arterien vom

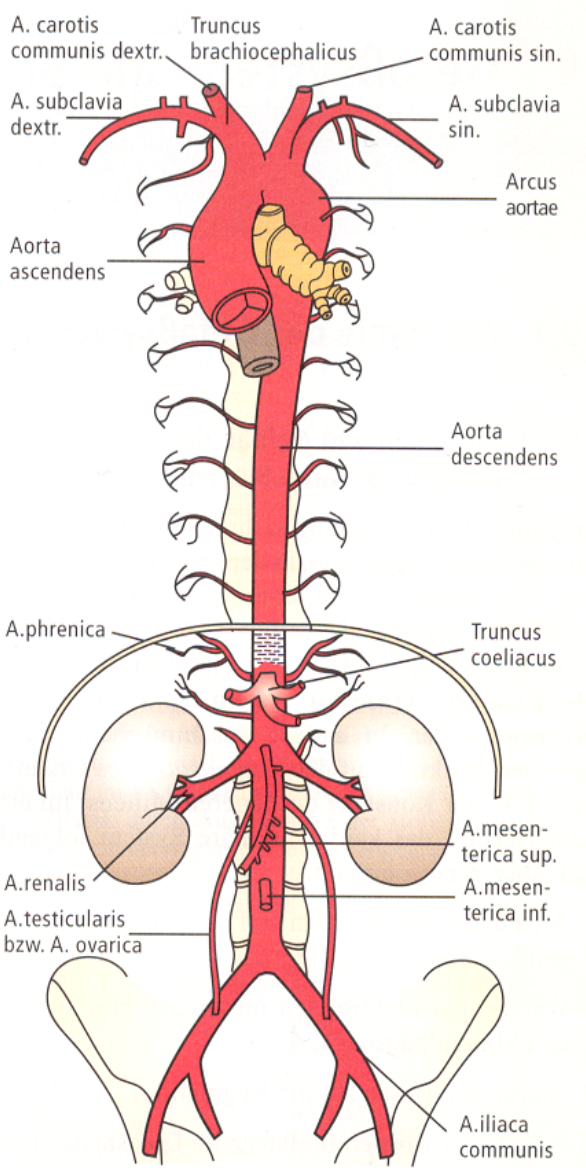

Abbildung 1.3: Arterienstämme des Körperkreislaufs (aus [14], S. 214)

elastischen Typ bezeichnet. Mit zunehmender Entfernung vom Herzen nehmen die elastischen Bestandteile ab und die glatten Muskelfasern zu. Solche Gefäße mit geringer Wanddehnbarkeit bezeichnet man als Arterien vom muskulären Typ.

Viskosität des Blutes: Die Viskosität des Blutes beeinflusst den Strömungswiderstand in den Gefäßen. Aufgrund der inhomogenen Zusammensetzung des Blutes aus Zellen und Plasma ist die Viskosität keine konstante Größe, sondern wird durch verschiedene Parameter (Viskosität des Plasmas, Hämatokritwert, Schubspannung, Gefäßdurchmesser) bestimmt:
Die Viskosität des Blutplasmas ist normalerweise ca. doppelt so groß wie

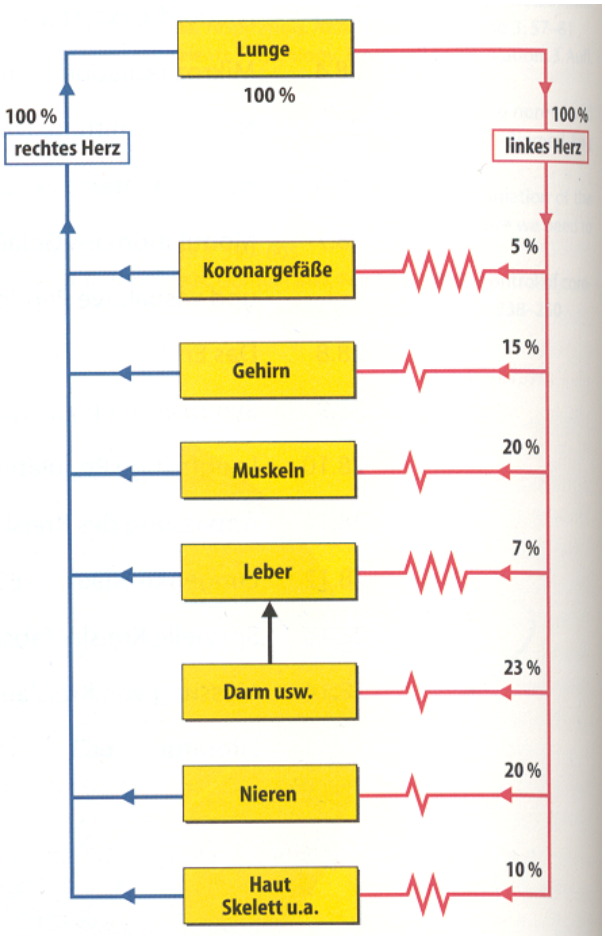

Abbildung 1.4: Blutkreislaufschema (aus [11], S. 604)

jene von Wasser. Oft wird die Blutviskosität als Vielfaches der Plasmaviskosität angegeben (relative Viskosität).
Unter dem Hämatokritwert versteht man den Anteil der Erythrozyten im Blut. Bei wachsendem Hämatokritwert steigt auch die Viskosität.
Die Schubspannung bezeichnet die tangentiale Kraft, die auf Flüssigkeitsschichten ausgeübt wird, wenn sie dem Geschwindigkeitsprofil entsprechend gegeneinander verschoben werden. Da die Erythrozyten bei schneller Strömung eine widerstandsarme Form annehmen, nimmt die Viskosität bei starker Strömung mit hoher Schubspannung ab.

Das Blut in der Aorta wird durch den Ausfluss des Schlagvolumens aus dem Herzen beschleunigt. Aufgrund der Massenträgheit des Blutes kommt es zu einem Druckanstieg am Beginn der Aorta. Dieser bleibt lokal begrenzt und fällt geringer als in einem starren Rohr aus, weil nicht die gesamte im Gefäßsystem enthaltene Blutsäule beschleunigt werden muss. Durch den Druckanstieg kommt es zu einer Dehnung der Arterienwand und folgedessen zu einer Querschnittserweiterung. Dadurch wird ein Teil des Blutvolumens gespeichert. Zwischen dem Anfangsbe-

reich der Aorta und den folgenden Teilen bildet sich ein Druckgradient aus, der eine Beschleunigung und Weiterbewegung des Blutvolumens bewirkt. Daraus ergeben sich Pulswellen, die sich mit einer gewissen Geschwindigkeit fortpflanzen. Dabei muss zwischen der Ausbreitungsgeschwindigkeit der Pulswelle und der wesentlich geringeren Strömungsgeschwindigkeit des Blutes unterschieden werden.

Passiert die Pulswelle einen Ort im System, so lassen sich drei Grundphänomene der Pulswelle beobachten, nämlich der Druckpuls, der Strompuls und der Querschnittpuls. Diese stellen eine Änderung des Wellendrucks, der Wellenströmung und des Gefäßquerschnitts dar. Laufen die Wellen in einem System nur in eine Richtung, so weisen die drei Pulsformen übereinstimmende Verläufe auf. Im Arteriensystem ist das allerdings nicht der Fall, weil es zu Wellenreflexionen kommt.

Im Arteriensystem kommt es immer wieder zu Verzweigungen oder zu Änderungen des Gefäßquerschnitts, der Wanddicke oder der Elastizität. Dadurch steigt der Wellenwiderstand teilweise gleichmäßig und teilweise sprunghaft an, was zu Wellenreflexionen führt. Daher kommt es zu Überlagerungen der peripherwärts laufenden und der herzwärts laufenden Wellen. Dabei addieren sich die Wellendrücke, während sich die Wellenstromstärken subtrahieren. Deshalb sind die Druck- und Strompulse von unterschiedlicher Gestalt.

Wie bereits erwähnt, unterscheiden sich die Ausbreitungsgeschwindigkeiten der Strompulse und Druckpulse stark. Die mittlere Strömungsgeschwindigkeit beträgt 15-20 cm/s. Am Ende der Systole hat das soeben aus dem Herzen ausgeworfene Blut daher ca. 20 cm zurückgelegt, dagegen hat die Pulswelle zu diesem Zeitpunkt bereits das Arteriensystem vollständig durchlaufen und es kehren reflektierte Wellen zum Herzen zurück. In der elastischen Aorta von Jugendlichen liegt die Pulswellengeschwindigkeit bei 4-6 m/s. In den weniger dehnbaren Arterien vom muskulären Typ steigt sie auf 7-12 m/s.

Die Abbildung 1.5 (aus [11], S. 611) bietet einige Beispiele von Druck- und Strompulsformen an verschiedenen Stellen des arteriellen Kreislaufabschnitts. Die Abbildung 1.6 (aus [11], S. 612) zeigt den Blutdruck, den Gesamtquerschnitt der Gefäße und die mittlere Strömungsgeschwindigkeit in den einzelnen Abschnitten des Kreislaufsystems.

Mit zunehmendem mittleren Blutdruck steigt die Pulswellengeschwindigkeit, weil der Elastizitätsmodul mit wachsender Dehnung der Arterien größer wird. Auch im höheren Lebensalter nimmt die Pulswellengeschwindigkeit zu, vor allem in den elastischen Arterien. Es kommt nämlich zu einem altersbedingten Umbau der Gefäßwand, bei der das elastische Gewebe ab- und das kollagene Gewebe zunimmt.

Abschließend soll Abbildung 1.7 (aus [3], S. 189) nochmals einen Überblick über verschiedene Phänomene im menschlichen Blutkreislauf bieten. Zu oberst ist das

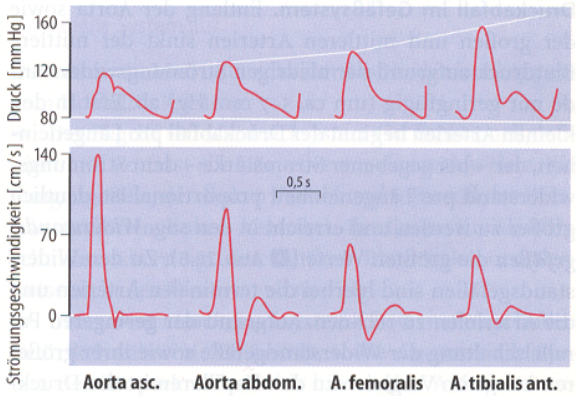

Abbildung 1.5: Druck- und Strompulsformen in verschiedenen Arterien (aus [11], S. 611)

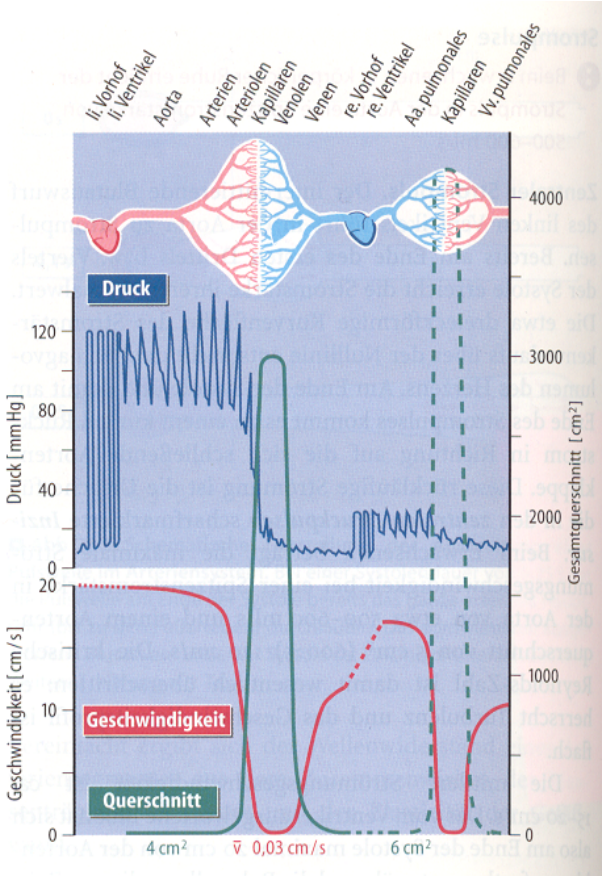

Abbildung 1.6: Blutdruck, Gesamtquerschnitt und mittlere Strömungsgeschwindigkeit im kardiovaskulären System (aus [11], S. 612)

System skizziert, gefolgt von der ungefähren Anzahl der Gefäße im jeweiligen Systemabschnitt. Das mit 1 markierte Segment listet den prozentuellen Anteil

der Gefäßabschnitte am totalen peripheren Widerstand auf. Danach kommt der mittlere Blutdruck und die mittlere Strömungsgeschwindigkeit. In Abschnitt 4 wird der Gefäßdurchmesser angegeben, danach folgt die Gesamtquerschnittsfläche der parallelen Gefäße. Zuletzt wird die Verteilung der Blutmenge im Kreislaufsystem dargestellt.

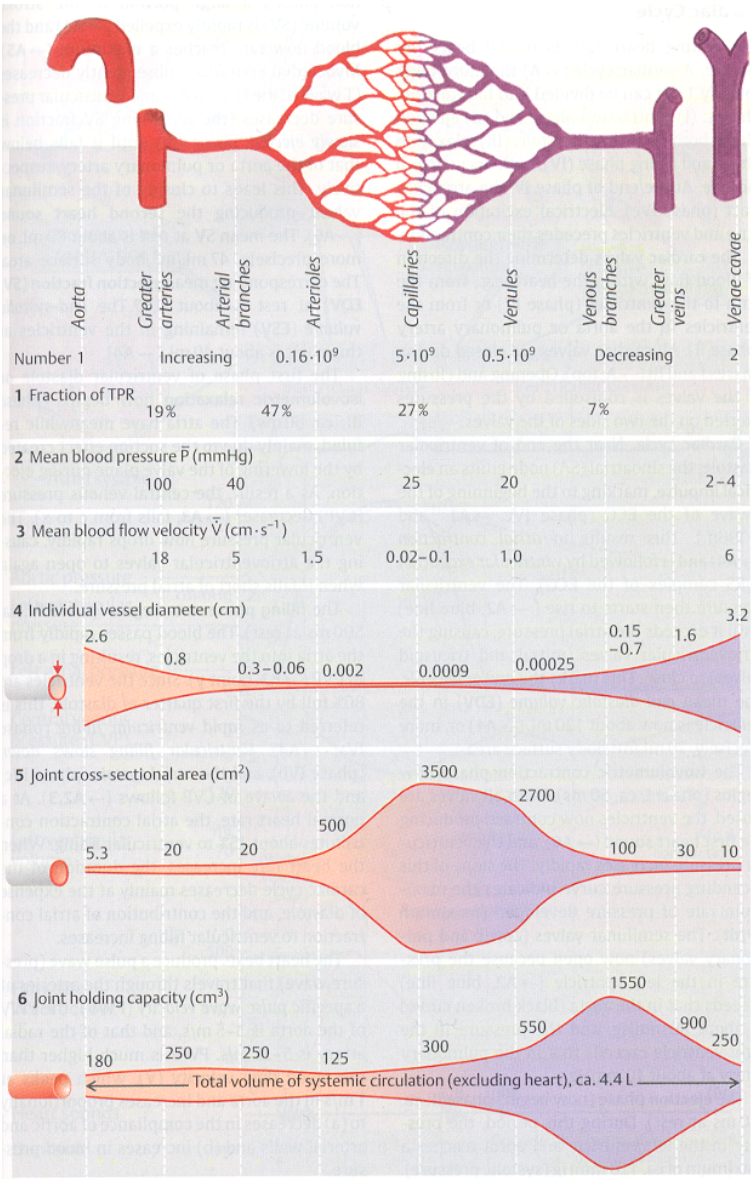

Abbildung 1.7: Eigenschaften der Gefäßabschnitte (aus [3], S. 189)

Kapitel 2

Finite Elemente Methode

Dieses Kapitel soll eine Einführung in die Finite Elemente Methode (FEM) bieten, welche die grundlegende Lösungsmethode dieser Arbeit darstellt. Das Ziel ist darzulegen, wie und warum dieses Verfahren funktioniert, dabei wird auf einige wichtige mathematische Definitionen und Sätze zurückgegriffen. Um die Übersichtlichkeit zu bewahren, wird auf Definitionen grundlegender mathematischer Begriffe sowie auf Beweise größtenteils verzichtet. Diese können in den Quellen zu diesem Kapitel, nämlich in dem für diese Arbeit sehr hilfreichen Buch von Pavel Solin [13] sowie in Notizen zu einer Vorlesung von Jens Markus Melenk über die Finite Elemte Methode [9], als auch in vielen anderen Lehrbüchern über die Finite Elemente Methode nachgelesen werden.

2.1 Bilinearformen und das Lex-Milgram Lemma

Definition (Linearform)
Sei V ein Vektorraum. Eine Abbildung $l : V \to \mathbb{R}$ heißt Linearform $\iff$

1. $l(v + w) = l(v) + l(w) \quad \forall v, w \in V$
2. $l(\lambda v) = \lambda l(v) \quad \forall v \in V,\ \lambda \in \mathbb{R}$

Definition (Bilinearform)
Sei V ein Vektorraum. Eine Abbildung $B : V \times V \to \mathbb{R}$ heißt Bilinearform $\iff$
$\forall\, w \in V$: *die Abbildungen*

$$v \mapsto B(w, v) \qquad und \qquad v \mapsto B(v, w)$$

sind Linearformen im Sinne der obigen Definition.

Definition
Sei $(V, \|\cdot\|)$ *ein normierter Raum.*
Eine Bilinearform $B : V \times V \to \mathbb{R}$ *heißt* stetig $\iff$

$$\exists\, M > 0 : \quad |B(u,v)| \leq M \, \|u\| \, \|v\| \quad \forall\, u, v \in V$$

Die Bilinearform heißt symmetrisch $\iff$

$$B(u,v) = B(v,u) \quad \forall\, u, v \in V$$

Die Bilinearform heißt koerziv *oder* V-elliptisch $\iff$

$$\exists\, \alpha > 0 : \quad \alpha \, \|u\|^2 \leq B(u,u) \quad \forall\, u \in V$$

Jeder Bilinearform B kann in eindeutiger Weise ein linearer Operator $B_{OP} : V \to V'$ zugeordnet werden. Dieser ist definiert durch:

$$(B_{OP} u)(v) = B(u,v) \quad \forall\, u, v \in V$$

Diese Relation definiert einen bijektiven Zusammenhang zwischen stetigen Bilinearformen $B : V \times V \to \mathbb{R}$ und linearen stetigen Operatoren $B_{OP} : V \to V'$.

Definition (Energienorm)
Sei V *ein Hilbertraum und* $B : V \times V \to \mathbb{R}$ *eine stetige symmetrische koerzive Bilinearform. Dann definiert die Bilinearform* B *ein Skalarprodukt:*

$$(u,v)_e = B(u,v)$$

Die durch das Skalaprodukt induzierte Norm,

$$\|u\|_e = \sqrt{(u,u)_e},$$

heißt Energienorm.

Satz (Lax-Milgram Lemma)
Sei V *ein Hilbertraum und* $B : V \times V \to \mathbb{R}$ *eine stetige koerzive Bilinearform. Weiters sei* $l \in V'$*. Dann existiert eine eindeutige Lösung* $u \in V$ *für das Variationsproblem*

$$B(u,v) = l(v) \quad \forall v \in V \tag{2.1}$$

2.2 Schwache Formulierung eines Modellproblems

Im folgenden Abschnitt wird die so genannte schwache Formulierung für ein Modellproblem hergeleitet. Wegen der Komplexität des eigentlichen Blutflussmodells wird hier bewusst ein einfacheres Modellproblem betrachtet. Bei der Auswahl des Modellproblems wird ein bestmöglicher Kompromiss versucht, um einerseits ein möglichst einfaches Beispiel zu erhalten und andererseits doch eine

gewisse Ähnlichkeit mit dem Blutflussmodell aufrecht zu erhalten.
Da das Blutflussmodell eindimensional bezüglich des Ortes ist, erscheint es ausreichend, auch hier nur ein eindimensionales Modell zu betrachten. Ebenfalls in Anlehnung an das Blutflussmodell wird die unabhängige Variable mit z bezeichnet, weil dort die einzige Ortsdimension ebenfalls mit z benannt wird.

Das Modellproblem ist eine partielle Differentialgleichung zweiter Ordnung, die mit Dirichlet-Randbedingungen versehen ist. Es lautet folgendermaßen:

$$\begin{aligned} -\left(a(z)u'(z)\right)' + c(z)u(z) &= f(z) \quad \text{für } z \in \Omega = (0,1) \\ u(0) = u(1) &= 0 \end{aligned} \tag{2.2}$$

Die klassische Lösung dieses Problems ist eine Funktion $u \in C^2(\Omega) \cap C(\overline{\Omega})$, welche die Gleichung 2.2 in ganz Ω sowie die Randbedingungen erfüllt.

Als Nächstes wird die so genannte schwache Formulierung des Problems 2.2 eingeführt. Deren Herleitung kann in folgende Schritte unterteilt werden:

1. Multiplikation mit einer Testfunktion $v \in C_0^\infty(\Omega)$

$$-\left(a(z)u'(z)\right)' v(z) + c(z)u(z)v(z) = f(z)v(z)$$

2. Integration über Ω

$$-\int_\Omega \left(a(z)u'(z)\right)' v(z)\, dz + \int_\Omega c(z)u(z)v(z)\, dz = \int_\Omega f(z)v(z)\, dz$$

3. Partielle Integration des ersten Teils der Gleichung

$$\begin{aligned} -\int_\Omega \left(a(z)u'(z)\right)' v(z)\, dz &= -a(z)u'(z)v(z)|_0^1 + \int_\Omega a(z)u'(z)v'(z)\, dz = \\ &= \int_\Omega a(z)u'(z)v'(z)\, dz \quad \text{weil } v(0) = v(1) = 0 \end{aligned}$$

Somit ergibt sich

$$\int_\Omega a(z)u'(z)v'(z)\, dz + \int_\Omega c(z)u(z)v(z)\, dz = \int_\Omega f(z)v(z)\, dz \tag{2.3}$$

Nun können die Regularitätsanforderungen abgeschwächt werden; denn die obigen Integrale bleiben auch dann endlich, wenn die Funktionen folgendermaßen gewählt werden:

$$u, v \in H_0^1(\Omega), \quad f \in L^2(\Omega) \tag{2.4}$$

Dabei bezeichnet $H_0^1(\Omega)$ einen so genannten Sobolev Raum, der wie folgt definiert ist:

$$\begin{aligned} H^1(\Omega) :=& \{u \in L^2(\Omega) \mid u' \in L^2(\Omega)\} \\ H_0^1(\Omega) :=& \{u \in H^1(\Omega) \mid u(0) = u(1) = 0\} \end{aligned} \tag{2.5}$$

Es ist leicht einzusehen, dass die klassische Lösung auch die schwache Formulierung löst. Umgekehrt gilt: Ist eine schwache Lösung von 2.3 zusammen mit 2.4 hinreichend glatt, d.h. $u \in C^2(\Omega) \cap C(\overline{\Omega})$, dann erfüllt sie auch die klassische Formulierung des Modellproblems. Für eine genauere Ausführung dieses Sachverhalts sei auf das Kapitel 8.1.2 in [4] verwiesen.

Jetzt kann eine Verbindung der schwachen Formulierung zu den Linear- bzw. Bilinearformen des vorigen Abschnitts hergestellt werden.
Sei $V = H_0^1(\Omega)$. Dann kann eine Bilinearform $B : V \times V \to \mathbb{R}$ definiert werden als:

$$B(u, v) := \int_\Omega a(z)u'(z)v'(z)\, dz + \int_\Omega c(z)u(z)v(z)\, dz \tag{2.6}$$

Weiters lässt sich auch eine Linearform $l \in V'$ definieren:

$$l(v) := \int_\Omega f(z)v(z)\, dz \tag{2.7}$$

Die schwache Formulierung des Modellproblems lässt sich somit auch mit Hilfe einer Linear- sowie Bilinearform realisieren. Gesucht wird also eine Funktion $u \in V$, welche die Gleichung

$$B(u, v) = l(v) \tag{2.8}$$

für alle $v \in V$ erfüllt.

Eine Aussage über die Existenz und Eindeutigkeit einer Lösung bietet unter entsprechenden Voraussetzungen das Lax-Milgram Lemma 2.1.

2.3 Galerkin-Verfahren

Die exakte Lösung der schwachen Formulierung des Modellproblems 2.8 wird in einem unendlichdimensionalen Vektorraum V gesucht. Im Allgemeinen ist es daher nicht möglich, diese exakte Lösung auch zu finden. Daher kommt die Idee, das Problem im Endlichdimensionalen zu approximieren. Ihren Namen hat dieses Verfahren von Boris Grigorievich Galerkin bekommen, der sich als erster systematisch damit befasst hat.

Das Galerkin Approximationsproblem wird oft auch als diskretes Problem bezeichnet und kann folgendermaßen formuliert werden:

Sei V_h ein endlichdimensionaler Unterraum von V, also $V_h \subset V,\ dim(V_h) = N < \infty$. Gesucht wird nun eine Lösung $u_h \in V_h$, welche die Gleichung

$$B(u_h, v) = l(v) \tag{2.9}$$

für alle $v \in V_h$ erfüllt.

Sind für das schwache Problem die Bedingungen des Lax-Milgram Lemmas 2.1 erfüllt, d.h. ist die Bilinearform $B(\cdot,\cdot)$ stetig und koerziv, dann gilt dies auch für das diskrete Problem. Daraus ergibt sich folgende Aussage:

Satz (Eindeutige Lösbarkeit)
Sei $V_h \subset V,\ dim(V_h) = N < \infty$. Dann existiert eine eindeutige Funktion $u_h \in V_h$, welche die Gleichung

$$B(u_h, v) = l(v)$$

für alle $v \in V_h$ löst.

Da der Raum V_h eine endliche Basis $\{v_i\}_{i=1}^N$ besitzt, kann die Lösung $u_h \in V_h$ des diskreten Problems 2.9 explizit gefunden werden. Sie kann als Linearkombination der Basisfunktionen dargestellt werden:

$$u_h = \sum_{i=1}^{N} u_i v_i \tag{2.10}$$

Setzt man nun diesen Lösungsansatz 2.10 in das diskrete Problem 2.9 ein, so erhält man

$$B\left(\sum_{i=1}^{N} u_i v_i, v\right) = l(v) \quad \forall\, v \in V_h. \tag{2.11}$$

Mit Hilfe der Linearität von $B(\cdot,\cdot)$ kommt man weiters zu

$$\sum_{i=1}^{N} u_i\, B\left(v_i, v\right) = l(v) \quad \forall\, v \in V_h. \tag{2.12}$$

Ersetzt man v durch die Basisfunktionen $v_1, v_2, \ldots, v_N$, so erhält man

$$\sum_{i=1}^{N} u_i\, B\left(v_i, v_j\right) = l(v_j) \quad j = 1, 2, \ldots, N. \tag{2.13}$$

Obwohl in 2.13 nur mehr die Basisfunktionen betrachtet werden, sind die beiden letzten Formulierungen 2.12 und 2.13 des diskreten Problems äquivalent.

