



# Visualisierung von Simulationsensembles zur 4D- Phasenkontrast-MR-Datenassimilation

Masterarbeit vorgelegt von:

**Simon Leistikow**

Matrikelnummer: 406 838

Studiengang: M.Sc. Informatik

Thema gestellt von und betreut durch:

**Prof. Dr. Lars Linsen**

Zweitgutachter:

**Jun.-Prof. Dr. Benjamin Risse**

Münster, 27. August 2019



# Inhaltsverzeichnis

<b>1</b>	<b>Einleitung</b>	<b>1</b>
<b>2</b>	<b>Grundlagen</b>	<b>3</b>
2.1	Phasenkontrast-Magnetresonanz . . . . .	3
2.2	Numerische Strömungssimulationen . . . . .	5
2.2.1	Die Boltzmann-Gleichung . . . . .	6
2.2.2	Die Lattice-Boltzmann-Methode . . . . .	8
2.2.3	Randbedingungen . . . . .	10
2.2.4	Parametrisierung . . . . .	11
2.3	Klassische Multidimensionale Skalierung . . . . .	14
2.4	Das Voreen-Framework . . . . .	16
<b>3</b>	<b>Verwandte Arbeiten</b>	<b>19</b>
<b>4</b>	<b>Methodik</b>	<b>21</b>
4.1	Quantifizierung von Ähnlichkeit . . . . .	21
4.2	Die Analyseprozedur . . . . .	30
4.2.1	Vorverarbeitung . . . . .	30
4.2.2	Segmentierung . . . . .	32
4.2.3	Simulation . . . . .	33
4.2.4	Nachbearbeitung . . . . .	38
4.2.5	Analyse . . . . .	39
4.3	Die Anwendung . . . . .	41
<b>5</b>	<b>Ergebnisse</b>	<b>43</b>
5.1	Der Stabphantom-Datensatz . . . . .	44
5.2	Der Aneurysma-Phantom-Datensatz . . . . .	48
5.3	Diskussion . . . . .	50
<b>6</b>	<b>Fazit und Ausblick</b>	<b>51</b>



# 1 Einleitung

Die Analyse von Blutflussdaten spielt eine wichtige Rolle bei der Diagnose von Krankheiten des kardiovaskulären Systems und bei der Planung von deren Behandlung. Dafür werden zunehmend Daten mittels Phasenkontrast-Magnetresonanz (MR) generiert, welche räumliche und zeitliche Informationen über Richtung und Magnitude der lokalen Geschwindigkeit erfassen. Obwohl diese Technik in den letzten Jahren stets weiterentwickelt wurde, sind die Daten weiterhin aufgrund diverser Einschränkungen nicht oder nur bedingt für medizinisch relevante Aussagen betroffener Patienten verwendbar. Grund dafür ist die geringe zeitliche und räumliche Auflösung, zwischen denen technisch bedingt abgewägt werden muss.

Um das Problem zu umgehen und somit hochaufgelöste Daten zu erhalten, werden numerische Strömungssimulationen verwendet. Gegenstand der Messung sind zumeist Gefäße, nicht selten mit anatomischen Besonderheiten wie Abzweigungen und Aneurysmen. Deren innere Oberfläche wird als Simulationsgeometrie am Computer nachgebildet. Anschließend wird ein Modell für die Simulation gewählt, auf das in Kapitel 2 eingegangen wird. Dieses unterliegt jedoch einigen Annahmen und Vereinfachungen, sodass die Realität nur bedingt abgebildet werden kann. Dadurch können die simulierten Daten nicht für die Behandlung von Patienten herangezogen werden. Gegenstand aktueller Forschung ist daher, sowohl gemessene als auch simulierte Daten gleichermaßen zu verwenden bzw. zu kombinieren, was üblicherweise und im Folgenden als Datenassimilation bezeichnet wird. Die so erhaltenen Daten sollen dabei die tatsächlich vorliegenden Bedingungen in sowohl räumlich als auch zeitlich ausreichend hoher Auflösung widerspiegeln. Dies wird durch den Einsatz von Strömungssimulationen ermöglicht. Deren Durchführung ist zwar zeitlich aufwendig, jedoch steht dem eine angestrebte Reduzierung der Messdauer am lebenden Objekt gegenüber.

Es existieren bereits einige Ansätze zur Assimilation von Blutflussdaten, deren Kernbestandteil beispielsweise die Steuerung von Simulationen durch die Verwendung der gemessenen Daten ist. Diesbezüglich wird in Kapitel 3 ein Überblick über den aktuellen Forschungsstand gegeben. Ein grundsätzliches Problem ist weiterhin die Evaluation der Resultate aufgrund des Mangels an grundwahren Daten. Gemessene Daten bilden zwar die Wirklichkeit ab, sind allerdings aus genannten Gründen nicht als Referenz zu betrachten, ebenso wenig die rein simulierten Daten. Aus diesem Grund und durch die Vielzahl möglicher Werte ist das Parametrisieren des gewählten Modells kein triviales Unterfangen. Im Rahmen dieser Arbeit wird daher der Fokus auf die Wahl und

## 1 Einleitung

die Parametrisierung des Modells gelegt, um eine erste Vergleichbarkeit zu den gemessenen Daten schaffen zu können. Das so ermittelte Modell kann dann weiterverwendet werden, um eine fortgeschrittene Assimilation durchzuführen.

Um ein erstes Verständnis der Auswirkungen verschiedener Parameterkonfigurationen auf den Simulationsverlauf zu erlangen, werden die simulierten Daten zunächst unabhängig von den gemessenen Daten betrachtet. Da stetig neue Sequenzen zur Gewinnung von Phasenkontrast-MR-Daten entwickelt werden, wird ebenfalls untersucht, inwiefern sich Ensembles von gemessenen Daten unterscheiden bzw. ähneln. Der nächste Schritt ist dann die Zusammenführung der gemessenen und simulierten Ensembles. Zum Einsatz kommt dafür eine Ähnlichkeitsdarstellung [6], welche eine aggregierte Visualisierung der Ensembles darstellt, um auf einen Blick erste Vergleiche ziehen zu können. Diese konnten bereits erfolgreich auf räumlich und zeitlich veränderlichen Daten, insbesondere auf physikalischen Simulationen getestet werden [4, 19]. Grundlegend dafür ist klassische Multidimensionale Skalierung und verschiedene Metriken, die unterschiedliche Strömungseigenschaften der Daten erfassen. Da es sich um Vektorfelder handelt, werden die Unähnlichkeit der enthaltenen Richtungen und deren Magnituden sowohl getrennt voneinander betrachtet als auch in Kombination. Weitere interaktive Visualisierungen in mehreren koordinierten Ansichten geben Auskunft über den Ursprung der zuvor gezeigten Ähnlichkeiten. Dies und weitere Implementierungsdetails werden in Kapitel 4 besprochen und die erstellte Anwendung zur Generierung und Analyse von Simulationsensembles vorgestellt.

Die Implementierung erfolgt unter Verwendung von Voreen [22], ein quelloffenes Framework zur Volumenvisualisierung, vorgestellt in Unterkapitel 2.4, welches um eine Simulations- und Analyseprozedur erweitert wird. Die Strömungssimulationen werden dabei mit der Lattice-Boltzmann-Methode durchgeführt. Diese wird in Unterkapitel 2.2.2 erläutert.

Die entwickelten Methoden werden in Kapitel 5 auf zwei Datensätze unterschiedlicher Komplexität angewendet und ausgewertet, welche mittels Phasenkontrast-MR akquiriert wurden. Zusammenfassend soll im Rahmen dieser Arbeit unter Verwendung von Simulationsensembles und einer Ähnlichkeitsanalyse die Grundlage geschaffen werden, fortgeschrittene Techniken der Datenassimilation durchführen zu können. Dementsprechend folgt abschließend ein Ausblick in Kapitel 6.

## 2 Grundlagen

In diesem Kapitel werden zum Verständnis der Methodiken und Ergebnisse notwendige Grundlagen vorgestellt. Dabei wird auf die Technik zur Akquirierung der Daten, sowie auf die Lattice-Boltzmann-Methode und deren Parametrisierung eingegangen. Zum Zweck des Vergleichs von simulierten und gemessenen Daten wird die klassische Multidimensionale Skalierung erläutert. Da die Implementierung mithilfe des Voreen-Frameworks wesentlicher Bestandteil dieser Arbeit ist, wird es dazu ebenfalls eine Einführung geben, was zudem die Weiterentwicklung erleichtern soll.

### 2.1 Phasenkontrast-Magnetresonanz

Die zum Zeitpunkt und Zweck der Arbeit vorliegenden Daten wurden mittels Phasenkontrast-Magnetresonanz (MR) von der Arbeitsgruppe *Magnetresonanz* am Institut für Medizinische Mikrobiologie der Universität Jena generiert. Bei dieser Bildgebungstechnik führt das eingesetzte Magnetfeld zu einer gemeinsamen Ausrichtung der Protonen der Wasserstoffatome, sowohl in den Gefäßen als auch im umliegenden, stationären Gewebe. Über das Hauptmagnetfeld gelegte, bipolare Magnetfeldgradienten führen dann zu einer positionsabhängigen Phasenverschiebung der Protonen. Diese fällt durch das fließende Blut stärker innerhalb von Gefäßen aus als im stationären Gewebe. Die gemessene Phasenverschiebung hängt dabei unmittelbar von der Geschwindigkeit des Blutes ab und führt so in den Gefäßen zu einer hohen gemessenen Signalstärke. Bedingt durch den technischen Fortschritt im Bereich der Magnetresonanzbildgebung existieren heute mehrere Ausprägungen der Messtechnik. 2D-Phasenkontrast-MR wird im klinischen Umfeld bereits genutzt, um Blutfluss in einer 2D-Ebene zu messen, welche vor der Messung ausgerichtet wird. Eine quantitative Analyse der gewonnenen Daten durch das Auswerten der Flussraten, -geschwindigkeiten sowie dem gepumpten Blut pro Herzschlag erleichtert die Einschätzung der Herzfunktionen. Die Berechnungen sind jedoch stark von der zuvor gewählten Ausrichtung der Ebene abhängig [15]. Eine Erweiterung stellt daher die 3D- bzw. 4D-Phasenkontrast-MR dar, wobei in dadurch generierten Datensätzen für jede von drei orthogonalen Richtungen entsprechend der Auflösung Bilder der Phasen und Magnituden vorliegen. In zuletzt genannter Technik sind diese zudem zeitlich aufgelöst, sodass für jeden diskreten Zeitschritt ein 3D-Datensatz vorliegt, womit meist ein vollständiger Herzzyklus abgebildet wird. Die Daten werden dabei letztlich durch das Mit-

## 2 Grundlagen

teln mehrerer Herzzyklen gewonnen, welche durch ein Navigatorsignal synchronisiert werden. Die Phasenbilder werden dabei aus den Phasenunterschieden jeder Raumrichtung berechnet, von denen jedes den Geschwindigkeitsanteil der jeweiligen Richtung enthält. Dafür muss ein Parameter  $V_{enc}$  festgelegt werden, der die höchste erwartete Strömungsgeschwindigkeit je Richtung spezifiziert. Je höher dieser Parameter gewählt wird, desto größer ist der Kontrastverlust, weshalb in der Praxis zwischen der korrekten Detektion hoher Geschwindigkeiten und hohem Kontrast abgewägt werden muss. Die zuvor erwähnten, bipolaren Magnetfeldgradienten werden dann so angepasst, dass die maximale Phasenverschiebung der  $V_{enc}$  entspricht. Aus den Phasenbildern wird dann ein dreidimensionales, volumetrisches Vektorfeld der Geschwindigkeiten rekonstruiert. Diese enthalten auf einem regelmäßigen Gitter angeordnete Messpunkte, deren Anzahl für jede Dimension im Folgenden durch  $\vec{d} = (d_x, d_y, d_z)$  definiert sei. Der Abstand der Messpunkte in der physikalischen Domäne ist also nicht notwendigerweise für jede Dimension dieselbe, jedoch innerhalb jeder Dimension konstant und wird als *Spacing*  $\vec{s} = (s_x, s_y, s_z)$  definiert. Ferner wird der physikalische Abstand des ersten Messpunkts zum Ursprung des Koordinatensystems als *Offset*  $\vec{o} = (o_x, o_y, o_z)$  bezeichnet. Jeder Messpunkt enthält die lokale, gerichtete Geschwindigkeit  $\vec{u}$ , sowie deren Magnituden  $|\vec{u}|$  in  $\text{mm s}^{-1}$ . Da zwischen umliegenden Datenpunkten linear interpoliert oder der nächste benachbarte Datenpunkt verwendet werden kann, lassen sich die Daten als Funktionen auffassen. Damit seien die Geschwindigkeit  $\vec{u}$  und Magnitude  $|\vec{u}|$  durch

$$\vec{u} : \mathbb{R}^3 \rightarrow \mathbb{R}^3, \quad \text{bzw.} \quad |\vec{u}| : \mathbb{R}^3 \rightarrow \mathbb{R} \quad (2.1)$$

innerhalb der Domäne  $[o_x, o_y, o_z] \times [o_x + s_x \cdot d_x, o_y + s_y \cdot d_y, o_z + s_z \cdot d_z]$  definiert.

Bei der Rekonstruktion kann es mitunter zu Phasenumkehrungen<sup>1</sup> kommen, wenn  $V_{enc}$  zu niedrig gewählt wurde. Geschwindigkeiten, deren Magnitude größer ist als durch den Parameter spezifiziert, werden dann auf die andere Seite des Wertebereichs abgebildet. Es existieren allerdings Algorithmen, um derartige Artefakte bis zu einem gewissen Grad zu kompensieren [15]. Die Aufnahmen der Magnituden werden anatomische Bilder genannt und enthalten die ungerichtete Strömungsgeschwindigkeit. Sie sind weniger anfällig gegen unkorreliertes Rauschen als die Phasenbilder, weshalb diese verwendet werden, um beispielsweise Segmentierungen durchzuführen [15]. Diese Information wird ferner für die Implementierung in Kapitel 4 von Bedeutung sein. Bei den beschriebenen Messungen, wie auch im Allgemeinen, wird unterschieden zwischen solchen am lebenden Objekt (*in-vivo*) und an sogenannten Phantomen (*in-vitro*). Für zu behandelnde Patienten bzw. das Abbilden tatsächlichen Blutflusses sind Messungen am lebenden Objekt unabdingbar. Für Forschungszwecke ist dies jedoch meist zu teuer und aufwendig, bzw. ethisch fraglich, wenn Tiere als Messobjekt dienen. So wurden beispielsweise Mäuse für zwei der dieser Arbeit vorliegenden Datensätze etwa 60 Minuten narkotisiert und

---

<sup>1</sup>Phasenumkehrungen: von engl. *phase-wraps*



der Messprozedur unterzogen [2]. An diesem Punkt werden Phantome eingesetzt, insbesondere um neue Analysemethoden zu erforschen, ohne auf Messdaten verzichten zu müssen. Dabei kann es sich um abstrakte Gebilde handeln, wie etwa um einen Zylinder aus Acrylglas mit verschiedenen Einsätzen zur Untersuchung deren Strömungsbeeinflussung. Einen Kompromiss zwischen *in-vivo* und *in-vitro* Messungen können dagegen Nachbildungen tatsächlicher Gefäße und Gefäßstrukturen wie Aneurysmen darstellen. Diese werden am Computer anhand von Vorbildern modelliert und mit dem 3D-Drucker erstellt. Das Modell kann dann mit verschiedenen Materialien (Latex-Gel, Flüssiglatex) nachgeformt werden, um flexible Eigenschaften der Gefäßwände nachzuempfinden. Dies spiegelt jedoch nur bedingt die tatsächlichen Verhältnisse wider, da das umliegende Gewebe ebenfalls einen Einfluss auf die Bewegung der Gefäßwände hat, bzw. deren Eigenschaften nicht perfekt nachgebildet werden können. Überdies wird aus mehreren Gründen Wasser als Flüssigkeit für die *in-vitro* Messungen verwendet, anstatt menschliches Blut. Dabei wird sowohl einfaches Tafelwasser, als auch destilliertes Wasser, gemischt mit Glycerin, Natriumiodid und Natriumthiosulfat verwendet, um Eigenschaften des menschlichen Bluts nachzuempfinden [25].

## 2.2 Numerische Strömungssimulationen

Numerische Strömungssimulationen finden in vielen Bereichen mit den unterschiedlichsten Fluiden Verwendung, sei es im biomedizinischen Bereich, in der Industrie oder gar im Sport [27]. Sie sind notwendig zur Lösung der meisten Strömungsprobleme, da diese mit analytischen Methoden aufgrund der Komplexität der Randbedingungen nur sehr aufwendig oder gar nicht mehr lösbar sind. Grundlage für numerische Strömungssimulationen ist stets ein physikalisches Modell, welches für die Implementierung diskretisiert wird. In der Strömungsmechanik werden Fluide als makroskopische Phänomene betrachtet. Zwar wird das betrachtete Fluid als eine Menge von infinitesimalen Fluidelementen verstanden, diese sind jedoch noch immer weit größer als die enthaltenen Moleküle bzw. deren Abstände voneinander. Für jedes Fluidelement innerhalb eines Gesamtvolumens  $V$  wird eine Dichte  $\rho$ , der Druck  $p$  und eine Geschwindigkeit  $\vec{v}$  definiert. Unter Annahme eines nicht komprimierbaren Fluides, als welches Flüssigkeiten meist näherungsweise angenommen werden können, gilt  $\rho = \text{const}$  und es kann die inkompressible Navier-Stokes-Gleichung durch

$$\rho \dot{\vec{v}} = -\nabla p + \eta \Delta \vec{v} + \vec{F} \quad (2.2)$$

angegeben werden. Hierbei ist  $\Delta = \nabla \cdot \nabla$  der Laplace-Operator und  $\dot{\vec{v}}$  die partielle Ableitung der Geschwindigkeit des Fluidelements nach der Zeit. Die Gleichung enthält neben dem Einfluss externer Kräfte  $\vec{F}$ , wie etwa die Gravitation, noch die Scherviskosität  $\eta$ . Um diesen makroskopischen Ansatz in

## 2 Grundlagen

einer Simulation zu verwenden, müssen aufwendig Laplace-Gleichungen für jeden diskreten Zeitschritt gelöst werden [24]. Bei einem mikroskopischen Ansatz sind dagegen die kleinsten Partikel eines Fluides Gegenstand der Betrachtung. Es müssen also die verschiedensten Wechselwirkungen zwischen Molekülen betrachtet werden und daraus noch die notwendigen Informationen wie etwa der Druck abgeleitet werden. Dies stellt nicht zuletzt aufgrund der riesigen erforderlichen Menge von Partikeln oft keine sinnvolle Möglichkeit für eine durch Ressourcen begrenzte Computersimulation dar. Zwischen den beiden Skalen liegt die mesoskopische Betrachtung. Dafür werden mehrere Partikel als Einheit angenommen und deren gemeinsames Verhalten betrachtet. Diese Idee wird durch die Boltzmann-Gleichung aufgegriffen und in der Lattice-Boltzmann-Methode (LBM) implementiert. Die LBM vereint die Vorteile der mikro- und makroskopischen Betrachtung: sie lässt sich auf beliebig komplexe Simulationsdomänen anwenden und bildet wichtige Strömungsmerkmale wie Inkompressibilität und turbulente Strömungen ab. Vielmehr kann die Navier-Stokes-Gleichung unter gewissen Bedingungen aus der Boltzmann-Gleichung abgeleitet werden [24]. Wie bei jedem numerischen Verfahren muss auch bei der LBM von Ungenauigkeiten in der Berechnung ausgegangen werden, die von der Parametrisierung des Modells abhängig sind. Darauf wird in Unterkapitel 2.2.4 eingegangen. Bezüglich der Viskosität wird ferner zwischen newtonschen und nicht-newtonschen Fluiden unterschieden. Bei Blut handelt es sich um ein nicht-newtonsches Fluid, d. h. die Viskosität, also die Zähflüssigkeit, hängt von der auf das Fluid ausgeübten Kraft ab und ist insbesondere nicht konstant. Im Prinzip genügt kein Fluid exakt der Definition von newtonschen Fluiden, dennoch wird Wasser unter Normalbedingungen als ein solches angenommen. Selbst Blut genügt weitestgehend der Physik newtonscher Fluide und wird in der Literatur und im Folgenden als ein solches modelliert [23].