Um das lineare Gleichungssystem 2.13 auch in Matrixschreibweise anführen zu können, werden folgende Definitionen verwendet:

Definition (Steifigkeitsmatrix)
Die Matrix $\boldsymbol{S}$

$$\boldsymbol{S} := \begin{pmatrix} B(v_1, v_1) & B(v_1, v_2) & \cdots & B(v_1, v_N) \\ B(v_2, v_1) & B(v_2, v_2) & \cdots & B(v_2, v_N) \\ \vdots & \vdots & \ddots & \vdots \\ B(v_N, v_1) & B(v_N, v_2) & \cdots & B(v_N, v_N) \end{pmatrix}$$

heißt Steifigkeitsmatrix.

Definition (Lastvektor)
Der Vektor $\boldsymbol{f}$

$$\boldsymbol{f} := \begin{pmatrix} l(v_1) \\ l(v_2) \\ \vdots \\ l(v_N) \end{pmatrix}$$

heißt Lastvektor.

Ebenso kann man einen Vektor $\boldsymbol{u}$ für die unbekannten Koeffizienten des Lösungsansatzes definieren:

$$\boldsymbol{u} := \begin{pmatrix} u_1 \\ u_2 \\ \vdots \\ u_N \end{pmatrix}$$

Das System von linearen algebraischen Gleichungen 2.13 kann somit als lineares Gleichungssystem in Matrixschreibweise dargestellt werden:

$$\boldsymbol{S}\,\boldsymbol{u} = \boldsymbol{f} \tag{2.14}$$

Da es sich bei der Matrix $\boldsymbol{S}$ um eine quadratische Matrix handelt, ist das Gleichungssystem genau dann eindeutig lösbar, wenn $\boldsymbol{S}$ invertierbar ist.
Unter den Voraussetzungen, dass V_h endlichdimensional ist und die Bilinearform $B(\cdot,\cdot) : V_h \times V_h \to \mathbb{R}$ koerziv ist, lässt sich die positive Definitheit der Steifigkeitsmatrix $\boldsymbol{S}$ zeigen.
Wäre nun die Steifigkeitsmatrix $\boldsymbol{S}$ singulär, dann existiert ein nichttrivialer Vektor $\boldsymbol{x} \in \mathbb{R}^N$, sodass $\boldsymbol{S}\boldsymbol{x} = \boldsymbol{0}$. Notwendigerweise gilt dann auch $\boldsymbol{x}^{\boldsymbol{T}}\boldsymbol{S}\boldsymbol{x} = 0$, was im Widerspruch zur positiven Definitheit von $\boldsymbol{S}$ steht.
Somit ist die Steifigkeitsmatrix $\boldsymbol{S}$ invertierbar und das System 2.14 eindeutig lösbar.

Im letzten Abschnitt wurde ein Modellproblem 2.2 definiert und die schwache Formulierung dafür hergeleitet. Unter der Bezeichnung Galerkin-Verfahren wurde das Problem ins Endlichdimensionale überführt und es folgte die Darstellung 2.13. Davon konnte man das lineare Gleichungssystem 2.14 ableiten. Wenn man dieses Gleichungssystem lösen kann, erhält man also eine Approximation für die Lösung des Modellproblems. Wieweit diese Lösungen voneinander abweichen, also welcher Fehler durch die endlichdimensionale Approximation begangen wird, soll im nächsten Unterkapitel genauer beleuchtet werden.

2.4 Fehlerabschätzung – Lemma von Céa

Dieser Abschnitt beschäftigt sich mit dem Fehler, der durch die Diskretisierung der schwachen Formulierung entsteht.

Sei u die exakte Lösung für das kontinuierliche Problem 2.8, und u_h die exakte Lösung für das diskrete Problem 2.9. Durch die Subtraktion dieser beiden Lösungen erhält man den Fehler

$$e_h = u - u_h.$$

Der Fehler e_h besitzt folgende Orthogonalitätseigenschaft:

$$B(u - u_h, v) = 0 \quad \forall\, v \in V_h \tag{2.15}$$

Ist die Bilinearform $B(\cdot, \cdot)$ symmetrisch, so induziert sie ein Skalarprodukt (vgl. Definition Energienorm auf Seite 13):

$$(u, v)_e = B(u, v)$$

Zusammen mit 2.15 erhält man:

$$(e_n, v)_e = 0 \quad \forall\, v \in V_h$$

Der Fehler der Galerkin Approximation ist also orthogonal zum Unterraum V_h bezüglich des energetischen Skalarprodukts. Die approximierte Lösung $u_h \in V_h$ ist somit die orthogonale Projektion der exakten Lösung $u \in V$ in den Unterraum V_h bezüglich des energetischen Skalarprodukts. Daher ist u_h das am nächsten zur exakten Lösung u liegende Element in V_h bezüglich der Energienorm und somit die Bestapproximation:

$$\|u - u_h\|_e = \inf_{v \in V_h} \|u - v\| \tag{2.16}$$

Eine weitere Fehlerabschätzung bietet das Lemma von Céa, wo die Symmetrie der Bilinearform nicht vorausgesetzt werden muss.

Satz (Lemma von Céa)
Sei V ein Hilbertraum, $B(\cdot,\cdot) : V \times V \to \mathbb{R}$ eine stetige koerzive Bilinearform mit

$$|B(u,v)| \leq M \|u\| \|v\| \quad \forall\, u,v \in V$$
$$\alpha \|u\|^2 \leq B(u,u) \quad \forall\, u \in V,$$

dabei sind α und M positive Konstanten.

Sei $l \in V'$ und sei $V_h \subset V$ ein abgeschlossener Unterraum von V. Sei $u \in V$ die Lösung von 2.8 und $u_h \in V_h$ die Lösung von 2.9. Dann gilt:

$$\|u - u_h\| \leq \frac{M}{\alpha} \inf_{v \in V_h} \|u - v\| \tag{2.17}$$

Der Approximationsfehler e_n hängt laut dem Lemma von Céa von der Wahl des Unterraums V_h ab, nicht aber von der Wahl der Basis. Auch die numerischen Resultate sollten von den konkreten Basisfunktionen unabhängig sein. Auf die Kondition der Steifigkeitsmatrix S kann die Wahl der Basisfunktionen aber durchaus Einfluss nehmen.
Mit der Wahl eines geeineten Unterraums und der dazugehörigen Basis beschäftigt sich das folgende Unterkapitel.

2.5 Basisfunktionen

Auch wenn das Galerkin Verfahren keine spezielle Basis des Raums V_h vorschreibt, werden in der Regel Basisfunktionen verwendet, die nur einen kleinen Träger haben. Sind die Träger zweier Basisfunktionen v_j und v_k disjunkt, so ist der Eintrag in der Steifigkeitsmatrix $s_{jk} = B(v_j, v_k) = 0$. Der Vorteil solcher schwach besetzter Steifigkeitsmatrizen besteht darin, dass das entsprechende System leichter numerisch gelöst werden kann.

Der einfachste Fall ist wohl, im Raum V_h nur stückweise affine Funktionen zuzulassen. In unserem Modellproblem wird das Intervall $\Omega = (0,1)$ betrachtet. Dieses Intervall wird nun in $N-1$ Teilintervalle zerlegt:

$$K_j = (z_j, z_{j+1}),\ j = 1, \ldots, N \qquad 0 = z_1 < z_2 < \ldots < z_N = 1$$

Dabei bezeichnet man die Intervalle K_j als Elemente und die Punkte z_j als Knoten. Die Menge aller Elemente, $\mathcal{T} := \{K_j | j = 1, \ldots, N-1\}$, wird Gitter genannt, $h_j = z_{j+1} - z_j$ ist die Elementgröße. Im speziellen Fall der stückweise affinen Funktionen wird der Raum V_h auch als $S^{1,1}$ bzw. $S_0^{1,1}$ bezeichnet, der formal folgendermaßen definiert werden kann:

$$S^{1,1}(\mathcal{T}) := \{v \in H^1(\Omega) |\, v_{|K} \in \mathcal{P}_1 \quad \forall\, K \in \mathcal{T}\} \tag{2.18}$$
$$S_0^{1,1}(\mathcal{T}) := S^{1,1}(\mathcal{T}) \cap H_0^1(\Omega)$$

Die so genannten Hutfunktionen bilden eine Basis von $S^{1,1}(\mathcal{T})$:

$$\varphi_j(z) = \begin{cases} \frac{z-z_{j-1}}{h_{j-1}} & z \in (z_{j-1}, z_j) \\ \frac{z_{j+1}-z}{h_j} & z \in (z_j, z_{j+1}) \\ 0 & sonst \end{cases} \tag{2.19}$$

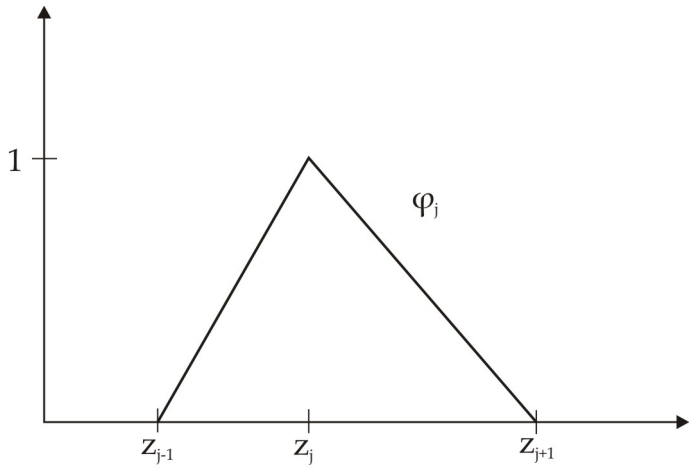

Abbildung 2.1: Hutfunktion

Diese Basisfunktionen φ_i erfüllen für alle Knoten z_j die Eigenschaft

$$\varphi_i(z_j) = \delta_{ij}.$$

Kapitel 3

FEM mit Matlab

Dieses Kapitel beschäftigt sich mit der Implementierung der Finiten Elemente Methode für ein eindimensionales Modellproblem in Matlab. Die Implementierung und somit auch diese Ausführungen folgen der Arbeitsunterlage [1]. Weiterführend sei auf einen Artikel von Jochen Alberty et al. [2] verwiesen, in dem die Vorgangsweise auch für mehrdimensionale Fälle beschrieben ist.

Die mathematische Theorie zur Vorgangsweise bei der Finiten Elemente Methode wurde bereits im letzten Kapitel entwickelt und wird daher hier vorausgesetzt, die Konzentration soll auf einer speziellen Art der Implementierung in Matlab liegen. Die Implementierung des eigentlichen Blutflussmodells basiert auf der Implementierung dieses Modellproblems, daher ist es empfehlenswert, sich mit den Merkmalen dieser Art der Implementierung bereits hier mit einem vergleichsweise einfachen Modellproblem vertraut zu machen.

3.1 Das Modellproblem

Wir greifen das bereits in Kapitel 2.2 definierte Modellproblem 2.2 wieder auf:

$$\begin{aligned} -\left(a(z)u'(z)\right)' + c(z)u(z) &= f(z) \quad \text{für } z \in \Omega = (0,1) \\ u(0) = u(1) &= 0 \end{aligned} \tag{3.1}$$

Über die schwache Formulierung (vgl. Kapitel 2.2) und die Galerkin Diskretisierung (vgl. Kapitel 2.3) kommt man zur diskreten Modellbeschreibung:

$$\int_0^1 a(z)u_h'(z)v_h'(z)\,dz + \int_0^1 c(z)u_h(z)v_h(z)\,dz = \int_0^1 f(z)v_h(z)\,dz \tag{3.2}$$

Als endlichdimensionaler Lösungsraum V_h sei der Raum der stückweise affinen Funktionen $S_0^{1,1}$ (vgl. 2.18) genommen, als Basis dieses Raums die in 2.19 definierten Hutfunktionen.

Setzt man nun die Basisfunktionen in 3.2 ein, so kann man die Koeffizientenmatrizen $\boldsymbol{A}$ und $\boldsymbol{C}$ (vgl. Definition der Steifigkeitsmatrix in Kapitel 2.3) sowie den Lastvektor $\boldsymbol{f}$, oft auch nur rechte Seite genannt (vgl. Definition des Lastvektors in Kapitel 2.3), folgendermaßen definieren:

$$\boldsymbol{A}_{ik} := \int_0^1 a(z)\varphi_i'(z)\varphi_k'(z)\,dz \tag{3.3}$$

$$\boldsymbol{C}_{ik} := \int_0^1 c(z)\varphi_i(z)\varphi_k(z)\,dz \tag{3.4}$$

$$\boldsymbol{f}_i := \int_0^1 f(z)\varphi_i(z)\,dz \tag{3.5}$$

Man erhält somit folgendes Gleichungssystem, das gelöst werden soll (vgl. 2.14):

$$(\boldsymbol{A} + \boldsymbol{C})\boldsymbol{u} = \boldsymbol{f} \tag{3.6}$$

3.2 Zerlegung in finite Elemente

Um die Hutfunktionen als Basis des $S_0^{1,1}$ verwenden zu können, muss das betrachtete Einheitsintervall $\Omega = (0, 1)$ in verschiedene Elemente, d.h. Teilintervalle, unterteilt werden. Dabei ist es am einfachsten, äquidistante Knoten zu setzen. Beispielsweise teilt man das Intervall $\Omega = (0, 1)$ also in 10 Teilintervalle mit der Länge $0,1$. Hat man wie im Modell angenommen homogene Dirichlet Randbedingungen, so erhält man dadurch neun Basisfunktionen. Sind die Dirichlet Randbedingungen nicht homogen, muss man zwei zusätzliche „halbe“ Hutfunktionen bei den Knoten z_1 und z_{11} einführen.

Die Informationen über die Diskretisierung des Raums werden in drei Dateien im ASCII Format mit der Endung .dat abgespeichert. Diese Dateien können von Matlab eingelesen werden.
Im der ersten Datei `coordinates.dat` speichert man die Ortskoordinaten der Knoten in Spaltenform:

$$\begin{matrix} 0.0 \\ 0.1 \\ 0.2 \\ \vdots \\ 1.0 \end{matrix} \tag{3.7}$$

In einer zweiten Datei `elements.dat` werden in der j-ten Zeile die Nummern der Knoten gespeichert, die das j-te Element beranden. Dabei muss beachtet werden, dass in Matlab die Vektoren immer mit dem Index 1 beginnen. So liegt z.B. das erste Element K_1 zwischen den Knoten z_1 und z_2, die dazugehörigen Koordinaten sind die ersten beiden Elemente in der Datei `coordinates.dat`. Die erste Zeile der Datei `elements.dat` lautet daher $(1 \quad 2)$. Daraus ergibt sich für `elements.dat`:

$$\begin{matrix} 1 & 2 \\ 2 & 3 \\ 3 & 4 \\ \vdots & \vdots \\ 10 & 11 \end{matrix} \tag{3.8}$$

Schlussendlich werden in einer dritten Datei `dirichlet.dat` die Nummern der Knoten am Rand angegeben, in unserem Fall:

$$\begin{matrix} 1 \\ 11 \end{matrix} \tag{3.9}$$

3.3 Aufstellen der Matrizen

Die Berechnung der Koeffizientenmatrizen $\boldsymbol{A}$ (vgl. 3.3) und $\boldsymbol{C}$ (vgl. 3.4) kann vereinfacht werden, indem man die Integrale nur auf den jeweiligen Elementen berechnet und dann über alle Elemente aufsummiert.

$$\boldsymbol{A}_{ik} = \sum_{j=1}^{10} \int_{K_j} a(z)\varphi_i'(z)\varphi_k'(z)\, dz \tag{3.10}$$

$$\boldsymbol{C}_{ik} = \sum_{j=1}^{10} \int_{K_j} c(z)\varphi_i(z)\varphi_k(z)\, dz \tag{3.11}$$

Es werden nun für ein Element K_j alle Matrixeinträge $\boldsymbol{A}_{ik}(K_j)$ und $\boldsymbol{C}_{ik}(K_j)$ berechnet, die sich aus dem betrachteten Element ergeben. Da als Basis die Hutfunktionen gewählt wurden, überschneiden sich in einem Element K_j nur zwei Basisfunktionen, nämlich φ_j und φ_{j+1}. Daher bekommt man Einträge für die Matrixelemente $\boldsymbol{A}_{j\,j}$, $\boldsymbol{A}_{j\,j+1}$, $\boldsymbol{A}_{j+1\,j}$ und $\boldsymbol{A}_{j+1\,j+1}$ bzw. $\boldsymbol{C}_{j\,j}$, $\boldsymbol{C}_{j\,j+1}$, $\boldsymbol{C}_{j+1\,j}$ und $\boldsymbol{C}_{j+1\,j+1}$.

Der in K_j liegende Teil der Hutfunktion φ_j hat die Form einer fallenden Geraden, die Funktion φ_{j+1} ist genau gegengleich eine steigende Gerade.

Auf dem Intervall (z_j, z_{j+1}) mit $h_j = z_{j+1} - z_j$ gilt also:

$$\varphi_j(z) = \frac{z_{j+1} - z}{h_j} \qquad \varphi_{j+1}(z) = \frac{z - z_j}{h_j}$$

Die Ableitungen lauten daher:

$$\varphi_j'(z) = -\frac{1}{h_j} \qquad \varphi_{j+1}'(z) = \frac{1}{h_j}$$

Damit können nun die Einträge für die Matrix $\boldsymbol{A}$ berechnet werden:

$$\begin{aligned}
\boldsymbol{A}_{j\,j}(K_j) &= \int_{K_j} a(z)\varphi_j'(z)\varphi_j'(z)\,dz = \int_{K_j} a(z)\frac{-1}{h_j}\frac{-1}{h_j}\,dz = \frac{1}{h_j^2}\int_{K_j} a(z)\,dz \\
\boldsymbol{A}_{j\,j+1}(K_j) &= \int_{K_j} a(z)\varphi_j'(z)\varphi_{j+1}'(z)\,dz = \int_{K_j} a(z)\frac{-1}{h_j}\frac{1}{h_j}\,dz = \frac{-1}{h_j^2}\int_{K_j} a(z)\,dz \\
\boldsymbol{A}_{j+1\,j}(K_j) &= \int_{K_j} a(z)\varphi_{j+1}'(z)\varphi_j'(z)\,dz = \int_{K_j} a(z)\frac{1}{h_j}\frac{-1}{h_j}\,dz = \frac{-1}{h_j^2}\int_{K_j} a(z)\,dz \\
\boldsymbol{A}_{j+1\,j+1}(K_j) &= \int_{K_j} a(z)\varphi_{j+1}'(z)\varphi_{j+1}'(z)\,dz = \int_{K_j} a(z)\frac{1}{h_j}\frac{1}{h_j}\,dz = \frac{1}{h_j^2}\int_{K_j} a(z)\,dz
\end{aligned}$$

Die Einträge für die Matrix $\boldsymbol{C}$ ergeben sich folgendermaßen:

$$\begin{aligned}
\boldsymbol{C}_{j\,j}(K_j) &= \int_{K_j} c(z)\varphi_j(z)\varphi_j(z)\,dz = \\
&= \int_{K_j} c(z)\frac{z_{j+1} - z}{h_j}\frac{z_{j+1} - z}{h_j}\,dz = \frac{1}{h_j^2}\int_{K_j} c(z)(z_{j+1} - z)^2\,dz \\
\boldsymbol{C}_{j\,j+1}(K_j) &= \int_{K_j} c(z)\varphi_j(z)\varphi_{j+1}(z)\,dz = \\
&= \int_{K_j} c(z)\frac{z_{j+1} - z}{h_j}\frac{z - z_j}{h_j}\,dz = \frac{1}{h_j^2}\int_{K_j} c(z)(z_{j+1} - z)(z - z_j)\,dz \\
\boldsymbol{C}_{j+1\,j}(K_j) &= \int_{K_j} c(z)\varphi_{j+1}(z)\varphi_j(z)\,dz = \\
&= \int_{K_j} c(z)\frac{z - z_j}{h_j}\frac{z_{j+1} - z}{h_j}\,dz = \frac{1}{h_j^2}\int_{K_j} c(z)(z_{j+1} - z)(z - z_j)\,dz \\
\boldsymbol{C}_{j+1\,j+1}(K_j) &= \int_{K_j} c(z)\varphi_{j+1}(z)\varphi_{j+1}(z)\,dz = \\
&= \int_{K_j} c(z)\frac{z - z_j}{h_j}\frac{z - z_j}{h_j}\,dz = \frac{1}{h_j^2}\int_{K_j} c(z)(z - z_j)^2\,dz
\end{aligned}$$

Die nächsten Zeilen zeigen, wie das Aufstellen der Matrizen mit Matlab realisiert wird.

```
for j=1:size(elements,1)
    z=coordinates(elements(j,:));
    A(elements(j,:),elements(j,:)) = A(elements(j,:),elements(j,:)) + quad('a',
        z(1),z(2)) / (z(2)-z(1))^2 * [1 -1;-1 1];
    C(elements(j,:),elements(j,:)) = C(elements(j,:),elements(j,:)) + coeffC(z)
        ;
end
```

Mittels der for-Schleife werden die einzelnen Elemente durchlaufen. Im Falle des Modellproblems läuft j also von 1 bis 10. Dabei werden dem Vektor z jeweils die Nummern der aktuellen Knoten übergeben, für das erste Element, also für $j = 1$, besteht z daher aus $(1 \quad 2)$.

Danach werden einer 2×2 Teilmatrix der Matrix $\boldsymbol{A}$ neue Werte zugewiesen. Diese Teilmatrix enthält zwei Elemente der Diagonale und die zwei danebenliegenden Elemente der ersten Nebendiagonalen. Im Schritt j entspricht das genau den Elementen $\boldsymbol{A}_{j\,j}$, $\boldsymbol{A}_{j\,j+1}$, $\boldsymbol{A}_{j+1\,j}$ und $\boldsymbol{A}_{j+1\,j+1}$. Man beachte, dass die Matrixelemente nicht durch die neu berechneten Werte überschrieben, sondern addiert werden. Das geschieht deshalb, weil sich die meisten Basisfunktionen über zwei Elemente erstrecken und daher mehrere Bestandteile addiert werden müssen. Die Matrix $\boldsymbol{A}$ wird am Beginn des Programms als Matrix nur mit Nulleinträgen initialisiert.

Die Integration wird mit Hilfe der Matlab-Funktion quad durchgeführt. In der Datei `a.m` wird eine Funktion definiert, die der gewünschten mathematischen Funktion $a(z)$ entspricht. Da sich die Anteile für $\boldsymbol{A}_{j\,j}$, $\boldsymbol{A}_{j\,j+1}$, $\boldsymbol{A}_{j+1\,j}$ und $\boldsymbol{A}_{j+1\,j+1}$ auf einem bestimmten Element K_j nur durch ihr Vorzeichen unterscheiden, genügt es, die Integration nur einmal auszuführen. Die passenden Vorzeichen bekommt man dann durch eine Multiplikation mit einer 2×2 Matrix, die ± 1 als Einträge besitzt.

Die entsprechenden Einträge $\boldsymbol{C}_{j\,j}$, $\boldsymbol{C}_{j\,j+1}$, $\boldsymbol{C}_{j+1\,j}$ und $\boldsymbol{C}_{j+1\,j+1}$ für die Matrix $\boldsymbol{C}$ werden auf ähnliche Weise erzeugt. Allerdings wird die Integration nicht direkt im Hauptprogramm, sondern mittels einer eigenen Matlab-Funktion coeffC ausgeführt.
Der Code der Datei `coeffC.m` lautet folgendermaßen:

```
function M=coeffC(z)
for j=1:2
    for k=1:2
        M(j,k)=(-1)^j*(-1)^k*quad('integrand',z(1),z(2) ,[],[], 'c',1,-z(1+mod(j
            ,2)),1,-z(1+mod(k,2)))/(z(2)-z(1))^2;
    end
```

```
end
```

Die vier verschiedenen Integrale werden durch die Verschachtelung zweier Schleifen wieder mit Hilfe des quad Befehls berechnet. Dabei werden der Funktion integrand die Funktion c sowie vier Parameter übergeben, welche die Koeffizienten im Integral darstellen (siehe unten im Code von `integrand.m`). Diese vier Parameter sind je nach Wert von j und k verschieden, nämlich entweder der Wert von z_j oder z_{j+1}, die im Matlab-Code ja in jedem Schritt als z(1) bzw. z(2) vorliegen. Der Wechsel zwischen z(1) und z(2) wird mit der Modulo-Funktion mod vollzogen.

Die wesentlichen Zeilen der Datei `integrand.m` lauten:

```
function val=integrand(z,func,c1,c0,d1,d0)
val=eval([func,'(z)'])*(c1*z+c0)*(d1*z+d0);
```

Die Funktion c wird unter dem Namen func übernommen und mittels eval bei z ausgewertet. Das Ergebnis wird mit zwei Polynomen erster Ordnung multipliziert, deren Koeffizienten wie zuerst beschrieben zugewiesen werden. Mit dieser Vorgangsweise können alle vier benötigten Integranden dargestellt werden.

Durch das Abarbeiten der Schleife im Hauptprogramm werden auf die soeben geschilderte Weise zwei Matrizen erzeugt, die zusammen die Steifigkeitsmatrix des Modellproblems bilden. Eine ähnliche Vorgangsweise wird auch beim Erstellen des Lastvektors verfolgt, wie im nächsten Abschnitt ausgeführt wird.

3.4 Aufstellen des Lastvektors

Beim Aufstellen der rechten Seite ist die Verfahrensweise mit jener bei den Matrizen zuvor vergleichbar. In jedem Element K_j liegen zwei Basisfunktionen, nämlich φ_j und φ_{j+1}. Daraus ergeben sich Änderungen bei den zwei entsprechenden Einträgen $\boldsymbol{f}_j$ und $\boldsymbol{f}_{j+1}$ des Lastvektors $\boldsymbol{f}$.

$$\boldsymbol{f}_j(K_j) = \int_{K_j} f(z)\varphi_j(z)\,dz = \frac{1}{h_j}\int_{K_j} f(z)(z_{j+1}-z)\,dz$$
$$\boldsymbol{f}_{j+1}(K_j) = \int_{K_j} f(z)\varphi_{j+1}(z)\,dz = \frac{1}{h_j}\int_{K_j} f(z)(z-z_j)\,dz$$

Der dazugehörige Matlab-Code lautet folgendermaßen:

```
for j=1:size(elements,1)
    z=coordinates(elements(j,:));
    for k=1:2
        rhs(elements(j,k))=rhs(elements(j,k)) + (-1)^k* quad('integrand',z(1),
            z(2) ,[],[], 'f',0,1,1,-z(1+mod(k,2)))/(x(2)-x(1));
    end
end
```

Die finiten Elemente werden mit einer for-Schleife durchlaufen, dabei werden jeweils in der Variable z die Nummern der entsprechenden Knoten gespeichert. Die zwei sich ändernden Einträge des Lastvektors werden in einer zweiten for-Schleife berechnet. Zur Integration wird wieder der Befehl quad benutzt. Der Integrand wird wieder mit Hilfe der bereits im letzten Abschnitt eingeführten Funktion `integrand.m` (siehe S. 26) beschrieben. Dieser werden die Funktion f sowie die passenden Koeffizienten übergeben.

Nachdem die Matrizen der linken Seite und der Lastvektor der rechten Seite auf diese Weise erstellt wurden, ist das Gleichungssystem vollständig. Die Aufgabe lautet nun, dieses Gleichungssystem unter Berücksichtigung der Randbedingungen zu lösen.

3.5 Randbedingungen

Bis jetzt haben die Randbedingungen in das Gleichungssystem keinen Eingang gefunden. Liegen homogene Dirichlet Randbedingungen vor, hat dies auch keine Auswirkungen auf den Lösungsvorgang. Sind die Dirichlet Randbedingungen aber ungleich Null, so muss das Gleichungssystem vor dem Lösen umgewandelt werden, um den Randbedingungen Rechnung zu tragen.
Wie etwas später zu sehen ist, kann dies durch eine vergleichsweise einfache Umformung, die nur eine Zeile Matlab Code beansprucht, durchgeführt werden. Zunächst soll aber dargelegt werden, wie es zu der Änderung des Lastvektors kommt.

Das Modellproblem besteht aus den zwei quadratischen Matrizen $\boldsymbol{A}$ und $\boldsymbol{C}$ mit jeweils $n = 11$ Zeilen bzw. Spalten, sowie einem Lastvektor $\boldsymbol{f}$, ebenfalls mit der Dimension $n = 11$. Der Lösungsvektor hat daher ebenfalls die gleiche Dimension, wobei die beiden äußeren Einträge u_1 und u_{11} durch die Randbedingungen bestimmt werden. Die inneren Komponenten $u_2, u_3, \ldots, u_{10}$ müssen berech-

net werden. Die Indizes dieser inneren Vektoreinträge entsprechen jenen der so genannten freien Knoten.

Im Sinne einer allgemeineren Betrachtung wird im Weiteren eine Steifigkeitsmatrix $\boldsymbol{S}$ $(= \boldsymbol{A} + \boldsymbol{C})$ der Dimension $n \times n$ verwendet. Das Gleichungssystem lautet also

$$\boldsymbol{S}\boldsymbol{u} = \boldsymbol{f}$$

Dabei kann die Matrix $\boldsymbol{S}$ folgendermaßen aufgeteilt werden:

$$\left(\begin{array}{c|cccc|c} s_{1\,1} & s_{1\,2} & s_{1\,3} & \cdots & s_{1\,n-1} & s_{1\,n} \\ \hline s_{2\,1} & s_{2\,2} & s_{2\,3} & \cdots & s_{2\,n-1} & s_{2\,n} \\ s_{3\,1} & s_{3\,2} & s_{3\,3} & \cdots & s_{3\,n-1} & s_{3\,n} \\ \vdots & \vdots & \vdots & \ddots & \vdots & \vdots \\ s_{n-1\,1} & s_{n-1\,2} & s_{n-1\,3} & \cdots & s_{n-1\,n-1} & s_{n-1\,n} \\ \hline s_{n\,1} & s_{n\,2} & s_{n\,3} & \cdots & s_{n\,n-1} & s_{n\,n} \end{array}\right)$$

Schreibt man die Matrix bzw. die Vektoren in Blockform an (wobei der Index M für Mitte steht) so hat das Gleichungssystem folgendes Aussehen:

$$\left(\begin{array}{c|c|c} \boldsymbol{S}_{1\,1} & \boldsymbol{S}_{1\,M} & \boldsymbol{S}_{1\,n} \\ \hline \boldsymbol{S}_{M\,1} & \boldsymbol{S}_{M\,M} & \boldsymbol{S}_{M\,n} \\ \hline \boldsymbol{S}_{n\,1} & \boldsymbol{S}_{n\,M} & \boldsymbol{S}_{n\,n} \end{array}\right) \left(\begin{array}{c} \boldsymbol{u}_1 \\ \hline \boldsymbol{u}_M \\ \hline \boldsymbol{u}_n \end{array}\right) = \left(\begin{array}{c} \boldsymbol{f}_1 \\ \hline \boldsymbol{f}_M \\ \hline \boldsymbol{f}_n \end{array}\right)$$

Führt man nun die Multiplikation auf der linken Seite durch, erhält man:

$$\left(\begin{array}{c} \boldsymbol{S}_{1\,1}\boldsymbol{u}_1 + \boldsymbol{S}_{1\,M}\boldsymbol{u}_M + \boldsymbol{S}_{1\,n}\boldsymbol{u}_n \\ \hline \boldsymbol{S}_{M\,1}\boldsymbol{u}_1 + \boldsymbol{S}_{M\,M}\boldsymbol{u}_M + \boldsymbol{S}_{M\,n}\boldsymbol{u}_n \\ \hline \boldsymbol{S}_{n\,1}\boldsymbol{u}_1 + \boldsymbol{S}_{n\,M}\boldsymbol{u}_M + \boldsymbol{S}_{n\,n}\boldsymbol{u}_n \end{array}\right) = \left(\begin{array}{c} \boldsymbol{f}_1 \\ \hline \boldsymbol{f}_M \\ \hline \boldsymbol{f}_n \end{array}\right)$$

Danach werden alle bekannten Teile auf die rechte Seite gebracht. Da $\boldsymbol{u}_1$ und $\boldsymbol{u}_n$ ja durch die Randbedingungen bestimmt werden, sind das alle Ausdrücke, die $\boldsymbol{u}_M$ nicht enthalten.

$$\left(\begin{array}{c} \boldsymbol{S}_{1\,M}\boldsymbol{u}_M \\ \hline \boldsymbol{S}_{M\,M}\boldsymbol{u}_M \\ \hline \boldsymbol{S}_{n\,M}\boldsymbol{u}_M \end{array}\right) = \left(\begin{array}{c} \boldsymbol{f}_1 - (\boldsymbol{S}_{1\,1}\boldsymbol{u}_1 + \boldsymbol{S}_{1\,n}\boldsymbol{u}_n) \\ \hline \boldsymbol{f}_M - (\boldsymbol{S}_{M\,1}\boldsymbol{u}_1 + \boldsymbol{S}_{M\,n}\boldsymbol{u}_n) \\ \hline \boldsymbol{f}_n - (\boldsymbol{S}_{n\,1}\boldsymbol{u}_1 + \boldsymbol{S}_{n\,n}\boldsymbol{u}_n) \end{array}\right)$$

Löst man die Multiplikationen auf der linken Seite aus den einzelnen Blöcken heraus und fasst sie zusammen, zeigt sich folgende Struktur:

$$\left(\begin{array}{c} \boldsymbol{S}_{1\,M} \\ \hline \boldsymbol{S}_{M\,M} \\ \hline \boldsymbol{S}_{n\,M} \end{array}\right) \left(\begin{array}{c} \\ \boldsymbol{u}_M \\ \\ \end{array}\right) = \left(\begin{array}{c} \boldsymbol{f}_1 - (\boldsymbol{S}_{1\,1}\boldsymbol{u}_1 + \boldsymbol{S}_{1\,n}\boldsymbol{u}_n) \\ \hline \boldsymbol{f}_M - (\boldsymbol{S}_{M\,1}\boldsymbol{u}_1 + \boldsymbol{S}_{M\,n}\boldsymbol{u}_n) \\ \hline \boldsymbol{f}_n - (\boldsymbol{S}_{n\,1}\boldsymbol{u}_1 + \boldsymbol{S}_{n\,n}\boldsymbol{u}_n) \end{array}\right)$$

Wird nun noch auf die Blockdarstellung verzichtet, kann man das Gleichungssystem folgendermaßen anschreiben:

$$\begin{pmatrix} s_{1\,2} & s_{1\,3} & \cdots & s_{1\,n-1} \\ \hline s_{2\,2} & s_{2\,3} & \cdots & s_{2\,n-1} \\ s_{3\,2} & s_{3\,3} & \cdots & s_{3\,n-1} \\ \vdots & \vdots & \ddots & \vdots \\ s_{n-1\,2} & s_{n-1\,3} & \cdots & s_{n-1\,n-1} \\ \hline s_{n\,2} & s_{n\,3} & \cdots & s_{n\,n-1} \end{pmatrix} \begin{pmatrix} u_2 \\ u_3 \\ \vdots \\ u_{n-1} \end{pmatrix} = \begin{pmatrix} f_1 \\ \hline f_2 \\ f_3 \\ \vdots \\ f_{n-1} \\ \hline f_n \end{pmatrix} - u_1 \begin{pmatrix} s_{1\,1} \\ \hline s_{2\,1} \\ s_{3\,1} \\ \vdots \\ s_{n-1\,1} \\ \hline s_{n\,1} \end{pmatrix} - u_n \begin{pmatrix} s_{1\,n} \\ \hline s_{2\,n} \\ s_{3\,n} \\ \vdots \\ s_{n-1\,n} \\ \hline s_{n\,n} \end{pmatrix}$$

Die Matrix der linken Seite hat also die Dimension $n \times n-2$. Um ein eindeutiges Gleichungssystem für die Unbekannten $u_2, u_3, \ldots, u_{n-1}$ zu bekommen, benötigt man aber nur eine Matrix der Dimension $n-2 \times n-2$. Wie man leicht erkennen kann, erhält man ein derartiges Gleichungssystem durch das Weglassen der ersten und letzten Zeile.

In Matlab lassen sich die Umformungen der rechten Seite auf diese Weise durchführen:

```
u = sparse(length(coordinates),1);
u( dirichlet ) = u_d(coordinates( dirichlet ));
rhs = rhs-(A+C)*u;
```

Zuerst wird der Vektor u initialisiert und es werden ihm die Randbedingungen übergeben. D.h. der erste und der letzte Vektoreintrag beinhalten nun die Randwerte, die inneren Einträge sind Null.
Aufrund dieser Beschaffenheit des Vektors u entspricht die Multiplikation (A+C)*u genau jenen beiden in obiger Formel, wo u_1 mit der ersten Spalte der Matrix $\boldsymbol{S}$ und u_n mit der letzten Spalte multipliziert wird. Subtrahiert man nun noch das Ergebnis vom ursprünglichen Lastvektor, so erhält man die neue rechte Seite.

3.6 Lösen des Gleichungssystems

Um die Lösung für das diskrete Problem des Galerkin Verfahrens zu erhalten, muss somit folgendes Gleichungssystem gelöst werden, das im letzten Abschnitt

hergeleitet wurde:

$$\begin{pmatrix} s_{2\,2} & s_{2\,3} & \cdots & s_{2\,n-1} \\ s_{3\,2} & s_{3\,3} & \cdots & s_{3\,n-1} \\ \vdots & \vdots & \ddots & \vdots \\ s_{n-1\,2} & s_{n-1\,3} & \cdots & s_{n-1\,n-1} \end{pmatrix} \begin{pmatrix} u_2 \\ u_3 \\ \vdots \\ u_{n-1} \end{pmatrix} = \begin{pmatrix} f_2 \\ f_3 \\ \vdots \\ f_{n-1} \end{pmatrix} - u_1 \begin{pmatrix} s_{2\,1} \\ s_{3\,1} \\ \vdots \\ s_{n-1\,1} \end{pmatrix} - u_n \begin{pmatrix} s_{2\,n} \\ s_{3\,n} \\ \vdots \\ s_{n-1\,n} \end{pmatrix}$$

Die Reduktion des Gleichungssystems wird in Matlab so realisiert, dass nur mehr Einträge der verschiedenen Matrizen und Vektoren verwendet werden, die den freien Knoten zugeordnet werden. Daher werden die entsprechenden Indizes unter FreeNodes abgespeichert.
Um das Gleichungssystem numerisch zu lösen, wird der Backslash-Operator verwendet. Dieser ist äquivalent zur Funktion mldivide. Je nach Gestalt des Gleichungssystems verwendet Matlab dabei unterschiedliche Verfahren, um die Lösung zu berechnen.

```
u(FreeNodes)=(A(FreeNodes,FreeNodes)+(FreeNodes,FreeNodes))\rhs(
    FreeNodes);
```

Somit beinhaltet der Vektor $\boldsymbol{u}$ die Lösung der Approximation des Modellproblems. Genauer gesagt beinhaltet er die Koeffizienten, die noch mit den entsprechenden Basisfunktionen multipliziert werden müssen, danach muss nur noch aufsummiert werden, um die approximierte Lösungsfunktion zu erhalten.

Da als Basisfunktionen die Hutfunktionen gewählt wurden, ist eine graphische Darstellung der approximierten Lösung leicht möglich. Es müssen nur die Einträge von $\boldsymbol{u}$ über den Knoten aufgezeichnet und linear verbunden werden.

Die Abbildung 3.1 zeigt eine exakte sowie eine approximierte Lösung des Modellproblems. Die auftretenden Funktionen wurden dafür folgendermaßen gewählt:

$$a(z) = 1 \qquad c(z) = \frac{1}{100} \qquad f(z) = 1$$

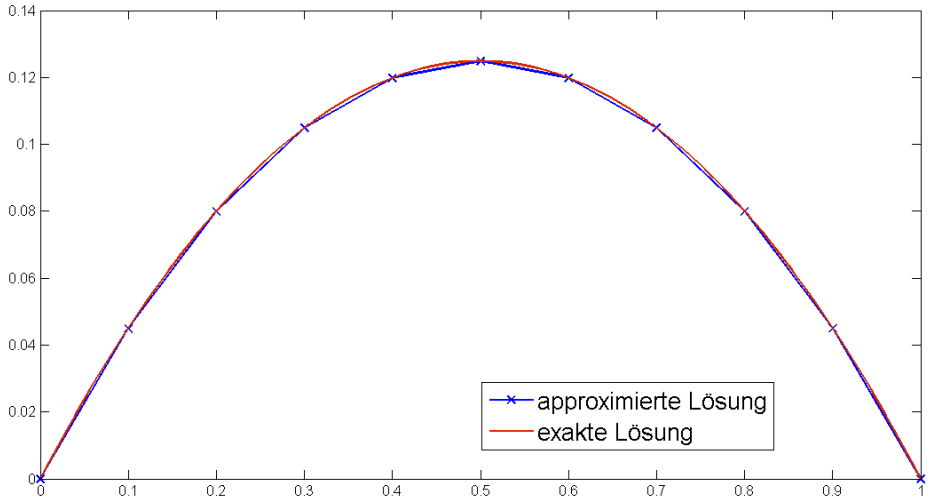

Abbildung 3.1: Vergleich zwischen exakter und approximierter Lösung des Modellproblems

Kapitel 4

Blutflussmodell

4.1 Hyperbolische partielle Differentialgleichungen erster Ordnung

Bevor sich der nächste Abschnitt dem Blutflussmodell widmet, sollen hier zunächst einige Begriffe geklärt werden, die im Weiteren noch eine wichtige Rolle spielen werden. Es ist hier nicht der geeignete Platz um eine Vorstellung und Klassifizierung aller Arten von partiellen Differentialgleichungen zu geben, ja selbst von den partiellen Differentialgleichungen erster Ordnung wird im Folgenden nur eine spezielle Klasse betrachtet. Dies geschieht natürlich im Hinblick auf das eigentliche Modell zur Beschreibung des Blutflusses, und daher ist die Notation schon an einen Artikel von Luca Formaggia et al. über das Blutflussmodell [5] angelehnt. Ansonsten folgt die Beschreibung wieder dem Buch von Pavel Solin [13].

Betrachtet werden soll eine partielle Differentialgleichung erster Ordnung in Erhaltungsform mit einer Raumdimension von folgender Art:

$$\frac{\partial \boldsymbol{u}(z,t)}{\partial t} + \frac{\partial \boldsymbol{F}\left(\boldsymbol{u}(z,t)\right)}{\partial z} = \boldsymbol{B}\left(\boldsymbol{u}(z,t)\right) \tag{4.1}$$

Dabei beinhaltet der n-dimensionale Vektor $\boldsymbol{u}$ die so genannten Erhaltungsgrößen, die zugleich die Zustandsvariablen des Systems darstellen. Die ebenso n-dimensionale Funktion $\boldsymbol{F}$ bezeichnet man als Flussfunktion, die rechte Seite $\boldsymbol{B}$ wird als Quellterm tituliert.

Man spricht bei $\boldsymbol{u}$ von einer Erhaltungsgröße, wenn alle Komponenten u_i folgende Eigenschaft erfüllen:

$$\int_{\mathbb{R}} u_i(z,t)\, dz = const_i$$

Eine mögliche Anforderung an das System 4.1 ist, dass es für $z \in \mathbb{R}$ und für $t \geq 0$ erfüllt ist. Um das Problem vollständig zu definieren, ist dann nur mehr eine Anfangsbedingung der Form

$$\boldsymbol{u}(z,0) = \boldsymbol{u}_0(z), \quad z \in \mathbb{R} \tag{4.2}$$

notwendig. In diesem Fall spricht man von einem Cauchy Problem.

Definition (hyperbolisch)
Das System 4.1 heißt hyperbolisch *, wenn die Flussfunktion $\boldsymbol{F}(\boldsymbol{u})$ stetig differenzierbar und die Jacobi Matrix $D\boldsymbol{F}(\boldsymbol{u})/D\boldsymbol{u}$ der Dimension $n \times n$ diagonalisierbar ist und nur reelle Eigenwerte besitzt.*
Das System 4.1 heißt streng hyperbolisch *, wenn die Eigenwerte $\lambda_i,\ i = 1, \ldots, n$ der Jacobi Matrix verschieden sind.*

Als Nächstes sei ein Blick auf die so genannten charakteristischen Kurven bzw. auf Charakteristiken geworfen, die eine spezifische Eigenschaft von hyperbolischen partiellen Differentialgleichungen sind. Sei das System von der Form 4.1, wobei die Ortskoordinate z auf dem Bereich Ω und die Zeitkoordinate t auf dem Intervall $(0,T)$ definiert ist. Unter diesen Voraussetzungen sind die charakteristische Kurven Raum-Zeit Kurven, die bestimmen, wie die Werte der Rand- und Anfangsbedingungen durch den Raum-Zeit Zylinder $Q_T = \Omega \times (0,T)$ verbreitet werden.

Um die Grundgedanken möglichst einfach darlegen zu können, sollen im Folgenden nur lineare Systeme betrachtet werden, am Beginn sogar beschränkt auf ein eindimensionales System der Form

$$u_t + a u_z = 0 \quad \forall\, z \in \mathbb{R}, t > 0. \tag{4.3}$$

Für eine vollständige Problemstellung benötigt man noch die Anfangsbedingung

$$u(z,0) = u_0(z) \quad \forall\, z \in \mathbb{R}. \tag{4.4}$$

Definition (Charakteristik)
Unter einer Charakteristik *bzw. einer* charakteristischen Kurve *der Gleichung 4.3, die durch den Punkt $(z_0, 0)$ geht, versteht man den Graphen der Lösung der gewöhnlichen Differentialgleichung*

$$z'(t) = a, \quad z(0) = z_0 \quad \forall\, t > 0. \tag{4.5}$$

Die Lösungen von 4.3 zusammen mit den Anfangsbedingungen sind konstant entlang den Charakteristiken $z(t)$ und vollständig durch die Anfangsbedingungen bestimmt, nämlich als

$$u(z,t) = u_0(z - at). \tag{4.6}$$

Man kann also sagen, dass die Gleichung 4.3 keine neuen Informationen hervorbringt, sondern nur einen Fluss beschreibt. Für $a > 0$ werden die Anfangsbedingungen nach rechts verschoben, für $a <$ kommt es zu einer Verschiebung nach links.

Für die Erläuterung des Begriffs der charakteristischen Variablen wird nun ein System von linearen partiellen Differentialgleichungen betrachtet, das von folgender Form ist:

$$\begin{aligned} \boldsymbol{u}_t(z,t) + \boldsymbol{A}\boldsymbol{u}_z(z,t) &= 0 \\ \boldsymbol{u}(z,0) &= \boldsymbol{u}_0(z) \end{aligned} \tag{4.7}$$

dabei ist $\boldsymbol{u} : \mathbb{R} \times \mathbb{R} \to \mathbb{R}^n$ und $\boldsymbol{A} \in \mathbb{R}^n \times \mathbb{R}^n$ eine konstante Matrix.

Ist das System hyperbolisch, so lässt sich die Matrix $\boldsymbol{A}$ diagonalisieren und hat ausschließlich reelle Eigenwerte. Es existiert daher eine nichtsinguläre Matrix $\boldsymbol{R}$, sodass gilt:

$$\boldsymbol{A} = \boldsymbol{R}\boldsymbol{E}_\lambda\boldsymbol{R}^{-1} \tag{4.8}$$

Hier ist $\boldsymbol{E}_\lambda$ eine Diagonalmatrix mit den Eigenwerten als Einträgen. Die Matrix $\boldsymbol{R}$ beinhaltet die rechten Eigenvektoren von $\boldsymbol{A}$ als Spalten.

Multipliziert man das System 4.7 mit $\boldsymbol{R}^{-1}$ von links und berücksichtigt Beziehung 4.8, so erhält man

$$\boldsymbol{R}^{-1}\boldsymbol{u}_t(z,t) + \boldsymbol{E}_\lambda\boldsymbol{R}^{-1}\boldsymbol{u}_z(z,t) = 0. \tag{4.9}$$

Um nun das System zu lösen, kann auf die so genannten charakteristischen Variablen übergegangen werden.

Definition (Charakteristische Variable)
Unter den charakteristischen Variablen $\boldsymbol{w}$ versteht man

$$\boldsymbol{w} = \boldsymbol{R}^{-1}\boldsymbol{u}. \tag{4.10}$$

Dadurch lässt sich das Gleichungssystem 4.9 vereinfacht darstellen als

$$\boldsymbol{w}_t + \boldsymbol{E}_\lambda \boldsymbol{w}_z = 0. \tag{4.11}$$

Da die Matrix $\boldsymbol{E}_\lambda$ diagonal ist, lässt sich das System in n unabhängige lineare Gleichungen aufspalten, die zusammen mit der entsprechenden Anfangsbedingung von folgender Form sind:

$$(w_i)_t + \lambda_i (w_i)_x = 0, \quad w_i(0) = w_{0,i} \tag{4.12}$$

Wie man zuvor gesehen hat, ist die Lösung einer solchen linearen partiellen Differentialgleichung von folgender Form (vgl. 4.6):

$$w(z,t) = w_{0,i}(z - \lambda_i t). \tag{4.13}$$

Damit erhält man als Lösung für das System 4.7:

$$\boldsymbol{u}(z,t) = \boldsymbol{R}\boldsymbol{w}(z,t) = \sum_{i=1}^{n} w_i(z,t)\boldsymbol{r}_i \tag{4.14}$$

Die Lösung ist also eine Superposition von n unabhängigen linearen Wellen, die sich mit der Geschwindigkeit λ_i ausbreiten.

Charakteristische Variable werden bei der Bestimmung der Randbedingungen von einem Zeitschritt auf den nächstem bei der numerischen Lösung des Blutflussmodells eine entscheidende Rolle spielen. Dort sind die Charakteristiken zwar nicht linear, werden aber in erster Näherung als linear angesehen. Die Idee soll nun kurz dargelegt werden.

Es wird angenommen, dass die Lösung bzw. die charakteristischen Variablen eines linearen Systems wie 4.7 zu einem Zeitpunkt t_n für alle Punkte z im betrachteten Abschnitt bekannt sind. Um die Lösung im Zeitpunkt $t_{n+1} = t_n + \Delta t$ an einem bestimmten Ort z_L zu erhalten, ist folgende Vorgangsweise möglich.

$$\begin{aligned} w(z,t_n) =& w_0(z - \lambda t_n) \\ w(z_L,t_{n+1}) =& w_0(z_L - (\lambda(t_n + \Delta t)) = w_0(z_L - \lambda\Delta t - \lambda t_n) = \\ =& w(z_L - \lambda\Delta t, t_n) \end{aligned} \tag{4.15}$$

4.2 Modellherleitung

In diesem Abschnitt wird ein Modell für den Blutfluss in Arterien hergeleitet. Den Ausgangspunkt bilden die Navier-Stokes-Gleichungen. Unter Berücksichtigung

einiger Annahmen wird daraus ein eindimensionales Modell abgeleitet. Auf eine genaue Abhandlung dieser Ableitung wird hier verzichtet, sie kann bei Alfio Quarteroni et al. [10] nachgelesen werden. Neben diesen Ausführungen dient in noch größerem Maße der Artikel „One dimensional models for blood flow in arteries“ von Luca Formaggia et al. [5] als Vorlage für die folgenden Darstellungen. Er kann als Basis für diese Arbeit gesehen werden, daher werden die Bezeichnungen für die auftretenden Größen mehrheitlich von diesem Artikel übernommen.

Wie in [10] beschrieben wird, gibt es verschiedene Möglichkeiten, um zu dem in weiterer Folge benutzten eindimensionalen Modell zu kommen. Es werden hier als Ausgangspunkt die Navier-Stokes-Gleichungen für inkompressible Flüssigkeiten verwendet, die folgendermaßen lauten:

$$\frac{\partial \boldsymbol{u}}{\partial t} + (\boldsymbol{u} \cdot \nabla)\, \boldsymbol{u} + \frac{1}{\rho} \nabla P - \operatorname{div}\left[\nu \left(\nabla \boldsymbol{u} + (\nabla \boldsymbol{u})^T\right)\right] = 0 \tag{4.16}$$

$$\operatorname{div} \boldsymbol{u} = 0 \tag{4.17}$$

Als zugrunde liegender Definitionsbereich dient ein zylindrischer Bereich Ω_t, der sich auf Grund der durch den Fluss verursachten Wandbewegung über die Zeit t verändert. (x, y, z) bezeichnet das kartesische Koordinatensystem. Der Vektor $\boldsymbol{u}$ stellt die Flussgeschwindigkeit dar, somit beschreiben seine drei Komponenten die Flussgeschwindigkeit in x-, y- und z-Richtung. Weiters ist P der Druck, ν die kinematische Viskosität und ρ die Blutdichte.
Die Gleichung 4.16 wird als Momentengleichung bezeichnet, die Gleichung 4.17 als Kontinuitätsgleichung. Diese Namen leiten sich von den physikalischen Eigenschaften ab, welche durch die beiden Gleichungen beschrieben werden. Eine ausführliche Herleitung der Navier-Stokes-Gleichungen findet sich in [10].

Die obigen Gleichungen beruhen schon auf einer sehr wesentlichen Vereinfachung, nämlich dieser, dass eine Arterie oder zumindest ein Arterienstück als zylindrisches verformbares Rohr angenommen wird, wobei die Gefäßwand als flüssigkeitsundurchlässig angesehen wird (siehe dazu Abb. 4.1, aus [5], S.253). Da ist es natürlich naheliegend, als nächsten Schritt zylindrische Koordinaten einzuführen, die mit (r, θ, z) bezeichnet werden.
Weiters unterstellt man dem Gefäß eine konstante Länge L, d.h. in unserem Koordinatensystem hat die Arterie eine Ausdehnung von $z = 0$ bis $z = L$.