### 2.2.1 Die Boltzmann-Gleichung

Grundlage für die LBM ist die Boltzmann-Gleichung, deren Herleitung nun folgt. Fundamental für diesen mesoskopischen Ansatz ist eine Verteilungsfunktion  $f(\vec{x}, \vec{v}, t)$ , da Partikel Gegenstand der Betrachtung sind. Sie kann als eine Verallgemeinerung der Dichte  $\rho(\vec{x}, t)$  angesehen werden, welche ebenfalls die mikroskopische Partikelgeschwindigkeit  $\vec{v}$  berücksichtigt. Im Gegensatz zu  $\rho(\vec{x}, t)$ , welche die Massendichte im physikalischen Raum angibt, repräsentiert  $f(\vec{x}, \vec{v}, t)$  zusätzlich die Massendichte im Geschwindigkeitsraum. Sie gibt also die Dichte aller Partikel an Position  $\vec{x}$  zur Zeit  $t$  mit Geschwindigkeit  $\vec{v}$  an [18]. Wird von kräftefreien Gasparkeln in einem sonst leeren aber begrenzten Raum ausgegangen, kann angenommen werden, dass sich nach ausreichender Zeit eine bestimmte, als konstant anzunehmende Verteilung der Gasparkel bzw. Moleküle eingestellt hat. Das System befindet sich also im Gleichgewicht, weshalb die daraus resultierende Verteilung  $f^{eq}(\vec{x}, \vec{v}, t)$  auch *Gleichgewichtsverteilung* genannt wird. Die Funktionswerte der Verteilung sind

zu diesem Zeitpunkt nicht notwendig für das weitere Verständnis und werden daher als gegeben angenommen. Nun kann betrachtet werden, wie sich die Verteilungsfunktion  $f(\vec{x}, \vec{v}, t)$  über die Zeit entwickelt. Bildet man das totale Differential in Bezug zur Zeit  $t$  so erhält man

$$\frac{df}{dt} = \left( \frac{\partial f}{\partial t} \right) \frac{dt}{dt} + \left( \frac{\partial f}{\partial x_\alpha} \right) \frac{dx_\alpha}{dt} + \left( \frac{\partial f}{\partial v_\alpha} \right) \frac{dv_\alpha}{dt}. \quad (2.3)$$

Das  $\alpha$  denotiert dabei einen festen aber beliebigen Index aller vektoriellen Größen, also beispielsweise  $x_\alpha \in \{x_x, x_y, x_z\}$  und wird verwendet, da sich die so implizierten Gleichungen anderweitig nicht unterscheiden. Die rechte Seite kann mit  $\frac{dt}{dt} = 1$ ,  $\frac{dx_\alpha}{dt} = v_\alpha$  und  $\frac{dv_\alpha}{dt} = \frac{F_\alpha}{\rho}$  weiter vereinfacht werden zu

$$\frac{\partial f}{\partial t} + v_\alpha \frac{\partial f}{\partial x_\alpha} + \frac{F_\alpha}{\rho} \frac{\partial f}{\partial v_\alpha} = \Omega(f). \quad (2.4)$$

Dies ist die **Boltzmann-Gleichung**, wobei  $\Omega(f) = \frac{df}{dt}$  die übliche Notation für totale Differentiale darstellt [18]. Für kräftefreie Systeme, also  $F_\alpha = 0$ , vereinfacht sich die Gleichung weiter zu

$$\frac{\partial f}{\partial t} + v_\alpha \frac{\partial f}{\partial x_\alpha} = \Omega(f) \quad (2.5)$$

Zwar ist bei den Messungen immer auch die Gravitation als äußere Kraft vorhanden, die Kräfte sind jedoch vernachlässigbar klein in Relation zu den hohen Geschwindigkeiten der Flüssigkeit, welche die sehr kleinen Phantome in kurzer Zeit durchströmt. Ebenfalls existiert kein Temperaturgradient, der einen relevanten Effekt haben könnte.

Gleichung 2.5 kann nun als Advektionsgleichung verstanden werden. Die linke Seite beschreibt die Veränderung der Partikelverteilung in Abhängigkeit der Geschwindigkeiten ihrer Partikel und die rechte Seite die lokale Neuverteilung aufgrund von Kollisionen der Partikel. Aus diesem Grund wird der Term  $\Omega(f)$  auch Kollisionsoperator genannt. Kollisionen innerhalb der betrachteten Skalierung genügen der Newtonschen Mechanik und erhalten Masse, Impuls, sowie die Gesamtenergie und innere Energie [18].

In der LBM wird üblicherweise der *BKG*-Kollisionsoperator<sup>2</sup> verwendet. Dieser ist gegeben durch

$$\Omega(f) = -\frac{1}{\tau}(f - f^{eq}) \quad (2.6)$$

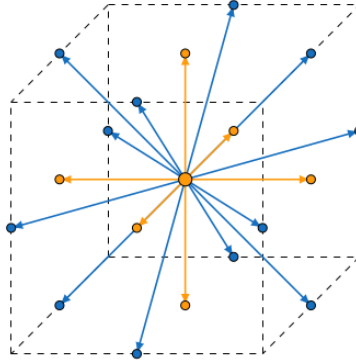
und beinhaltet die Relaxationszeit  $\tau$  von der gegenwärtigen Verteilung  $f$  zur Gleichgewichtsverteilung  $f^{eq}$ . Dies ist zugleich der simpelste Operator, der die oben genannten Größen erhält. Die Wahl der Relaxationszeit  $\tau$  spielt für die Stabilität und Genauigkeit von Simulationen eine große Rolle und wird in Abschnitt 2.2.4 erläutert. Der Operator wurde mit dem Smagorinsky-Modell

---

<sup>2</sup>BKG: benannt nach seinen Erfindern, Bhatnagar–Gross–Krook

## 2 Grundlagen

verbessert, sodass turbulente Strömungen auch bei geringen Auflösungen stabil simuliert werden können, welches somit auch für diese Arbeit verwendet wird [16, 26]. Dabei wird aus der molekularen Relaxationszeit  $\tau$  und einer turbulenten Komponente eine effektive Relaxationszeit zusammengesetzt. Für deren Berechnung muss ein Parameter, die sogenannte Smagorinsky-Konstante, festgelegt werden.



**Abbildung 2.1:** Das D3Q19-Schema [16]. Die umliegenden Zellen in Pfeilrichtung werden für den Informationsaustausch verwendet.

### 2.2.2 Die Lattice-Boltzmann-Methode

In der Lattice-Boltzmann-Methode (LBM) werden Gleichung 2.5 und 2.6 zunächst vereint zu

$$\frac{\partial f}{\partial t} + v_\alpha \frac{\partial f}{\partial x_\alpha} = -\frac{1}{\tau}(f - f^{eq}). \quad (2.7)$$

Ferner wird die Simulationsdomäne durch ein uniformes, regelmäßiges Gitter mit *Spacing*  $\Delta x$  diskretisiert. Die so entstehenden Einheiten des diskretisierten Raumes werden als Zellen bezeichnet. Diese müssen nun für den Austausch an Informationen, also Partikeln und deren Eigenschaften, in Beziehung zueinander gesetzt werden. Die übliche Nomenklatur zur Charakterisierung der Beziehungen ist dabei  $DnQm$ , wobei  $n \in \{1, 2, 3\}$  die Anzahl der Dimensionen angibt. Die Diskretisierung des Geschwindigkeitsraumes  $v$  wird durch  $m$  angegeben und gibt somit zeitgleich die Anzahl der Verbindungen zu benachbarten Zellen an. Die möglichen Werte für  $m$  sind abhängig von  $n$ . Für die vorliegenden Daten gilt also  $n = 3$  und für  $m$  kommen prinzipiell mehrere Möglichkeiten in Betracht, namentlich 13, 15, 19 und 27 [18]. Bei Letzterer besteht damit eine Verbindung zu jeder der 26 umliegenden Zellen, wobei zusätzlich die Informationen der Zelle selbst mitberücksichtigt werden. Die Laufzeit und der Speicherbedarf der Simulation steigen jedoch zusammen mit der Genauigkeit der Approximation in  $m$ . Als Kompromiss wird häufig und daher auch für diese Arbeit  $m = 19$  für  $n = 3$  gewählt [16]. Das sich so ergebende Schema  $D3Q19$  ist in Abbildung 2.1 dargestellt. Die Zellen repräsentieren eine

Partikelverteilung  $f_i(\vec{x}, t)$  mit  $i = 1..m$  auf mesoskopischer Skala. Durch das gleichmäßige Gitter kann die Annahme getroffen werden, dass innerhalb eines diskreten Zeitschrittes  $\Delta t$  jedes Partikel immer exakt eine der im Sinne des  $DnQm$ -Schemas benachbarten Zellen erreichen kann. Eine nicht uniforme bzw. nicht gleichmäßige Diskretisierung ist denkbar, jedoch nicht üblich, da insbesondere der sonst inhärente Vorteil der Parallelisierbarkeit entfällt. Gleichung 2.7 wird ferner als gültig entlang jeder der  $m$  definierten Richtungen angenommen, woraus sich

$$\frac{\partial f_i}{\partial t} + v_{i\alpha} \frac{\partial f_i}{\partial x_\alpha} = -\frac{1}{\tau}(f_i - f_i^{eq}) \quad (2.8)$$

ergibt. Diese kann mittels finiter Differenzen approximiert werden durch

$$\frac{f_i(\vec{x} + \vec{v}_i \Delta t, t + \Delta t) - f_i(\vec{x}, t)}{\Delta t} = -\frac{1}{\tau}(f_i - f_i^{eq}) \quad (2.9)$$

und umgestellt werden zu

$$f_i(\vec{x} + \vec{v}_i \Delta t, t + \Delta t) = f_i(\vec{x}, t) - \frac{\Delta t}{\tau}(f_i - f_i^{eq}). \quad (2.10)$$

Dies ist die sogenannte **Lattice-Boltzmann-Gleichung** in Kombination mit dem diskreten BKG-Kollisionsoperator [18, 24]. Obwohl es sich dabei eigentlich um eine Approximation der ersten Ordnung in Bezug zur Zeit handelt, bietet diese dennoch dieselbe Genauigkeit wie eine Approximation zweiter Ordnung [18].

Die so erhaltene Gleichung kann nun erneut in zwei logischen Komponenten zerlegt werden: die Kollisions- und Strömungsschritt<sup>3</sup>. Bei der Berechnung jedes Zeitschrittes  $\Delta t$  erhält also jede Partikelpopulation  $f_i(\vec{x}, t)$  einen Anteil aus dem Kollisions- bzw. Relaxationsschritt, also dem Anstreben der Gleichgewichtsverteilung  $f_i^{eq}(\vec{x}, t)$ . Dieser Schritt findet gänzlich lokal für jede Zelle statt, es müssen also keine Informationen aus umliegenden Zellen verwendet oder verändert werden. Der Zustand  $f_i^*(\vec{x}, t)$  der Verteilung nach dem Kollisionsschritt kann also durch

$$f_i^*(\vec{x}, t) = f_i(\vec{x}, t) - \frac{\Delta t}{\tau}(f_i(\vec{x}, t) - f_i^{eq}(\vec{x}, t)) \quad (2.11)$$

angegeben werden. Im Schritt des Propagierens verteilt sich die neue Partikelpopulation  $f_i^*(\vec{x}, t)$  dann auf die umliegenden, durch das  $DnQm$ -Schema definierten, Zellen. Für diese gilt somit

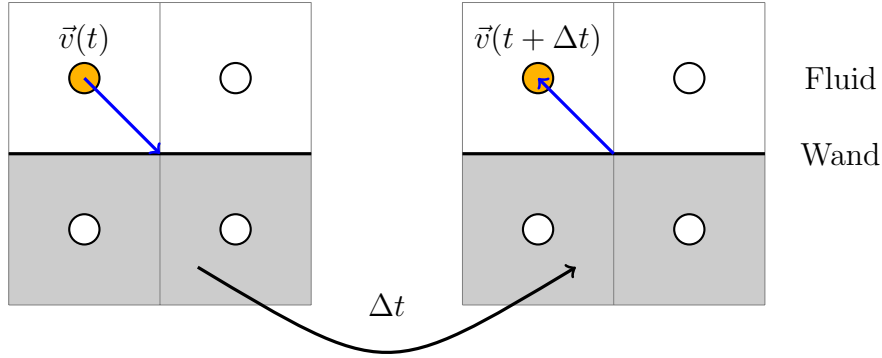
$$f_i(\vec{x} + \vec{v}_i \Delta t, t + \Delta t) = f_i^*(\vec{x}, t), \quad (2.12)$$

sodass es sich dabei um keine lokale Operation mehr handelt, da die umliegenden Zellen modifiziert werden [18]. Diese beiden Schritte können beliebig wiederholt werden, bis ein Abbruchkriterium die Simulation beendet.

---

<sup>3</sup>Strömungsschritt: von engl. *streaming step*

## 2 Grundlagen



**Abbildung 2.2:** Skizze eine Partikelpopulation (orange), welche mit einer starren Wand (grau) kollidiert, nachdem sie über eine Verbindung (blau) mit Geschwindigkeit  $\vec{v}$  in Richtung einer benachbarten Zelle propagiert. Bei dem Rückprallansatz gilt  $\vec{v}(t) = -\vec{v}(t + \Delta t)$ . Es findet also keine Bewegung unmittelbar entlang der Wand statt, womit Haftung modelliert wird. Entlehnt an [18].

Dabei können zu jedem Zeitschritt  $t$  die makroskopischen Größen, also die Massendichte  $\rho$  durch

$$\rho(\vec{x}, t) = \sum_{i=1}^m f_i(\vec{x}, t) \quad (2.13)$$

und die Geschwindigkeit  $\vec{u}$  durch

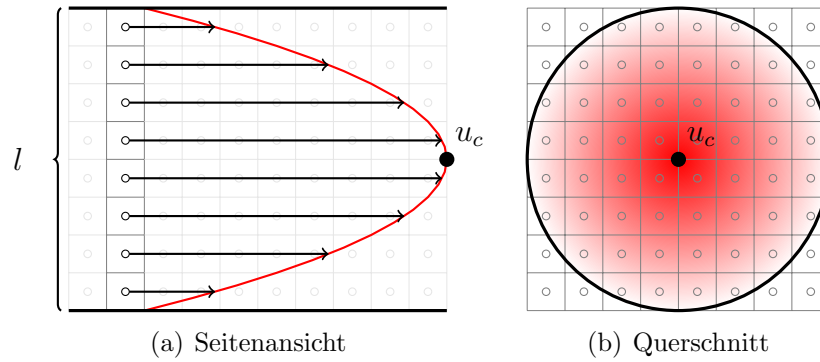
$$\vec{u}(\vec{x}, t) = \frac{1}{\rho(\vec{x}, t)} \sum_{i=1}^m \vec{v}_i \cdot f_i(\vec{x}, t) \quad (2.14)$$

aus der Partikelpopulation berechnet werden. Aus diesen lassen sich dann weitere Größen ableiten.

### 2.2.3 Randbedingungen

Ein weiterer Aspekt in der LBM sind Randbedingungen. Dabei sind zum einen das Verhalten der Partikel beim Kollidieren mit der Simulationsgeometrie, zum anderen die Bedingungen beim Ein- und Auslass der Partikel zu betrachten. Auch hierfür gilt es, jeweils ein möglichst geeignetes Modell für das vorliegende Problem zu wählen. Es kann die Eigenschaft festgehalten werden, dass die Simulationsgeometrie (Gefäßwand) im Allgemeinen beliebig ist. Sie verläuft also insbesondere weder genau durch das Zentrum, noch genau zwischen den Zellen. Bei der Kollision von Partikeln mit der Gefäßwand wird üblicherweise ein Rückprallansatz<sup>4</sup> gewählt, skizziert in Abbildung 2.2 [18]. Die Implikation ist das Modellieren einer impermeablen, also für das Fluid undurchlässigen Gefäßwand, sowie das Ausbleiben einer transversalen Bewegung zwischen den

<sup>4</sup>Rückprallansatz: von engl. *bounce-back*



**Abbildung 2.3:** Skizze eines parabolischen Poiseuille-Geschwindigkeitsprofils (rot) in einer zylindrischen Simulationsgeometrie (schwarz) mit charakteristischer Länge  $l$ . Die Länge der Pfeile (links) bzw. die Intensität der roten Farbe (rechts) kodiert die Magnitude der lokalen Geschwindigkeit, die in  $u_c$  ihr Maximum hat.

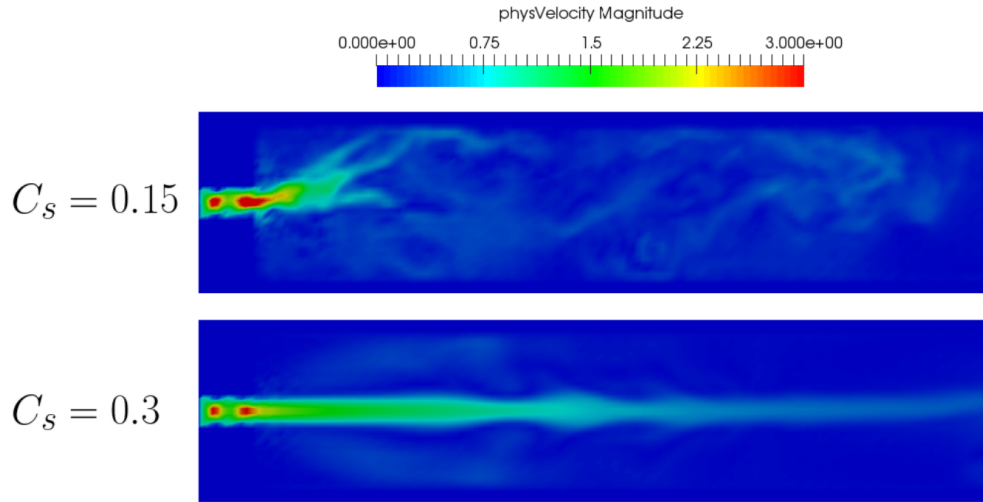
beiden Kollisionsteilnehmern. Für die Gefäßwand werden also Haftrandbedingungen<sup>5</sup> gewählt [18]. Diese Methode ist in der Implementierung numerisch stabil, jedoch nur anwendbar, wenn die Geometrie durch treppenartige Formen approximiert wird, damit eine Ausrichtung am Gitter vorliegt. Diese Methode kann für nicht entsprechend ausgerichtete Geometrien durch lineare Interpolation verbessert werden, wie durch Bouzidi *et al.* [1] gezeigt wurde. Dieser Ansatz wurde zudem in dem verwendeten Framework OpenLB implementiert [17] und in einer klinischen Studie validiert [23], sodass er für diese Arbeit ebenfalls gewählt wird. Durch die genannten Randbedingungen können zudem bewegte Wände modelliert werden. Dabei wird abhängig von der Bewegung der Wand relativ zum Fluid eine Geschwindigkeitskomponente hinzugefügt, um das entsprechende Verhalten zu simulieren. Dies sollte jedoch nicht mit deformierbaren Wänden verwechselt werden, deren Modellierung weitaus komplexer ist und daher nicht Teil dieser Arbeit ist. Zudem werden vom verwendeten Framework bisher nur starre Wände unterstützt. Weitere Randbedingungen werden an den Einlässen und Ausflüssen der Simulation gewählt. Für die Einlässe wird unter erneuter Annahme eines newtonschen Fluides ein Poiseuille-Geschwindigkeitsprofil definiert, welches in Abbildung 2.3 skizziert ist, dessen Maximum durch  $u_c$  angegeben sei. Für die Ausläufe wird als Bedingung konstanter Druck gefordert.

### 2.2.4 Parametrisierung

Um eine Strömungssimulation basierend auf der LBM zu konfigurieren, muss neben verschiedenen Modellentscheidungen (wie im vorherigen Unterkapitel diskutiert) noch eine Parametrisierung erfolgen. Die zu berücksichtigenden

<sup>5</sup>Haftungseigenschaften: von engl. *no-slip*

## 2 Grundlagen



**Abbildung 2.4:** Der Effekt unterschiedlicher Werte für die Smagorinsky-Konstante auf turbulente Strömung bei sonst identischen Bedingungen [16].

Parameter sind hierbei im Folgenden aufgeführt. Mit \* annotierte Parameter geben dabei die entsprechenden dimensionslosen Größen an, mit denen in der LBM typischerweise gearbeitet wird.