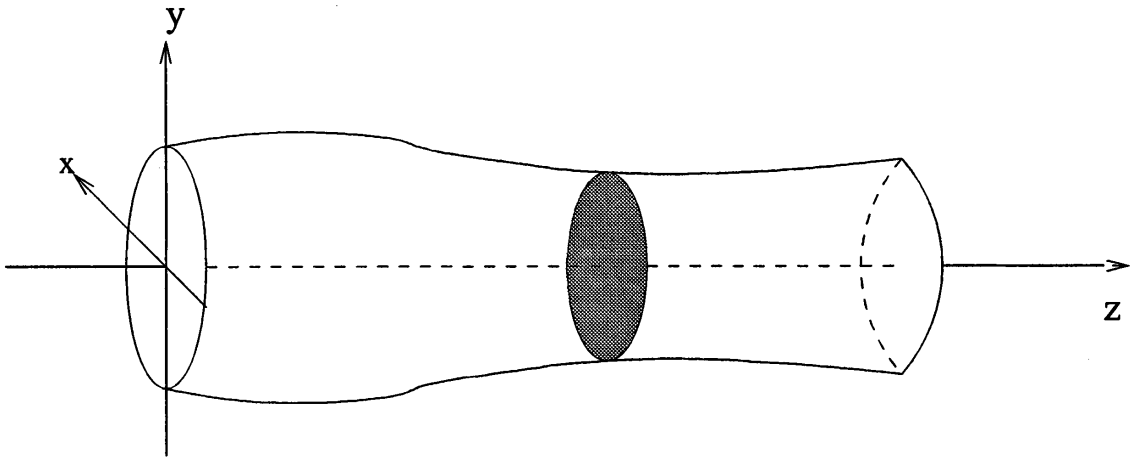

Abbildung 4.1: Darstellung eines Arterienstücks als zylindrisches Rohr (aus [5], S.253)

Bevor das eindimensionale Modell hergeleitet wird, werden noch weitere Annahmen getroffen:

Axiale Symmetrie: Alle auftretenden Größen sind unabhängig vom Winkel θ. Das bedeutet u.a., dass jeder axiale Querschnitt $z =$ const über den ganzen Zeitverlauf kreisförmig bleibt. Der Radius R des Rohrs ist somit eine Funktion des Ortes z und der Zeit t.

Radiale Verschiebung: Die Arterienwand verschiebt sich nur in radialer Richtung. Sei R_0 ein Referenzradius, dann kann die Auslenkung der Gefäßwand durch $\eta = R - R_0$ beschrieben werden.

Konstanter Druck: Der Druck ist auf jedem Querschnitt konstant, d.h. er ist nur von z und t abhängig.

Keine Volumenkräfte: Volumenkräfte wie z.B. die Gravitation werden vernachlässigt.

Dominanz der axialen Geschwindigkeit: Geschwindigkeitskomponenten orthogonal zur z-Achse können vernachlässigt werden im Vergleich zu der Geschwindigkeitskomponente entlang der Hauptausbreitungsrichtung (z-Achse). Bezeichnet man diese mit u_z, so ergibt sich für die Geschwindigkeit:

$$u_z(t, r, z) = \bar{u}(t, z)\, s\left(\frac{r}{R(t, z)}\right) \tag{4.18}$$

Dabei ist $\bar{u}$ die mittlere Geschwindigkeit auf jedem axialen Querschnitt, und die Funktion $s : \mathbb{R} \to \mathbb{R}$ beschreibt ein Geschwindigkeitsprofil.

Eine axiale Querschnittsfläche wird in weiterer Folge mit S bezeichnet, wobei die Zylinderkoordinaten folgende Beschreibung erlauben:

$$S = S(z,t) = \{(r,\theta,z) : 0 \leq r \leq R(z,t),\ 0 \leq \theta < 2\pi\} \tag{4.19}$$

Um nun von einem dreidimensionalen System (bezüglich der Ortskoordinaten) zu einem eindimensionalen System zu kommen, wird über eine allgemeine Querschnittsfläche $S(z,t)$ integriert. Wie sich die Rechenschritte bei den einzelnen Termen der Navier-Stokes-Gleichungen gestalten, kann wiederum aus [10] entnommen werden. Als Ergebnis erhält man das folgende System:

$$\frac{\partial A}{\partial t} + \frac{\partial Q}{\partial z} = 0 \tag{4.20}$$

$$\frac{\partial Q}{\partial t} + \frac{\partial}{\partial z}\left(\alpha \frac{Q^2}{A}\right) + \frac{A}{\rho}\frac{\partial P}{\partial z} + K_R \frac{Q}{A} = 0 \tag{4.21}$$

Diese beiden Gleichungen beinhalten drei Zustandsgrößen:

Querschnittsfläche A:

$$A(z,t) = \int_{S(z,t)} dz \tag{4.22}$$

Mittlerer Volumenfluss Q:

$$Q(z,t) = \int_{S(z,t)} u_z(x,y,z,t)\, dx\, dy = A\bar{u} \tag{4.23}$$

Mittlerer Druck P:

$$P(z,t) = \frac{1}{A(z,t)} \int_{S(z,t)} P(x,y,z,t)\, dx\, dy \tag{4.24}$$

Neben der Blutdichte ρ treten auch noch zwei neue Größen auf:

Coriolis Koeffizient α:

$$\alpha = \frac{\int_S u_z^2\, d\gamma}{A\bar{u}^2} = \frac{\int_S s^2\, d\gamma}{A} \tag{4.25}$$

Im Allgemeinen variiert der Coriolis Koeffizient mit der Zeit bzw. mit dem Ort. In diesem Modell wird er allerdings als konstant angenommen. Einem parabolischen Geschwindigkeitsprofil entspricht ein Coriolis Koeffizient von $\alpha = \frac{4}{3}$. Allerdings kann in Messungen ein etwas flacheres Geschwindigkeitsprofil festgestellt werden, daher wäre ein Wert von $\alpha = 1,1$

sinnvoll. In unserem Modell wird aber $\alpha = 1$ verwendet, was einem flachen Profil entspricht. Der Grund liegt in den sich dadurch ergebenden Modellvereinfachungen.

Widerstandsparameter K_R:

$$K_R = -2\pi\nu s'(1) \tag{4.26}$$

K_R wird auch als Reibungsparameter bezeichnet. Wie α hängt auch K_R vom gewählten Geschwindigkeitsprofil ab. Der normalerweise verwendete Wert ist $K_R = 8\pi\nu$, was einem parabolischen Profil entspricht.

Das betrachtete System besteht aus den beiden Gleichungen 4.20 und 4.21, beinhaltet aber drei Zustandsgrößen A, Q und P. Um das System lösen zu können, braucht man daher eine dritte Gleichung. Diese bekommt man aus einem mechanischen Modell für die Verschiebung der Gefäßwand. In [10] beschäftigt sich ein ganzes Kapitel mit dieser Thematik. Es wird hier der Zusammenhang zwischen Druck und Fläche mit dieser algebraischen Gleichung beschrieben:

$$P - P_{ext} = \beta \frac{\sqrt{A} - \sqrt{A_0}}{A_0} \tag{4.27}$$

Auch in dieser Gleichung treten wieder einige neue Parameter auf:

Referenzfläche A_0: Dieser Wert entspricht jener Querschnittsfläche, die das Rohr mit dem Referenzradius R_0 aufweist. Es gilt also $A_0 = R_0^2\pi$. Bei den Berechnungen werden R_0 bzw. A_0 mit den Anfangswerten belegt.

β:

$$\beta = Eh_0\sqrt{\pi} \tag{4.28}$$

Der Parameter β ist ebenso wie A_0 eine Funktion des Ortes z, dagegen wird angenommen, dass sich beide nicht über die Zeit ändern.

Elastizitätsmodul (Youngscher Modul) E: Der Elastizitätsmodul beschreibt die Steifigkeit der Gefäßwand. Eine steifere Wand entspricht dabei einem größeren Wert von E.

Wandstärke h_0: Dieser Parameter gibt an, wie dick die Gefäßwand in einem Referenzzustand ist.

Externer Druck P_{ext}: In weiterer Folge wird der externe Druck o. B. d. A. mit $P_{ext} = 0$ angenommen.

Zur Vereinfachung der weiteren Darstellung wird noch die Größe c_1 eingeführt:

$$c_1 = \sqrt{\frac{\beta}{2\rho A_0}} A^{\frac{1}{4}} \tag{4.29}$$

Als Nächstes wird nun den Druck P aus den Systemgleichungen eliminiert. Daher ersetzt man P in Gleichung 4.21 durch die Beziehung 4.27. Man erhält dadurch ein partielles Differentialgleichungssystem mit den beiden Zustandsgrößen A und Q. Nach einigen Umformungen kann es in Erhaltungsform folgendermaßen dargestellt werden:

$$\boldsymbol{U} = \begin{pmatrix} A \\ Q \end{pmatrix} \tag{4.30}$$

$$\frac{\partial \boldsymbol{U}}{\partial t} + \frac{\partial \boldsymbol{F}}{\partial z}(\boldsymbol{U}) = \boldsymbol{B}(\boldsymbol{U}), \quad z \in (0, L),\ t > 0 \tag{4.31}$$

$$\boldsymbol{F}(\boldsymbol{U}) = \begin{pmatrix} Q \\ \alpha \frac{Q^2}{A} + \frac{\beta}{3\rho A_0} A^{\frac{3}{2}} \end{pmatrix} \tag{4.32}$$

$$\boldsymbol{B}(\boldsymbol{U}) = \begin{pmatrix} 0 \\ -K_R \frac{Q}{A} - \frac{A}{A_0 \rho}\left(\frac{2}{3}A^{\frac{1}{2}} - A_0^{\frac{1}{2}}\right)\frac{\partial \beta}{\partial z} + \frac{\beta}{\rho}\frac{A}{A_0^2}\left(\frac{2}{3}A^{\frac{1}{2}} - \frac{1}{2}A_0^{\frac{1}{2}}\right)\frac{\partial A_0}{\partial z} \end{pmatrix} \tag{4.33}$$

Die Jacobi Matrix des Flussvektors $\boldsymbol{F}$, die in weiterer Folge noch mehrmals benötigt wird, berechnet sich leicht als

$$\boldsymbol{H}(\boldsymbol{U}) = \frac{\partial \boldsymbol{F}}{\partial \boldsymbol{U}} = \begin{pmatrix} 0 & 1 \\ -\alpha \frac{Q^2}{A^2} + \frac{\beta}{2\rho A_0} A^{\frac{1}{2}} & 2\alpha \frac{Q}{A} \end{pmatrix}. \tag{4.34}$$

Wie in [10] nachgerechnet wird, besitzt das System 4.31 für $A > 0$ zwei verschiedene reelle Eigenwerte, es ist daher ein streng hyperbolisches partielles Differentialgleichungssystem (vgl. Definition S. 33).
Die beiden Eigenwerte lauten folgendermaßen:

$$\lambda_{1,2} = \alpha \frac{Q}{A} \pm \sqrt{c_1^2 + \alpha(\alpha - 1)\frac{Q^2}{A^2}} \tag{4.35}$$

Setzt man nun $\alpha = 1$, so kann man die charakteristischen Variablen $W_{1,2}$, wie in [10] beschrieben, berechnen.

$$W_{1,2} = \frac{Q}{A} \pm 4\sqrt{\frac{\beta}{2\rho A_0}} A^{\frac{1}{4}} \tag{4.36}$$

Diese Beziehung zwischen den charakteristischen Variablen und den Zustandsgrößen kann nun auch invertiert werden, um A und Q aus W_1 und W_2 berechnen zu können. Bei der Berechnung der Randbedingungen wird von diesem Umstand Gebrauch gemacht.

$$A = \left(\frac{2\rho A_0}{\beta}\right)^2 \left(\frac{W_1 - W_2}{8}\right)^4 \tag{4.37}$$

$$Q = A\frac{W_1 + W_2}{2} \tag{4.38}$$

Damit sind alle wichtigen Größen des Modells hergeleitet. Bereits im nächsten Abschnitt, wo die Diskretisierung des Modells in Angriff genommen wird, wird auf viele der angeführten Gleichungen wieder zurückgegriffen.

4.3 Diskretisierung

Im vorigen Unterkapitel wurde ein eindimensionales Blutflussmodell hergeleitet. Um diese Gleichungen zu lösen, muss ein numerisches Verfahren angewandt werden. Es stehen verschiedene Methoden zur Verfügung, so werden für Problemstellungen dieser Art oft Finite Volumen Verfahren benutzt. Dagegen verwenden Sherwin et al. in [12] unter anderem eine unstetige Galerkin Methode. Hier wird eine Finite Elemente Methode, genauer ein Taylor-Galerkin Verfahren zweiter Ordnung benutzt, wie es auch in [5] verwendet wird. Die dortigen Ausführungen dienen auch wieder als Richtlinie für die kommenden Überlegungen.

4.3.1 Taylorentwicklung

Zunächst soll die Taylorentwicklung zweiter Ordnung nach der Zeit t von $\boldsymbol{U}$ aufgestellt werden. Dazu werden die ersten beiden Ableitungen von $\boldsymbol{U}$ benötigt. Die erste Ableitung nach t erhält man, indem der zweite Term in der Gleichung 4.31 auf die andere Seite gebracht wird.

$$\frac{\partial \boldsymbol{U}}{\partial t} = \boldsymbol{B} - \frac{\partial \boldsymbol{F}}{\partial z} \tag{4.39}$$

Für die zweite Ableitung nach der Zeit muss dieser Ausdruck nun nochmals differenziert werden. Dabei wird u.a. die Kettenregel angewendet, wo der Ausdruck

$\boldsymbol{B_U} = \frac{\partial \boldsymbol{B}}{\partial \boldsymbol{U}}$ seinen Ursprung hat.

$$\frac{\partial^2 \boldsymbol{U}}{\partial t^2} = \boldsymbol{B_U} \frac{\partial \boldsymbol{U}}{\partial t} - \frac{\partial}{\partial z}\left(\boldsymbol{H}\frac{\partial \boldsymbol{U}}{\partial t}\right) \tag{4.40}$$

Um die ersten Ableitungen von $\boldsymbol{U}$ zu eliminieren, setzt man nun 4.39 in die rechte Seite der obigen Gleichung ein. Dadurch erhält man:

$$\frac{\partial^2 \boldsymbol{U}}{\partial t^2} = \boldsymbol{B_U}\left(\boldsymbol{B} - \frac{\partial \boldsymbol{F}}{\partial z}\right) - \frac{\partial(\boldsymbol{H}\boldsymbol{B})}{\partial z} + \frac{\partial}{\partial z}\left(\boldsymbol{H}\frac{\partial \boldsymbol{F}}{\partial z}\right) \tag{4.41}$$

Die Zeitachse wird in Zeitschritte der Länge Δt diskretisiert. Nach n Zeitschritten ist also der Zeitpunkt $t^n = n\Delta t$ erreicht. Um den Wert von $\boldsymbol{U}$ zu einem nächsten Zeitpunkt t^{n+1} zu erhalten, verwendet man eine Taylorentwicklung zweiter Ordnung.

$$\boldsymbol{U}^{n+1} = \boldsymbol{U}^n + \Delta t \frac{\partial \boldsymbol{U}^n}{\partial t} + \frac{\Delta t^2}{2}\frac{\partial^2 \boldsymbol{U}^n}{\partial t^2} \tag{4.42}$$

Ersetzt man die auftretenden Ableitungen durch die Ausdrücke in den Gleichungen 4.39 und 4.41, so erhält man:

$$\begin{aligned}\boldsymbol{U}^{n+1} = \boldsymbol{U}^n + \Delta t &\left[\boldsymbol{B}^n - \frac{\partial \boldsymbol{F}^n}{\partial z}\right] + \\ &+ \frac{\Delta t^2}{2}\left[\boldsymbol{B}^n_{\boldsymbol{U}}\left(\boldsymbol{B}^n - \frac{\partial \boldsymbol{F}^n}{\partial z}\right) - \frac{\partial(\boldsymbol{H}^n\boldsymbol{B}^n)}{\partial z} + \frac{\partial}{\partial z}\left(\boldsymbol{H}^n\frac{\partial \boldsymbol{F}^n}{\partial z}\right)\right]\end{aligned} \tag{4.43}$$

Dabei steht $\boldsymbol{F}^n$ für $\boldsymbol{F}(\boldsymbol{U}^n)$, das gilt analog für die restlichen Ausdrücke. Mit Hilfe dieses semi-diskreten Systems erhält man eine Approximation $\boldsymbol{U}^n$ für den Funktionswert von $\boldsymbol{U}$ zum Zeitpunkt t^n, also $\boldsymbol{U}(t^n) \approx \boldsymbol{U}^n$.

Für die weitere Lösung wird folgende Darstellung von $\boldsymbol{U}^{n+1}$ verwendet, die aus 4.43 durch eine Umordnung der Terme hervorgeht.

$$\begin{aligned}\boldsymbol{U}^{n+1} = \boldsymbol{U}^n - \Delta t \frac{\partial}{\partial z}&\left[\boldsymbol{F}^n + \frac{\Delta t}{2}\boldsymbol{H}^n\boldsymbol{B}^n\right] - \\ &- \frac{\Delta t^2}{2}\left[\boldsymbol{B}^n_{\boldsymbol{U}}\frac{\partial \boldsymbol{F}^n}{\partial z} - \frac{\partial}{\partial z}\left(\boldsymbol{H}^n\frac{\partial \boldsymbol{F}^n}{\partial z}\right)\right] + \Delta t\left(\boldsymbol{B}^n + \frac{\Delta t}{2}\boldsymbol{B}^n_{\boldsymbol{U}}\boldsymbol{B}^n\right)\end{aligned} \tag{4.44}$$

4.3.2 Galerkin-Verfahren

Nachdem im vorigen Abschnitt eine Diskretisierung bezüglich der Zeit durchgeführt wurde und daher mit 4.44 ein semi-diskretes System vorliegt, soll nun

auch eine Diskretisierung bezüglich des Ortes vorgenommen werden. Das Galerkin-Verfahren wird dazu im Rahmen der Finiten Elemente Methode verwendet. Eine Einführung in die Thematik bietet das Kapitel 2, das Galerkin-Verfahren im Besonderen ist dabei im Unterkapitel 2.3 beschrieben.

Der Ort wird diskretisiert, indem man das Intervall $[0, L]$ in $N-1$ Elemente $[z_j, z_{j+1}]$ unterteilt. Die Länge eines Elements ist somit $h_j = z_{j+1} - z_j$.
Der Raum der stückweise linearen Funktionen soll mit V_h bezeichnet werden. Da eine Diskretisierung für beide Zustandsgrößen A und Q durchgeführt werden muss, werden folgende Räume eingeführt:

$$\begin{aligned}\boldsymbol{V}_h =& V_h^2\\ \boldsymbol{V}_h^0 =& \{\boldsymbol{v}_h \in \boldsymbol{V}_h |\ \boldsymbol{v}_h = \boldsymbol{0} \text{ für } z = 0,\ z = L\}\end{aligned}$$

Das Skalarprodukt wird über eine Integration realisiert und durch $(.,.)$ dargestellt:

$$(\boldsymbol{u}, \boldsymbol{v}) = \int_0^L \boldsymbol{u}\,\boldsymbol{v}\,dz$$

Um die weiteren Darstellungen übersichtlicher gestalten zu können, werden noch folgende Bezeichnungen eingeführt.

$$\begin{aligned}\boldsymbol{F}_{LW}(\boldsymbol{U}) &= \boldsymbol{F}(\boldsymbol{U}) + \frac{\Delta t}{2}\boldsymbol{H}(\boldsymbol{U})\boldsymbol{B}(\boldsymbol{U})\\ \boldsymbol{B}_{LW}(\boldsymbol{U}) &= \boldsymbol{B}(\boldsymbol{U}) + \frac{\Delta t}{2}\boldsymbol{B}_{\boldsymbol{U}}(\boldsymbol{U})\boldsymbol{B}(\boldsymbol{U})\end{aligned}$$

Bei der Finiten Elemente Methode sucht man nach einer so genannten schwachen Lösung (vgl. Kapitel 2.2). Dazu multipliziert man Gleichung 4.44 mit einer Testfunktion $\varphi_h \in \boldsymbol{V}_h^0$ und bildet mittels Integration das sich daraus ergebende Skalarprodukt.

$$\begin{aligned}\left(\boldsymbol{U}_h^{n+1}, \varphi_h\right) =& (\boldsymbol{U}_h^n, \varphi_h) - \Delta t\left(\frac{\partial}{\partial z}\boldsymbol{F}_{LW}(\boldsymbol{U}_h^n), \varphi_h\right) - \frac{\Delta t^2}{2}\left(\boldsymbol{B}_{\boldsymbol{U}}(\boldsymbol{U}_h^n)\frac{\partial \boldsymbol{F}(\boldsymbol{U}_h^n)}{\partial z}, \varphi_h\right) +\\ &+ \frac{\Delta t^2}{2}\left(\frac{\partial}{\partial z}\left(\boldsymbol{H}(\boldsymbol{U}_h^n)\frac{\partial \boldsymbol{F}(\boldsymbol{U}_h^n)}{\partial z}\right), \varphi_h\right) + \Delta t\left(\boldsymbol{B}_{LW}(\boldsymbol{U}_h^n), \varphi_h\right)\end{aligned} \tag{4.45}$$

Der erste und der vierte Term werden nun noch durch eine partielle Integration modifiziert. Dies soll mit einem der beiden Ausdrücke erläutert werden, beim

zweiten sind die Rechenschritte analog.

$$\begin{aligned}\left(\frac{\partial}{\partial z}\boldsymbol{F}_{LW}(\boldsymbol{U}_h^n), \varphi_h\right) &= \int_0^L \frac{\partial}{\partial z}\boldsymbol{F}_{LW}(\boldsymbol{U}_h^n)\,\varphi_h\,dz = \\ &= \boldsymbol{F}_{LW}(\boldsymbol{U}_h^n)\,\varphi_h|_0^L - \int_0^L \boldsymbol{F}_{LW}(\boldsymbol{U}_h^n)\,\frac{d\varphi_h}{dz}\,dz \\ &= -\int_0^L \boldsymbol{F}_{LW}(\boldsymbol{U}_h^n)\,\frac{d\varphi_h}{dz}\,dz, \quad \text{weil } \varphi_h(0) = \varphi_h(L) = 0 \\ &= \left(\boldsymbol{F}_{LW}(\boldsymbol{U}_h^n), \frac{d\varphi_h}{dz}\right)\end{aligned} \tag{4.46}$$

Somit kann das Problem für die approximierte Lösung formuliert werden. Gesucht ist also ein $\boldsymbol{U}_h^{n+1} \in \boldsymbol{V}_h$, das für alle $\varphi_h \in \boldsymbol{V}_h^0$ die folgenden Gleichungen sowie die zugehörigen Randbedingungen erfüllt.

$$\begin{aligned}\left(\boldsymbol{U}_h^{n+1}, \varphi_h\right) = (\boldsymbol{U}_h^n, \varphi_h) + \Delta t\left(\boldsymbol{F}_{LW}(\boldsymbol{U}_h^n), \frac{d\varphi_h}{dz}\right) - \frac{\Delta t^2}{2}\left(\boldsymbol{B}_{\boldsymbol{U}}(\boldsymbol{U}_h^n)\frac{\partial \boldsymbol{F}(\boldsymbol{U}_h^n)}{\partial z}, \varphi_h\right) + \\ + \frac{\Delta t^2}{2}\left(\boldsymbol{H}(\boldsymbol{U}_h^n)\frac{\partial \boldsymbol{F}(\boldsymbol{U}_h^n)}{\partial z}, \frac{d\varphi_h}{dz}\right) + \Delta t\,(\boldsymbol{B}_{LW}(\boldsymbol{U}_h^n), \varphi_h)\end{aligned} \tag{4.47}$$

Im folgenden Abschnitt wird diese Gleichung noch ausführlicher betrachtet. Dabei wird jedes Skalarprodukt genauer unter die Lupe genommen, um als nächsten Schritt die tatsächliche Implementierung und damit die numerische Lösung vornehmen zu können.

Als Basis des Raums $\boldsymbol{V}_h$ der stückweise linearen Funktionen werden Hutfunktionen verwendet (vgl. Kaptiel 2.5). Dadurch kommt man zu einem Gleichungssystem, dessen Lösung eine approximierte Lösung der Modellgleichungen in einem Zeitpunkt darstellt (vgl. Kapitel 2.3).

4.3.3 Berechnung der Matrizen

Da der Vektor $\boldsymbol{U}_h^{n+1}$ aus den beiden Komponenten A_h^{n+1} und Q_h^{n+1} besteht, müssen zwei Gleichungssysteme gelöst werden. Die Matrix der linken Seite ist aber für beide lineare Gleichungssysteme ident, weil für beide Komponenten die selbe Diskretisierung verwendet wird.

Die Hutfunktionen haben die Eigenschaft, dass sich ihr Träger nur über ein oder zwei Elemente der Zerlegung erstreckt. Es ist daher naheliegend, die auftretenden Skalarprodukte auf jedem Element einzeln zu berechnen und die Ergebnisse

entsprechend zu summieren. Die auftretenden Skalarprodukte sind in der Definition der Steifigkeitsmatrix auf Seite 17 genauer erläutert. Über die genaue Vorgangsweise bei der Berechnung gibt Kapitel 3.3 Auskunft.

Auf einem Element K_j überschneiden sich immer zwei Hutfunktionen. Bezeichnet man die Matrix der linken Seite des Gleichungssystems für A_h^{n+1} mit $\boldsymbol{S}$, so ergeben sich folgende Einträge (dabei steht φ_i für eine Basisfunktion in V_h):

$$
\begin{aligned}
\boldsymbol{S}_{j\,j}(K_j) &= \int_{K_j} \varphi_j(z)\varphi_j(z)\,dz = \\
&= \int_{K_j} \frac{z_{j+1}-z}{h_j}\frac{z_{j+1}-z}{h_j}\,dz = \frac{1}{h_j^2}\int_{K_j}(z_{j+1}-z)^2\,dz \\
\boldsymbol{S}_{j\,j+1}(K_j) &= \int_{K_j} \varphi_j(z)\varphi_{j+1}(z)\,dz = \\
&= \int_{K_j} \frac{z_{j+1}-z}{h_j}\frac{z-z_j}{h_j}\,dz = \frac{1}{h_j^2}\int_{K_j}(z_{j+1}-z)(z-z_j)\,dz \\
\boldsymbol{S}_{j+1\,j}(K_j) &= \int_{K_j} \varphi_{j+1}(z)\varphi_j(z)\,dz \\
&= \int_{K_j} \frac{z-z_j}{h_j}\frac{z_{j+1}-z}{h_j}\,dz = \frac{1}{h_j^2}\int_{K_j}(z_{j+1}-z)(z-z_j)\,dz \\
\boldsymbol{S}_{j+1\,j+1}(K_j) &= \int_{K_j} \varphi_{j+1}(z)\varphi_{j+1}(z)\,dz = \\
&= \int_{K_j} \frac{z-z_j}{h_j}\frac{z-z_j}{h_j}\,dz = \frac{1}{h_j^2}\int_{K_j}(z-z_j)^2\,dz
\end{aligned}
$$

Summiert man alle so berechneten Einträge, erhält man die Matrix $\boldsymbol{S}$ der linken Seite, die für alle Zeitschritte gleich bleibt und daher nur einmal berechnet werden muss.

Um zu einer Matrix $\boldsymbol{T}$ der linken Seite für die Komponente Q_h^{n+1} zu gelangen, sind die genau gleichen Überlegungen und Rechnungen durchzuführen. Es gilt daher $\boldsymbol{T} = \boldsymbol{S}$.

4.3.4 Berechnung des Lastvektors

Nach der linken Seite sollen nun die einzelnen Terme der rechten Seite der Gleichung 4.47 betrachtet werden. Die Skalarprodukte werden im Folgenden jeweils in ihrer Integralform dargestellt und die auftretenden Größen dabei genau ausgeführt.

Wie zuvor werden die Überlegungen für ein allgemeines Intervall K_j durchgeführt. Daraus ergibt sich ein Eintrag in der j-ten und $j+1$-ten Zeile des Lastvektors. Die vollständige rechte Seite erhält man, indem man die jeweiligen Ergebnisse der Skalarprodukte mit den dazugehörigen Zeitschritten multipliziert und die Terme danach summiert.