- Die Gitterkonstante  $\Delta x$  definiert den Abstand zwischen zwei benachbarten Zellen, angegeben in Metern m. Sie wird üblicherweise anhand der charakteristischen Länge  $l$  der Simulationsgeometrie, welche in etwa den größten enthaltenen Durchmesser angibt, sowie deren Auflösung  $N$  berechnet, also  $\Delta x = \frac{l}{N}$ .
- Die physikalische Länge eines diskreten Zeitschrittes  $\Delta t$  wird in Sekunden s angegeben.
- Die BGK-Relaxationszeit  $\tau$  wird ebenfalls in Sekunden angegeben. Oftmals wird auch der dimensionslose Relaxationsparameter  $\tau^*$  verwendet, welcher über  $\tau^* = \frac{\tau}{\Delta t}$  mit der Relaxationszeit in Beziehung steht [18].
- Die physikalische Dichte  $\rho$  des Fluides wird in  $\text{kg m}^{-3}$  angegeben und kann der Literatur entnommen werden.
- Die dimensionslose Dichte  $\rho^*$  ist im Falle eines inkompressiblen Fluides im Gegensatz zu der physikalischen Dichte  $\rho$  Fluktuationen unterworfen, welche sich in der LBM auf Druckunterschiede beziehen. Sie fungiert daher lediglich als Referenzwert und kann letztlich einen beliebigen Wert erhalten. Ihr Durchschnittswert wird üblicherweise auf den Wert 1 gesetzt [18].



- Die Scherviskosität des Fluides  $\eta$  wird in  $\text{kg m}^{-1} \text{s}^{-1}$  angegeben. Alternativ kann auch die kinematische Viskosität  $\nu$  über die Beziehung  $\eta = \nu \cdot \rho$ , angegeben in  $\text{m}^2 \text{s}^{-1}$ , verwendet werden. Die temperaturabhängigen Werte sind experimentell ermittelt worden und können der Literatur entnommen werden.
- Die typisch simulierte Geschwindigkeit  $u_c$  bzw. ihr dimensionsloses Gegenstück  $u_c^* = u_c \frac{\Delta t}{\Delta x}$  ist eine Ausgabe der Simulation, kann jedoch auch für Randbedingungen und somit ebenfalls als Eingabe verwendet werden, wie beispielsweise in Abbildung 2.3 skizziert.
- Die dimensionslose Schallgeschwindigkeit  $c_s^*$  wird üblicherweise und auch im verwendeten Framework auf den Wert  $\sqrt{1/3} \approx 0.577$  gesetzt [17]. Für inkompressible Fluide müssen die simulierten Geschwindigkeiten deutlich kleiner sein, also  $u_c^* \ll c_s^*$  [18].
- Die Smagorinsky-Konstante  $C_s$ , welche üblicherweise auf den Wert  $0.25 \pm 0.2$  festgelegt wird, definiert das Verhalten turbulenter Strömung. Höhere Werte führen dabei zu begrdigten Turbulenzen, zu sehen in Abbildung 2.4.

Die Parameter sind dabei über

$$\nu = \frac{\eta}{\rho} = c_s^{*2} \left( \tau^* - \frac{1}{2} \right) \frac{\Delta x^2}{\Delta t} \quad (2.15)$$

in Beziehung zueinander gesetzt [18]. Es existieren folglich prinzipiell beliebig viele Konfigurationen aus  $\tau^*$ ,  $\Delta t$  und  $\Delta x$ , um die physikalische korrekte Viskosität zu erhalten. Die Wahl ist jedoch nicht beliebig in Hinblick auf die Genauigkeit, Stabilität und Effizienz der Simulation. Für eine hohe Genauigkeit muss im Wesentlichen beachtet werden, dass der räumliche und zeitliche Diskretisierungsfehler mit  $\Delta x^2$  bzw.  $\Delta t^2$  skaliert [18]. Die beiden sollten also so klein wie möglich gewählt werden, was jedoch die Effizienz stark beeinträchtigt. Bei der Anpassung von  $\Delta x$  oder  $\Delta t$  sollte im vorliegenden Falle von inkompressiblen Fluiden die sogenannte akustische Skalierung  $\Delta t \propto \Delta x$  eingehalten werden [18]. Ferner gelten weitere Einschränkungen, die jedoch die Aussage an dieser Stelle nicht ändern. Für die Effizienz sei unmittelbar angeführt, dass die Gesamtlaufzeit  $T$  der Simulation und der Speicherbedarf  $M$  im vorliegenden, dreidimensionalen Fall durch

$$T \propto \frac{1}{\Delta x^3 \cdot \Delta t}, M \propto \frac{1}{\Delta x^3} \quad (2.16)$$

mit  $\Delta x$  bzw.  $\Delta t$  skalieren [18]. Die untere Grenze der beiden Parameter hängt also von der verwendeten Hardware und der zur Verfügung stehenden Zeit ab. Generell sollte  $\Delta x$  jedoch so klein gewählt werden, dass alle Strömungsmerkmale wie Turbulenzen ausreichend aufgelöst werden können. Wie diese ausfallen,

## 2 Grundlagen

hängt von der Simulationsgeometrie mit charakteristischer Länge  $l$  ab und wird durch die dimensionslosen Reynolds-Zahl  $Re$  gekennzeichnet. Diese ist definiert durch

$$Re = \frac{l \cdot u_c}{\nu}, \quad (2.17)$$

wobei hohe Werte auf turbulente und niedrige Werte auf laminare Strömungen hindeuten. Eine andere, ähnliche Geometrie weist dabei identisches Turbulenzverhalten auf, wenn auch diese Kennzahl den gleichen Wert hat. Liegt eine hohe Reynolds-Zahl vor, ist  $\Delta x$  folglich entsprechend klein zu wählen, damit die Simulation stabil bleibt. Instabilität ist in der LBM als das exponentielle Wachstum numerischer Fehler definiert, die irgendwann zu ungültigen Werten ( $NaN^6$ ) führen. Eine in diesem Sinne stabile Simulation zu erreichen, ist daher von großer Bedeutung und kann ferner durch weitere Eigenschaften begünstigt werden. Zum einen sollte der Relaxationsparameter  $\tau^*$  nicht zu nah an  $\frac{1}{2}$  gewählt werden, zum anderen sollten je nach erwarteter, typischer Geschwindigkeit  $u_c$  die Bedingungen

$$u_c^* < \begin{cases} \alpha(\tau^* - \frac{1}{2}) & \text{wenn } \tau^* < 0.55 \\ 0.4 & \text{sonst} \end{cases} \quad (2.18)$$

beachtet werden, wobei  $\alpha$  eine numerische Konstante in der Größenordnung von  $\frac{1}{8}$  ist [18]. Bei allen Beschränkungen handelt es sich jedoch nur um grobe Richtlinien, die Simulationen können dennoch instabil werden.

Damit ist diesem Unterkapitel zweifelsfrei zu entnehmen, dass es viele Parameter gibt, welche einen noch unbekannten Effekt auf den Simulationsverlauf in Kombination mit anderen Parametern haben. Insbesondere ist selbst bei einer wohldurchdachten Parametrisierung nicht klar, wie gut die resultierende Simulation die Wirklichkeit repräsentiert. Aus diesem Grund werden im Rahmen dieser Arbeit zunächst Ensembles von Simulationen mit einer Vielzahl von Parameterkonfigurationen und -kombinationen angefertigt, um deren Einfluss auf das Simulationsergebnis einschätzen zu können.

## 2.3 Klassische Multidimensionale Skalierung

Kernpunkt dieser Arbeit ist der Vergleich mehrerer Simulationsläufe mit gemessenen Daten zwecks Optimierung des verwendeten Modells bzw. der Parametrisierung. Für jeden Simulationslauf können dabei mehrere Zeitschritte vorhanden sein. Es soll letztlich eine Visualisierung gewählt werden, bei der jeder Zeitschritt jedes Simulationslaufs bzw. der gemessenen Daten mit jedem anderen verglichen werden kann. Zunächst ergibt sich aus dieser Betrachtung

---

<sup>6</sup>NaN: Not a Number - eine nicht darstellbare Zahl

eine *Distanzmatrix*

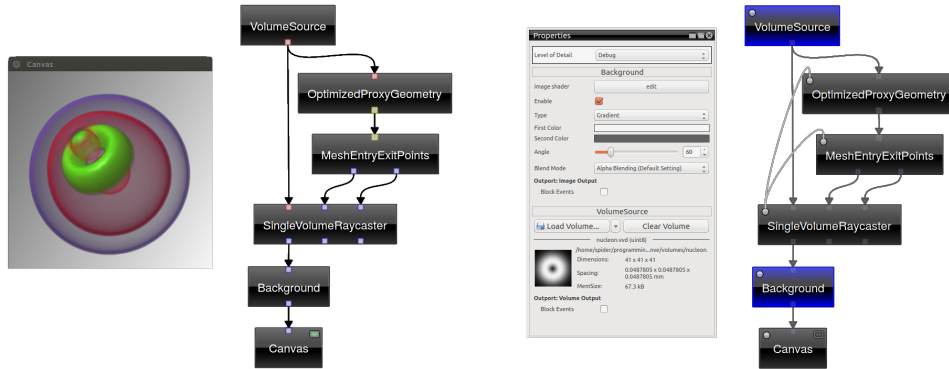
$$D = \begin{pmatrix} d_{11} & \dots & d_{1k} & d_{1k+1} & \dots & d_{1n} \\ d_{21} & \dots & d_{2k} & d_{2k+1} & \dots & d_{2n} \\ \vdots & & \vdots & \vdots & \ddots & \vdots \\ d_{n1} & \dots & d_{nk} & d_{nk+1} & \dots & d_{nn} \end{pmatrix} \quad (2.19)$$

mit Einträgen  $d_{ij}$ , wobei  $n$  die Gesamtzahl aller Zeitschritte aus allen Simulationsläufen angibt. Die Distanzen können auf Grundlage verschiedener Strömungseigenschaften und Metriken berechnet und somit als (Un)Ähnlichkeiten zu den anderen Daten verstanden werden. Die Einträge  $d_{i1}$  bis  $d_{ik}$  sind dabei beispielsweise den Ähnlichkeiten zwischen dem ersten Simulationslauf mit  $k$  Zeitschritten und einem beliebigen Zeitschritt  $i$  zuzuordnen. Aus der Verwendung von Metriken folgt zudem die Symmetrie der Matrix  $D$ , sodass letztlich nur die obere, bzw. untere Dreiecksmatrix berechnet werden muss. Um nun eine Visualisierung zu generieren, welche die Distanzen entsprechend berücksichtigt, bietet sich Multidimensionale Skalierung (MDS) an. Grundsätzlich kann mit dieser Technik ein hochdimensionaler Raum mit Dimension  $n$  in einen Raum mit niedrigerer Dimensionalität  $m$  projiziert werden. Dafür wird für jeden Zeitschritt  $i$  repräsentativ eine Position im Zielraum gefunden, sodass dessen Abstand zu den projizierten Positionen aller Zeitschritte  $j$  die korrespondierenden Distanzen  $d_{ij}$  widerspiegelt. Da die Visualisierung im Wesentlichen von Domainexperten verstanden werden soll, wird als Abstandsmaß die euklidische Distanz gewählt. Die so verwendete Technik wird auch klassisches MDS genannt. Auf die Wahl von  $m$  und die Berechnung der Distanzen wird in Kapitel 4 eingegangen. Hier wird zunächst der MDS-Ansatz nach Wickelmaier beschrieben [28]. Dieser wurde bereits erfolgreich auf Daten angewendet, welche den Vorliegenden ähnlich sind, d. h. es liegen viele Simulationsläufe ( $\mathcal{O}(10)$ ) mit jeweils vielen Zeitschritten ( $\mathcal{O}(100)$ ) und somit Distanzmatrizen mit großen  $n$  vor [6, 19]. Für die Projektion wird nun zur Distanzmatrix  $D$  eine Koordinatenmatrix  $X$  gesucht, deren Berechnung wie folgt zusammengefasst werden kann.

1. Berechnen der quadrierten Distanzen  $P = [d_{ij}^2]$ .
2. Anwenden doppelter Zentrierung mit  $B = -\frac{1}{2} \cdot J \cdot P \cdot J$ , wobei  $J = I - \frac{1}{n} \cdot U$ .  $I$  sei die Einheitsmatrix und  $U$  die Matrix, deren Einträge ausschließlich den Wert 1 haben.
3. Extrahieren der  $m$  größten, positiven Eigenwerte  $\lambda_1, \dots, \lambda_m$  von  $B$  und der korrespondierenden Eigenvektoren  $e_1, \dots, e_m$ .
4. Sei  $E_m$  die Matrix aus den  $m$  Eigenvektoren und  $\Lambda_m$  die Diagonalmatrix der  $m$  Eigenwerte, dann lässt sich die gesuchte Matrix berechnen durch  $X = E_m \cdot \Lambda_m^{1/2}$ .

## 2 Grundlagen

Die erhaltene Matrix  $X$  ist dabei nur eine mögliche räumliche Konfiguration, welche die Distanzmatrix  $D$  widerspiegelt. Damit ist die Projektion insbesondere nicht interpretationsfähig bezüglich Rotation und Spiegelung, sowie uniformer Skalierung.



(a) Netzwerkdarstellung des Datenflusses mit dessen Ausgabe im dazugehörigen Canvas. (b) Darstellung der Links (hellgrau) zum Synchronisieren der Kameraeinstellungen, sowie die Einstellungsmöglichkeiten von den in blau markierten Prozessoren.

**Abbildung 2.5:** Zu sehen ist ein simples Beispielnetzwerk in Voreen. Ein Volumendatensatz wird geladen und mittels Raycasting visualisiert.

## 2.4 Das Voreen-Framework

Voreen (Volume Rendering Engine) ist ein Framework zur Visualisierung medizinischer Volumendaten [22]. Es wurde von der Arbeitsgruppe *Visualisierung und Computergrafik* (VisCG) an der Universität Münster initiiert und wird zum Zeitpunkt der Erstellung dieser Arbeit kollaborativ von der Arbeitsgruppe *Pattern Recognition and Image Analysis* (PRIA) sowie *VISualization & graphIX* (VISIX) weiterentwickelt.

Das Framework basiert auf der Programmiersprache C++ und der OpenGL-API und ist somit plattformunabhängig. Die bereitgestellte Funktionalität ist in Module unterteilt, welche das Basis-Framework erweitern. Ferner existiert eine graphische Benutzeroberfläche basierend auf Qt5, welche die implementierte Datenfluss-Netzwerk-Metapher repräsentiert. Komplexe Aufgaben können mit so einem Netzwerk abgebildet werden, welches selbst aus meist simplen Komponenten besteht. Konkret handelt es sich um einen gerichteten Datenfluss, bei dem die Daten über Verbindungen von einem Knoten zum nächsten, im Folgenden *Prozessoren* genannt, weitergereicht, bzw. dort weiterverarbeitet werden. Die unterschiedlichen Datentypen sind dabei farblich kodiert: rote Anschlüsse akzeptieren Volumendaten, blaue hingegen Bilddaten. Viele weitere Datentypen sind implementiert.

Dies sei anhand eines Beispielnetzwerks, zu sehen in Abbildung 2.5, erklärt. Beginnend bei einer Datenquelle (*VolumeSource*) wird beispielhaft der Nukleon-Volumendatensatz eingeladen und über ihren ausgehenden Anschluss zur Verfügung gestellt. Dieser Datensatz wird nun für ein Raycasting verwendet, hinter dessen Ausgabe noch ein Hintergrund mit von grau nach weiß verlaufendem Gradienten gelegt wird. Das Resultat wird über ein Canvas, dem Endpunkt des Netzwerks, ausgegeben. Jeder Prozessor besitzt ferner einige Einstellungsmöglichkeiten zur Konfiguration seiner eigenen Logik. So lässt sich für die Datenquelle der Pfad bestimmen und beim Raycaster beispielsweise die Transferfunktion einstellen. All diese Einstellungen können auch über sogenannte Links miteinander synchronisiert werden. Das Netzwerk wird dabei nach jeder Änderung bzw. in regelmäßigen Abständen in Datenflussrichtung ausgewertet. Das Ausrichten der Kamera durch die Interaktion mit der Maus führt so beispielsweise umgehend zu einer Auswertung des Netzwerks, wodurch die Ausgabe entsprechend aktualisiert wird. Neben der grafischen Ausgabe ist auch das Schreiben von Dateien möglich und auch Python-Skripte können in den Datenfluss integriert werden. Des Weiteren sei auf die Möglichkeit von asynchronen und parallelisierten Berechnungen hingewiesen, um andere Teile des Netzwerkes nicht unnötig zu beeinträchtigen. Mit diesem Framework lässt sich damit prinzipiell jedes Problem in ein Netzwerk überführen und daraus eine Anwendung mit Benutzeroberfläche erstellen. Dies ist natürlich nicht für jedes Problem sinnvoll, im Rahmen dieser Arbeit wird Voreen jedoch insbesondere aufgrund der intensiven Arbeit mit Volumendaten gewählt. Damit kann bereits einige Funktionalitäten als gegeben angenommen werden, da es sich wie bereits erwähnt bei den meisten Prozessoren um triviale Operationen auf Volumen- und Bilddaten wie beispielsweise Schwellenwertoperationen handelt.

Um die Arbeit von Domainexperten zu unterstützen, können die mit der Anwendung erstellten und konfigurierten Netzwerke in sogenannten *Workspaces* gespeichert und zusammen mit einer Dokumentation weitergereicht werden. Da die Netzwerke schnell an Größe zunehmen und damit an Übersichtlichkeit verlieren können, kann für den Endanwender ein sogenannter Anwendungsmodus konfiguriert werden. Dieser stellt nur die wichtigsten Einstellungsmöglichkeiten und nur die wesentliche Ausgabe des Netzwerks in möglichst intuitiver Weise zur Verfügung.



## 3 Verwandte Arbeiten

In diesem Kapitel werden aktuelle Arbeiten aus thematisch verwandten Bereichen vorgestellt und diskutiert. Dazu gehören numerischen Strömungssimulationen, die Verarbeitung von Ensembles und die Assimilation von mittels 4D-Phasenkontrast-MR akquirierten Blutflussdaten. Dadurch soll zum einen die Wahl der verwendeten Methoden begründet und motiviert, zum anderen ein Überblick über den aktuellen Forschungsstand gegeben werden.

**Strömungssimulationen** In dieser Arbeit wird die Lattice-Boltzmann-Methode (LBM) für das Durchführen numerischer Strömungssimulationen verwendet. Es existiert jedoch eine ganze Bandbreite alternativer, bereits etablierter Methoden. Die bekanntesten sind das Finite-Elemente-Verfahren, Finite-Volumen-Verfahren sowie sogenannte Navier-Stokes-Solver. Dabei müssen aufwendige partielle Differentialgleichungen gelöst werden. Im Gegensatz dazu ist die LBM ist ein relativ junges Verfahren, welches zunehmend an Popularität gewinnt [18]. Grund dafür ist die rasante Entwicklung der Hardware und insbesondere die Zunahme von parallelen Berechnungen, da das Verfahren durch die vielen unabhängigen, lokalen Operationen inhärent parallelisierbar ist. Die Community hat im Zuge dessen das OpenLB-Framework geschaffen, welches in dieser Arbeit verwendet wird [17]. Ein anderes bekanntes Framework zum Konfigurieren von Strömungssimulationen ist OpenFOAM [13], welches ein breites Spektrum an Solvern bietet. Die Simulationsdomäne wird dabei, im Gegensatz zum regelmäßigen Gitter in der LBM, durch Tetraeder ausgeschöpft, wodurch der Vorteil der Parallelisierbarkeit gemindert wird und das Konzept eine aufwendige Einarbeitung erfordert. Der LBM wohnt dagegen ein verhältnismäßig simples Konzept inne, was jedoch keine Einschränkung für die Anwendbarkeit bedeutet. So konnte das Verfahren bereits auf Simulationsdomänen hoher Komplexität, wie etwa auf dem Aortenbogen und Aneurysmen, im Zusammenspiel mit 4D-Phasenkontrast-MR-Daten unter Berücksichtigung des Herzzyklus validiert werden [9, 10, 16, 23].

**Ensembleansatz** Der in dieser Arbeit verfolgte Ansatz, ein Ensemble von Simulationen zu generieren, wurde in ähnlicher Form bereits im Framework HemeLB [21] umgesetzt. Dabei wurde eine Simulationsgeometrie anhand von statistischen Daten verschiedener Patienten, wie etwa der Puls und Blutdruck, konfiguriert und jeweils eine Simulation generiert. Die Ergebnisse wurden anhand der Position und Stärke hoher Scherkräfte an den Gefäßwänden miteinander verglichen. In-vivo Messungen des Blutflusses haben dabei jedoch nicht

### 3 Verwandte Arbeiten

stattgefunden.

Jiang *et al.* [14] haben in ihrer Arbeit hingegen mehrere Simulation unter Verwendung derselben Geometrie durchgeführt, welche durch die Segmentierung von mittels 4D-Phasenkontrast-MR gemessenen Daten akquiriert wurde. Die Datensätze wurden, wie zuvor, anhand der Scherkräfte an den Gefäßwänden miteinander verglichen. Dabei konnten hohe Korrelationen zwischen simulierten und gemessenen Daten nachgewiesen werden. Zusätzlich wurde von den Autoren eine Metrik zum Vergleich von Vektorfeldern entwickelt, welche in Kapitel 4 gegen weitere Vergleichen wird. Dies ist ferner ein Beispiel für die Verwendung eines Navier-Stokes-Solvers, anstatt der LBM.