Die Basisfunktionen von $\boldsymbol{V}_h$ sowie von V_h werden jeweils mit φ_i bezeichnet. Es sollte im jeweiligen Zusammenhang klar sein, welche Basis gemeint ist.
Im Sinne der Übersichtlichkeit wird auf die Kennzeichnung des Zeitpunkts verzichtet. Es liegt allen kommenden Berechnungen die Annahme zu Grunde, dass vom Zeitpunkt t^n ausgehend mit einem Zeitschritt Δt die Werte für den Zeitpunkt t^{n+1} berechnet werden.

Die Ergebnisse der Berechnungen der rechten Seite werden in einem Vektor $\boldsymbol{f}$ zusammengefasst. Dabei muss wiederum darauf Rücksicht genommen werden, dass $\boldsymbol{U}$ aus den beiden Komponenten A und Q besteht. Daher wird zwischen den beiden Vektoren ${}_A\boldsymbol{f}$ und ${}_Q\boldsymbol{f}$ unterschieden.

Die Gleichung 4.47 besteht auf der rechten Seite aus fünf Skalarprodukten. Diese fünf Teile werden im Folgenden einzeln betrachtet. Dies wird beim Vektor $\boldsymbol{f}$ durch eine weitere Indizierung festgehalten, wir unterscheiden also zwischen ${}^k\boldsymbol{f},\ k = 1, \ldots, 5$.

Es wird nun angenommen, dass für den Zeitpunkt t^n die linear approximierte Lösung bereits berechnet wurde. Es wird zunächst nur die Komponente A betrachtet. Der Koeffizient zur Basisfunktion φ_j stimmt auf dem Knoten z_j mit dem Funktionswert von A_h überein. Er wird in den weiteren Darstellungen mit $A_j = A_j^n$ bezeichnet. Auf einem Element K_j ist die Funktion A_h linear und kann mit Hilfe der Basisfunktion folgendermaßen dargestellt werden.

$$A_h = A_j\varphi_j + A_{j+1}\varphi_{j+1} \quad \text{auf } K_j$$

Analoge Überlegungen und Darstellungskonventionen gelten auch für Q.

Den Beginn macht nun das erste Skalarprodukt der rechten Seite von 4.47:

$$(\boldsymbol{U}, \varphi)$$

Es folgt:

$$
\begin{aligned}
{}^1_A\boldsymbol{f}_j(K_j) &= \int_{K_j} \left(A_j\varphi_j(z) + A_{j+1}\varphi_{j+1}(z)\right)\varphi_j\, dz = \\
&= \int_{K_j} \left(A_j\frac{z_{j+1}-z}{h_j} + A_{j+1}\frac{z-z_j}{h_j}\right)\frac{z_{j+1}-z}{h_j}\, dz \qquad (4.48)\\
{}^1_A\boldsymbol{f}_{j+1}(K_j) &= \int_{K_j} \left(A_j\varphi_j(z) + A_{j+1}\varphi_{j+1}(z)\right)\varphi_{j+1}\, dz = \\
&= \int_{K_j} \left(A_j\frac{z_{j+1}-z}{h_j} + A_{j+1}\frac{z-z_j}{h_j}\right)\frac{z-z_j}{h_j}\, dz \qquad (4.49)
\end{aligned}
$$

$$
\begin{aligned}
{}^1_Q\boldsymbol{f}_j(K_j) &= \int_{K_j} \left(Q_j\varphi_j(z) + Q_{j+1}\varphi_{j+1}(z)\right)\varphi_j(z)\, dz = \\
&= \int_{K_j} \left(Q_j\frac{z_{j+1}-z}{h_j} + Q_{j+1}\frac{z-z_j}{h_j}\right)\frac{z_{j+1}-z}{h_j}\, dz \qquad (4.50)\\
{}^1_Q\boldsymbol{f}_{j+1}(K_j) &= \int_{K_j} \left(Q_j\varphi_j(z) + Q_{j+1}\varphi_{j+1}(z)\right)\varphi_{j+1}(z)\, dz = \\
&= \int_{K_j} \left(Q_j\frac{z_{j+1}-z}{h_j} + Q_{j+1}\frac{z-z_j}{h_j}\right)\frac{z-z_j}{h_j}\, dz \qquad (4.51)
\end{aligned}
$$

Das zweite Skalarprodukt ist bereits von etwas komplexerer Form:

$$\left(\boldsymbol{F}_{LW}(\boldsymbol{U}), \frac{d\varphi}{dz}\right) \qquad (4.52)$$

$$\boldsymbol{F}_{LW}(\boldsymbol{U}) = \boldsymbol{F}(\boldsymbol{U}) + \frac{\Delta t}{2}\boldsymbol{H}(\boldsymbol{U})\boldsymbol{B}(\boldsymbol{U})$$

Die in dieser Gleichung auftretenden Terme $\boldsymbol{F}$, $\boldsymbol{H}$ und $\boldsymbol{B}$ wurden in Kapitel 4.2 hergeleitet.

$$\boldsymbol{F}(\boldsymbol{U}) = \begin{pmatrix} Q \\ \alpha\frac{Q^2}{A} + \frac{\beta}{3\rho A_0}A^{\frac{3}{2}} \end{pmatrix} \qquad (4.53)$$

$$\boldsymbol{H}(\boldsymbol{U}) = \begin{pmatrix} 0 & 1 \\ -\alpha\frac{Q^2}{A^2} + \frac{\beta}{2\rho A_0}A^{\frac{1}{2}} & 2\alpha\frac{Q}{A} \end{pmatrix} \qquad (4.54)$$

$$\boldsymbol{B}(\boldsymbol{U}) = \begin{pmatrix} 0 \\ -K_R\frac{Q}{A} - \frac{A}{A_0\rho}\left(\frac{2}{3}A^{\frac{1}{2}} - A_0^{\frac{1}{2}}\right)\frac{\partial\beta}{\partial z} + \frac{\beta}{\rho}\frac{A}{A_0^2}\left(\frac{2}{3}A^{\frac{1}{2}} - \frac{1}{2}A_0^{\frac{1}{2}}\right)\frac{\partial A_0}{\partial z} \end{pmatrix} \qquad (4.55)$$

Um den Ausdruck $\boldsymbol{B}$ zu vereinfachen, wird in der Folge angenommen, dass A_0 sowie β konstant entlang des ganzen Arterienstücks sind. Dadurch erhält man

für $\boldsymbol{B}$:

$$\boldsymbol{B}(\boldsymbol{U}) = \begin{pmatrix} 0 \\ -K_R \frac{Q}{A} \end{pmatrix} \tag{4.56}$$

Multipliziert man die Jacobi Matrix $\boldsymbol{H}$ mit dem vereinfachten Vektor $\boldsymbol{B}$, so ergibt das:

$$\boldsymbol{H}(\boldsymbol{U})\,\boldsymbol{B}(\boldsymbol{U}) = \begin{pmatrix} -K_R \frac{Q}{A} \\ -2\alpha K_R \frac{Q^2}{A^2} \end{pmatrix} \tag{4.57}$$

Zusammen mit $\boldsymbol{F}$ erhält man daraus für $\boldsymbol{F}_{LW}$:

$$\boldsymbol{F}_{LW}(\boldsymbol{U}) = \begin{pmatrix} Q - \frac{\Delta t}{2} K_R \frac{Q}{A} \\ \alpha \frac{Q^2}{A} + \frac{\beta}{3\rho A_0} A^{\frac{3}{2}} - \Delta t \alpha K_R \frac{Q^2}{A^2} \end{pmatrix}$$

Wird nun dieser Ausdruck wieder in 4.52 eingesetzt, können die beiden Komponenten des Skalarprodukts folgendermaßen berechnet werden:

$$\begin{aligned} {}^2_A\boldsymbol{f}_j(K_j) &= \int_{K_j} \Bigg(Q_j\varphi_j(z) + Q_{j+1}\varphi_{j+1}(z) - \\ &\qquad - \frac{\Delta t}{2} K_R \frac{Q_j\varphi_j(z) + Q_{j+1}\varphi_{j+1}(z)}{A_j\varphi_j(z) + A_{j+1}\varphi_{j+1}(z)} \Bigg) \varphi_j'(z)dz = \\ &= \int_{K_j} \Bigg(Q_j \frac{z_{j+1} - z}{h_j} + Q_{j+1} \frac{z - z_j}{h_j} - \\ &\qquad - \frac{\Delta t}{2} K_R \frac{Q_j(z_{j+1} - z) + Q_{j+1}(z - z_j)}{A_j(z_{j+1} - z) + A_{j+1}(z - z_j)} \Bigg) \left(\frac{-1}{h_j} \right) dz \end{aligned} \tag{4.58}$$

$${}^2_A\boldsymbol{f}_{j+1}(K_j) = -\,{}^2_A\boldsymbol{f}_j(K_j) \tag{4.59}$$

$$\begin{aligned} {}^2_Q\boldsymbol{f}_j(K_j) &= \int_{K_j} \Bigg(\alpha \frac{(Q_j\varphi_j(z) + Q_{j+1}\varphi_{j+1}(z))^2}{A_j\varphi_j(z) + A_{j+1}\varphi_{j+1}(z)} + \frac{\beta}{3\rho A_0} (A_j\varphi_j(z) + A_{j+1}\varphi_{j+1}(z))^{\frac{3}{2}} - \\ &\qquad - \Delta t\, \alpha K_R \frac{(Q_j\varphi_j(z) + Q_{j+1}\varphi_{j+1}(z))^2}{(A_j\varphi_j(z) + A_{j+1}\varphi_{j+1}(z))^2} \Bigg) \varphi_j'(z)dz = \\ &= \int_{K_j} \Bigg(\alpha \frac{1}{h_j} \frac{(Q_j(z_{j+1} - z) + Q_{j+1}(z - z_j))^2}{A_j(z_{j+1} - z) + A_{j+1}(z - z_j)} + \\ &\qquad + \frac{\beta}{3\rho A_0} \left(\frac{A_j(z_{j+1} - z) + A_{j+1}(z - z_j)}{h_j} \right)^{\frac{3}{2}} - \\ &\qquad - \Delta t\, \alpha K_R \frac{(Q_j(z_{j+1} - z) + Q_{j+1}(z - z_j))^2}{(A_j(z_{j+1} - z) + A_{j+1}(z - z_j))^2} \Bigg) \left(\frac{-1}{h_j} \right) dz \end{aligned} \tag{4.60}$$

$${}^2_Q\boldsymbol{f}_{j+1}(K_j) = -\,{}^2_Q\boldsymbol{f}_j(K_j) \tag{4.61}$$

Jetzt soll das dritte Skalarprodukt betrachtet werden:

$$\left(\boldsymbol{B_U}(\boldsymbol{U})\frac{\partial \boldsymbol{F}(\boldsymbol{U})}{\partial z}, \varphi\right)$$

Dabei verwendet man für $\boldsymbol{B}$ wieder die vereinfachte Form 4.56. Differenziert man nun den Vektor $\boldsymbol{B}$ nach den beiden Zustandsgrößen A und Q, so erhält man folgende Jacobi Matrix $\boldsymbol{B_U}$.

$$\boldsymbol{B_U} = \begin{pmatrix} 0 & 0 \\ K_R\frac{Q}{A^2} & -K_R\frac{1}{A} \end{pmatrix} \tag{4.62}$$

Weiters muss der Vektor $\boldsymbol{F}$ (siehe Gleichung 4.53) nach dem Ort z differenziert werden.

$$\frac{\partial \boldsymbol{F}}{\partial z} = \begin{pmatrix} Q' \\ \alpha\frac{2QQ'A-Q^2A'}{A^2} + \frac{\beta}{3\rho A_0}\frac{3}{2}A^{\frac{1}{2}}A' \end{pmatrix} \tag{4.63}$$

Diese beiden Terme werden nun miteinander multipliziert.

$$\boldsymbol{B_U}\frac{\partial \boldsymbol{F}}{\partial z} = \begin{pmatrix} 0 \\ K_R\frac{Q}{A^2}Q' - K_R\frac{1}{A}\left(\alpha\frac{2QQ'A-Q^2A'}{A^2} + \frac{\beta}{2\rho A_0}A^{\frac{1}{2}}A'\right) \end{pmatrix} \tag{4.64}$$

Da die erste Komponente nur aus einem Nulleintrag besteht, gilt dies auch für die entsprechenden Skalarprodukte.

$${}^3_A\boldsymbol{f}_j(K_j) = {}^3_A\,\boldsymbol{f}_{j+1}(K_j) = 0 \tag{4.65}$$

Die Berechnungen für die zweite Komponente sind dagegen aufwändiger.

$$
\begin{aligned}
&{}^{3}_{Q}\boldsymbol{f}_j(K_j) = \\
&= \int_{K_j} \Bigg[K_R \frac{Q_j\varphi_j(z) + Q_{j+1}\varphi_{j+1}(z)}{\left(A_j\varphi_j(z) + A_{j+1}\varphi_{j+1}(z)\right)^2} \left(Q_j\varphi_j(z) + Q_{j+1}\varphi_{j+1}(z)\right)' - \\
&\qquad - K_R \frac{1}{A_j\varphi_j(z) + A_{j+1}\varphi_{j+1}(z)} \Bigg(\alpha \frac{Q_j\varphi_j(z) + Q_{j+1}\varphi_{j+1}(z)}{\left(A_j\varphi_j(z) + A_{j+1}\varphi_{j+1}(z)\right)^2} \\
&\qquad\quad \big(2\left(Q_j\varphi_j(z) + Q_{j+1}\varphi_{j+1}(z)\right)' \left(A_j\varphi_j(z) + A_{j+1}\varphi_{j+1}(z)\right) - \\
&\qquad\qquad - \left(Q_j\varphi_j(z) + Q_{j+1}\varphi_{j+1}(z)\right) \left(A_j\varphi_j(z) + A_{j+1}\varphi_{j+1}(z)\right)'\big) + \\
&\qquad + \frac{\beta}{2\rho A_0} \left(A_j\varphi_j(z) + A_{j+1}\varphi_{j+1}(z)\right)^{\frac{1}{2}} \left(A_j\varphi_j(z) + A_{j+1}\varphi_{j+1}(z)\right)' \Bigg) \Bigg] \varphi_j(z)\, dz \\
&= \int_{K_j} \Bigg[K_R \frac{Q_j\left(z_{j+1} - z\right) + Q_{j+1}\left(z - z_j\right)}{\left(A_j\left(z_{j+1} - z\right) + A_{j+1}\left(z - z_j\right)\right)^2} \left(-Q_j + Q_{j+1}\right) - \qquad (4.66) \\
&\qquad - K_R \frac{1}{A_j\left(z_{j+1} - z\right) + A_{j+1}\left(z - z_j\right)} h_j \Bigg(\alpha \frac{Q_j\left(z_{j+1} - z\right) + Q_{j+1}\left(z - z_j\right)}{\left(A_j\left(z_{j+1} - z\right) + A_{j+1}\left(z - z_j\right)\right)^2} h_j \\
&\qquad\quad \Bigg(2\left(-Q_j + Q_{j+1}\right)\left(A_j\left(z_{j+1} - z\right) + A_{j+1}\left(z - z_j\right)\right) \frac{1}{h_j^2} - \\
&\qquad\qquad - \left(Q_j\left(z_{j+1} - z\right) + Q_{j+1}\left(z - z_j\right)\right)\left(-A_j + A_{j+1}\right) \frac{1}{h_j^2} \Bigg) + \\
&\qquad + \frac{\beta}{2\rho A_0} \left(\frac{\left(A_j\left(z_{j+1} - z\right) + A_{j+1}\left(z - z_j\right)\right)}{h_j} \right)^{\frac{1}{2}} \left(-A_j + A_{j+1}\right) \frac{1}{h_j} \Bigg) \Bigg] \frac{z_{j+1} - z}{h_j}\, dz
\end{aligned}
$$

$$
\begin{aligned}
&{}^{3}_{Q}\boldsymbol{f}_{j+1}(K_j) \\
&= \int_{K_j} \Bigg[K_R \frac{Q_j\varphi_j(z) + Q_{j+1}\varphi_{j+1}(z)}{\left(A_j\varphi_j(z) + A_{j+1}\varphi_{j+1}(z)\right)^2} \left(Q_j\varphi_j(z) + Q_{j+1}\varphi_{j+1}(z)\right)' - \\
&\quad - K_R \frac{1}{A_j\varphi_j(z) + A_{j+1}\varphi_{j+1}(z)} \Bigg(\alpha \frac{Q_j\varphi_j(z) + Q_{j+1}\varphi_{j+1}(z)}{\left(A_j\varphi_j(z) + A_{j+1}\varphi_{j+1}(z)\right)^2} \\
&\quad \Big(2\left(Q_j\varphi_j(z) + Q_{j+1}\varphi_{j+1}(z)\right)' \left(A_j\varphi_j(z) + A_{j+1}\varphi_{j+1}(z)\right) - \\
&\quad - \left(Q_j\varphi_j(z) + Q_{j+1}\varphi_{j+1}(z)\right) \left(A_j\varphi_j(z) + A_{j+1}\varphi_{j+1}(z)\right)'\Big) + \\
&\quad + \frac{\beta}{2\rho A_0} \left(A_j\varphi_j(z) + A_{j+1}\varphi_{j+1}(z)\right)^{\frac{1}{2}} \left(A_j\varphi_j(z) + A_{j+1}\varphi_{j+1}(z)\right)' \Bigg) \Bigg] \varphi_{j+1}(z)\, dz \\
&= \int_{K_j} \Bigg[K_R \frac{Q_j\left(z_{j+1} - z\right) + Q_{j+1}\left(z - z_j\right)}{\left(A_j\left(z_{j+1} - z\right) + A_{j+1}\left(z - z_j\right)\right)^2} \left(-Q_j + Q_{j+1}\right) - \qquad (4.67) \\
&\quad - K_R \frac{1}{A_j\left(z_{j+1} - z\right) + A_{j+1}\left(z - z_j\right)} h_j \Bigg(\alpha \frac{Q_j\left(z_{j+1} - z\right) + Q_{j+1}\left(z - z_j\right)}{\left(A_j\left(z_{j+1} - z\right) + A_{j+1}\left(z - z_j\right)\right)^2} h_j \\
&\quad \Bigg(2\left(-Q_j + Q_{j+1}\right)\left(A_j\left(z_{j+1} - z\right) + A_{j+1}\left(z - z_j\right)\right) \frac{1}{h_j^2} - \\
&\quad - \left(Q_j\left(z_{j+1} - z\right) + Q_{j+1}\left(z - z_j\right)\right)\left(-A_j + A_{j+1}\right) \frac{1}{h_j^2} \Bigg) + \\
&\quad + \frac{\beta}{2\rho A_0} \left(\frac{\left(A_j\left(z_{j+1} - z\right) + A_{j+1}\left(z - z_j\right)\right)}{h_j} \right)^{\frac{1}{2}} \left(-A_j + A_{j+1}\right) \frac{1}{h} \Bigg) \Bigg] \frac{z - z_j}{h_j}\, dz
\end{aligned}
$$

Auch im vierten Skalarprodukt tritt die Ableitung von $\boldsymbol{F}$ auf.

$$
\left(\boldsymbol{H}(\boldsymbol{U}) \frac{\partial \boldsymbol{F}(\boldsymbol{U})}{\partial z}, \frac{d\varphi}{dz} \right)
$$

Multipliziert man $\boldsymbol{H}$ (siehe Gleichung 4.54) mit der nach z differenzierten Flussfunktion $\boldsymbol{F}$ (siehe Gleichung 4.63), so erhält man

$$
\boldsymbol{H} \frac{\partial \boldsymbol{F}}{\partial z} = \begin{pmatrix} \alpha \frac{2QQ'A - Q^2A'}{A^2} + \frac{\beta}{2\rho A_0} A^{\frac{1}{2}} A' \\ Q' \left(-\alpha \frac{Q^2}{A^2} + \frac{\beta}{2\rho A_0} A^{\frac{1}{2}} \right) + 2\alpha \frac{Q}{A} \left(\alpha \frac{2QQ'A - Q^2A'}{A^2} + \frac{\beta}{2\rho A_0} A^{\frac{1}{2}} A' \right) \end{pmatrix} \qquad (4.68)
$$

Die beiden Komponenten werden nun wieder getrennt betrachtet.

$$
\begin{aligned}
&{}^{4}_{A}\boldsymbol{f}_j(K_j) = \\
&\quad = \int_{K_j} \Bigg(\alpha \frac{Q_j\varphi_j(z) + Q_{j+1}\varphi_{j+1}(z)}{\left(A_j\varphi_j(z) + A_{j+1}\varphi_{j+1}(z)\right)^2} \\
&\qquad\qquad \big(2\left(Q_j\varphi_j(z) + Q_{j+1}\varphi_{j+1}(z)\right)' \left(A_j\varphi_j(z) + A_{j+1}\varphi_{j+1}(z)\right) - \\
&\qquad\qquad - \left(Q_j\varphi_j(z) + Q_{j+1}\varphi_{j+1}(z)\right)\left(A_j\varphi_j(z) + A_{j+1}\varphi_{j+1}(z)\right)'\big) + \\
&\qquad\quad + \frac{\beta}{2\rho A_0}\left(A_j\varphi_j(z) + A_{j+1}\varphi_{j+1}(z)\right)^{\frac{1}{2}} \left(A_j\varphi_j(z) + A_{j+1}\varphi_{j+1}(z)\right)' \Bigg) \varphi_j'(z)\, dz \\
&\quad = \int_{K_j} \Bigg(\alpha \frac{Q_j\left(z_{j+1} - z\right) + Q_{j+1}\left(z - z_j\right)}{\left(A_j\left(z_{j+1} - z\right) + A_{j+1}\left(z - z_j\right)\right)^2} h_j \qquad (4.69) \\
&\qquad\qquad \Bigg(2\left(-Q_j + Q_{j+1}\right)\left(A_j\left(z_{j+1} - z\right) + A_{j+1}\left(z - z_j\right)\right)\frac{1}{h_j^2} - \\
&\qquad\qquad - \left(Q_j\left(z_{j+1} - z\right) + Q_{j+1}\left(z - z_j\right)\right)\left(-A_j + A_{j+1}\right)\frac{1}{h_j^2}\Bigg) + \\
&\qquad\quad + \frac{\beta}{2\rho A_0}\left(\frac{\left(A_j\left(z_{j+1} - z\right) + A_{j+1}\left(z - z_j\right)\right)}{h_j}\right)^{\frac{1}{2}} \left(-A_j + A_{j+1}\right)\frac{1}{h}\Bigg)\frac{-1}{h}\, dz \\
&{}^{4}_{A}\boldsymbol{f}_{j+1}(K_j) = -\,{}^{4}_{A}\boldsymbol{f}_j(K_j) \qquad (4.70)
\end{aligned}
$$

$$
\begin{aligned}
&{}^{4}_{Q}\boldsymbol{f}_j(K_j) = \\
&= \int_{K_j} \Bigg[(Q_j\varphi_j(z) + Q_{j+1}\varphi_{j+1}(z))' \Bigg(-\alpha \frac{(Q_j\varphi_j(z) + Q_{j+1}\varphi_{j+1}(z))^2}{(A_j\varphi_j(z) + A_{j+1}\varphi_{j+1}(z))^2} + \\
&\quad + \frac{\beta}{2\rho A_0} (A_j\varphi_j(z) + A_{j+1}\varphi_{j+1}(z))^{\frac{1}{2}} \Bigg) + \\
&\quad + 2\alpha \frac{Q_j\varphi_j(z) + Q_{j+1}\varphi_{j+1}(z)}{A_j\varphi_j(z) + A_{j+1}\varphi_{j+1}(z)} \Bigg(\alpha \frac{Q_j\varphi_j(z) + Q_{j+1}\varphi_{j+1}(z)}{(A_j\varphi_j(z) + A_{j+1}\varphi_{j+1}(z))^2} \\
&\qquad \big(2\,(Q_j\varphi_j(z) + Q_{j+1}\varphi_{j+1}(z))'\,(A_j\varphi_j(z) + A_{j+1}\varphi_{j+1}(z)) - \\
&\qquad - (Q_j\varphi_j(z) + Q_{j+1}\varphi_{j+1}(z))\,(A_j\varphi_j(z) + A_{j+1}\varphi_{j+1}(z))' \big) + \\
&\qquad + \frac{\beta}{2\rho A_0} (A_j\varphi_j(z) + A_{j+1}\varphi_{j+1}(z))^{\frac{1}{2}} (A_j\varphi_j(z) + A_{j+1}\varphi_{j+1}(z))' \Bigg) \Bigg] \varphi_j'(z)\, dz \\
&= \int_{K_j} \Bigg[(-Q_j + Q_{j+1}) \frac{1}{h_j} \Bigg(-\alpha \frac{(Q_j\,(z_{j+1} - z) + Q_{j+1}\,(z - z_j))^2}{(A_j\,(z_{j+1} - z) + A_{j+1}\,(z - z_j))^2} + \qquad (4.71) \\
&\quad + \frac{\beta}{2\rho A_0} \left(\frac{A_j\,(z_{j+1} - z) + A_{j+1}\,(z - z_j)}{h_j} \right)^{\frac{1}{2}} \Bigg) + \\
&\quad + 2\alpha \frac{Q_j\,(z_{j+1} - z) + Q_{j+1}\,(z - z_j)}{A_j\,(z_{j+1} - z) + A_{j+1}\,(z - z_j)} \Bigg(\alpha \frac{Q_j\,(z_{j+1} - z) + Q_{j+1}\,(z - z_j)}{(A_j\,(z_{j+1} - z) + A_{j+1}\,(z - z_j))^2} h_j \\
&\qquad \Bigg(2\,(-Q_j + Q_{j+1})\,(A_j\,(z_{j+1} - z) + A_{j+1}\,(z - z_j))\,\frac{1}{h_j^2} - \\
&\qquad - (Q_j\,(z_{j+1} - z) + Q_{j+1}\,(z - z_j))\,(-A_j + A_{j+1})\,\frac{1}{h_j^2} \Bigg) + \\
&\quad + \frac{\beta}{2\rho A_0} \left(\frac{(A_j\,(z_{j+1} - z) + A_{j+1}\,(z - z_j)}{h_j} \right)^{\frac{1}{2}} (-A_j + A_{j+1})\,\frac{1}{h_j} \Bigg) \Bigg] \frac{-1}{h_j}\, dz \\
&{}^{4}_{Q}\boldsymbol{f}_{j+1}(K_j) = -\,{}^{4}_{Q}\boldsymbol{f}_j(K_j) \qquad (4.72)
\end{aligned}
$$

Zuletzt folgt noch das fünfte Skalarprodukt.

$$(\boldsymbol{B}_{LW}(\boldsymbol{U}), \varphi)$$

$$\boldsymbol{B}_{LW}(\boldsymbol{U}) = \boldsymbol{B}(\boldsymbol{U}) + \frac{\Delta t}{2} \boldsymbol{B}_{\boldsymbol{U}}(\boldsymbol{U}) \boldsymbol{B}(\boldsymbol{U})$$

Verwendet man wieder das vereinfachte $\boldsymbol{B}$ (siehe Gleichung 4.56), so ergibt sich die gleiche Jacobi Matrix $\boldsymbol{B}_{\boldsymbol{U}}$ wie beim dritten Skalarprodukt (siehe Gleichung 4.62).

Werden diese beiden Größen miteinander multipliziert, so erhält man:

$$\boldsymbol{B_U B} = \begin{pmatrix} 0 \\ K_R^2 \frac{Q^2}{A^2} \end{pmatrix} \tag{4.73}$$

Damit ergibt sich für den gesamten Ausdruck $\boldsymbol{B}_{LW}$:

$$\boldsymbol{B}_{LW} = \begin{pmatrix} 0 \\ -K_R \frac{Q}{A} + \frac{\Delta t}{2} K_R^2 \frac{Q}{A^2} \end{pmatrix} \tag{4.74}$$

Da die erste Komponente dieses Ausdrucks Null ist, sind auch die dazugehörigen Skalarprodukte Null. Es verbleiben also noch die Berechnungen für die zweite Komponente.

$$\begin{aligned} {}^5_Q\boldsymbol{f}_j(K_j) &= \int_{K_j} \Bigg(-K_R \frac{Q_j\varphi_j(z) + Q_{j+1}\varphi_{j+1}(z)}{A_j\varphi_j(z) + A_{j+1}\varphi_{j+1}(z)} + \\ &\qquad + \frac{\Delta t}{2} K_R^2 \frac{Q_j\varphi_j(z) + Q_{j+1}\varphi_{j+1}(z)}{\left(A_j\varphi_j(z) + A_{j+1}\varphi_{j+1}(z)\right)^2} \Bigg) \varphi_j(z)\, dz \\ &= \int_{K_j} \Bigg(-K_R \frac{Q_j\left(z_{j+1} - z\right) + Q_{j+1}\left(z - z_j\right)}{A_j\left(z_{j+1} - z\right) + A_{j+1}\left(z - z_j\right)} + \\ &\qquad + \frac{\Delta t}{2} K_R^2 \frac{Q_j\left(z_{j+1} - z\right) + Q_{j+1}\left(z - z_j\right)}{\left(A_j\left(z_{j+1} - z\right) + A_{j+1}\left(z - z_j\right)\right)^2} h_j \Bigg) \frac{z_{j+1} - z}{h_j}\, dz \end{aligned} \tag{4.75}$$

$$\begin{aligned} {}^5_Q\boldsymbol{f}_{j+1}(K_j) &= \int_{K_j} \Bigg(-K_R \frac{Q_j\varphi_j(z) + Q_{j+1}\varphi_{j+1}(z)}{A_j\varphi_j(z) + A_{j+1}\varphi_{j+1}(z)} + \\ &\qquad + \frac{\Delta t}{2} K_R^2 \frac{Q_j\varphi_j(z) + Q_{j+1}\varphi_{j+1}(z)}{\left(A_j\varphi_j(z) + A_{j+1}\varphi_{j+1}(z)\right)^2} \Bigg) \varphi_{j+1}(z)\, dz \\ &= \int_{K_j} \Bigg(-K_R \frac{Q_j\left(z_{j+1} - z\right) + Q_{j+1}\left(z - z_j\right)}{A_j\left(z_{j+1} - z\right) + A_{j+1}\left(z - z_j\right)} + \\ &\qquad + \frac{\Delta t}{2} K_R^2 \frac{Q_j\left(z_{j+1} - z\right) + Q_{j+1}\left(z - z_j\right)}{\left(A_j\left(z_{j+1} - z\right) + A_{j+1}\left(z - z_j\right)\right)^2} h_j \Bigg) \frac{z - z_j}{h_j}\, dz \end{aligned} \tag{4.76}$$

Somit wurden alle Teile der rechten Seite der Gleichung 4.47 berechnet. Um zu einem Vektor der rechten Seite zu gelangen, müssen sie nun noch passend summiert werden.

$${}^k_A\boldsymbol{f}_j = {}^k_A\, \boldsymbol{f}_j(K_j) + {}^k_A\, \boldsymbol{f}_j(K_{j-1}) \tag{4.77}$$

Um die fünf Vektoren ${}^{k}_{A}\boldsymbol{f},\ k = 1, \ldots, 5$ zu einem Vektor zu vereinigen, müssen sie mit den jeweiligen Zeitschritten multipliziert und danach addiert werden.

Für ${}_{Q}\boldsymbol{f}$ gelten analoge Überlegungen.

Es wurden damit alle Formeln hergeleitet, die man für die Implementierung des Blutflussmodells braucht. Eine solche Implementierung in Matlab wird im nächsten Kapitel vorgestellt, dabei wird auf die hier berechneten Gleichungen zurückgegriffen.

Kapitel 5

Implementierung

In der Folge wird eine Implementierung des im vorigen Kapitel hergeleiteten Blutflussmodells vorgestellt. Als Software dient die auf der Technischen Universität Wien im Rahmen einer Studentenlizenz erhältliche Version R2007a von Matlab. Die Vorgangsweise der Umsetzung lehnt sich stark an die im Kapitel 3 beschriebene Verfahrensweise bei einer Implementierung der Finiten Elemente Methode in Matlab an.

Es werden im weiteren Verlauf immer wieder Programmbestandteile zu sehen sein, in denen die vorkommenden Variablen mit gewissen Werten belegt sind. Die bei den jeweiligen Berechnungen tatsächlich verwendeten Parameterwerte werden bei der Vorstellung der Simulationsergebnisse in Kapitel 6 festgehalten.

5.1 Aufbau des Programms

Vor den eigentlichen Berechnungen werden drei Dateien erzeugt, in denen sich die Ortsdiskretisierung widerspiegelt. In der Datei `coordinates.dat` werden die z-Koordinaten der Knoten gespeichert, in `elements.dat` sind für alle Elemente der Diskretisierung die Indexnummern der Knoten festgehalten, welche das jeweilige Element beranden. Dazu kommt noch die Datei `dirichlet.dat`, in der die Nummern der Knoten am Rand gespeichert sind.
Eine ausführlichere Beschreibung dazu ist in Kapitel 3.2 nachzulesen.

Der dazugehörige Matlab-Code hat folgende Gestalt:

```
a =[0:0.1:10]';
```

```
csvwrite('coordinates.dat',a);

b(:,1) =[1:1:100]';
b(:,2) =[2:1:101]';
csvwrite('elements.dat',b);

c=[1 101]';
csvwrite(' dirichlet .dat',c);
```

Um den Inhalt der drei Dateien auch verwenden zu können, müssen diese zuerst in den Matlab Workspace geladen werden. Dies geschieht mit folgenden Befehlen:

```
load coordinates.dat;
load elements.dat;
load  dirichlet .dat;
```

Zu Beginn des Programms werden auch einigen Variablen die entsprechenden Parameterwerte zugewiesen.

```
alpha = 1;
KR = 8*pi*0.035;
rho = 1;
h0 = 0.05;
E0 = 3*10^6;
beta = E0*h0*sqrt(pi);
R0 = 0.5;
A0 = R0^2*pi;
Q0 = 0;
```

Bevor im Code die Herzstücke des Programms folgen, werden noch einige Initialisierungen und Variablenzuweisungen durchgeführt. Diese werden in der weiteren Abhandlung bei den entsprechenden Teilen erwähnt, um sie in den passenden Zusammenhang zu bringen.

5.2 Aufstellen der Matrizen

Da der Vektor $\boldsymbol{U}$ der Zustandsgrößen aus den zwei Komponenten A und Q besteht, wird das Galerkin Verfahren auf zwei Gleichungssysteme aufgeteilt, deren Matrizen der linken Seite mit S und T bezeichnet werden.

Für beide Komponenten werden die selbe Diskretisierung und die selben Basisfunktionen verwendet, daher sind auch die Steifigkeitsmatrizen S und T ident. Weiters bleiben diese Gegebenheiten über den ganzen Zeitverlauf gleich, es muss daher nur einmal am Beginn der Berechnungen eine Steifigkeitsmatrix aufgestellt werden.

Bevor die tatsächliche Berechnung der Steifigkeitsmatrix S beginnt, werden noch einige Initialisierungen vorgenommen.

```
S=sparse(length(coordinates),length(coordinates));
T=sparse(length(coordinates),length(coordinates));

M=[0 0; 0 0];
```

Dabei wird die Matrix M zur Speicherung von Zwischenergebnissen bei der Berechnung verwendet.

Vergleicht man die Skalarprodukte der linken Seite des Blutflussmodells (siehe Abschnitt 4.3.3) mit den Skalarprodukten der linken Seite des Modellproblems (siehe Kapitel 3.3), so zeigt sich, dass die Berechnung von S aus dem Blutflussmodell mit der Berechnung von C aus dem Modellproblem übereinstimmt, wenn $c(z) = 1$ gesetzt wird.

Daher wird auch die ab Seite 24 beschriebene Vorgangsweise großteils übernommen. Einziger Unterschied ist, dass die Berechnung der Skalarprodukte wegen der vereinfachten Form der Funktion c nicht in eine eigene Matlab-Funktion ausgelagert wird, sondern direkt im Hauptprogramm vonstatten geht.

```
for j=1:size(elements,1)
    z=coordinates(elements(j,:));
    h=z(2)-z(1);
    M(1,1) = quad('integrand1',z(1),z(2) ,[],[],-1, z(2),-1,z(2))/h^2;
    M(1,2) = quad('integrand1',z(1),z(2) ,[],[],-1, z(2),1,-z(1))/h^2;
    M(2,1) = quad('integrand1',z(1),z(2) ,[],[],-1, z(2),1,-z(1))/h^2;
    M(2,2) = quad('integrand1',z(1),z(2) ,[],[],1,- z(1),1,-z(1))/h^2;

    S(elements(j,:),elements(j,:)) = S(elements(j,:),elements(j,:)) + M;
end
```

Die Werte der Matrix T werden in der Folge von S übernommen.

```
T = S;
```

5.3 Aufstellen der rechten Seite

Der Lastvektor der rechten Seite des Gleichungssystems beim Galerkin Verfahren muss bei der Lösung des Blutflussmodells in jedem Zeitschritt neu berechnet werden. Daher werden die einzelnen Zeitpunkte mit Hilfe einer for Schleife durchlaufen. Noch vor der Schleife müssen der Anfangswert der Zeit t und der Zeitschritt dt festgelegt werden.

```
t=0;
dt = 1*10^(-4);
```

Danach kann mit der Schleife begonnen werden.

```
for i = 2:301
    t=t+dt;
    rhsA=sparse(length(coordinates),1);
    rhsQ=sparse(length(coordinates),1);
```

Die Schleife beginnt mit dem Laufindex 2, weil der erste Index den Anfangswerten vorbehalten ist.
Zuerst wird der Zeitpunkt vom letzten Durchlauf um den Zeitschritt dt erhöht. Danach werden zwei Vektoren initialisiert, in denen die rechten Seiten der beiden Gleichungssysteme für A und Q abgespeichert werden.

In einer zweiten Schleife wird nun der Lastvektor der rechten Seite berechnet. Die Vorgangsweise folgt dabei jener von Kapitel 3.4. Da die rechte Seite des Blutflussmodells aber sehr viel komplexer als jene des Modellproblems von Kapitel 3 ist, wurde sie bereits bei der Diskretisierung im Abschnitt 4.3.4 in fünf Teile aufgespalten. Diese Aufteilung wird auch bei der Implementierung weitergeführt.
Ein weiterer kleiner Unterschied bei der Programmierung liegt darin, dass auf die innere for Schleife und den mod Befehl des Codes in Kapitel 3.4 verzichtet wird. Stattdessen werden die auftretenden Terme explizit angeschrieben.

Im Weiteren werden alle fünf Teile einzeln betrachtet, dabei wird auf etwaige Besonderheiten hingewiesen. Beim folgenden Ausschnitt zum ersten Skalarprodukt ist am Beginn auch die Schleife ersichtlich.

```
    for j=1:size(elements,1)
        z=coordinates(elements(j,:));

        NA(1)=quad('erstesSP',z(1),z(2),tol ,[], A(j),A(j+1),z(1),z(2),-1,z(2));
        NA(2)=quad('erstesSP',z(1),z(2),tol ,[], A(j),A(j+1),z(1),z(2),1,-z(1));
```

```
rhsA(elements(j,1))=rhsA(elements(j,1)) + NA(1);
rhsA(elements(j,2))=rhsA(elements(j,2)) + NA(2);
NQ(1)=quad('erstesSP',z(1),z(2),tol ,[], Q(j),Q(j+1),z(1),z(2),-1,z(2));
NQ(2)=quad('erstesSP',z(1),z(2),tol ,[], Q(j),Q(j+1),z(1),z(2),1,-z(1));
rhsQ(elements(j,1))=rhsQ(elements(j,1)) + NQ(1);
rhsQ(elements(j,2))=rhsQ(elements(j,2)) + NQ(2);
```

Betrachtet man die Integrale 4.48 und 4.50, welche die ersten Skalarprodukte der beiden Komponenten darstellen, so stellt man fest, dass sich die beiden Komponenten nur durch die Koeffizienten A_j und Q_j bzw. A_{j+1} und Q_{j+1} unterscheiden. Es reicht daher aus, für beide Komponenten den gleichen Integranden zu verwenden und beim Funktionsaufruf die passenden Koeffizienten zu übergeben.

Der Integrand ist in der Datei `erstesSP.m` gespeichert, welche folgendes Aussehen hat:

```
function val=erstesSP(z,a1,a2,z1,z2,d1,d0)
h=z2-z1;
val=(a1.*(z2-z)+a2.*(z-z1))/h.*(d1*z+d0)./h;
```

Das zweite Skalarprodukt kann nach den Gleichungen 4.58 bis 4.61 folgendermaßen berechnet werden.

```
NA(1)=quad('zweitesSP1K',z(1),z(2),tol,[], A(j),A(j+1),Q(j),Q(j+1),z(1),
    z(2),dt,KR);
NA(2)=-NA(1);
rhsA(elements(j,1))=rhsA(elements(j,1)) + NA(1);
rhsA(elements(j,2))=rhsA(elements(j,2)) + NA(2);
NQ(1)=quad('zweitesSP2K',z(1),z(2),tol,[], A(j),A(j+1),Q(j),Q(j+1),z(1),
    z(2),dt,alpha,rho,KR,A0,beta);
NQ(2)=-NQ(1);
rhsQ(elements(j,1))=rhsQ(elements(j,1)) + NQ(1);
rhsQ(elements(j,2))=rhsQ(elements(j,2)) + NQ(2);
```

Dabei ist die Funktion zweitesSP1K :

```
function val=zweitesSP1K(z,a1,a2,q1,q2,z1,z2,dt,KR)
h=z2-z1;
val=dt*(((q1.*(z2-z)+q2.*(z-z1))/h-dt/2*KR*(q1.*(z2-z)+q2.*(z-z1))/(a1.*(
    z2-z)+a2.*(z-z1)))/(-h));
```

Für die zweite Komponente ist der Integrand von der Form:

```
function val=zweitesSP2K(z,a1,a2,q1,q2,z1,z2,dt,alpha,rho,KR,A0,beta)
h=z2-z1;
val= dt*((alpha/h*(q1*(z2-z)+q2*(z-z1)).^2/(a1*(z2-z)+a2*(z-z1))+beta/(3*
    rho*A0)*((a1*(z2-z)+a2*(z-z1))/h).^(3/2)-dt*alpha*KR*(q1*(z2-z)+q2*(z
    -z1)).^2/(a1*(z2-z)+a2*(z-z1)).^2)/(-h));
```

Beim dritten Skalarprodukt ist die erste Komponente Null, es gibt daher entsprechend den Integralen 4.66 und 4.67 nur Beiträge für die rechte Seite des zweiten Gleichungssystems.

```
NQ(1)=quad('drittesSP2K',z(1),z(2),tol ,[], A(j),A(j+1),Q(j),Q(j+1),z(1),
    z(2),alpha,rho,KR,A0,beta,-1,z(2),dt);
NQ(2)=quad('drittesSP2K',z(1),z(2),tol ,[], A(j),A(j+1),Q(j),Q(j+1),z(1),
    z(2),alpha,rho,KR,A0,beta,1,-z(1),dt);
rhsQ(elements(j,1))=rhsQ(elements(j,1)) + NQ(1);
rhsQ(elements(j,2))=rhsQ(elements(j,2)) + NQ(2);
```

Für die Integration wird die Funktion drittesSP2K verwendet.

```
function val=drittesSP2K(z,a1,a2,q1,q2,z1,z2,alpha,rho,KR,A0,beta,d1,d0,dt)
h=z2-z1;
val= -dt^2/2*((KR.*(q1.*(z2-z)+q2.*(z-z1))./(a1.*(z2-z)+a2.*(z-z1)).^2.*(-
    q1+q2)-KR./(a1.*(z2-z)+a2.*(z-z1)).*h.*(alpha.*(q1.*(z2-z)+q2.*(z-z1))
    ./(a1.*(z2-z)+a2.*(z-z1)).^2.*h.*(2.*(-q1+q2).*(a1.*(z2-z)+a2.*(z-z1))./h
    ^2.-(q1.*(z2-z)+q2.*(z-z1))*(-a1+a2)./h^2)+beta./(2.*rho.*A0).*(a1.*(z2-
    z)+a2.*(z-z1)).^(1/2)./h^(1/2).*(-a1+a2)./h)).*(d1*z+d0)./h);
```

Da sich die auf einem Element der Diskretisierung ergebenden Einträge beim vierten Skalarprodukt nur um ein Vorzeichen unterscheiden (vgl. 4.69 - 4.72), müssen nur zwei Integrationen ausgeführt werden.

```
NA(1)=quad('viertesSP1K',z(1),z(2),tol ,[], A(j),A(j+1),Q(j),Q(j+1),z(1),
    z(2),alpha,rho,A0,beta,dt);
NA(2)=-NA(1);
rhsA(elements(j,1))=rhsA(elements(j,1)) + NA(1);
rhsA(elements(j,2))=rhsA(elements(j,2)) + NA(2);
NQ(1)=quad('viertesSP2K',z(1),z(2),tol ,[], A(j),A(j+1),Q(j),Q(j+1),z(1),
    z(2),alpha,rho,A0,beta,dt);
NQ(2)=-NQ(1);
rhsQ(elements(j,1))=rhsQ(elements(j,1)) + NQ(1);
rhsQ(elements(j,2))=rhsQ(elements(j,2)) + NQ(2);
```

Dazu benötigt man folgende Funktionen:

```
function val=viertesSP1K(z,a1,a2,q1,q2,z1,z2,alpha,rho,A0,beta,dt)
h=z2-z1;
val= -dt^2/2*((alpha*(q1.*(z2-z)+q2.*(z-z1))/(a1.*(z2-z)+a2.*(z-z1)).^2*h
    *(2*(-q1+q2)*(a1.*(z2-z)+a2.*(z-z1))/h^2-(q1.*(z2-z)+q2.*(z-z1))*(-
    a1+a2)/h^2)+beta/(2*rho*A0)*(a1.*(z2-z)+a2.*(z-z1)).^(1/2)/h^(1/2)*(-a1+
    a2)/h)/(-h));
```

```
function val=viertesSP2K(z,a1,a2,q1,q2,z1,z2,alpha,rho,A0,beta,dt)
h=z2-z1;
val= -dt^2/2*(((-q1+q2)/h*(-alpha*(q1.*(z2-z)+q2.*(z-z1)).^2/(a1.*(z2-z)+
    a2.*(z-z1)).^2+beta/(2*rho*A0)*(a1.*(z2-z)+a2.*(z-z1)).^(1/2)/h^(1/2))+2*
    alpha*(q1.*(z2-z)+q2.*(z-z1))/(a1.*(z2-z)+a2.*(z-z1))*(alpha*(q1.*(z2-
    z)+q2.*(z-z1))/(a1.*(z2-z)+a2.*(z-z1)).^2/h *(2*(-q1+q2)*(a1.*(z2-z)+a2
    .*(z-z1))-(q1.*(z2-z)+q2.*(z-z1))*(-a1+a2))+beta/(2*rho*A0)*(a1.*(z2-
    z)+a2.*(z-z1)).^(1/2)/h^(1/2)*(-a1+a2)/h))/(-h));
```

Im letzten Skalarprodukt beschränken sich die Berechnungen wegen des Nulleintrags in der ersten wieder auf die zweite Komponente (vgl. 4.75 und 4.76). Nach den letzen Berechnungen für die rechte Seite kann die Schleife wieder geschlossen werden.

```
        NQ(1)=quad('fuenftesSP2K',z(1),z(2),tol ,[], A(j) ,A(j+1),Q(j) ,Q(j+1),z(1)
            ,z(2) ,dt ,KR,-1,z(2));
        NQ(2)=quad('fuenftesSP2K',z(1),z(2),tol ,[], A(j) ,A(j+1),Q(j) ,Q(j+1),z(1)
            ,z(2) ,dt ,KR,1,-z(1));
        rhsQ(elements(j,1))=rhsQ(elements(j,1)) + NQ(1);
        rhsQ(elements(j,2))=rhsQ(elements(j,2)) + NQ(2);
    end
```

In der Datei `fuenftesSP2K.m` befindet sich der dazugehörige Integrand.

```
function val=fuenftesSP2K(z,a1,a2,q1,q2,z1,z2,dt,KR,d1,d0)
h=z2-z1;
val= dt*((-KR*(q1.*(z2-z)+q2.*(z-z1))/(a1.*(z2-z)+a2.*(z-z1))+dt/2*KR^2*h
    *(q1.*(z2-z)+q2.*(z-z1))/(a1.*(z2-z)+a2.*(z-z1)).^2)*(d1*z+d0)/h);
```

Damit sind alle Bestandteile der beiden linearen Gleichungssysteme bestimmt. Bevor die Lösung in Angriff genommen wird, müssen noch die passenden Randbedingungen bestimmt werden.

5.4 Randbedingungen

In den bisherigen Modellbetrachtungen im Kapitel 4 wurden die Randwerte vernachlässigt. Um mit Hilfe der Finiten Elemente Methode zu einer Lösung zu gelangen, ist die Vorgabe von Randwerten bei $z = 0$ und $z = L$ aber unumgänglich. Außerdem kann erst mit geeigneten Randbedingungen das betrachtete Arterienstück in den Blutkreislauf des Körpers eingebettet werden.

Da die beiden Zustandsgrößen des Blutflussmodells die Querschnittsfläche A und der mittlere Volumenfluss Q sind, müssen für diese beide Größen bei jedem Zeitschritt geeignete Werte bei $z = 0$ und $z = L$ vorgegeben werden.

Allerdings hat man für die beiden Zustandsgrößen oft keine passenden Messwerte oder andere Daten, um die Randwerte an den benötigten Punkten vorgeben zu können. Dagegen ist der entsprechende Druck eine häufiger bekannte Systemgröße. Daher verwendet man die charakteristischen Variablen des Systems, um die erforderlichen Randwerte berechnen zu können.

Eine kurze Einführung in die Thematik der Charakteristiken und der charakteristischen Variablen bietet Kapitel 4.1.

Der Zusammenhang zwischen den Zustandsgrößen und den charakteristischen Variablen wurde bereits mit den Gleichungen 4.36 bis 4.38 beschrieben und soll hier wiederholt werden:

$$W_{1,2} = \frac{Q}{A} \pm 4\sqrt{\frac{\beta}{2\rho A_0}} A^{\frac{1}{4}} \tag{5.1}$$

$$A = \left(\frac{2\rho A_0}{\beta}\right)^2 \left(\frac{W_1 - W_2}{8}\right)^4 \tag{5.2}$$

$$Q = A\frac{W_1 + W_2}{2} \tag{5.3}$$

Diesen Gleichungen liegt die Annahme $\alpha = 1$ zugrunde.

Die weiteren Überlegungen folgen den Ausführungen in [5] und [7].

Um die Randbedingungen in einen Zusammenhang mit den physiologischen Vorgängen des Körpers zu bringen, wird angenommen, dass bei $z = 0$ das proximale und bei $z = L$ das distale Ende des betrachteten Arterienstücks liegt.

Für die betrachteten Parameterbereiche liegt mit 4.31 ein hyperbolisches System vor, dessen zwei Eigenwerte unterschiedliches Vorzeichen haben, wobei wir den positiven Eigenwert mit dem Index 1 versehen.
Somit gibt es an beiden Enden des Arterienstücks eine einlaufende charakteristische Variable. Eine Möglichkeit ist nun, deren Werte durch eine Funktion zu beschreiben.

$$W_1(t) = g_1(t) \quad \text{bei } z = 0 \tag{5.4}$$
$$W_2(t) = g_2(t) \quad \text{bei } z = L \tag{5.5}$$

Um die Randbedingungen zu vervollständigen, werden noch die Werte der ausgehenden charakteristischen Variablen benötigt. Diese können mit Hilfe einer linearen Extrapolation geschätzt werden (vgl. dazu Kapitel 4.1, insbesondere Gleichung 4.15).

Es sei nun angenommen, dass die Werte für $\boldsymbol{U}$ und somit die Werte der charakteristischen Variablen zum Zeitpunkt t^n bekannt sind (im Zuge der Berechnungen werden die approximierten Werte bekannt sein). Der Wert von W_2 zum Zeitpunkt t^{n+1} bei $z = 0$ kann dann durch den Wert von W_2 zum Zeitpunkt t^n bei $z = -\lambda_2^n(0)\Delta t$ geschätzt werden. Dabei ist $\lambda_2^n(0)$ der Eigenwert zum Zeitpunkt t^n an der Stelle $z = 0$. Für W_1 bei $z = L$ wird eine analoge Vorgangsweise gewählt. Dies führt zu:

$$W_2^{n+1}(0) := W_2^n(-\lambda_2^n(0)\Delta t) \tag{5.6}$$
$$W_1^{n+1}(L) := W_1^n(L - \lambda_1^n(L)\Delta t) \tag{5.7}$$

Nun soll die Funktion für die eingehende charakteristische Variable am proximalen Arterienstückende nochmals genauer betrachtet werden. In der Praxis wird oft nicht direkt eine Funktion für die charakteristische Variable sondern eine Druckkurve $p(t)$ vorgegeben. Dazu muss die Funktion $g_1(t)$ entsprechend angepasst werden.

Aus 5.1 erhält man:

$$W_1 = W_2 + 8\sqrt{\frac{\beta}{2\rho A_0} A^{\frac{1}{2}}} \tag{5.8}$$

Um A durch die Funktion p ersetzen zu können, verwendet man den Zusammenhang 4.27 zwischen der Querschnittsfläche A und dem Druck P. Dadurch erhält man:

$$A^{\frac{1}{2}} = \frac{A_0}{\beta} p + A_0^{\frac{1}{2}} \tag{5.9}$$

Setzt man das in Gleichung 5.8 ein, so ergibt das:

$$W_1 = W_2 + 8\sqrt{\frac{1}{2\rho} p + \frac{\beta}{2\rho\sqrt{A_0}}} \tag{5.10}$$

Als weitere Vereinfachung wird bei der numerischen Lösung der Wert der zweiten charakteristischen Variablen in dieser Formel konstant am Anfangswert belassen.

$$W_1 = W_2^0 + 8\sqrt{\frac{1}{2\rho} p + \frac{\beta}{2\rho\sqrt{A_0}}} \tag{5.11}$$

Ebenfalls über den ganzen Zeitverlauf wird der Wert von W_2 bei $z = L$ mit dem Anfangswert belegt. Dadurch wird ein unendlich langes Rohr simuliert.

$$W_2(t) = g_2(t) = W_2^0 \quad \text{bei } z = L \tag{5.12}$$

Mit den Gleichungen 5.11 und 5.12 sowie 5.6 und 5.7 sind zu jedem Zeitschritt des Lösungsverfahrens die Werte für die charakteristischen Variablen am Rand definiert. Aus 5.2 und 5.3 folgen die entsprechenden Randwerte für A und Q.