**Datenassimilation** Die Assimilation von 4D-Phasenkontrast-Daten mittels numerischer Strömungssimulationen ist ebenfalls ein sehr junger Forschungsbereich. De Hoon *et al.* [11] haben ein auf der inkompressiblen Navier-Stokes-Gleichung (vgl. Gleichung 2.2) basierendes Simulationsverfahren definiert, welches gemessene Daten als Eingabe verwendet. Somit lässt sich die Simulation steuern, indem für jeden simulierten Zeitpunkt, zu dem keine Messung vorhanden ist, die Domäne durch einen Korrekturterm wieder in Richtung der gemessenen Daten verschoben wird. Dies wurde von den Autoren mit einer Federkraft verglichen, welche die Simulation hin zu den gemessenen Daten lenkt. Das Verfahren wurde ferner von denselben verbessert, indem die Simulation durch die Reversibilität der Navier-Stokes-Gleichung sowohl vorwärts als auch rückwärts durchgeführt wurde [12]. Das Ergebnis wurde sowohl gegen synthetische Daten, als auch gegen eine ungesteuert Simulation verglichen. Dem diente erneut die Scherkräfte an der Wand sowie der lokale Unterschied der Magnitude und Richtung über den gesamten Querschnitt des verwendeten Aorten-Datensatzes.

Funke *et al.* [7] verwendeten ein ebenfalls auf der inkompressiblen Navier-Stokes-Gleichung basierendes Optimierungsverfahren, um die initialen Randbedingungen des Modells in den mittels Phasenkontrast-MR akquirierten Daten wiederherzustellen. Das Verfahren stellte sich als sehr robust im Falle von 2D- und 3D-Daten heraus. Ein Vergleich der gemessenen zu den simulierten und assimilierten Daten wurde erneut durch Betrachtung der Scherkräfte an den Gefäßwänden, sowie der Strömungsgeschwindigkeit durchgeführt.

Dies zeigt, dass Datenassimilation auf verschiedenen Ebenen stattfinden kann. Die Grundlage ist dabei stets eine numerische Strömungssimulation sowie der Vergleich von simulierten, gemessenen und assimilierten Daten.



## 4 Methodik

Zu Beginn dieses Kapitels sei erneut das Ziel dieser Arbeit aufgegriffen. Es sollen Strömungssimulationen so konfiguriert und parametrisiert werden, dass diese möglichst gut die gemessenen Daten widerspiegeln, um letztlich Datenassimilation betreiben zu können. Kernbestandteil dieser Aufgabe ist also das Quantifizieren der Ähnlichkeit zwischen gemessenen und simulierten Datensätzen, worauf im Folgenden detailliert eingegangen wird.

### 4.1 Quantifizierung von Ähnlichkeit

Um ein Maß für Ähnlichkeit definieren zu können, muss zunächst gewährleistet sein, dass die Domäne vergleichbar ist. Die gemessenen Daten dienen hierbei als Referenz, da diese für die Segmentierung der enthaltenen Gefäße bzw. Phantome verwendet werden. Wie in Unterabschnitt 2.1 bereits erläutert, werden bei Messungen mittels 4D-Phasenkontrast-MR anatomische Bilder erstellt und für die Segmentierung verwendet. Letztere definiert damit die Domäne, in welcher die Datensätze miteinander verglichen werden können. Bei den Simulationen ist dies offensichtlich, da diese auf der durch die Segmentierung erhaltenen Geometrie initialisiert und durchgeführt wird. Ferner existieren in vielen Fällen noch mehr Datensätze desselben Phantoms, also im Falle dieser Arbeit mehrere Messungen mit unterschiedlichen Einstellung der Pumpe bei sonst unveränderten Bedingungen.

Um Ähnlichkeiten verschiedener Datensätze bestimmen zu können, wird eine Ähnlichkeitsdarstellung<sup>1</sup> verwendet. Dabei handelt es sich um eine aggregierte Projektion bzw. Einbettung des gesamten Ensembles aus gemessenen und simulierten Daten in eine einzelne Visualisierung. Die Projektion wird mittels klassischer multidimensionaler Skalierung (MDS) durchgeführt, wie in Abschnitt 2.3 erklärt. Für die Projektion muss jedoch zunächst eine Distanz- bzw., in diesem Kontext auch Ähnlichkeitsmatrix genannt, definiert werden. Diese beinhaltet die paarweisen Ähnlichkeiten jedes Zeitschrittes zu jedem anderen, bzw. zu sich selbst. Ein Zeitschritt ist hierbei zu verstehen als das Vektorfeld  $\vec{u}$  (vgl. Gleichung 2.1) der Geschwindigkeiten zu einem simulierten bzw. gemessenen Zeitpunkt. Effektiv ist das Ziel also, eine Metrik zur Quantifizierung der Ähnlichkeit von Vektorfeldern zu entwickeln. Ein solches Vektorfeld hat zwei offensichtliche Größen, denen man Beachtung schenken kann, namentlich die Magnitude und die Richtung. Wird zunächst nur die Magnitude  $|\vec{u}|$

---

<sup>1</sup>Ähnlichkeitsdarstellung: von engl. *Similarity Plot*

## 4 Methodik

betrachtet, liegt ein dreidimensionales Skalarfeld vor, für welches Fofonov *et al.* [5] bereits eine effektive Metrik zur Bestimmung der Ähnlichkeit entwickelt haben. Diese basiert auf einer Verallgemeinerung des Vergleichs von zwei Isokonturen, welche je ein Skalarfeld beschreiben. Die Anwendbarkeit dieser Methode wurde bereits gut untersucht und bestätigt [4]. Es werden jedoch selten alle verfügbaren Datenpunkte, im vorliegenden Falle also jeder Voxel des Datensatzes, in die Berechnung miteinbezogen. Grund dafür ist die mit der Anzahl der Datenpunkte steigende Dauer und der benötigte Speicherplatz. Vielmehr wird ein quasi-Monte Carlo-Ansatz (qMC) gewählt, bei dem  $N$  quasi-zufällige Datenpunkte innerhalb der gemeinsamen, durch die Segmentierung definierte Domäne gewählt werden. Die Anzahl der Datenpunkte lässt sich so auf etwa  $N = 16384$  oder gar auf die Hälfte reduzieren, ohne, dass wichtige Eigenschaften verloren gehen. Dies konnte durch die Konvergenz der Unterschiede der Distanzmatrizen bei linearer Erhöhung der Anzahl der Datenpunkte auf Daten überprüft werden, die den vorliegenden ähnlich sind [4]. Vielmehr wäre deduktiv bereits eine geringere Anzahl ausreichend, da die Domäne der Strömungsdaten sich nicht auf das kubische, das Phantom umschließende Volumen, sondern nur auf das enthaltene Phantom bezieht. Dessen Volumen ist jedoch nicht ohne Weiteres bekannt, sodass die Anzahl der Datenpunkte berechnet wird, als würden die umschließenden Volumen miteinander verglichen werden. Dies erlaubt die einmalige Wahl eines Wertes für alle Phantome, da diese eine ähnliche Auflösung aufweisen und es kann auf einen erneuten Nachweis der Konvergenz verzichtet werden.

Seien  $\vec{p}_i$  mit  $i = 1, \dots, N$  nun die durch den qMC-Ansatz bestimmten Datenpunkte und  $h$  ein Isowert innerhalb des Wertebereichs der Magnituden aller Felder, also  $h \in [u_{min}, u_{max}]$ . Dann wird jeder Datenpunkt nun gegen den Isowert getestet und dadurch eine Isokontur  $C$  bestimmt mit  $C = [c_i]$  und

$$c_i = \begin{cases} 1 & \text{wenn } h < |\vec{u}(\vec{p}_i)| \\ 0 & \text{sonst.} \end{cases} \quad (4.1)$$

Es gilt nun, die Isokonturen miteinander zu vergleichen. Der Jaccard-Index ist ein effektives Maß zur Berechnung der Ähnlichkeit  $d$  zweier Mengen  $A$  und  $B$  und kann über

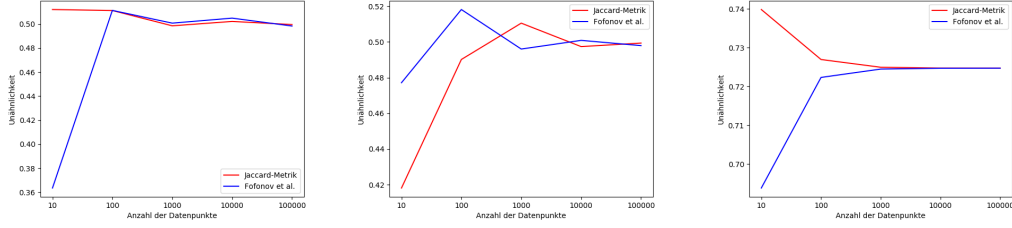
$$d_{AB} = 1 - \frac{|A \cap B|}{|A \cup B|} \quad (4.2)$$

als Metrik verwendet werden. Diese basiert auf dem Verhältnis von den in beiden zu den in mindestens einer der beiden Mengen enthaltenen Elemente. Seien  $C^A$  und  $C^B$  nun die Isokonturen zweier Zeitschritte  $A$  und  $B$ , dann kann Gleichung 4.2 unter Beachtung der Reihenfolge der Elemente auch geschrieben werden als

$$d_{AB} = 1 - \frac{\sum_{i=1}^N \min(c_i^A, c_i^B)}{\sum_{i=1}^N \max(c_i^A, c_i^B)}, \quad (4.3)$$

da die  $c_i$  nur entweder den Wert 0 oder 1 haben können. Diese Gleichung kann jedoch als eine Verallgemeinerung der Jaccard-Metrik verstanden werden, die

## 4.1 Quantifizierung von Ähnlichkeit



(a)  $C^A$  und  $C^B$  wurden für beide Abbildungen jeweils randomisiert mit einer Uniformverteilung auf dem Intervall  $[0, 1]$  initialisiert.  
 (b)  $C^A$  und  $C^B$  wurden mit einer linearen Funktion, bzw. mit einer Sinus-Periode im Intervall  $[0, 1]$  initialisiert.

**Abbildung 4.1:** Darstellung der Unähnlichkeit zweier Konturen  $C^A$  und  $C^B$ , ermittelt mit der generalisierten Jaccard-Metrik (rot) sowie der Verallgemeinerung von Fofonov *et al.* [5] (blau). Für eine große Anzahl  $N$  der Datenpunkte konvergieren beide auf denselben Wert.

sich nicht mehr nur auf Isokonturen anwenden lässt. Normalisiert man zunächst alle Werte der Skalarfelder mit

$$\hat{c}_i = \frac{|\vec{u}(\vec{p}_i)| - u_{\min}}{u_{\max} - u_{\min}}, \quad (4.4)$$

anstatt einen Isowert zu bestimmen und einen Vergleich der Isokonturen durchzuführen, lässt sich die verallgemeinerte Metrik nun direkt auf die Werte der Datenpunkte  $\hat{c}_i$  über  $\hat{C}_i = [\hat{c}_i]$  anwenden. Fofonov *et al.* haben die Metrik 4.2 jedoch mit einer Argumentation über unendlich viele Isokonturen verallgemeinert zu

$$d_{AB} = 1 - \frac{\sum_{i=1}^N 1 - \max(\hat{c}_i^A, \hat{c}_i^B)}{\sum_{i=1}^N 1 - \min(\hat{c}_i^A, \hat{c}_i^B)}. \quad (4.5)$$

Metrik 4.3 und 4.5 sind offensichtlich nicht äquivalent, konvergieren jedoch für große  $N$ , wie sie auf den vorliegenden Daten verwendet werden. Dies wurde durch die Visualisieren der beiden Metriken für synthetische Daten untersucht, wie in Abbildung 4.1 zu sehen. Um keine quantitative Untersuchung anschließen zu müssen, wird im Folgenden die durch Fofonov *et al.* definierte Metrik verwendet, welche eine Verbesserung zur Jaccard-Metrik, angewendet auf Isokonturen, darstellt [5]. Hierbei ist insbesondere für folgende Bestimmungen der Ähnlichkeiten relevant, dass die Einträge der Distanzmatrix  $D = [d_{ij}]$  stets normiert sind, also  $d_{ij} \in [0, 1]$ . Die Jaccard-Metrik und ihre Verallgemeinerungen garantieren diese Normierung aufgrund der berechneten Quotienten unmittelbar. Das Subtrahieren von 1 transformiert dabei jeweils die Ähnlichkeit der beiden zu deren Unähnlichkeit. Letztere wird für die gewünschte Anwendung der MDS gefordert.

Die verallgemeinerte Metrik 4.5 wurde von den Autoren auf Daten astrophysikalischer Simulationen gegen andere bekannte Metriken verglichen. Nament-

#### 4 Methodik

lich sind dies die zuvor genannte Ähnlichkeit der Isokonturen, die gemittelt normalisierte Differenz der Gradienten an den Datenpunkten über

$$d_{grad(A,B)} = \frac{\sum_{i=1}^N \|\nabla \hat{c}_i^A - \nabla \hat{c}_i^B\|}{2N}, \quad (4.6)$$

sowie gegen die Unähnlichkeiten basierend auf dem Korrelationskoeffizienten der beiden durch Gleichung 4.4 normalisierten Mengen zweier Konturen durch

$$d_{corr(A,B)} = \frac{1}{2} \left( 1 - \frac{\sum_{i=1}^N (\hat{c}_i^A - \bar{\hat{c}}^A)(\hat{c}_i^B - \bar{\hat{c}}^B)}{\sqrt{\sum_{i=1}^N (\hat{c}_i^A - \bar{\hat{c}}^A)^2 (\hat{c}_i^B - \bar{\hat{c}}^B)^2}} \right). \quad (4.7)$$

Dabei konnte gezeigt werden, dass bestimmte Muster, wie beispielsweise Rotationen, durch die zuletzt genannten Metriken besser erfasst werden können. Jedoch waren die Projektionen mehr von Verdeckung und Unordnung betroffen und Ausreißen konnten nicht gut ausgemacht werden. Da das Ziel dieser Arbeit ist, den gemessenen Daten sehr ähnliche Simulationen zu generieren und somit Ausreißer auszuschließen, wird der Fokus auf die durch Fofonov *et al.* vorgestellte Metrik gelegt.

Soll nun jedoch die Ähnlichkeit der Richtungen der Vektorfelder betrachtet werden, kann diese Metrik nicht mehr ohne Weiteres angewendet werden. Letztlich soll jedoch erneut eine Metrik gefunden werden, welche die Unähnlichkeiten der Richtungen  $d_{\angle(A,B)}$  auf das Intervall  $[0, 1]$  abbildet. Ein trivialer Ansatz ist hierbei, für jeden Datenpunkt den Winkel zwischen beiden Richtungsvektoren  $\vec{u}^A$  und  $\vec{u}^B$  der paarweise betrachteten Felder  $A$  und  $B$  als Maß für die Unähnlichkeit zu verwenden. Dann lässt sich die Metrik für einen einzelnen Datenpunkt bereits wie folgt beschreiben:

$$\varphi_{AB}(\vec{p}) = \begin{cases} \frac{1}{\pi} \arccos \left( \frac{\vec{u}^A(\vec{p}) \cdot \vec{u}^B(\vec{p})}{|\vec{u}^A(\vec{p})| \cdot |\vec{u}^B(\vec{p})|} \right) & \text{wenn } |\vec{u}^A(\vec{p})| \neq 0 \wedge |\vec{u}^B(\vec{p})| \neq 0 \\ 0 & \text{wenn } |\vec{u}^A(\vec{p})| = 0 \wedge |\vec{u}^B(\vec{p})| = 0 \\ 1 & \text{sonst.} \end{cases} \quad (4.8)$$

Es wird folglich der auf das Intervall  $[0, 1]$  normalisierte Winkel zwischen beiden Richtungsvektoren verwendet, wenn diese vom Nullvektor verschieden sind. Liegt beide Male der Nullvektor vor, so sind sich beide maximal ähnlich und die Metrik wertet zu 0 aus. Im Falle dessen, dass es sich genau bei einem der beiden um den Nullvektor handelt, so wertet die Metrik zur maximalen Unähnlichkeit, also 1 aus. Diese Entscheidung wurde getroffen, da die Metrik dem Erkennen von Ausreißern dienen soll. Bei einem Nullvektor innerhalb der Simulationsdomäne handelt es sich um eine absolute Seltenheit, mehr noch in den gemessenen Daten aufgrund des enthaltenen Rauschens. Um nun die Auswertungen für alle Datenpunkte in einer Metrik zu vereinen, wird letztlich mit

$$d_{\angle(A,B)} = \frac{\sum_{i=1}^N \varphi_{AB}(\vec{p}_i)}{N} \quad (4.9)$$

der Durchschnitt der einzelnen Winkelunähnlichkeiten gebildet. Schlussendlich können nun alle paarweisen Ähnlichkeiten aller Felder verwendet werden, um für jede Metrik eine Distanzmatrix  $D$  zu bestimmen, wie in Abschnitt 2.3 beschrieben.

Nun liegen die Unähnlichkeiten der Magnitude  $d_{AB}$  und der Richtungen  $d_{\angle(A,B)}$  in Form von separaten Distanzmatrizen vor. Diese Informationen sollen nun vereint werden, um letztlich eine einzelne Projektion generieren zu können. Der Umstand, dass die beiden Metriken auf dem Intervall  $[0, 1]$  definiert sind kann genutzt werden, um die Distanzmatrizen zu kombinieren. Eine Menge  $\mathcal{D} = \{D', D'', \dots\}$  von Distanzmatrizen kann vereint werden, indem für jeden Eintrag  $d_{ij}$  der resultierenden Matrix  $D$  eine Funktion  $f$  auf die Einträge derselben Position alle Matrizen angewendet wird mit  $d_{ij} = f(d'_{ij}, d''_{ij}, \dots)$ . Dies setzt natürlich dieselbe Größe der Matrizen voraus, welche unmittelbar gegeben ist, wenn alle Matrizen für dasselbe Ensemble generiert wurden. Die Funktion  $f$  sollte dabei der Definition

$$f(x_1, \dots, x_n) = \begin{cases} 1 & \text{wenn } \exists i : x_i = 1 \\ 0 & \text{wenn } \forall i : x_i = 0 \\ 0 \leq \gamma \leq 1 & \text{sonst} \end{cases} \quad (4.10)$$

genügen, damit die Interpretation der erzeugten Distanzmatrix sich nicht ändert. Bei  $\gamma$  handelt es sich dabei um eine monoton steigende Funktion auf dem Intervall  $[0, 1]$ , damit hohe Werte stets mit niedriger Ähnlichkeit, bzw. niedrige Werte mit hoher Ähnlichkeit assoziiert bleiben. Eine mögliche Funktion ist

$$f(x_1, \dots, x_n) = \max(x_1, \dots, x_n). \quad (4.11)$$

Hierbei wird jeweils nur der Eintrag an entsprechender Stelle aller Distanzmatrizen verwendet, der die größte Unähnlichkeit repräsentiert. Deren Anwendung ist sinnvoll, wenn letztlich nur Simulationen gesucht werden, die in allen Feldern ähnlich sind. Der Nachteil dieser Funktion ist jedoch, dass viele Informationen verloren gehen. Es ist insbesondere nicht klar, ob nur ein Feld für die Unähnlichkeit verantwortlich ist, oder mehrere. Um Magnituden- und Winkelunähnlichkeiten zu kombinieren, kann daher die Funktion






$$f(x_1, \dots, x_n) = 1 - \prod_{i=1}^n (1 - x_i) \quad (4.12)$$

verwendet werden. Dabei wird jedes Feld gleichermaßen in die neue Distanzmatrix miteinbezogen.