Bei der Umsetzung in Matlab müssen zunächst die charakteristischen Variablen zum Simulationsbeginn berechnet werden.

```
W2L_0=Q0/A0-4*sqrt(beta/(2*rho*A0))*A0^(1/4);
W20_0=W2L_0;
```

Dabei ist der Wert von W_2 über das ganze Arterienstück konstant, weil die Werte von A_0, Q_0 und β ebenfalls als konstant angenommen werden.

Alle folgenden Berechnungen werden innerhalb der Zeitschleife, also für jeden Zeitschritt ausgeführt.

Für die oben beschriebene lineare Extrapolation von W_2 bei $z = 0$ und W_1 bei $z = L$ müssen zunächst die dazugehörigen Eigenwerte nach Gleichung 4.35 bestimmt werden.

```
lambda2_0=Q(1)/A(1)-sqrt(beta/(2*rho*A0))*A(1)^(1/4);
lambda1_L=Q(lc)/A(lc)+sqrt(beta/(2*rho*A0))*A(lc)^(1/4);
```

Die Variable lc entspricht dem Index der Ortskoordinate bei $z = L$.

Danach werden die Stellen $z = -\lambda_2^n(0)\Delta t$ sowie $z = L - \lambda_1^n(L)\Delta t$ berechnet, an denen die charakteristischen Variablen W_2^n bzw. W_1^n ausgewertet werden sollen.

```
zs0=-lambda2_0*dt;
zsL=coordinates(lc)-lambda1_L*dt;
```

Die Auswertung wird mit folgendem Codestück realisiert:

```
Azs0=A(floor(zs0/h)+1)*mod(zs0,h)/h+A(floor(zs0/h)+2)*(1-mod(zs0,h)/h);
AzsL=A(floor(zsL/h)+1)*mod(zsL,h)/h+A(floor(zsL/h)+2)*(1-mod(zsL,h)/h);
Qzs0=Q(floor(zs0/h)+1)*mod(zs0,h)/h+Q(floor(zs0/h)+2)*(1-mod(zs0,h)/h);
QzsL=Q(floor(zsL/h)+1)*mod(zsL,h)/h+Q(floor(zsL/h)+2)*(1-mod(zsL,h)/h);
```

Da die Werte von A und Q zum Zeitpunkt t^n als lineare Interpolation der Hutfunktionen vorliegen, muss festgestellt werden, in welchen Elementen der Diskretisierung die Auswertungspunkte zs0 und zsL zu liegen kommen. Dazu wird eine äquidistante Zerlegung mit Elementgröße h vorausgesetzt.

Dividiert man die Auswertungsstelle durch die Elementgröße, also $\frac{z}{h}$, so ist das Ergebnis ohne Berücksichtigung der Nachkommastellen die Indexnummer des entsprechenden Elements, wobei man noch eine Verschiebung um 1 berücksichtigen muss, weil das erste Element der Matlab-Indizierung folgend mit eins und nicht mit null indiziert wird. Die Funktion floor liefert die nötige Abrundung.

Kennt man nun das Element der Zerlegung, so muss noch die passende Zwischenstelle gefunden werden, an der eine durch die Funktionswerte bei den berandenden Knoten gelegte Gerade ausgewertet werden soll. Dazu wird die Funktion mod verwendet, welche den Rest der Division von z durch h liefert.

Mit diesen Funktionswerten von A und Q können nun die beiden gesuchten charakteristischen Variablen berechnet werden.

```
W20=Qzs0/Azs0-4*sqrt(beta/(2*rho*A0))*Azs0^(1/4);
W1L=QzsL/AzsL+4*sqrt(beta/(2*rho*A0))*AzsL^(1/4);
```

Die beiden anderen charakteristischen Variablen werden entsprechend den Gleichungen 5.11 und 5.12 bestimmt.

```
W2L=W2L_0;
W10=W20_0+8*sqrt(1/(2*rho)*p(t)+beta/(2*rho*sqrt(A0)));
```

In der Funktion p ist zu diesem Zweck eine passende Druckkurve definiert.

Um zu den Randwerten zu gelangen, müssen jetzt noch die charakteristischen Variablen in die Zustandsgrößen umgewandelt werden (vgl. 5.2 und 5.3). Das Ergebnis wird in den beiden Vektoren uA und uQ gespeichert, die davor noch entsprechend initialisiert werden müssen.

```
uA(1)=(2*rho*A0/beta)^2*((W10-W20)/8)^4;
uQ(1)=uA(1)*(W10+W20)/2;
uA(lc)=(2*rho*A0/beta)^2*((W1L-W2L)/8)^4;
uQ(lc)=uA(lc)*(W1L+W2L)/2;
```

5.5 Lösen des Gleichungssystems

Die beiden linearen Gleichungssysteme für die Werte der Zustandsgrößen A und Q können nun, wie in den Kapiteln 3.5 und 3.6 beschrieben, gelöst werden.

Die Randwerte wurden bereits in den Lösungsvektoren gespeichert. Daher werden im nächsten Schritt die Gleichungssysteme so verändert, dass die Randwerte berücksichtigt werden.

```
rhsA = rhsA-S*uA;
rhsQ = rhsQ-T*uQ;
```

Somit können die Gleichungssysteme auf den freien Knoten gelöst werden.

```
uA(FreeNodes)=S(FreeNodes,FreeNodes)\rhsA(FreeNodes);
uQ(FreeNodes)=T(FreeNodes,FreeNodes)\rhsQ(FreeNodes);
```

Die Ergebnisse werden noch in zwei Matrizen abgespeichert und die Vektoren mit den Zustandsgrößen werden aktualisiert. Nach dem Löschen einiger Variablen kann die Schleife zum nächsten Zeitpunkt übergehen.

```
    AV(:,i)=uA;
    QV(:,i)=uQ;
    A=uA;
    Q=uQ;
    clear uA uQ rhsA rhsQ
end
```

Damit ist die Implementierung des Blutflussmodells abgeschlossen. Für eine Speicherung der Simulationsergebnisse oder deren grafische Ausgabe sind noch die entsprechenden Befehle hinzuzufügen, auf eine eingehende Erläuterung dieser Programmzeilen wird hier aber verzichtet.

5.6 Bifurkationen

Bis jetzt wurden nur einzelne Arterienstücke simuliert. Um das Modell zu erweitern liegt es nahe, auch Verzweigungen (= Bifurkationen) von einer Arterie in zwei weitere zu betrachten.
Die folgenden Überlegungen basieren wiederum auf [5] von Luca Formaggia et al., ganz ähnliche Ausführungen des Autors findet man auch in [6].

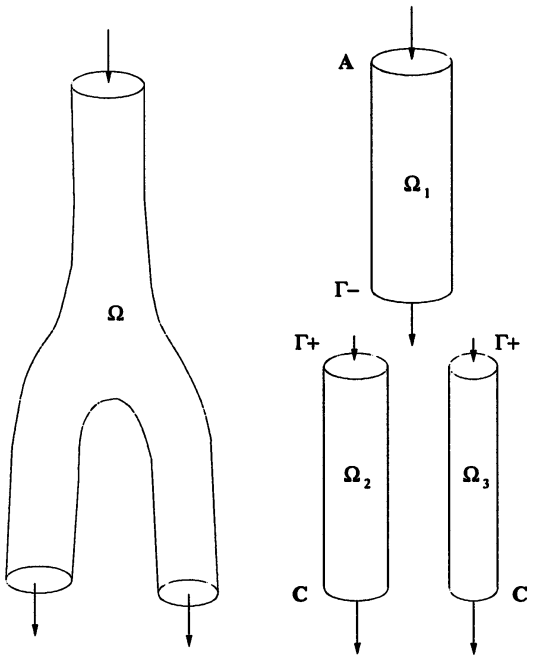

Abbildung 5.1: Skizze zur Modellierung einer Bifurkation (aus [5], S. 264)

Die Modellierung einer Verzweigung wird mit Abbildung 5.1 (aus [5], S. 264) verdeutlicht. Dabei wird in weiterer Folge die proximale Arterie als Arterie 1, die beiden von der Bifurkation ausgehenden mit Arterie 2 bzw. 3 bezeichnet.

Um die drei Arterien bei einer Bifurkation geeignet verbinden zu können, werden drei Bedingungen benötigt. Die erste ergibt sich aus der Forderung, dass der Volumenfluss über die Verzweigung erhalten bleibt.

$$Q^{(1)} = Q^{(2)} + Q^{(3)} \quad \text{bei } z = L^{(1)},\ t > 0 \tag{5.13}$$

Die zwei weiteren Bedingungen erhält man aus der Forderung, dass der Druck an der Bifurkation in allen drei Arterien gleich sein muss.

$$P^{(1)} = P^{(2)} = P^{(3)} \quad \text{bei } z = L^{(1)},\ t > 0 \tag{5.14}$$

Für die bei der Finiten Elemente Methode benötigten Randbedingungen werden noch drei weitere Werte hinzugefügt, die sich wieder aus der Extrapolation der ausgehenden charakteristischen Variablen ergeben.

Zusammen erhält man ein nichtlineares Gleichungssystem mit den sechs Unbekannten $A^{(1)}, A^{(2)}, A^{(3)}, Q^{(1)}, Q^{(2)}$ und $Q^{(3)}$. Dieses Gleichungssystem muss in jedem Zeitschritt gelöst werden, um die Randwerte bei der Bifurkation zu bestimmen.

$$\begin{aligned}
W_1^{(1)} =& \frac{Q^{(1)}}{A^{(1)}} + 4\sqrt{\frac{\beta^{(1)}}{2\rho A_0^{(1)}}}(A^{(1)})^{\frac{1}{4}} \\
W_2^{(2)} =& \frac{Q^{(2)}}{A^{(2)}} - 4\sqrt{\frac{\beta^{(2)}}{2\rho A_0^{(2)}}}(A^{(2)})^{\frac{1}{4}} \\
W_2^{(3)} =& \frac{Q^{(3)}}{A^{(3)}} - 4\sqrt{\frac{\beta^{(3)}}{2\rho A_0^{(3)}}}(A^{(3)})^{\frac{1}{4}} \\
Q^{(1)} =& Q^{(2)} + Q^{(3)} \\
\beta^{(1)}\frac{\sqrt{A^{(1)}} - \sqrt{A_0^{(1)}}}{A_0^{(1)}} =& \beta^{(2)}\frac{\sqrt{A^{(2)}} - \sqrt{A_0^{(2)}}}{A_0^{(2)}} \\
\beta^{(1)}\frac{\sqrt{A^{(1)}} - \sqrt{A_0^{(1)}}}{A_0^{(1)}} =& \beta^{(3)}\frac{\sqrt{A^{(3)}} - \sqrt{A_0^{(3)}}}{A_0^{(3)}}
\end{aligned} \tag{5.15}$$

Dabei wurde für die beiden letzten Gleichungen die Beziehung 4.27 verwendet, die den Druck P in Abhängigkeit zur Querschnittsfläche A darstellt.

Da nun drei Arterienstücke gleichzeitig zu berechnen sind, ist eine Umstrukturierung des bisherigen Programms notwendig. Die einzelnen Rechenschritte sind aber nach wie vor die gleichen, daher wird auf eine nochmalige Auflistung des Codes dieser Programmteile verzichtet, die Konzentration liegt auf den Neuerungen.

Die einzelnen Schritte der bisherigen Berechnungen sind in Funktionen zusammengefasst, die von einem Hauptprogramm aufgerufen werden.

Zu Beginn werden in gewohnter Weise die Dateien mit den Informationen der Ortsdiskretisierung erzeugt. Danach werden die Parameter und Anfangswerte definiert, wobei darauf geachtet werden muss, welche Werte in allen drei Arterien gleich sind und welche nicht.

Der erste Funktionsaufruf dient der Erzeugung der Steifigkeitsmatrizen.

```
[S_1, T_1]=linkeSeite(coordinates_1, elements_1);
[S_2, T_2]=linkeSeite(coordinates_2, elements_2);
[S_3, T_3]=linkeSeite(coordinates_3, elements_3);
```

Es folgt der Beginn der Zeitschleife mit den Berechnungen der Randwerte.

```
for i=2:2001
    t=t+dt;

    [uA1_1, uQ1_1, uAL_1, uQL_1, W20_1, W1L_1, W2L_1, W10_1]= randwerte(
        coordinates_1, A_1(:,i-1), Q_1(:,i-1), beta, rho, t, dt, A0_1, W2L_0_1,
        W20_0_1);
    [uA1_2, uQ1_2, uAL_2, uQL_2, W20_2, W1L_2, W2L_2, W10_2]= randwerte(
        coordinates_2, A_2(:,i-1), Q_2(:,i-1), beta, rho, t, dt, A0_2, W2L_0_2,
        W20_0_2);
    [uA1_3, uQ1_3, uAL_3, uQL_3, W20_3, W1L_3, W2L_3, W10_3]= randwerte(
        coordinates_3, A_3(:,i-1), Q_3(:,i-1), beta, rho, t, dt, A0_3, W2L_0_3,
        W20_0_3);
```

Dabei wird die Bifurkation noch nicht berücksichtigt, dies erfolgt im nächsten Schritt. Dazu muss zunächst das nichtlineare Gleichungssystem 5.15 definiert werden. Das wird mit einer im Hauptprogramm eingebetteten Funktion bewerkstelligt.

```
bif =@(ub) [ub(2)/ub(1)+4*sqrt(beta/(2*rho*A0_1))*(ub(1))^(1/4)-W1L_1;
    ub(4)/ub(3)-4*sqrt(beta/(2*rho*A0_2))*(ub(3))^(1/4)-W20_2; ub(6)/ub
    (5)-4*sqrt(beta/(2*rho*A0_3))*(ub(5))^(1/4)-W20_3; ub(4)+ub(6)-ub(2)
    ; beta*(sqrt(ub(3))-sqrt(A0_2))/A0_2 - beta*(sqrt(ub(1))-sqrt(A0_1))/
    A0_1; beta*(sqrt(ub(5))-sqrt(A0_3))/A0_3 - beta*(sqrt(ub(1))-sqrt(A0_1
    ))/A0_1];
```

Die sechs Unbekannten sind dabei durch den Vektor ub dargestellt.

Das nichtlineare Gleichungssystem wird mit Hilfe der Matlab-Funktion fsolve gelöst. Als Startwerte des Lösungsalgorithmus werden die Ergebnisse des vorherigen Zeitschritts verwendet, vor dem ersten Durchlauf muss daher der Vektor Ub mit Anfangswerten belegt werden.

```
[Ub]= fsolve(bif, Ub);
uAL_1=Ub(1);
uQL_1=Ub(2);
uA1_2=Ub(3);
uQ1_2=Ub(4);
uA1_3=Ub(5);
uQ1_3=Ub(6);
```

In einer letzten Funktion werden die rechten Seiten der Gleichungssysteme erzeugt und die Systeme werden gelöst. Dazu werden beim Funktionsaufruf u.a. die gerade berechneten Randwerte übergeben.

```
[A_1(:,i), Q_1(:,i)]=rhs_loesung(coordinates_1, elements_1, dirichlet_1,
    alpha, KR, rho, h0, E0, beta, t, dt, tol, A0_1, A_1(:,i-1), Q_1(:,i-1),
     uA1_1, uQ1_1, uAL_1, uQL_1, S_1, T_1);
[A_2(:,i), Q_2(:,i)]=rhs_loesung(coordinates_2, elements_2, dirichlet_2,
    alpha, KR, rho, h0, E0, beta, t, dt, tol, A0_2, A_2(:,i-1), Q_2(:,i-1),
     uA1_2, uQ1_2, uAL_2, uQL_2, S_2, T_2);
[A_3(:,i), Q_3(:,i)]=rhs_loesung(coordinates_3, elements_3, dirichlet_3,
    alpha, KR, rho, h0, E0, beta, t, dt, tol, A0_3, A_3(:,i-1), Q_3(:,i-1),
     uA1_3, uQ1_3, uAL_3, uQL_3, S_3, T_3);

end
```

Somit sind die Berechnungen eines Zeitschritts beendet und die Schleife kann von neuem durchlaufen werden.

Die Ergebnisse werden beim Aufruf der Funktion rhs_loesung in entsprechenden

Matrizen gespeichert. Nach Beendigung der Simulation können wieder verschiedene Grafiken zur Visualisierung erzeugt werden.

Die Ergebnisse von Simulationen verschiedener Szenarien sind im nächsten Kapitel zusammengefasst.

Kapitel 6

Simulationsergebnisse

In den folgenden Experimenten wird als proximaler Eingang jeweils eine Druckfunktion vorgegeben. Dabei wird immer eine halbe Sinusschwingung verwendet. Die Länge der Schwingung unterscheidet sich zwischen den Experimenten, die Amplitude ist dagegen bei allen Eingangsfunktionen die gleiche, nämlich $2 \cdot 10^4\ dyn/cm^2$.

Die weiteren Annahmen gelten für alle kommenden Experimente, außer es wird dort auf eine Änderung hingewiesen.
Das betrachtete Arterienstück hat eine Länge von $10\ cm$. Bei der Diskretisierung wird eine konstante Elementgröße von $h = 0.1\ cm$ verwendet. Der Zeitschritt Δt beträgt $1 \cdot 10^{-4}\ s$.

Außerdem gilt:

$$\begin{aligned}
\alpha =& 1 \\
\nu =& 0.035\ Poise \\
K_R =& 8\pi\nu \\
\rho =& 1\ g/cm^3 \\
h_0 =& 0.05\ cm \\
E_0 =& 3 \cdot 10^6\ dyn/cm^2 \\
R_0 =& 0.5\ cm
\end{aligned}$$

Bei den ersten Tests wird ein Eingangsimpuls der Länge $0.0025\ s$ benutzt. Dabei soll vor allem untersucht werden, wie sich die Veränderung von h bzw. Δt auf das Simulationsergebnis auswirkt.

In [7] wird folgende Stabilitätsbedingung angegeben:

$$\frac{\Delta t \sup_{0<z<L} \left(\max(|\lambda_1|, |\lambda_2|)\right)}{h} < \frac{1}{\sqrt{3}} \tag{6.1}$$

Die Eigenwerte liegen am Beginn der Berechnungen betragsmäßig im Bereich von $380\ cm/s$. Daraus ergibt sich bei $h = 0.1\ cm$ eine obere Grenze für den Zeitschritt von $\Delta t \approx 1.5 \cdot 10^{-4}\ s$.

Bei Simulationen mit $\Delta t = 2 \cdot 10^{-4}\ s$ konnten keine Ergebnisse erzielt werden, mit $\Delta t = 1 \cdot 10^{-4}\ s$ hingegen schon. Die angegebene Stabilitätsbedingung konnte somit auf diese Weise bestätigt werden. Daraus ergibt sich weiters, dass $\Delta t\,|\lambda| < h$ für die Berechnung erfüllt sein muss. Daher liegen die für die Extrapolation der charakteristischen Variablen notwendigen Zwischenpunkte (siehe Kapitel 5.4) immer in den äußersten Elementen der Zerlegung. Der Code zur Auswertung der Zustandsgrößen an diesen Zwischenpunkten kann daher vereinfacht werden.

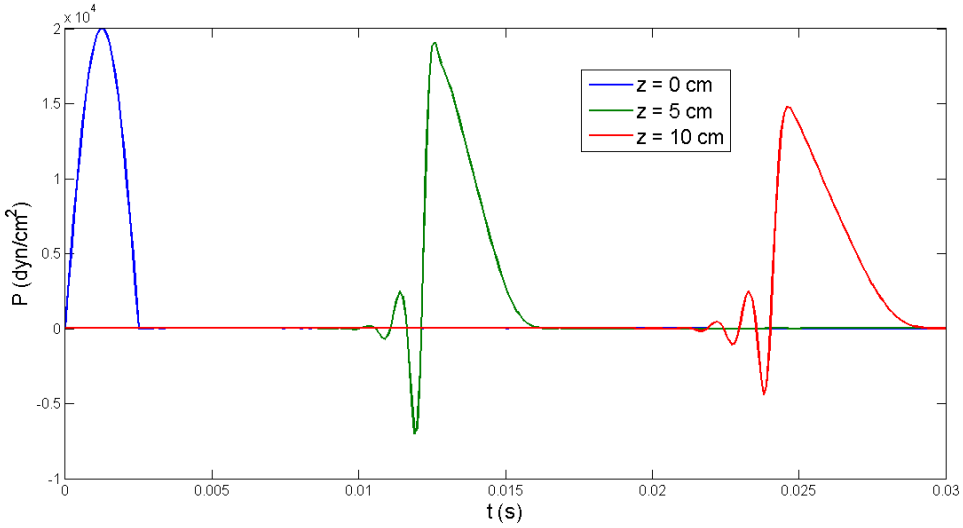

Abbildung 6.1: Druckverlauf an drei Punkten der Arterie

In Abbildung 6.1 ist der zeitliche Druckverlauf an drei Punkten der Arterie ($z = 0\ cm$, $z = 5\ cm$ und $z = 10\ cm$) dargestellt. Dabei erkennt man, dass es am Beginn der Druckänderung zu Oszillationen kommt, die sowohl auf die Breite als auch die Höhe des Sinusimpulses mit Fortdauer der Druckweiterleitung Einfluss nehmen.

Verkleinert man den Zeitschritt, so werden die Schwingungen sogar noch stärker, wie in Abbildung 6.2 ersichtlich ist. Da eine merkbare Vergrößerung des Zeit-

schritts bei fixer Elementgröße wegen der Stabilitätsbedingung nicht mehr möglich ist, stellt daher die Wahl von $\Delta t = 1 \cdot 10^{-4}\ s$ bereits die beste Möglichkeit dar.

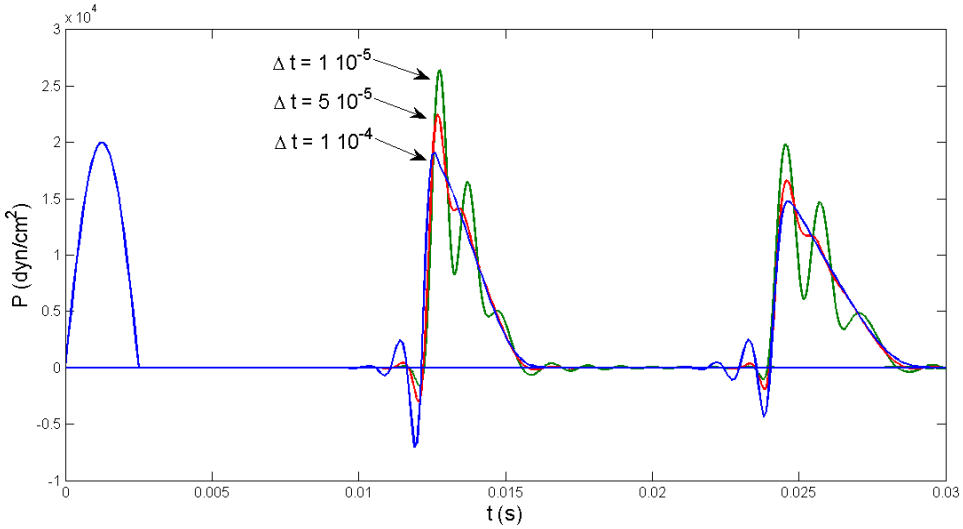

Abbildung 6.2: Vergleich des Drucks bei unterschiedlichen Zeitschritten

In einem zweiten Versuch wird der Zeitschritt bei $1 \cdot 10^{-4}\ s$ belassen, hingegen wird die Elementgröße h der Ortsdiskretisierung verändert. Dabei kann h nicht verkleinert werden, weil sonst die obige Stabilitätsbedingung verletzt wird und keine numerische Lösung möglich ist. Wird h vergrößert, so nehmen die Schwingungen ab, allerdings wird die Amplitude der Eingangskurve noch stärker verringert und es treten vermehrt Oszillationen nach dem eigentlichen Impuls auf (siehe Abbildung 6.3).

Vergleicht man die Simulationsergebnisse mit den in [7] veröffentlichten Ergebnissen, so kann eine Übereinstimmung bei der Ausbreitungsgeschwindigkeit festgestellt werden.

Für das nächste Experiment wird eine andere Druckkurve als Eingangsfunktion verwendet, deren Impulslänge $0.165\ s$ beträgt. Dies ist aus physiologischer Sichtweise realistischer als der kurze Impuls. Ein Vergleich zwischen den beiden Druckfunktionen zeigt Abbildung 6.4.

Durch die längere Periodendauer ist die Steigung der Kurve viel geringer. Daher treten bei der Simulation keine Oszillationen mehr auf. Das Ergebnis ist in Abbildung 6.5 ersichtlich, wo der Druckverlauf bei $z = 0\ cm$, $z = 5\ cm$ und $z = 10\ cm$ dargestellt ist. Vergleicht man dieses Resultat wieder mit den Simulationsergebnissen in [7], so kann neuerlich deren Gleichartigkeit bemerkt werden.

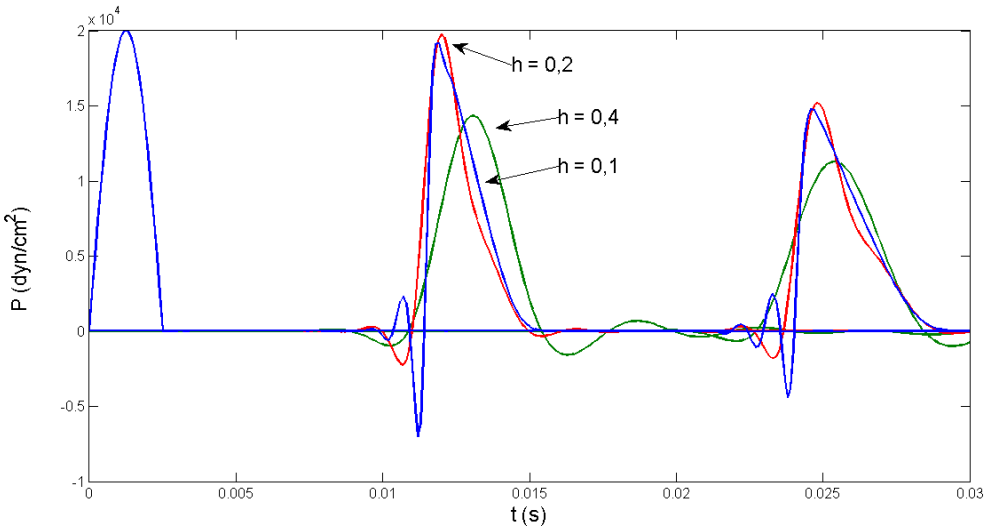

Abbildung 6.3: Vergleich des Drucks bei unterschiedlichen Elementgrößen

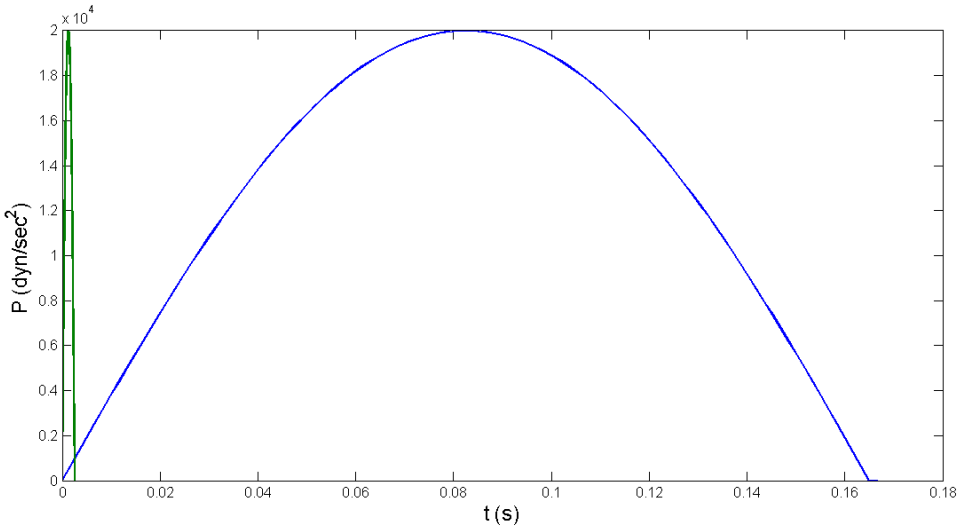

Abbildung 6.4: Vergleich zweier Druckfunktionen

In einem dritten Experiment wird das Verhalten bei einer Bifurkation untersucht. Dazu werden drei Arterienstücke mit einer Länge von je $10\ cm$ bei einer Verzweigung vereinigt. Die proximale Arterie hat dabei einen Referenzradius R_0 von $0,5\ cm$, die beiden weiteren Gefäße besitzen einen Anfangsradius von $0,4\ cm$ bzw. $0,3\ cm$.