Eine weitere Möglichkeit, die Magnitude und den Winkel in eine Distanzmatrix zu überführen, ist durch die Verwendung einer Metrik gegeben, die beide Größen unmittelbar miteinbezieht, ohne den Umweg über zwei Distanzmatrizen zu nehmen. Li und Shen haben zum Vergleich von Vektorfeldern eine gewichtete Metrik entwickelt [20], die von Jiang *et al.* [14] adaptiert wurde zu

$$S(\vec{u}_1, \vec{u}_2) = \alpha e^{-2 \arcsin(\frac{|\vec{u}_1 \cdot \vec{u}_2|}{|\vec{u}_1| \cdot |\vec{u}_2|})} + \beta e^{-\frac{|\vec{u}_1| - |\vec{u}_2|}{u_{max}}} \quad (4.13)$$

## 4 Methodik

Größe	Vektor	Farbe
Magnitudo	$(\theta, \theta)$	
Richtung	$(\cos(2\pi\theta), \sin(2\pi\theta))$	
Kombination	$(\theta \cdot \cos(2\pi\theta), \theta \cdot \sin(2\pi\theta))$	
Referenz 1	$(1, 0)$	
Referenz 2	$(-1, 0)$	

**Tabelle 4.1:** Beschreibung eines Ensembles synthetischer Datensätze unter Angabe der veränderten Größe, des repräsentativen Vektors und einer eindeutigen Farbe. Der Faktor  $\theta$  verläuft dabei linear im Intervall  $(0, 1]$ , welches in 40 Einheiten diskretisiert wurde.

mit  $\alpha, \beta \geq 0$  und  $\alpha + \beta = 1$ . Diese ist für Paare von Vektoren  $\vec{u}_1, \vec{u}_2$  definiert und wurde bereits von zuletzt genannten Autoren für den Vergleich zwischen Simulationen und mittels 4D-Phasenkontrast-MR gemessenen Daten verwendet. Dabei wurde  $\alpha = \beta = 0.5$  gewählt. Erneut werden letztlich die Werte aller Paare von Vektoren mit

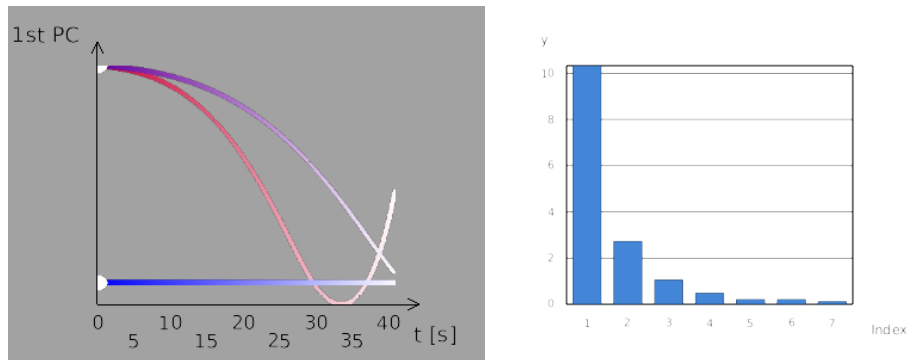
$$d_{S(A,B)} = \frac{\sum_{i=1}^N S(\vec{u}^A(\vec{p}_i), \vec{u}^B(\vec{p}_i))}{N} \quad (4.14)$$

gemittelt. Eine Verallgemeinerung dieser Metrik für eine beliebige Anzahl von Feldern, wie im Falle der mit Gleichung 4.10 vorgestellte Methode, ist jedoch nicht Möglich.

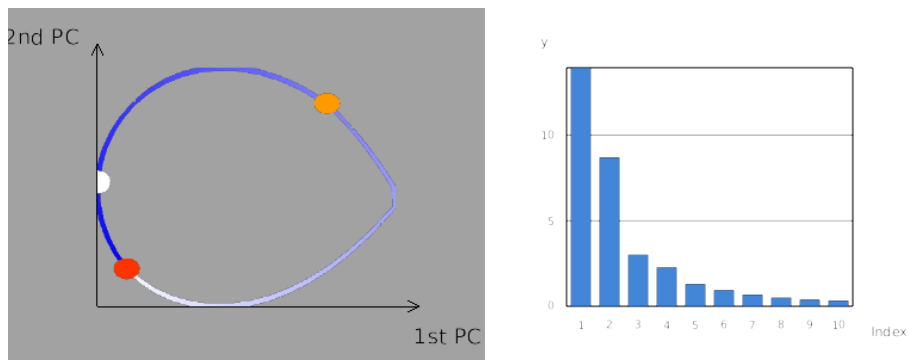
Die so erhaltenen Distanzmatrizen können mittels MDS (vgl. Abschnitt 2.3) auf wahlweise bis zu drei Dimensionen reduziert und projiziert werden. Dabei wird für jeden Zeitschritt entsprechend seiner Unähnlichkeiten zu allen übrigen eine Position im niederdimensionalen Raum derart ermittelt, dass deren euklidischen Abstände die Unähnlichkeiten widerspiegeln. Die Achsen des projizierten Raumes sind die Hauptrichtungen bzw. Eigenvektoren, denen jeweils ein Eigenwert zugeordnet ist. Nur solche mit großen Eigenwerten tragen dabei signifikant zur Projektion bei. Das Visualisieren der drei oder weniger Dimensionen mit den größten Eigenwerten ist dabei intuitiv. Projizierte Zeitschritte eines Simulationslaufes oder einer Messung können dabei in zeitlicher Reihenfolge zu einer Kurve verbunden und farblich kodiert werden. Eine voll gesättigte Farbe repräsentiert dabei eindeutig eine Zeitreihe. Da bei der gleichzeitigen Visualisierung der ersten zwei oder drei Hauptkomponenten die Zeit nicht als Dimension dargestellt wird, kann diese Information durch lineare Interpolation mit der Farbe weiß erneut hinzugefügt werden. Die so erzeugten Visualisierungen werden Ähnlichkeitsdarstellung genannt.

Um die Methoden zu evaluieren, wurde ein Ensemble synthetischer Datensätze generiert, welches in Tabelle 4.1 beschrieben wird. Dabei handelt es sich um homogene Vektorfelder der Dimension  $2 \times 2 \times 2$ . Die Größe ist arbiträr, da letztlich jeder Zeitschritt durch einen einzigen Vektor beschrieben wird. Das Ensemble enthält Datensätze, bei deren Vektoren ausschließlich

## 4.1 Quantifizierung von Ähnlichkeit



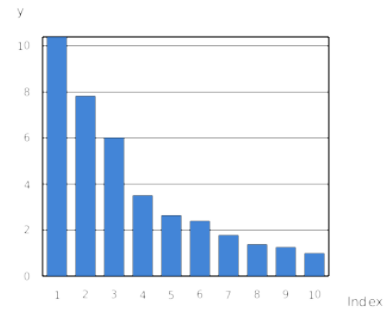
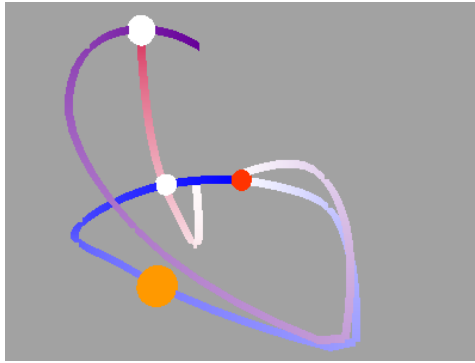
(a) Verwendung von Metrik 4.5 zur ausschließlichen Berücksichtigung der Magnitude. Als Achsen werden die erste Hauptkomponente und die Zeit verwendet.



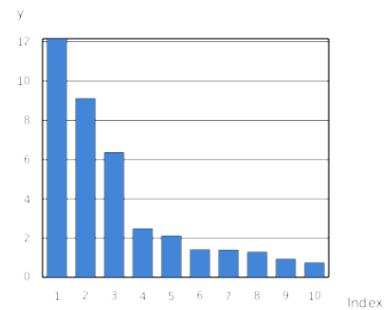
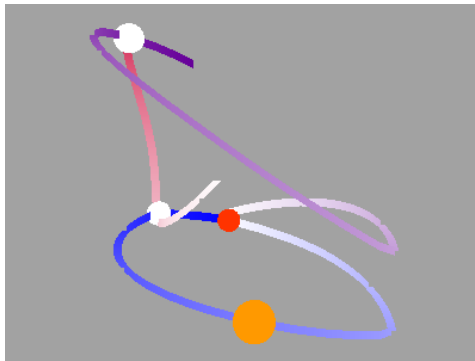
(b) Verwendung von Metrik 4.9 zur ausschließlichen Berücksichtigung der Richtung. Als Achsen werden die ersten beiden Hauptkomponenten verwendet.

**Abbildung 4.2:** Ähnlichkeitsdarstellungen der Metriken, welche jeweils entweder nur die Magnitude oder Richtung berücksichtigen. Zusätzlich sind die Eigenwerte der ersten zehn Hauptkomponenten dargestellt, was die jeweilige Wahl der Dimensionen begründet.

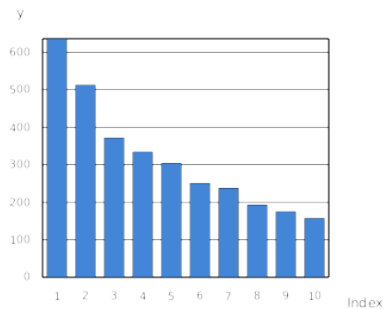
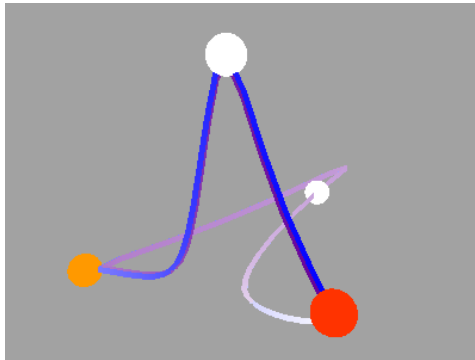
## 4 Methodik



(a) Verwendung von Metrik 4.11 zur Kombination der Magnituden- und Richtungs-  
informationen. Als Achsen werden die ersten drei Hauptkomponenten verwendet.



(b) Verwendung von Metrik 4.12 zur Kombination der Magnituden- und Richtungs-  
informationen. Als Achsen werden die ersten drei Hauptkomponenten verwendet.



(c) Verwendung von Metrik 4.14 mit  $\alpha = \beta = 0.5$  zur Kombination der Magnituden-  
und Richtungsinformationen. Als Achsen werden die ersten drei Hauptkomponenten  
verwendet.

**Abbildung 4.3:** Ähnlichkeitsdarstellungen der Metriken, welche sowohl die Magni-  
tude, als auch die Richtung berücksichtigen. Zusätzlich sind die  
Eigenwerte der ersten zehn Hauptkomponenten dargestellt, was  
die jeweilige Wahl der Dimensionen begründet.



die Magnitude bzw. die Richtung manipuliert wird, sowie eine Kombination daraus. Zusätzlich sind zwei konstante Referenzdatensätze enthalten, welche die Evaluation erleichtern sollen. Zunächst werden die Metriken 4.5 und 4.9 zur separaten Betrachtung von Magnitude und Richtung ausgewertet. Abbildung 4.2(a) zeigt eine konstante Kurve für den Datensatz mit sich verändernder Richtung, welche wie zu erwarten die von Referenz 1 und 2 überdeckt, da die Magnituden jeweils identisch sind. Ändert sich nur die Magnitude, so erfährt auch die entsprechende Kurve eine starke Änderung. Werden die Änderungen der Magnitude und der Richtung kombiniert, so verhält sich die resultierende Kurve ähnlich, da der Winkel keinen Einfluss hat. Der Verlauf ist jedoch nicht identisch, da sich die Magnituden um den Faktor  $\sqrt{2}$  unterscheiden. Dennoch erreicht die Kombination am Ende noch die konstante Kurve, da diese zum Endzeitpunkt dieselbe Magnitude aufweisen. Die Kurve des Datensatzes mit sich ändernder Magnitude entfernt sich dagegen erneut aus dem bereits genannten Grund der unterschiedlichen Magnitude. Die Position zum Endzeitpunkt ist dagegen nicht dieselbe wie zum Zeitpunkt  $\theta = 25$ , da die zweite Hauptkomponente ebenfalls einen nicht zu vernachlässigenden Einfluss besitzt. Letzteres ist anhand der dazugehörigen Eigenwerte nachzuvollziehen. In Abbildung 4.2(b) sind die kreisförmigen Kurven der beiden Datensätze mit sich ändernder Richtung identisch, was der Erwartung entspricht. Die Magnitude wird durch den weißen Punkt repräsentiert und ist konstant. Die Referenzpunkte liegen sich gegenüber, was ebenfalls der Erwartung entspricht. Die Kombinationen der Magnituden- und Richtungsinformationen über Metrik 4.11 und 4.12 zeigen ähnliche Ergebnisse für den synthetischen Datensatz in Abbildung 4.3(a) bzw. 4.3(b). Es finden sich dieselben markanten Muster aus vorherigen Abbildungen in Kombination wieder, d. h. eine kreisförmige Kurve in einer Ebene für die sich ändernde Richtung und eine beinahe lineare Bewegung durch die Änderung der Magnitude. Eine weitere Kurve stellt gut erkennbar den Verlauf der Kombination beider Größen dar: sie beginnt und endet in etwa am Start-, bzw. Endpunkt des Datensatzes der sich ändernden Magnituden bzw. Richtung. Die Referenzpunkte sind analog zur kreisförmigen Kurve zu Interpretieren. Die Metrik von Jiang *et al.* wurde wie von den Autoren vorgeschlagen mit  $\alpha = \beta = 0.5$  ausgewertet und liefert ein weniger interpretierbares Ergebnis. Zunächst legen die Eigenwerte nahe, dass die Verwendung der ersten drei Hauptkomponenten möglicherweise nicht ausreichend ist. In Abbildung 4.3(c) sind die beiden Kurven der Datensätze mit sich ändernder Richtung zu sehen, welche einen sehr ähnlichen Verlauf haben. Der Datensatz der sich ändernden Magnitude wird durch den oberen weißen Punkt dargestellt, bzw. approximiert, da die entsprechenden Änderungen in Relation zu den übrigen verschwindend gering sind. Dies kann durch das Ändern der Gewichtung kompensiert werden, was durch die Verwendung von  $\alpha = 0.1$  und  $\beta = 0.9$  überprüft wurde. Damit verliert die Metrik für den Zweck dieser Arbeit jedoch ihre Allgemeingültigkeit, da die Gewichtung für jeden Datensatz unterschiedlich ausfallen kann.

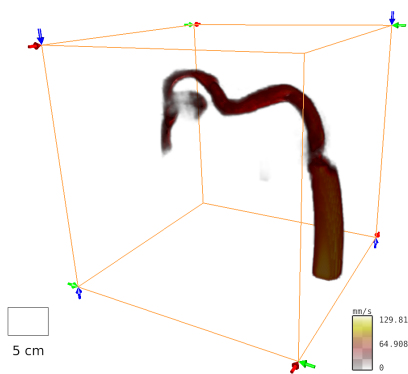
## 4.2 Die Analyseprozedur

Essentiell für die Datenassimilation ist das Akquirieren von Daten mittels Simulationen. Im Rahmen dieser Arbeit werden Ensembles von Simulationsläufen generiert, da die Auswirkung der Simulationsparameter in Hinblick auf die Ähnlichkeit zu den gemessenen Daten nicht offensichtlich ist. Für das Generieren von Ensembles und deren Analyse wird im Folgenden eine Analyseprozedur vorgestellt, die alle notwendigen Schritte in sinnvoller Reihenfolge kombiniert. Dafür wurde das in Abschnitt 2.4 vorgestellte Framework Voreen erweitert und daraus eine Anwendung erstellt. Mit deren Hilfe sind Domainexperten in der Lage, ihre mittels 4D-Phasenkontrast-MR gemessenen Daten zu importieren und Simulationen zu konfigurieren. Die Anwendung wurde auf die der Arbeit zugrundeliegenden Daten der Arbeitsgruppe *Magnetresonanz* am Institut für Medizinische Mikrobiologie der Universität Jena zugeschnitten und durch Feedback von Verena Hoerr und Ali Nahardani angepasst.

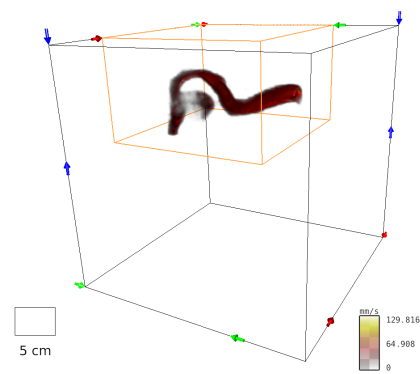
### 4.2.1 Vorverarbeitung

Die bereits erwähnten, der Arbeit zugrundeliegenden Daten wurden in unterschiedlichen Formaten zur Verfügung gestellt. Zunächst kam ein zuvor entwickeltes Format eines ehemaligen Mitarbeiters der Arbeitsgruppe, Philipp Rene Bovenkamp, zum Einsatz. Bei diesem wurden notwendige Metadaten, namentliche Auflösung, *Spacing* und die Anzahl der Zeitschritte (vgl. Abschnitt 2.1), in einer separaten Textdatei gespeichert. Es sei erwähnt, dass für diese Arbeit ausschließlich Datensätze mit statischen Strömungen verwendet wurden, sodass stets ein einzelner Zeitschritt je Messung zur Verfügung steht. Die anatomischen Daten, bzw. Rekonstruktionen der Phasenbilder für jede Dimension (vgl. Abschnitt 2.1) wurden in dem Format jeweils Scheibenweise in eine ASCII-Textdatei geschrieben. Das Lesen dieser Dateien ist sehr langsam, weshalb ein Vorverarbeitungsschritt eingeführt wurde, bei dem die Daten in das Voreen-eigene, XML-basierte Volumenformat konvertiert wurden. Darin sind die Rohdaten kompakt in Binärform gespeichert und es können weitere Metadaten hinzugefügt werden. Bei der Konvertierung wurde so für jeden Zeitschritt je ein Datensatz für Magnitude (aus dem anatomischen Daten) und Geschwindigkeit erzeugt. Dieser enthält neben der Auflösung und dem *Spacing* zusätzlich noch den Namen des Feldes, also Magnitude oder Geschwindigkeit, sowie einen Namen zur Identifikation der Messung. Daten aktuellerer Messungen wurden im Nifti-Format zur Verfügung gestellt. Diese enthalten jedoch ausschließlich die Rekonstruktionen der Phasenbilder. Hier wurde jeweils ein weiteres Volumen berechnet, welches die entsprechenden Magnituden beinhaltet, um dem Aufbau zuvor genannten Daten zu gleichen. Da es sich um große Ensembles mit vielen Simulationsläufen und Zeitschritten handeln kann, auf denen Algorithmen ausgeführt werden, spielt die Speicherverwaltung eine besondere Rolle für die Interaktivität der Anwendung. Bei einem Ensemble von

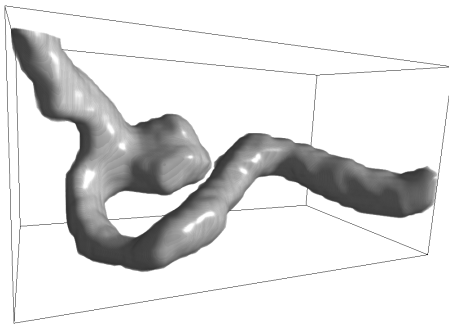
30 Simulationsläufen mit jeweils 20 Zeitschritten der Größe von 64MB, was einer Auflösung von  $256^3$  Voxeln entspricht, sind somit bereits mehr als 38GB zu bewältigen. Das Speichermanagement von Voreen implementiert zu diesen Zweck einen dynamischen Lademechanismus für das verwendete Format. Dabei werden Volumendaten erst dann aus dem meist langsamen Massenspeicher in den schnellen Haupt-, bzw. Grafikspeicher geladen, wenn diese benötigt werden. Wird die Kapazität des Hauptspeichers durch das sukzessive Einladen mehrere Volumen überschritten, so werden bereits eingeladene nach dem *LRU*-Prinzip<sup>2</sup> wieder aus dem Hauptspeicher verdrängt.



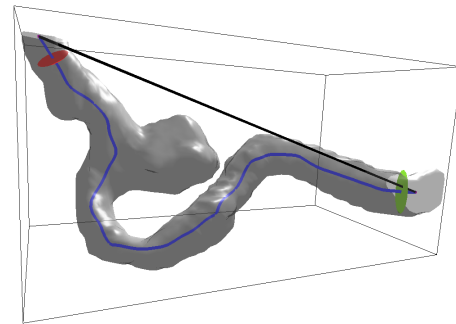
(a) Visualisierung des anatomischen Bildes vor der Segmentierung. Die Strömung im Aneurysma ist nicht klar zu erkennen und der Übergang vom Schlauch der Pumpe zum Phantom sollte nicht Teil der Simulation sein.



(b) Visualisierung des mit einer ROI (orange) zugeschnittenen Volumens. Der Übergang vom Schlauch der Pumpe zum Phantom wurde entfernt.



(c) Visualisierung nach Anwendung eines Medianfilters, Binarisierung und eines Gaußfilters.



(d) Visualisierung der extrahierten Isofläche mit Einlass (grün), Auslass (rot), Mittellinie (blau) und Gefäßgraphen (schwarz).

**Abbildung 4.4:** Die Segmentierung für einen bereits maskierten Datensatz erfolgt in vier Schritten am Beispiel eines Aneurysmen-Datensatzes.