Zuerst wird als Eingang wieder der kurze Impuls mit einer Länge von $0,0025\ s$ benutzt. Es treten dabei abermals die unerwünschten Schwingungen auf. Aller-

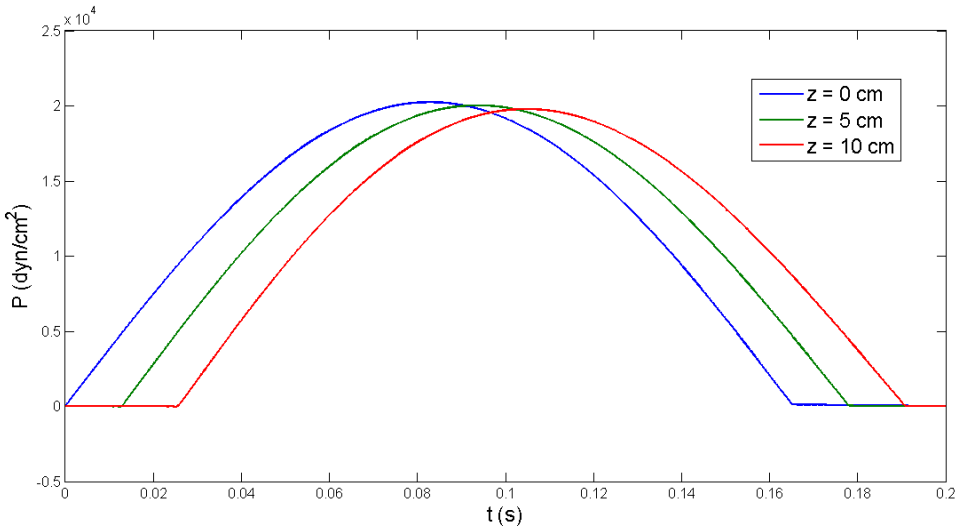

Abbildung 6.5: Druckverlauf an drei Punkten der Arterie

dings kann man bei dieser Simulation die Druckreflexionen an der Bifurkation gut erkennen. Dazu zeigt Abbildung 6.6 den Druckverlauf über die Zeit an zwei Punkten ($z = 3\ cm$, $z = 7\ cm$). Zuerst ereicht der Druckimpuls $z = 3\ cm$ (roter Kurvenausschlag), danach $z = 7\ cm$ (blau). Etwas später verlässt der Impuls die Arterie bei der Verzweigung, dabei kommt es zu einer Reflexion der Druckwelle. Diese kleinere Druckwelle setzt sich entgegen der ursprünglichen Ausbreitungsrichtung fort und erreicht zuerst wieder $z = 7\ cm$ (kleiner blauer Kurvenausschlag), um weiter in proximale Richtung zu wandern und etwas später $z = 3\ cm$ (rot) zu passieren.

Um die Simulationsergebnisse mit den in [5] dargestellten Ergebnissen vergleichen zu können, wird als Nächstes ein Eingangsimpuls der Länge $0,1\ s$ verwendet. Betrachtet man wieder den Druckverlauf in der ersten Arterie, so ist die Druckreflexion auch hier zu erkennen. Bei $z = 0\ cm$ verlängert sich der Druckimpuls durch die rücklaufende Druckwelle. Bei $z = 10\ cm$ fällt die Reflexion noch zur Gänze in die hinlaufende Druckwelle, als Auswirkung ist ein erhöhter Druck beobachtbar, in der Mitte der Arterie ist eine Kombination dieser beiden Effekte zu sehen (siehe Abbildung 6.7).

In den Arterien zwei und drei zeigen die Druckkurven das gleiche Verhalten, wie es beim vorigen Experiment mit einem einzigen Arterienstück sichtbar war (siehe Abbildungen 6.8 und 6.9).

Als Nächstes werden die charakteristischen Variablen in den drei Arterienstücken betrachtet. Man erkennt dabei, dass sich die Werte der charakteristischen Varia-

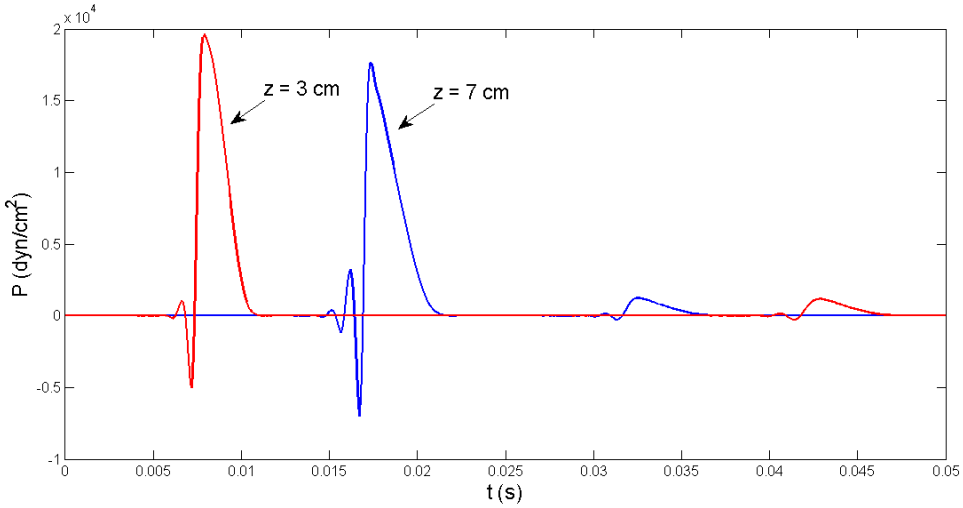

Abbildung 6.6: Druckverlauf an zwei Punkten der ersten Arterie

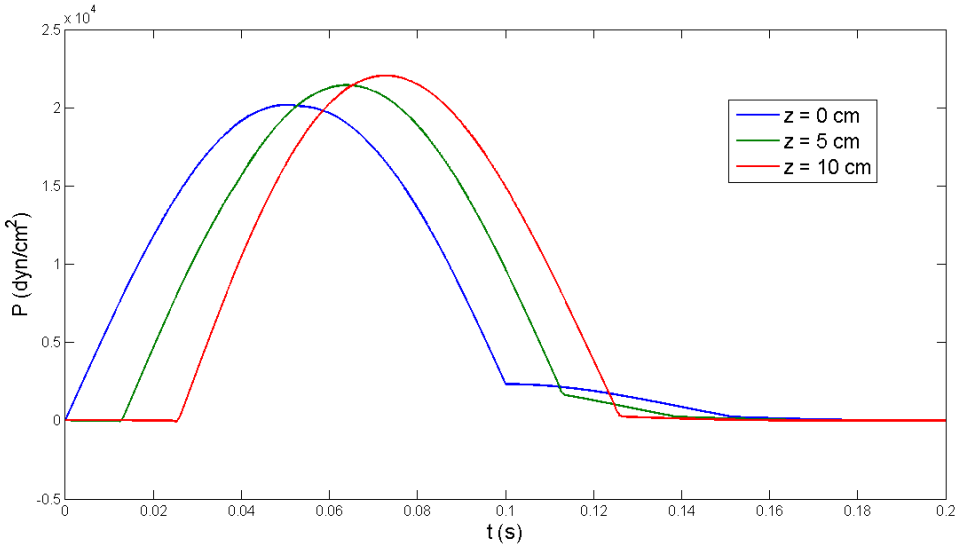

Abbildung 6.7: Druckverlauf an drei Punkten der ersten Arterie

blen, die eine Geschwindigkeit als Einheit besitzen, bei den engeren Arterien betragsmäßig erhöhen. Zur Verdeutlichung werden in den Abbildungen 6.10 und 6.11 die charakteristischen Variablen W_1 bzw. W_2 jeweils am Mittelpunkt des Arterienstücks ($z = 5\ cm$) graphisch über der Zeit dargestellt.

Man beachte, dass in Abbildung 6.10 die charakteristischen Variablen in allen drei Abschnitten einen ähnlichen Verlauf zeigen. Bei den charakteristischen Variablen entlang der rücklaufenden Charakteristiken in Abbildung 6.11 ist zu sehen, dass

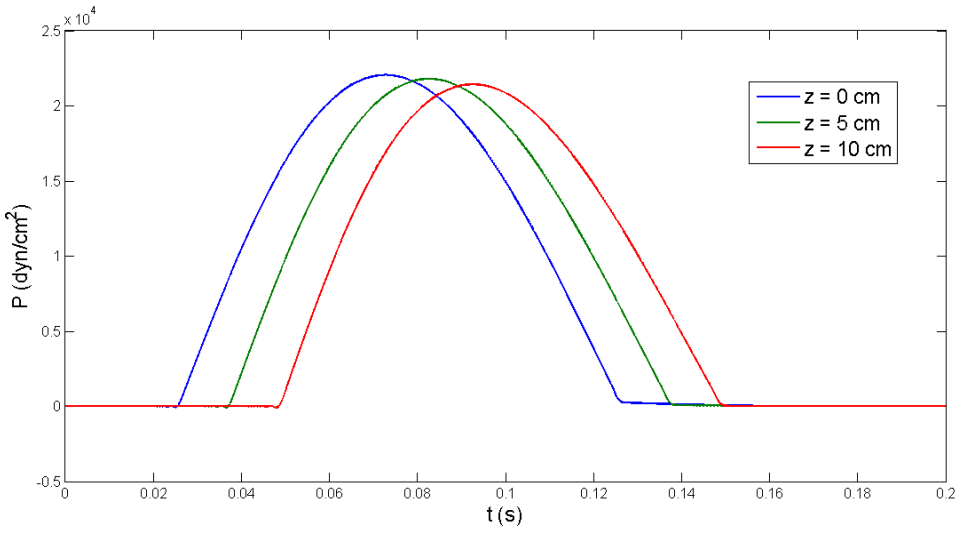

Abbildung 6.8: Druckverlauf an drei Punkten der zweiten Arterie

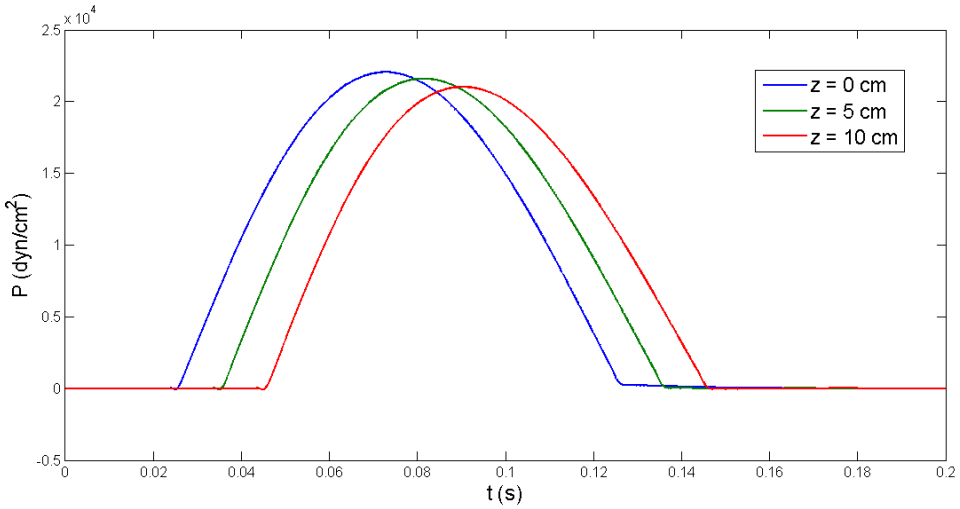

Abbildung 6.9: Druckverlauf an drei Punkten der dritten Arterie

die Werte in den beiden Arterienstücken nach der Verzweigung nahezu konstant bleiben. In der ersten Arterie hingegen ist eine Änderung der charakteristischen Variable sichtbar, in der sich die rücklaufende Druckwelle widerspiegelt.

Die Abbildung 6.12 zeigt den Volumenfluss in der Mitte der drei Arterien ($z = 5\ cm$) über die Zeit. Man erkennt, dass sich Q bei der Verzweigung in die beiden weitergehenden Gefäße aufteilt, dabei fließt mehr Blut durch die größere der

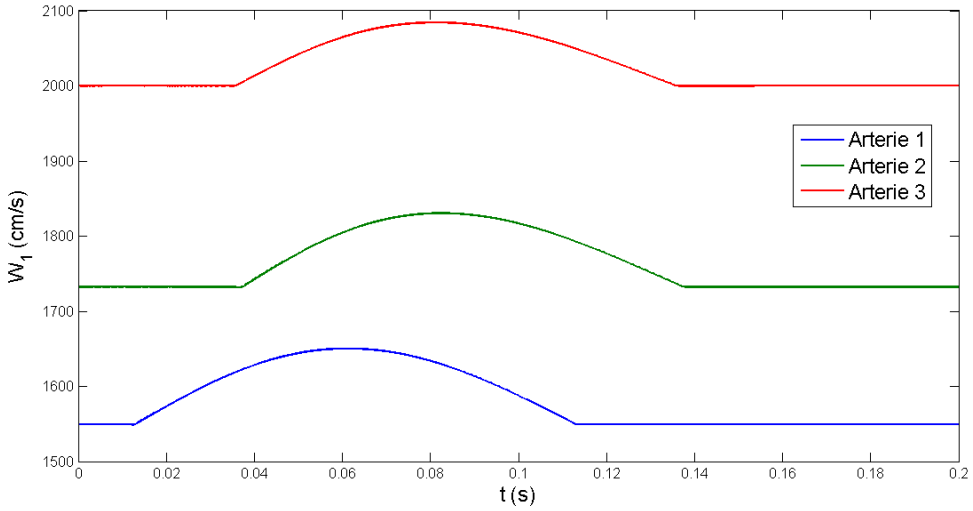

Abbildung 6.10: Vergleich der charakteristischen Variablen W_1 bei Arterienmitte

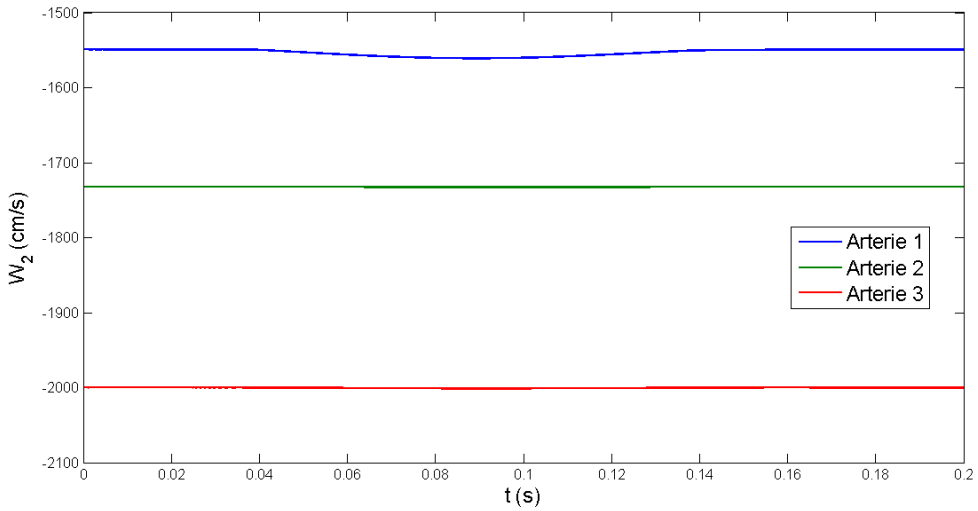

Abbildung 6.11: Vergleich der charakteristischen Variablen W_2 bei Arterienmitte

zwei Arterien. Man kann auch feststellen, dass der Fluss in der kleineren Arterie geringfügig früher bei $5\ cm$ eintrifft, die Ausbreitungsgeschwindigkeit ist also etwas höher. In der Flusskurve der proximalen Arterie ist ein geringer negativer Fluss sichtbar. Dies ist der Rückfluss, der durch die Reflexion an der Bifurkation entsteht.

Die Simulationsergebnisse stimmen sehr gut mit den Grafiken überein, die in [5] veröffentlicht wurden. So konnte die Implementierung des Blutflussmodells auf

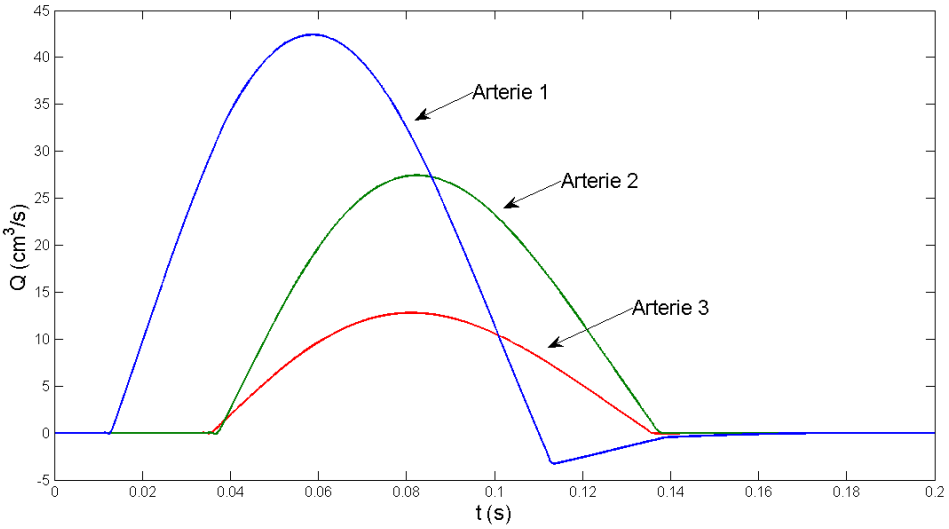

Abbildung 6.12: Volumenfluss an den drei Arterienstückmittelpunkten

diese Art verifiziert werden.

Kapitel 7

Ausblick

Das in den letzten Kapiteln vorgestellte Blutflussmodell und die dazugehörige Implementierung sollen als Grundlage für weitere Arbeiten dienen.

Eine erste mögliche Weiterentwicklung stellen Änderungen der Randwerte dar. Am distalen Ende des Arterienstücks können die Randbedingungen so gestaltet werden, dass sie nicht mehr ein unendlich langes Rohr simulieren, sondern den peripheren Widerstand des Gefäßsystems widerspiegeln.

Auch am proximalen Ende ist es sinnvoll, die Eingangskurve den realen Gegebenheiten im menschlichen Körper besser anzupassen. Dazu kann man die Randwerte mit Hilfe eines anderen Simulationsmodells, beispielsweise einem Modell des Herzens, vorgeben. Eine andere Möglichkeit stellt die Generierung einer Eingangskurve aus Messdaten dar.

Dieses Blutflussmodell für Arterien kann in verschiedenen Bereichen zur Anwendung gelangen. So können zum Beispiel Auswirkungen von Arterienveränderungen untersucht werden. Darunter fallen vom Körper selbst ausgebildete Veränderungen wie die Versteifung der Arterienwände oder die Verengung von Gefäßen. Es können aber auch chirurgische Maßnahmen, wie die Einsetzung eines Stents, simuliert werden. Dadurch kann eventuell schon vor einer Operation die bestmögliche Lage eines Stents festgestellt, und so die medizinische Versorgung des Patienten verbessert werden.

Neben diesen eher lokalen Betrachtungen dient das Modell aber vor allem der bereits in Kapitel 1.1 erwähnten Simulation der großen Arterien im Rahmen eines identifizierbaren und geregelten Herzkreislaufmodells. In der Arbeit von Johannes Kropf [8] wird für die Simulation des Blutflusses in den größeren Arte-

rien ein Modell benutzt, das mit Hilfe eines Finiten Volumen Verfahrens gelöst wird. Die hier vorliegende Arbeit bietet nun eine Alternative dazu. Es besteht die Möglichkeit, die beiden Teilmodelle im Rahmen des vollständigen Herzkreislaufmodells zu vergleichen und das bisherige Teilmodell eventuell zu ersetzen.

Abbildungsverzeichnis

1.1 Schema einer Herzkreislaufsimulation 2

1.2 Simulationskreislauf 5

1.3 Arterienstämme des Körperkreislaufs 7

1.4 Blutkreislaufschema 8

1.5 Druck- und Strompulsformen in verschiedenen Arterien 10

1.6 Blutdruck, Gesamtquerschnitt und mittlere Strömungsgeschwindigkeit im kardiovaskulären System 10

1.7 Eigenschaften der Gefäßabschnitte 11

2.1 Hutfunktion 20

3.1 Vergleich zwischen exakter und approximierter Lösung des Modellproblems 31

4.1 Darstellung eines Arterienstücks als zylindrisches Rohr 37

5.1 Skizze zur Modellierung einer Bifurkation 68

6.1 Druckverlauf an drei Punkten der Arterie 74

6.2 Vergleich des Drucks bei unterschiedlichen Zeitschritten 75

6.3 Vergleich des Drucks bei unterschiedlichen Elementgrößen 76

6.4 Vergleich zweier Druckfunktionen 76

6.5 Druckverlauf an drei Punkten der Arterie 77

6.6 Druckverlauf an zwei Punkten der ersten Arterie 78

6.7 Druckverlauf an drei Punkten der ersten Arterie 78

6.8 Druckverlauf an drei Punkten der zweiten Arterie 79

6.9 Druckverlauf an drei Punkten der dritten Arterie 79

6.10 Vergleich der charakteristischen Variablen W_1 bei Arterienmitte . . 80

6.11 Vergleich der charakteristischen Variablen W_2 bei Arterienmitte . . 80

6.12 Volumenfluss an den drei Arterienstückmittelpunkten 81

Literaturverzeichnis

[1] CARSTENSEN, CARSTEN: *A finite element implementation for 1d-problems in Matlab*. Unterlagen zur Vorlesung Adaptive Finite Elemente Methode, Technische Universität Wien.

[2] CARSTENSEN, CARSTEN, STEFAN A. FUNKEN und JOCHEN ALBERTY: *Remarks around 50 lines of Matlab: Short finite element implementation*. Numerical Algorithms, 20(2-3):117–137, 1999.

[3] DESPOPOULOS, AGAMEMNON und STEFAN SILBERNAGL: *Color Atlas of Physiology*. Thieme, Stuttgart, Deutschland, 5. Auflage, 2003.

[4] ERIKSSON, KENNETH, DON ESTEP, CLAES JOHNSON und PETER HANSBO: *Computational Differential Equations*. Cambridge University Press, New York, NY, USA, 1996.

[5] FORMAGGIA, LUCA, DANIELE LAMPONI und ALFIO QUARTERONI: *One-dimensional models for blood flow in arteries*. Journal of Engineering Mathematics, 47(3-4):251–276, Dezember 2003.

[6] FORMAGGIA, LUCA, DANIELE LAMPONI, MASSIMILIANO TUVERI und ALESSANDRO VENEZIANI: *Numerical Modeling of 1D Arterial Networks Coupled with a Lumped Parameters Description of the Heart*. Computer Methods in Biomechanics and Biomedical Engineering, 9(5):273–288, 2006.

[7] FORMAGGIA, LUCA, FABIO NOBILE und ALFIO QUARTERONI: *A One Dimensional Model for Blood Flow: Application to Vascular Prosthesis*. Mathematical Modeling and Numerical Simulation in Continuum Mechanics, 19:137–153, 2001.

[8] KROPF, JOHANNES: *Multiscale Blood Flow Modelling and Simulation*. Doktorarbeit, Technische Universität Wien, 2007.

[9] MELENK, JENS MARKUS: *Notizen zur Vorlesung Finite Elemente Methoden*. Technische Universität Wien, 2006.

[10] QUARTERONI, ALFIO und LUCA FORMAGGIA: *Mathematical Modelling and Numerical Simulation of the Cardiovascular System*. In: AYACHE, N. und P. G. CIARLET (Herausgeber): *Handbook of Numerical Analysis*, Band Special Volume: Computational Models for the Human Body, Seiten 3–127. Elsevier B. V., 2004.

[11] SCHMIDT, ROBERT, FLORIAN LANG und GERHARD THEWS: *Physiologie des Menschen mit Pathophysiolgie*. Springer Medizin Verlag, Heidelberg, Deutschland, 29. Auflage, 2005.

[12] SHERWIN, SPENCER J., LUCA FORMAGGIA, JOAQUIM PEIRÓ und VICTORIA FRANKE: *Computational modelling of 1D blood flow with variable mechanical properties and its application to the simulation of wave propagation in the human arterial system*. International Journal for numerical mehtods in fluids, 43(6-7):673–700, 2003.

[13] SOLIN, PAVEL: *Partial Differential Equations and the Finite Element Method*. John Wiley & Sons, Inc., Hoboken, NJ, USA, 2006.

[14] THEWS, GERHARD: *Anatomie, Physiologie, Pathophysiologie des Menschen*. Wissenschaftliche Verlagsgesellschaft mbH, Stuttgart, Deutschland, 5. Auflage, 1999.

[15] TROCH, INGE und FELIX BREITENECKER: *Modellbildung und Simulation*. Technische Universität Wien, 2004.

[16] URBAS, ELFRIEDE, BARBARA LEITNER, MONIKA HACKL, JEANETTE KLIMONT und CHRISTIAN VUTUC: *Österreichischer Todesursachenantlas 1998/2004*. Statsitik Austria, 2008.

Index

A_0 (Referenzfläche), 39
α (Coriolis Koeffizient), 38
β, 39
h_0, 39

A (Querschnittsfläche), 38

Bifurkation, 68
Bilinearform, 12
Blutkreislauf, 6

Céa Lemma, 19
Cauchy Problem, 33
Charakteristik, 33
charakteristische Kurve, 33
charakteristische Variable, 34

diskretes Problem, 16
Druckpuls, 9

Elastizitätsmodul, 39
Energienorm, 13
Erhaltungsgröße, 32

Flussfunktion, 32

Galerkin-Verfahren, 15

Hutfunktion, 20
hyperbolisch, 33

koerziv, 13
Kontinuitätsgleichung, 36

Lastvektor, 17
Lax-Milgram Lemma, 13
Linearform, 12

Modellproblem, 14, 21
Momentengleichung, 36

Navier-Stokes-Gleichungen, 36

Orthogonalität des Fehlers, 18

P (Druck), 38

Q (Volumenfluss), 38
Quellterm, 32
Querschnittpuls, 9
Querschnittsfläche, 38

schwache Formulierung, 14
Sobolev Raum, 15
Steifigkeitsmatrix, 17
Strompuls, 9

V-elliptisch, 13
Verzweigung, 68
Viskosität, 7

Wellenreflexionen, 9

Youngscher Modul, 39