<sup>2</sup>LRU: *Least Recently Used* - am längsten nicht verwendet

### 4.2.2 Segmentierung

Die Segmentierung der Phantome bzw. deren Inneren dient als Simulationsdomäne, sodass ihr Ergebnis den Ausgang der Simulation mitbestimmt. In den älteren Datensätzen wurden die entsprechenden Regionen bereits maskiert, sodass eine Segmentierung beinahe unmittelbar vorliegt. Dennoch enthalten diese durch die Messtechnik bedingte, unsaubere Oberflächen, da die Maskierung nicht optimal ist (siehe Abbildung 4.4(a)). Die aktuelleren Datensätze enthalten große, uninteressante Regionen und starkes Rauschen. Die Anwendung erlaubt daher zunächst das interaktive Auswählen einer quaderförmigen Region (ROI<sup>3</sup>), womit bereits ein gewisser Teil des unerwünschten Inhalts entfernt werden kann (siehe Abbildung 4.4(b)). Anschließend wird eine Kombination von Filtern auf die Region angewendet, deren Ziel das Erzeugen einer Maske für das enthaltende Phantom ist. Dafür kommt ein 3x3-Medianfilter und eine Binarisierung zum Einsatz, wodurch das Rauschen bereits entfernt werden kann. Der Schwellenwert muss dabei abhängig von der Stärke des Rauschens gewählt werden und unterscheidet sich insbesondere für die alten und neuen Daten. Ferner wird ein dreidimensionaler Gauß-Filter angewendet, sodass eine gewisse Glättung erreicht wird, ohne zu viele Details zu verlieren. Die Anwendung der Filter wurde durch eine qualitative Einschätzung des Ergebnisses (siehe Abbildung 4.4(c)) festgelegt. Dabei wurde die Verwendung möglichst weniger Filter angestrebt, um deren jeweiligen Effekt auf das Ergebnis gut einschätzen zu können. Im nächsten Schritt kann mit dem *Marching Cubes*-Algorithmus eine Isofläche der erhaltenen Maske berechnet werden, welche als Menge von Dreiecken vorliegt. Durch das Binarisieren und Glätten sollte der Isowert im Intervall  $(0, 1]$  so gewählt werden, dass das Volumen möglichst groß ausfällt, jedoch keine unerwünschten Verbindungen entstehen (siehe Abbildung 4.4(d)). Sollte die so erhaltene Isofläche dennoch nicht optimal sein, können weitere Filter verwendet oder die vorhandenen angepasst werden. Phasenumkehrungen (vgl. Abschnitt 2.1) können beispielsweise vereinzelt in Bereichen hoher Magnituden sehr niedrige zur Folge haben. Je nach Größe des betroffenen Bereiches müsste ein entsprechender Algorithmus angewendet werden, um die tatsächlichen Magnituden wiederherzustellen, was in dieser Arbeit nicht behandelt wird. Ist dies jedoch nur vereinzelt der Fall, kann es ausreichen, eine größere Nachbarschaft für den Medianfilter zu verwenden. Da dies jedoch unerwünschte Nebeneffekte an anderer Stelle zur Folge haben kann, können auch morphologische Operationen, also Dilatation und anschließende Erosion auf das bereits binarisierte Ergebnis angewendet werden. Sollten in der Region hingegen freistehende, durch Rauschen verursachte Artefakte vorhanden sein, können diese mit den in umgekehrter Reihenfolge angewendeten morphologischen Operationen entfernt werden. Die Isoflächen werden von dem verwendeten Simulationsframework OpenLB benötigt, um für das in der LBM genutzte Gitter zu initialisieren. Dabei wird entsprechend der gewählten Auflösung

---

<sup>3</sup>ROI: *Region Of Interest* - eine Region von Interesse

Fluid	Viskosität [ $\times 10^{-3} \text{m}^2 \text{s}^{-1}$ ]	Dichte [ $\text{kg m}^{-3}$ ]
Wasser (ca. 30 °C -10 °C)	0.8 – 1.35	995.6 – 1000
Blut (variable Bedingungen)	3 – 4	1043 – 1057

**Tabelle 4.2:** Literaturwerte der Viskosität und Dichte von Wasser und Blut.

für jeden Gitterpunkt getestet, ob dieser zur Simulationsdomäne gehört. Die Isofläche wird also als Simulationsgeometrie verwendet. Der verwendete Algorithmus schießt dafür einen Strahl in jede der drei Raumrichtungen und zählt die Anzahl der Schnittpunkte mit der Geometrie. Ist diese ungerade, so wird der Gitterpunkt zur Simulationsdomäne gezählt. Der Algorithmus besitzt zwar eine gewisse Robustheit gegenüber Löchern in der Geometrie, es werden jedoch unter Umständen auch inkorrekte Ergebnisse erzeugt. Zunächst wird daher die Möglichkeit geboten, das Resultat des Tests zu visualisieren. Sind Probleme offensichtlich, so können anschließend Löcher in der Geometrie automatisiert geschlossen werden. Dafür wird jede zusammenhängende Menge von Seiten der durch den *Marching Cube*-Algorithmus erzeugten Dreiecke iteriert, welche diese Seiten mit keinem anderen Dreieck teilen. Für jedes so definierte Loch in der Geometrie wird ein neuer Punkt eingefügt. Für alle beteiligten Seiten wird ein neues Dreieck ergänzt, dessen dritter Punkt durch den Schwerpunkt aller Endpunkte definiert wird. Implementiert wurde dies mithilfe der *Half-Edge*-Datenstruktur, welche die benötigten Informationen der Konnektivität zwischen den Dreiecken bereitstellt. Es sei jedoch erwähnt, dass das Überführen in die Datenstruktur bei komplexen Geometrien ein sehr rechenintensiver Schritt ist. Für die vorliegenden Daten hat das Schließen der Löcher keinen Vorteil bedeutet, jedoch sei hiermit auf die Möglichkeit dieser Optimierung verwiesen.

Die Segmentierung wurde für eine einzelne Geometrie beschrieben, jedoch liegen zu den Phantomen mehrere Messungen, bzw. zukünftig mehrere Zeitschritte für jede Messung vor. Für die in dieser Arbeit verwendeten Daten unterscheiden sich die Phantome je Messung jedoch nicht oder kaum, da diese aus einem (relativ) statischen Material angefertigt wurden (vgl. Kapitel 5). Aus diesem Grund wurde für die Segmentierung eine beliebige Messung gewählt und im Folgenden als Referenz verwendet. Die übrigen Messungen kommen in einem späteren Schritt ebenfalls zum Einsatz.

### 4.2.3 Simulation

Nachdem die Simulationsgeometrie wie im vorangegangenen Unterabschnitt aus der Segmentierung eines Referenzdatensatzes generiert wurde, wird nun die Simulation selbst bzw. das Erstellen eines Ensembles thematisiert. Da letzterem unterschiedlicher Parametrisierungen zugrunde liegen (vgl. Unterabschnitt 2.2.4), spielt die Wahl der Parameter eine gesonderte Rolle. Die Geometrie wird dabei zur Initialisierung des Simulationsgitters verwendet, wo-

durch die Simulationsdomäne initialisiert wird. Jeder Voxel des Gitters wird dabei letztlich durch ein Material bzw., allgemeiner, durch einen Typen repräsentiert. So werden alle innerhalb der Geometrie liegenden Voxel zunächst als ein Fluid initialisiert, die unmittelbar außerhalb liegenden als Wand. Die Eigenschaften des Fluides sind dabei Teil der Parametrisierung und durch deren Viskosität  $\eta$  und Dichte  $\rho$  definiert. Gültige Werte sind dabei für die verwendeten Flüssigkeiten der Literatur zu entnehmen, zu sehen in Tabelle 4.2. In der implementierten Anwendung werden dabei Intervalle für Wasser und Blut zur Auswahl gestellt. Die Messungen zur Akquirierung der in dieser Arbeit verwendeten Datensätze wurden dabei zwar ausschließlich mit Wasser durchgeführt, allerdings soll die Implementation in Zukunft auch für die nicht verwendeten, aber bereits vorliegenden Messungen von Mäusen genutzt werden können. Um innerhalb der Geometrie eine Strömung zu simulieren, müssen Ein- und Auslässe definiert werden, welche zudem deren Richtung festlegen. Zu diesem Zweck wird die bei der Segmentierung erhaltene Maske wiederverwendet. Mithilfe des von Drees *et al.* [3] implementierten Algorithmus wird zu der Geometrie in einem effizienten, iterativen Prozess ein Gefäßgraph und die Mittellinie ermittelt (siehe Abbildung 4.4(d)). Die Endknoten des Gefäßgraphen, also jene Knoten mit Grad 1, stellen Kandidaten für Ein- und Auslässe dar. Prinzipiell lässt sich eine Simulation in beide Richtungen durchführen. Da die simulierten Daten den gemessenen jedoch ähneln sollen, sollte die Richtung also identisch gewählt sein. Zu diesem Zweck wurde eine Heuristik implementiert, die automatisiert zu den Endknoten die Richtung ermittelt. Dafür wird in dem gemessenen Referenzdatensatz an der Position der jeweiligen Knoten die normalisierte Richtung  $\vec{d}$  der lokalen Strömung ausgelesen und mit dem Normalenvektor  $\vec{n}$  vom Knoten entlang der Mittellinie verglichen. Unter Verwendung eines Schwellenwertes  $h \in [0, \frac{\pi}{2}]$  kann durch

$$Typ = \begin{cases} \text{Einlass} & \text{wenn } \arccos(\vec{d} \cdot \vec{n}) < h \\ \text{Auslass} & \text{wenn } \arccos(\vec{d} \cdot \vec{n}) \geq \pi - h \\ \text{unbekannt} & \text{sonst} \end{cases} \quad (4.15)$$

der Typ des Knotens ermittelt werden. Weisen die gemessenen Daten im Bereich der Kandidaten turbulente Strömungen auf, kann die simple Heuristik den Typen nicht klassifizieren und der Nutzer muss mithilfe der Benutzeroberfläche eine manuelle Auswahl treffen. Dafür ist die Kenntnis des Datensatzes erforderlich oder es wird zuvor auf eine Visualisierung des gemessenen Vektorfeldes zurückgegriffen. Zur Auswahl stehen zur Zeit Strömungslinien, deren lokale Richtung farblich kodiert wurde, sowie eine Visualisierung mittels Pfeil-Glyphen. Eine Verbesserung der Heuristik kann jedoch beispielsweise durch Einbeziehen umliegender Voxel und Durchschnittsbildung erreicht werden. Die durch den Algorithmus und die Heuristik initialisierten Kandidaten müssen nicht zwangsläufig das gewünschte Ergebnis repräsentieren, es können auch falsch positive gefunden werden. Diese lassen sich ebenfalls mithilfe der

Benutzeroberfläche deaktivieren. Die Eigenschaften der gewünschten Kandidaten können dagegen noch modifiziert werden. Es wird das Verschieben entlang der Mittellinie und das Anpassen des lokalen Durchmessers unterstützt, dessen möglichst genaue Angleichung an die umliegende Gefäßwand für die Initialisierung des Poiseuille-Geschwindigkeitsprofils wichtig ist. Zusätzlich kann für jeden Einlass eine Anlaufphase definiert werden, bei dem die Strömungsgeschwindigkeit bis hin zur Zielgeschwindigkeit  $u_c$  langsam erhöht wird. Die gesamte Simulationsdomäne wird zu Beginn stets mit der Geschwindigkeit  $\vec{0}$  initialisiert, was bei unmittelbar anliegender Zielgeschwindigkeit zu einem abrupten Dichteunterschied führt und Instabilität zur Folge haben kann [16]. Aus diesem Grund wird in der Literatur für die Anlaufphase meist eine sinusoidale Geschwindigkeitsentwicklung gewählt, welche auch in der implementierten Anwendung zur Verfügung gestellt wird. Sei  $t_d$  die angestrebte Dauer der Simulation in Sekunden, dann wird die Dauer der Anlaufphase  $t_s$  initialisiert durch  $t_s = \frac{1}{4}t_d$ , ist aber beliebig im Intervall  $[0, t_d]$  wählbar. Damit hat sie standardmäßig einen nicht zu vernachlässigenden Anteil an der Simulationsdauer. Die Magnitude der aktuellen Maximalgeschwindigkeit  $u(t)$  wird dann für jeden Einlass berechnet durch

$$u(t) = \begin{cases} u_c \cdot \frac{\sin(-\frac{\pi}{2} + \frac{t}{t_s} \cdot \pi) + 1}{2} & \text{wenn } t < t_s \\ u_c & \text{sonst.} \end{cases} \quad (4.16)$$

Aktuell wird für jeden Einlass dieselbe Zielgeschwindigkeit angestrebt, da für die verwendeten Datensätze jeweils nur genau ein Einlass notwendig ist. Das Geschwindigkeitsprofil hängt allerdings vom jeweiligen Durchmesser ab. Als Referenzwert für die charakteristische Geschwindigkeit  $u_c$  wird automatisch die maximale Geschwindigkeit innerhalb des Referenzdatensatzes gewählt (vgl. Unterabschnitt 2.2.3).

Ferner lassen sich weitere Informationen aus dem Gefäßgraphen ermitteln, welche als Anhaltspunkt für die Parametrisierung dienen können. So wird unter anderem der minimale, maximale und durchschnittliche Durchmesser durch den Algorithmus von Drees *et al.* ermittelt. In der Anwendung wird dabei automatisch der durchschnittliche Durchmesser als Referenzwert für die charakteristische Länge gewählt, sie erlaubt aber auch das manuelle Ausmessen eines beliebigen Abstands. Dafür kann interaktiv ein Referenzpunkt der Messung gewählt werden, dessen Abstand zu einem beliebigen anderen durch Führen des Cursors angezeigt wird. Der Gitterabstand  $\Delta x$  kann nun implizit durch die charakteristische Länge  $l$ , sowie die Wahl der Auflösung  $N$  mit  $\Delta x = \frac{l}{N}$  angegeben werden. Da inkompressible Fluide simuliert werden sollen, muss die akustische Skalierung  $\Delta t \propto \Delta x$  eingehalten werden. Verdoppelt man also die räumliche Auflösung, muss die Länge eines Zeitschritts  $\Delta t$  ebenfalls verdoppelt werden (vgl. Unterabschnitt 2.2.4). Aus diesem Grund ist eine zeitliche Auflösung  $M$  Teil der Parametrisierung, mit welcher die Länge eines Zeitschritts durch  $\Delta t = \frac{\Delta x}{M}$  berechnet werden kann. Der dimensionslose

## 4 Methodik

Relaxationsparameter  $\tau^*$  wird dann vom verwendeten Framework anhand von Gleichung 2.15 automatisch berechnet durch

$$\tau^* = \frac{1}{c_s^{*2}} \frac{\eta}{\rho} \frac{\Delta t}{\Delta x^2}. \quad (4.17)$$

Ferner wird die dimensionslose Dichte  $\rho^*$  wie in der LBM üblich auf den Wert 1 gesetzt. Der letzte festzulegende Parameter ist die Smagorinsky-Konstante  $C_s$ , deren Effekt auf die Simulation nicht vorhergesagt werden kann. Aus diesem Grund wird ein Ensemble von Parametrisierungen erstellt. Dafür wird für jeden der zuvor genannten Parameter ( $N, M, l, u_c, \eta, \rho, C_s$ ) ein Intervall festgelegt. Die Grenzen sind dabei im Falle der Viskosität und Dichte durch die Literaturwerte, im Falle der charakteristischen Länge und Geschwindigkeit durch den Datensatz und andernfalls durch an der Literatur orientierte Werte definiert. Die Intervalle können dann beliebig aber gleichmäßig diskretisiert werden. Damit wird das Problem offensichtlich, dass aufgrund der hohen Parameterzahl eine Vielzahl von Simulationen notwendig werden würde. Bereits bei einer Unterteilung der Intervalle in drei Werte wären letztlich  $3^7 = 2187$  Simulationen notwendig, um das Zusammenspiel der Parameter zu analysieren. Tatsächlich müssen jedoch nicht alle Kombinationen betrachtet werden. Die charakteristische Länge und räumliche Auflösung definieren in Kombination nur den Gitterabstand und werden lediglich beide verwendet, um die Konfiguration intuitiver zu gestalten. Des Weiteren sind der Gitterabstand und die zeitliche Auflösung zwei Parameter, welche für die verwendete Simulationsgeometrie einmalig bestimmt und dann für weitere Simulationen konstant gehalten werden können. Sobald der Einfluss eines weiteren Parameters auf die Simulation in Hinblick auf die Ähnlichkeit zu den gemessenen Daten ermittelt wurde, kann dann anstelle des Intervalls ein einzelner Wert definiert werden. Auf weitere Details diesbezüglich wird im Analyseabschnitt eingegangen.

Die Parametrisierungen erhalten dabei zur Identifizierung einen eindeutigen Namen. Die Diskretisierung der Intervalle wird dabei auf höchstens 26 Einheiten beschränkt, da mehr ohnehin nicht sinnvoll wären. So lässt sich jede Einheit jedes Parameters mit einem Buchstaben des Alphabets kennzeichnen, wodurch sich beispielsweise für zwei Parameter und zwei Einheiten je Intervall die Zeichenfolgen  $AA$ ,  $AB$ ,  $BA$  und  $BB$  ergeben. Diese können ferner mit einem Präfix versehen werden, um verschiedene Gruppen von Simulationsläufen innerhalb eines Ensembles zu unterscheiden. Das Ensemble selbst erhält ebenfalls einen Namen, um die Auswertung zu vereinfachen.

Die Ausgabe der Simulation kann ebenfalls konfiguriert werden. So kann je nach Zweck für die Analyse gewählt werden, ob die makroskopische Geschwindigkeit, deren Magnitude und der Druck aus dem Simulationsgitter rekonstruiert werden sollen. Diese Liste ist prinzipiell beliebig erweiterbar, beispielsweise um die Scherung an den Grenzen der Simulationsdomäne, bzw. an den Gefäßwänden. In dieser Arbeit werden jedoch nur die Geschwindigkeit und deren Magnitude verwendet, da für diese ein unmittelbares Äquivalent



in den gemessenen Daten vorliegt. Bei einer Simulationsgeometrie der Seitenlänge  $50\text{mm}$ , einer charakteristischen Länge von  $5\text{mm}$ , einer räumlichen Auflösung von  $N = 40$  und einer zeitlichen Auflösung von  $M = 20$  würden für eine Simulationsdauer von  $t_d = 2\text{s}$  insgesamt 320 Zeitschritte mit jeweils  $400^3$  Voxeln berechnet werden. Da letztlich jedoch nur statische Strömungen gemessen wurden und somit simuliert werden soll, ist damit aber nur ein Bruchteil der Daten notwendig, um die Ähnlichkeit untersuchen zu können. Von besonderem Interesse ist dabei der zuletzt simulierte Zeitschritt, sowie einige gleichmäßig verteilte, vorangegangene Zeitschritte, um den Verlauf der Simulation einschätzen zu können. So ist beispielsweise ein früher Abbruch möglich, wenn eine Simulation instabil wurde oder aber sie konvergiert nicht zu einer statischen Strömung. Dabei wurden zwei Bedingungen implementiert, die zu vorzeitigem Abbruch führen können. Zum einen darf in der LBM eine maximale Geschwindigkeit von  $u_c^*$  gemäß Bedingung 2.18 nicht überschritten werden. Zum anderen können Parametrisierungen dazu führen, dass entweder gar keine Strömung entsteht oder diese konvergiert. Dafür wird die Standardabweichung  $\sigma$  der durchschnittlichen Energie  $\phi$  aller  $N$  Zeitschritte im Zeitraum der letzten simulierten Sekunde über

$$\sigma(\phi) = \sqrt{\frac{1}{N+1} \sum_{i=1}^N (\phi_i - \bar{\phi})^2} < \epsilon \bar{\phi} \quad (4.18)$$

mit einem Schwellenwert  $\epsilon = 10^{-5}$ , multipliziert mit dem Erwartungswert  $\bar{\phi}$ , verglichen [16]. Die durchschnittliche Energie berechnet sich dabei anhand der quadrierten Geschwindigkeiten aller Zellen und bietet sich daher als Merkmal für die Änderung des Systems an [16]. Im Falle der Konvergenz wird die Simulation vorzeitig abgebrochen. Letztlich werden somit etwa 20 gleichmäßig über die Simulationsdauer verteilte Zeitschritte ausgegeben. Eine weitere Möglichkeit, die Ausgabegröße zu reduzieren, ist durch das Vermindern der räumlichen Auflösung gegeben. Die Simulationen werden zunächst mit einer hohen Auflösung durchgeführt, um ein physikalisch möglichst akkurates Ergebnis zu liefern. Für die Analyse kann jedoch auch eine deutlich geringere Auflösung vollkommen ausreichend sein. Diese muss jedoch zunächst ermittelt werden, worauf in einem folgenden Unterkapitel eingegangen wird.

Die Parametrisierungen können dabei interaktiv in einer zur Verfügung gestellten Benutzeroberfläche konfiguriert werden. Zum Durchführen der Simulationen wurden zwei Möglichkeiten implementiert. Die Konfiguration und Simulation kann auf demselben Computersystem durchgeführt werden. Dabei kann jede Parametrisierung einzeln ausgewählt und simuliert werden, was insbesondere für die Fehlersuche sinnvoll sein kann. Schließlich ist es kein triviales Problem, eine funktionierende und gleichzeitig realistische Parametrisierung zu finden. Da die Laufzeit für große Ensembles oder sogar einzelne Simulationen je nach Größe, Parametrisierung und verwendeter Hardware mehrere Stunden bis Tage betragen kann, wurde zudem eine Schnittstelle zum Palma II Re-

chencluster der WWU Münster implementiert. Dieses stellt 434 Rechenknoten mit jeweils 72 logischen Rechenkernen und mindestens 92GB RAM bereit, auf welchem sowohl eine Intel-, als auch eine GCC-Toolchain zur Verfügung steht. Die LBM bietet durch das uniforme Gitter optimale Eigenschaften zur Parallelisierung, welche von dem verwendeten Framework OpenLB mittels OpenMP, MPI, sowie einem hybriden Betrieb ausgeschöpft werden können. Aus der lokal eingestellten Parametrisierung können Jobs für das Rechencluster erstellt werden, was die potentielle praktische Größe von Ensembles immens anhebt. Dafür wurde auf dem Cluster eine modifizierte Version des Simulationsquellcodes für beide Toolchains kompiliert. Für das gesamte Ensemble wird dann eine Konfigurationsverzeichnis erstellt, welches die Simulationsgeometrie und die einzelnen Parametrisierungen enthält. Dieses wird per SSH auf das Cluster übertragen und dann für jede Parametrisierung ein Job übermittelt, welcher in die Warteschlange des System eingereicht wird. Für die Jobs können je Ensemble die benötigten bzw. zu verwendenden Ressourcen eingestellt werden. Dabei wird eine breite Verteilung der Jobs angestrebt, d. h. jedem Job werden wenige Ressourcen zugeordnet, wohingegen viele Jobs gleichzeitig laufen können. Dies hat bei großen Ensembles den Vorteil, dass die Jobs in etwa gleichzeitig abgeschlossen sein werden und die Weiterverarbeitung unmittelbar erfolgen kann. Das gleichzeitige Simulieren mehrerer Ensembles ist ebenfalls möglich. Letztlich ist die Schnittstelle für andere Cluster adaptierbar, sodass diese bereits am Institut für Medizinische Mikrobiologie der Universität Jena auf der dortigen IT-Infrastruktur eingesetzt werden konnte.

### 4.2.4 Nachbearbeitung

Nach Beendigung aller Jobs eines Ensembles können die Ergebnisse per SSH auf das lokale System übertragen und dort für die weitere Analyse präpariert werden. Dafür wird für jede rekonstruierte Größe, also z. B. die Geschwindigkeit, aus den vorliegenden Zeitschritten zunächst ein neues Ensemble erstellt. Dies ist notwendig, da auch Größen rekonstruiert werden können, für die zur Zeit kein Äquivalent der gemessenen Daten zur Verfügung steht. Auf der anderen Seite muss nicht notwendigerweise jede der mit den gemessenen Daten zur Verfügung gestellten Größen auch aus den Simulationen rekonstruiert werden. In dieser neuen Struktur können somit gemessene und simulierte Daten zusammengeführt werden, was für deren Vergleich unerlässlich ist. Ein weitere Vorteil der Trennung nach den Größen ist, dass beispielsweise aus dem Ensemble, welches die Geschwindigkeiten enthält, andere Größen abgeleitet werden können und somit ein neues Ensemble erzeugen. Die Analyseprozedur kann an dieser Stelle unter der Verwendung von Python-Skripten beliebig erweitert werden. Einige Möglichkeiten werden im Ausblick diskutiert. Des Weiteren können an dieser Stelle auch Simulationsläufe aussortiert werden, welche nur einen Zeitschritt enthalten, also frühzeitig abgebrochen sind oder es können Ensembles derselben rekonstruierten Größe beliebig kombiniert werden. Für jedes Ensemble

ble kann daraufhin, unter Verwendung einer der bereits aufgeführten Metriken, eine Ähnlichkeitsdarstellung generiert werden, aber auch für die Kombination mehrerer Ensembles unterschiedlicher Größen. Beispielsweise kann so die Geschwindigkeit, welche selbst eine Kombination von Magnitude und Richtung darstellt, mit dem Druck und weiteren Strömungsrelevanten Größen kombiniert werden.

### 4.2.5 Analyse

Nachdem die Daten in ihrer finalen Form vorliegen, kann nun deren Analyse mithilfe der implementierten Anwendung erfolgen. Da es letztlich um den Vergleich der Ähnlichkeiten der Ensembleteilnehmer untereinander geht, wird der verbreitete Ansatz gewählt, zunächst einen Überblick über die Daten zu bekommen und Details nur nach Notwendigkeit zu erhalten [4, 19]. Da zu Beginn keine Kenntnis über die Auswirkung der einzelnen Simulationsparameter besteht, müssen einige initiale Simulationen durchgeführt werden, um einen Überblick zu bekommen. Dazu muss zunächst die zu simulierende zeitliche und räumliche Auflösung gefunden werden, welche eine hinreichende physikalische Präzision bietet, sowie die räumliche Auflösung der Ausgabe. Dafür werden alle Parameter mit einem beliebigen, aber plausiblen Wert initialisiert (vgl. Unterabschnitt 4.2.3). Anschließend wird lokal eine Testsimulation gestartet, um die zeitliche Auflösung festzulegen, wobei im Wesentlichen die Stabilität der Simulation als Kriterium zu verwenden ist. Wurde eine stabile Konfiguration gefunden, kann nun durch die akustische Skalierung die räumliche Auflösung beliebig verändert werden. Die Ausgabegröße des so generierten Ensembles wird nun nicht im Vorhinein beschränkt, stattdessen werden zusätzlich niedrig aufgelöste Versionen generiert und in das Ensemble eingefügt. Die Auflösung sollte dabei zunächst so hoch wie möglich gewählt werden, bei gleichzeitiger Erhaltung der Interaktivität der Anwendung. Für das so erhaltene Ensemble wird dann eine Ähnlichkeitsdarstellung generiert und es kann visuell überprüft werden, welche räumliche Auflösung für den Datensatz ausreichend ist. Im Idealfall liegt eine Konvergenz der Ähnlichkeiten bei konstanter Erhöhung der Auflösung vor und die in Zukunft verwendete kann durch das Abwägen der Ähnlichkeit zur nächst höheren bestimmt werden. Da es sich hierbei um einen qualitativen Prozess handelt, sollte im Zweifel eher eine höhere anstatt einer niedrigeren Auflösung gewählt werden. Zusätzlich kann nun die Ähnlichkeit der gewählten Simulation zu ihrer geringer aufgelösten Version betrachtet werden. Fällt diese ebenfalls gering in Relation zur Ähnlichkeit zu anderen Auflösungen aus, so wurde eine angemessene Auflösung für die Ausgabe gefunden. Falls nicht, kann diese erhöht werden, bis ein zufriedenstellendes Ergebnis erreicht wurde. Damit wird die Ähnlichkeitsdarstellung an den Anfang der Analyse gestellt und bietet einen Überblick über das gesamte Ensemble. In einem nächsten Schritt kann wie zuvor beschrieben nun auch der Einfluss der einzelnen Parameter auf die Ähnlichkeit zu den gemessenen Daten bestimmt werden. Dafür werden

nun die räumliche und zeitliche Auflösung fixiert und die Dichte, Viskosität, charakteristische Geschwindigkeit und Smagorinsky-Konstante variiert, sodass möglicherweise weitere Parameter fixiert werden können.

Die Ähnlichkeitsdarstellung bietet somit schon eine Grundlage, den gemessenen Daten ähnliche Simulationen zu generieren. Allerdings ist nicht klar, woraus Unähnlichkeiten resultieren. Zur Beantwortung dieser Frage werden interaktive Werkzeuge in koordinierten Ansichten geboten, deren Zusammenspiel im Folgenden erläutert wird.

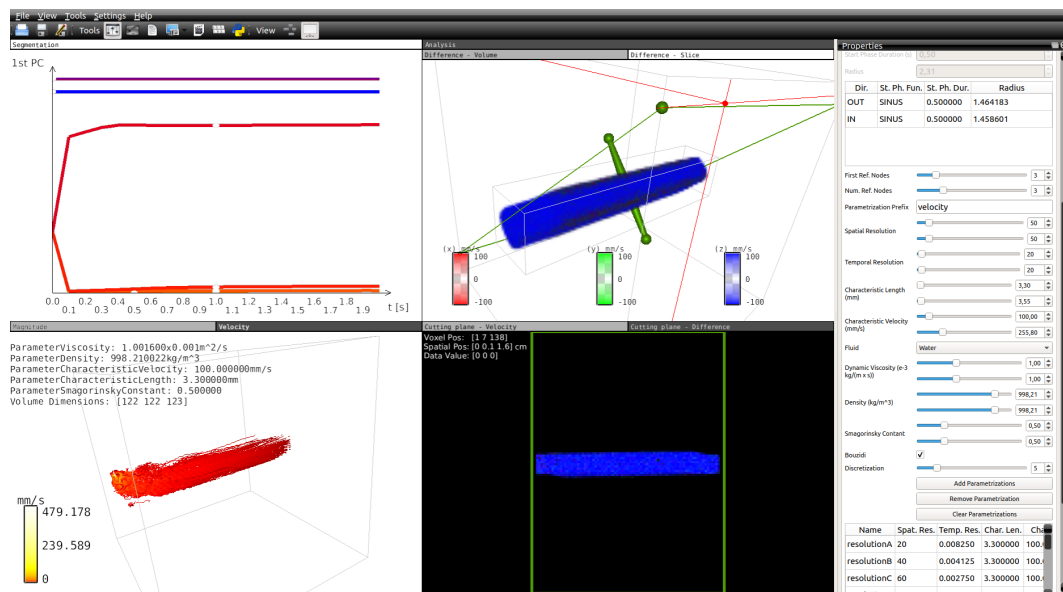
**Volumenvisualisierung** Mittels DVR-Raycasting<sup>4</sup> und einer passenden Transferfunktion kann die Magnitude eines einzelnen oder mehrerer Zeitschritte desselben oder verschiedener Ensembleteilnehmer nebeneinandergestellt visualisiert werden. Als Transferfunktion kann beispielsweise eine *Heatmap*, also ein mit zunehmender Magnitude von rot über gelb nach weiß verlaufender Gradient, verwendet werden. Zusätzlich können Informationen über den jeweiligen Zeitschritt in Textform eingeblendet werden, wie etwa die Simulationsparameter, der Zeitschritt und der Name des Simulationslaufs, bzw. der Messung. Da die Magnitude keine Auskunft über die Richtung gibt, kann auch für jeden Kanal eine eigene Transferfunktion gewählt werden. Da so jedoch das Problem der Verdeckung von im Inneren liegenden Strömungen besteht, wird zusätzlich eine Ebene visualisiert, welche anhand einer Griff-Metapher beliebig ausgerichtet werden kann. Der Schnitt des Volumens mit der Ebene wird dann in einer weiteren Ansicht dargestellt. Durch das Bewegen des Mauscursors kann zu jedem Pixel des Schnitts die lokale Geschwindigkeit angezeigt werden. Da die Ausrichtung des Schnittbildes jedoch nicht eindeutig ist, wird während der Bewegung des Cursors gleichzeitig die entsprechende Position in der dreidimensionalen Volumendarstellung angezeigt.

**Strömungslinien** Für jeden Zeitschritt können durch klassische Runge-Kutta-Integration Strömungslinien generiert und als Menge von kolorierten Pfaden visualisiert werden. Dabei können die lokale Richtung und Magnitude farblich kodiert werden. Anhand der Strömungslinien zweier Zeitschritte lassen sich optisch leicht Unterschiede in Hinblick auf Turbulenzen feststellen. Zusätzlich können diese mit dem *Quickbundles*-Algorithmus [8] gebündelt werden, sofern es aufgrund der Menge an Strömungslinien zur Verdeckung interessanter Regionen kommt.

**Differenzvisualisierung** Das Visualisieren zweier Zeitschritte mit den genannten Methoden kann bereits zu einer schnellen Feststellung der Herkunft der Unähnlichkeit führen. Unterschiede können jedoch auch unauffällig sein, da sie sich über die gesamte Domäne verteilen oder aber lokal begrenzt sind. Dafür kann durch Bilden der Differenz beider repräsentativer Vektorfelder ein

---

<sup>4</sup>DVR: Direct Volume Rendering



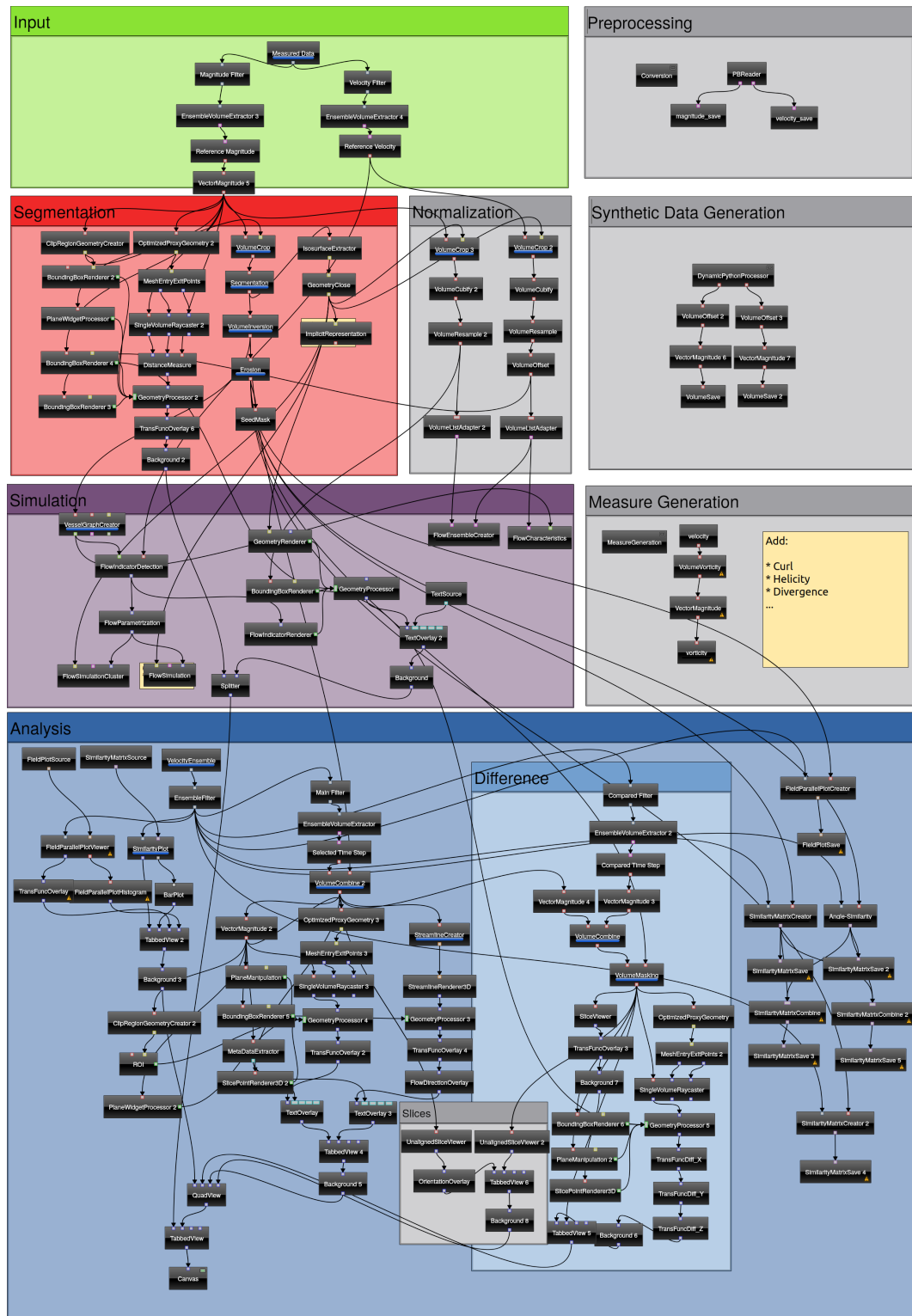
**Abbildung 4.5:** Die Benutzeroberfläche der Anwendung für den Endanwender. Zu sehen sind die koordinierten Ansichten, welche über Interaktion mit der Maus und der Tastatur zu bedienen sind. Auf der rechten Seite befinden sich die logisch gruppierten Einstellungsmöglichkeiten.

neues erzeugt und erneut mittels einer Volumenvisualisierung dargestellt werden. Dafür sollte jeder Kanal eine unterscheidbare Transferfunktion erhalten, sodass Regionen geringer Unterschiede eine hohe Transparenz und jene mit großen Unterschieden eine hoher Opazität erhalten. Bewährt hat sich dabei die Verwendung der Farben rot, grün und blau für die x-, y- und z-Richtung. Auch hier kann, analog zur Volumenvisualisierung eines Zeitschrittes, ein beliebiger Schnitt durch das Volumen dargestellt werden, um Details im Inneren erkennbar zu machen.

## 4.3 Die Anwendung

Basierend auf Voreen (vgl. Abschnitt 2.4) wurden alle genannten Werkzeuge in einer interaktiven Anwendung mit koordinierten Ansichten vereint. Die Logik wurde dabei durch vorhandene, modifizierte und zum Zwecke dieser Arbeit erstellte Prozessoren in einem komplexen Netzwerk zusammengeführt, dargestellt in Abbildung 4.6. Für den Endanwender ist die Komplexität des Netzwerks deutlich zu hoch, weshalb im sogenannten Anwendungsmodus nur die wichtigsten Einstellungsmöglichkeiten logisch gruppiert zur Verfügung gestellt werden. Die so erhaltene Anwendung ist in Abbildung 4.5 dargestellt.

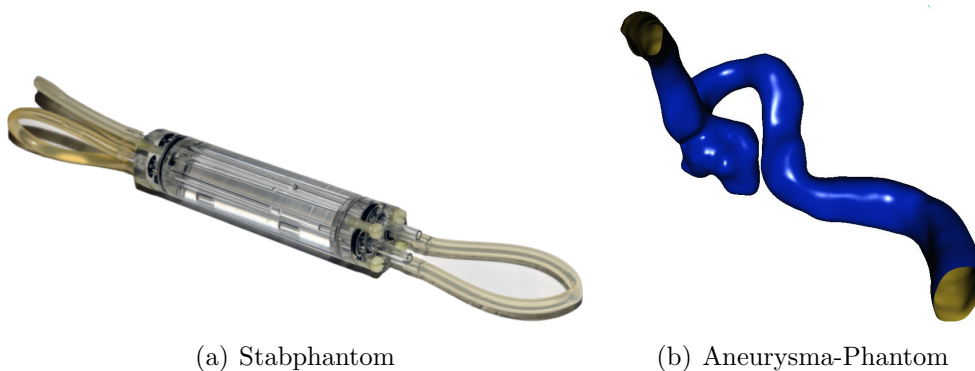
## 4 Methodik



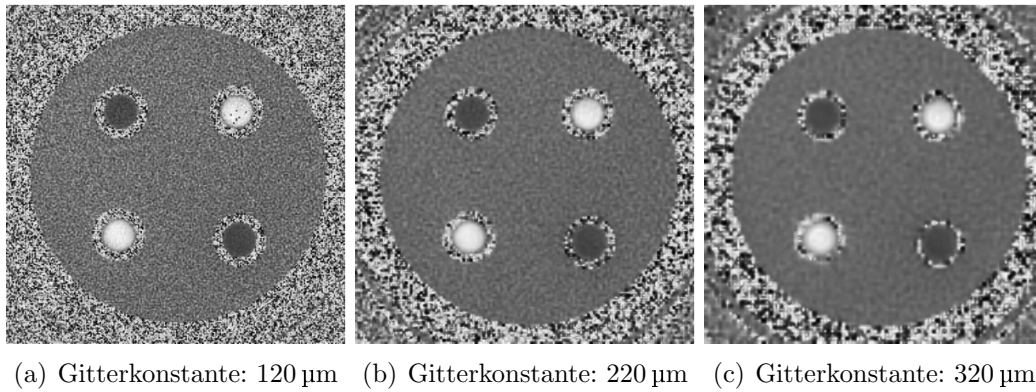
**Abbildung 4.6:** Das Netzwerk hinter der Anwendung. Einzelne Prozessoren bilden Teilnetzwerke, beispielsweise zur Segmentierung oder Simulation und sind entsprechend gruppiert. Zusätzlich sind viele Einstellungen der Prozessoren über Links miteinander synchronisiert, welche der Übersicht halber ausgeblendet wurden.

## 5 Ergebnisse

Gegenstand dieses Kapitels ist das Anwenden der zuvor vorgestellten Analyseprozedur auf zwei vorliegende Datensätze unterschiedlicher Komplexität, welche mittels Phasenkontrast-MR unter Verwendung von Phantomen akquiriert wurden. Diese sind in Abbildung 5.1 dargestellt. Als Fluid wurde jeweils Leitungswasser bei Zimmertemperatur und Normaldruck verwendet, da die Daten im Wesentlichen für die Erforschung neuer Strömungsmerkmale generiert wurden. Mithilfe dieser Merkmale sollen kritische Veränderungen des kardiovaskulären Systems frühzeitig erkannt werden. Da ein Teil der notwendigen Schritte für die Analyse beider Datensätze identisch abläuft, werden diese exemplarisch am Beispiel des ersten erklärt. Bei diesem handelt es sich um den denkbar simplen Fall laminarer Strömung in einer, bzw. mehrerer identischer Röhren, für den sich zudem eine analytische Lösung herleiten lässt. Aus diesem Grund wurden zum Zweck der Validierung der Methoden synthetische Daten generiert, auf welche im folgenden Unterabschnitt eingegangen wird. Der zweite Datensatz ist dagegen komplex in Bezug auf die Simulationsgeometrie, da es sich um ein Aneurysma-Phantom handelt. Da jeweils eine statische Strömung angestrebt wurde, ist nur ein einzelner Zeitschritt je Messung vorhanden. Abschließend folgt eine Diskussion, in der die Erkenntnisse der Analyse herausgestellt werden.



**Abbildung 5.1:** Die zwei in dieser Arbeit betrachteten Phantome. Zur Verfügung gestellt von Verena Hoerr, Ali Nahardani und Philipp Rene Bovenkamp [2].

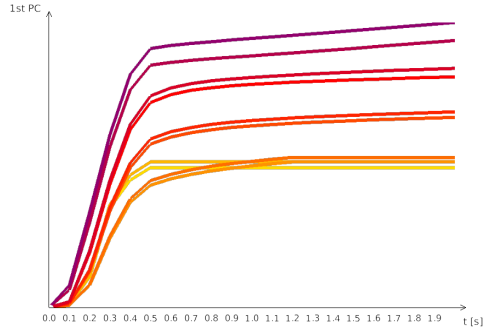


**Abbildung 5.2:** Querschnitt durch den Stabphantom-Datensatz, welcher drei Messungen unterschiedlicher Auflösungen bei sonst identischen Bedingungen enthält. Im Inneren sind die vier kleinen Röhrrchen gut zu erkennen, ebenso wie die unterschiedlichen Strömungsrichtungen, welche mit weiß bzw. mit schwarz farblich kodiert sind. Die Daten wurden von Ali Nahardani akquiriert.

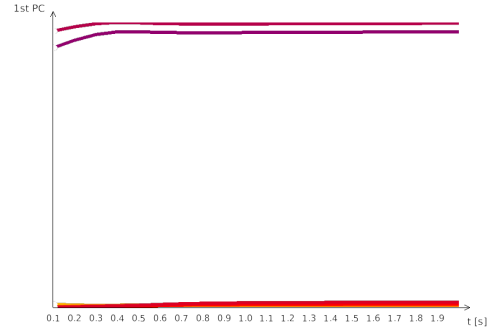
## 5.1 Der Stabphantom-Datensatz

Beim sogenannten Stabphantom handelt es sich um einen Zylinder aus Acrylglas, welcher vier kleinere Röhrrchen desselben Materials enthält. Zwei dieser Röhrrchen sind auf einer Seite über Schläuche an eine Pumpe angeschlossen und die übrigen Enden wurden so miteinander verbunden, dass ein geschlossener Kreislauf entsteht. Die Messung wurde mit einer von Ali Nahardani entwickelten Sequenz in einem 9.4 T-MR-Scanner mit verschiedenen Auflösungen durchgeführt. Aus den so erhaltenen Daten wurde für jede Auflösung das Geschwindigkeitsfeld rekonstruiert, deren Querschnitte in Abbildung 5.2 zu sehen sind. Regionen mit starkem Rauschen waren während der Messung mit Luft gefüllt, jene mit schwachem Rauschen hingegen mit stehendem Wasser. Für die folgende Analyse wird dabei nun repräsentativ eines der vier inneren Röhrrchen verwendet. Zu den Daten existieren hierbei keine anatomischen Bilder, die für eine Segmentierung herangezogen werden können. Daher wurde der Datensatz mit der höchsten Auflösung gewählt und die Magnitude aus dem Vektorfeld berechnet. Die Segmentierung wurde zunächst auf dem gesamten Datensatz durchgeführt, analog zu dem in Unterabschnitt 4.2.2 vorgestellten Verfahren. Im Detail wurden dabei Gauß- und Medianfilter zum Entfernen des Rauschens angewendet, gefolgt von einer Binarisierung und einer Erosion, um das hohe Rauschen rings um die kleinen Röhrrchen auszuschneiden. Die Ein- und Auslässe wurden dann automatisch detektiert und die charakteristische Länge und Geschwindigkeit auf den inneren Durchmesser der Röhrrchen, bzw. auf die höchste Magnitude im Datensatz gesetzt. Da sich die Strömungen zwischen den vier Röhrrchen nicht oder nur geringfügig unterscheiden, wird letztlich nur eines repräsentativ für die weitere Analyse verwendet.

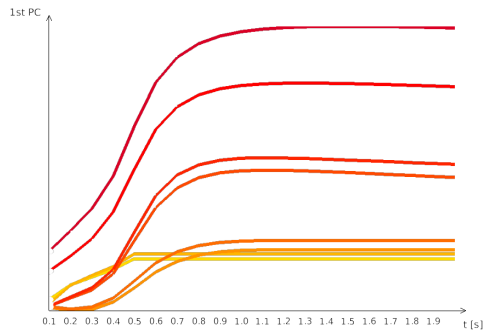




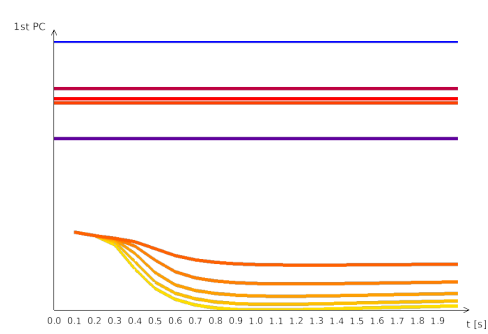
(a) Ähnlichkeit der Magnitude bei Änderung der Simulationsauflösung



(b) Ähnlichkeit der Richtung bei Änderung der Simulationsauflösung



(c) Kombinierte Ähnlichkeit (Metrik 4.12) von Magnitude und Richtung nach Entfernen der Ausreißer und bei Änderung der Simulationsauflösung



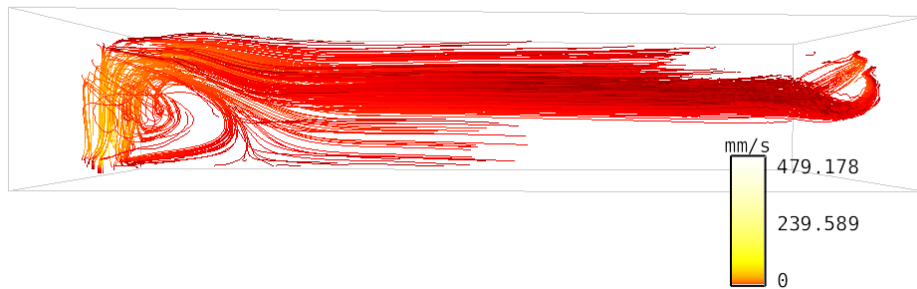
(d) Kombinierte Ähnlichkeit (Metrik 4.12) bei Änderung der charakteristischen Geschwindigkeit

**Abbildung 5.3:** Ähnlichkeitsdarstellungen zur Analyse des Stabphantom-Datensatzes.

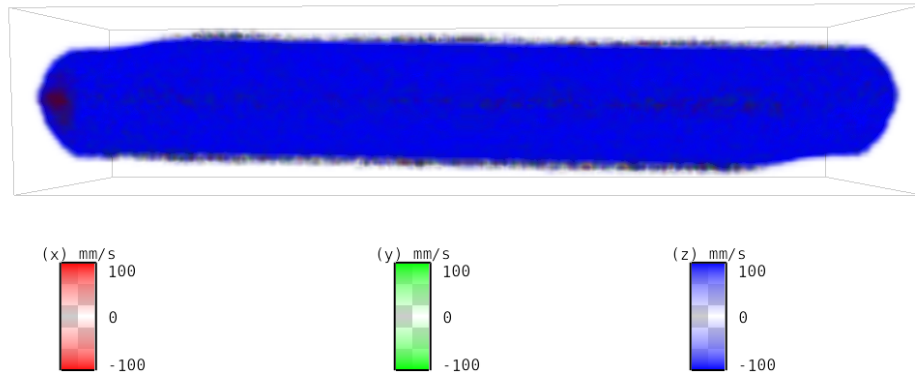
Zunächst muss die räumliche Simulations- und Ausgabeauflösung bestimmt werden. Dafür wurde ein Ensemble generiert, bei dem die räumliche Simulationsauflösung variiert und die Ausgabeauflösung nicht begrenzt wurde. Alle anderen Parameter wurden auf plausible Wert gesetzt, d. h. für die Dichte und Viskosität wurden die Literaturwerte für Wasser bei 20 °C gesetzt. Die Smagorinsky-Konstante wurde auf den Wert 0.5 gesetzt, was bereits recht laminare Strömungen zur Folge haben kann (vgl. Unterabschnitt 2.2.4). Für die räumliche Auflösung, welche sich auf die charakteristische Länge bezieht, wurden die Werte 20, 40, 60, 80 und 100 gewählt, was entsprechend der physikalischen Dimensionen des Datensatzes eine effektive Auflösung von 122, 245, 368, 491 und 614 Voxeln je Zeitschritt zur Folge hat. Als Simulationsergebnisse wurden dabei ausschließlich die Vektorfelder der Geschwindigkeiten gewählt. Die zeitliche Auflösung wurde initial auf 20 gesetzt, was unmittelbar stabile Simulationen lieferte. Die so erhaltenen Simulationen wurden anschließend auf eine Größe von  $128^3$  Voxeln reduziert, da mit Auflösung jenseits von  $256^3$  Voxeln je Zeitschritt keine interaktive Analyse mehr stattfinden kann. Aus die-

sen und den hochaufgelösten Simulationsläufen wurde ein Ensemble erstellt, welches mit der Ähnlichkeitsdarstellung, zu sehen in Abbildung 5.3, visualisiert wurde. Die Dimensionen der Projektion wurden dabei stets so gewählt, dass die verwendeten Hauptkomponenten den größten Teil der Informationen bewahren. Die Betrachtung der Ähnlichkeiten in Bezug zur Magnitude offenbart eine Konvergenz bei steigender Auflösung: Abbildung 5.3(a) zeigt fünf Paare von Linien. Diese stellen die Entwicklung der Ähnlichkeit der fünf hoch aufgelösten und dazugehörigen, niedrig aufgelösten Simulationsläufe dar. Der Farbgradient von violett über rot und orange zu gelb ist dem Anstieg der Auflösung zuzuordnen. In jedem Paar stellt die untere der beiden Linien die niedrig aufgelöste Version dar. Wichtig ist an dieser Stelle, dass der Verlauf nicht durch das Verringern der Auflösung beeinflusst wird. Dieser lässt insbesondere den Einfluss der sinusoiden Anlaufphase bis  $t = 0.5\text{ s}$  erkennen, auf die ein weitestgehend konstanter Verlauf folgt. Interessant ist hierbei der Simulationslauf der höchsten Auflösung, dargestellt in gelb, welcher einen anderen Verlauf aufweist, als ausgehend von den übrigen erwartet werden würde. Dabei sei angemerkt, dass die Simulation aufgrund dem Überschreiten der maximalen Rechenzeit frühzeitig abgebrochen wurde. Der letzte berechnete Zeitschritt wurde dafür als konstante Linie weitergeführt, um einen besseren Vergleich treffen zu können. Der Fehler sollte dabei zu vernachlässigen sein, da das Ende der Anlaufphase noch erreicht wurde. Als nächstes wird der Verlauf der Ähnlichkeit bezüglich der Richtung betrachtet, welcher in Abbildung 5.3(b) dargestellt ist. Hier ist eindeutig der Simulationslauf mit der geringsten Auflösung als Ausreißer zu erkennen, was auf einen zu großen numerischen Fehler hindeutet. Dieser Datensatz wurde daher aus dem Ensemble entfernt und anschließend die Ähnlichkeit mit Metrik 4.12 berechnet, zu sehen in Abbildung 5.3(c). Damit offenbart sich erneut die Konvergenz der Ähnlichkeiten. Zusätzlich ist eine Minderung des Unterschiedes zwischen den niedrig aufgelösten Versionen zu ihren hoch aufgelösten Äquivalenten bei steigender Auflösung zu verzeichnen. Damit wird die räumlichen Auflösung der Ausgabe für folgende Simulationen auf  $128^3$  beschränkt und die der Simulation selbst auf  $491^3$  Voxel.

In einem nächsten Schritt gilt es ein neues Ensemble zu generieren, in dem andere Parameter variiert werden. Zur Auswahl stehen die Dichte und Viskosität, die charakteristische Geschwindigkeit und die Smagorinsky-Konstante. Da es sich um eine simple Geometrie handelt, welche eine laminare Strömung enthalten soll, werden die Dichte und Viskosität weiterhin bei ihren Literaturwerten belassen. Die Smagorinsky-Konstante hat einen wesentlichen Effekt auf das Verhalten turbulenter Strömung, welche bei der vorliegenden Geometrie nicht zu erwarten ist. Daher wird diese weiterhin auf den Wert 0.5 gesetzt. Vielmehr hat bei einer laminaren Strömung in einem Zylinder die Geschwindigkeit am Einlass einen sehr großen Einfluss auf das resultierende Vektorfeld. Aus diesem Grund wird nun ausschließlich die charakteristische Geschwindigkeit variiert. Ferner wurden synthetische Daten zum Zwecke der Evaluation erstellt.



(a) Visualisierung der Strömungslinien im Vektorfeld des Simulationslaufs mit der höchsten charakteristischen Geschwindigkeit



(b) Visualisierung der Differenz der Vektorfelder eines gemessenen und der Simulationslaufs mit der höchsten charakteristischen Geschwindigkeit.

**Abbildung 5.4:** Visualisierung der lokalen Unähnlichkeit anhand von Strömungslinien und der Differenz zweier Vektorfelder.

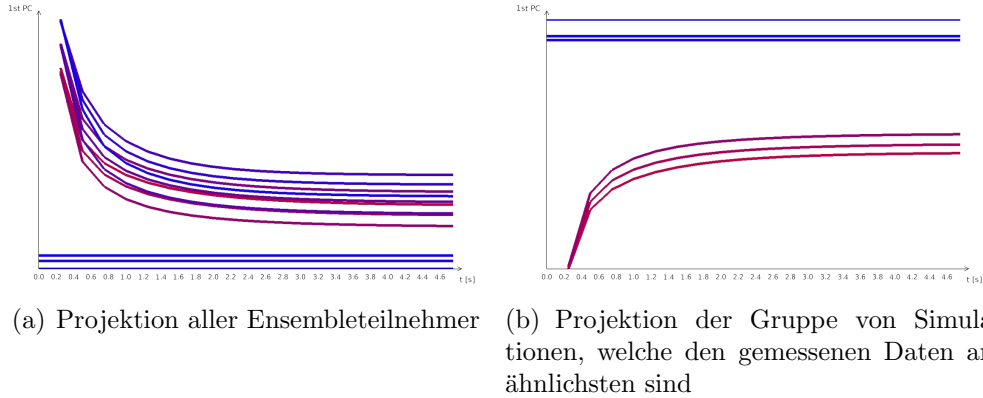
Dabei wurde das Poiseuille-Geschwindigkeitsprofil (vgl. Unterabschnitt 2.2.3) entlang der segmentierten Geometrie ausgewertet, sodass im Mittelpunkt des Querschnitts die höchste Magnitude vorliegt und zum Rand hin quadratisch abnimmt. Zusätzlich wurde ein Datensatz hinzugefügt, welcher konstant über den gesamten Querschnitt die höchste Magnitude annimmt. Letztlich wurde aus all den genannten Datensätzen ein Ensemble erstellt. Um einen Überblick über dieses zu erhalten, wird erneut zu Beginn die Ähnlichkeitsvisualisierung generiert. Nun und im Folgenden wird dabei stets Metrik 4.12 verwendet. In Abbildung 5.4 sind der konstante (blau) und der Poisseuille-Datensatz (violett) zu sehen, zwischen denen die Messungen verlaufen. In der unteren Hälfte befinden sich die Kurven der simulierten Datensätze, wobei die linear

von  $100 \text{ mm s}^{-1}$  bis  $300 \text{ mm s}^{-1}$  zunehmende, charakteristische Geschwindigkeit durch den Gradienten von orange nach gelb repräsentiert wird. Wider Erwarten nimmt die Ähnlichkeit der Simulationen zu den gemessenen und synthetischen Daten mit zunehmender Magnitude ab. Die Ursache dafür findet sich durch das Betrachten der Strömungslinien in den simulierten Daten, welche exemplarisch für den Datensatz mit der höchsten Geschwindigkeit in Abbildung 5.4(a) dargestellt sind. Offenbar treten unerwünschte Effekte beim Einlass auf, welche zu einer lokal begrenzten Verwirbelung führen. Die Verwendung der Differenzvisualisierung, zu sehen in Abbildung 5.4(b), zeigt diesen Effekt ebenfalls. Zudem ist dort ein relativ homogener Unterschied der Magnituden zu sehen, welcher auf die Längsachse des Zylinders beschränkt ist. Dies zeigt, dass die charakteristische Geschwindigkeit zu gering gewählt wurde. In einer weiteren Iteration kann nun ein weiteres Ensemble von Simulationen mit erhöhter charakteristischer generiert werden. Dies führt zu diesem Zeitpunkt jedoch zu keiner neuen Erkenntnis. Stattdessen kann festgehalten werden, dass an den Einlässen und teilweise auch an den Auslässen unerwartete Effekte auftreten können, welche mithilfe der Anwendung detektiert werden konnten.

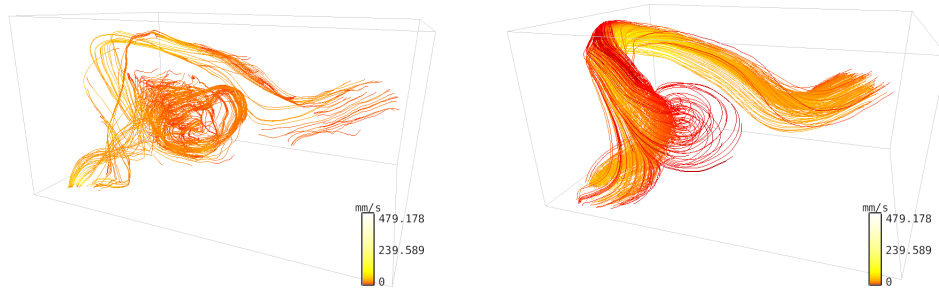
### 5.2 Der Aneurysma-Phantom-Datensatz

Repräsentativ für eine komplexe Geometrie wurde die Nachbildung eines Aneurysmas verwendet, welche im 3D-Druck erstellt und in Flüssiglatex getaucht wurde, um eine gewisse Elastizität von Gefäßwänden nachzuempfinden. Die Bewegung bzw. Expansion der Gefäßwände findet allerdings unterhalb der Genauigkeit der Messtechnik statt, sodass dieser Effekt für die Strömungssimulationen vernachlässigt werden kann. Mit diesem Phantom wurden drei Messungen von Philipp Rene Bovenkamp angefertigt. Aus der Analyse des Stabphantom-Datensatzes ist bekannt, wie das Bestimmen der räumlichen Simulations- und Ausgabeauflösung erfolgen kann und wird daher als gegeben angenommen. Die Segmentierung wurde wie bereits in Abschnitt 4.2.2 erläutert und in Abbildung 4.4 dargestellt, durchgeführt. Für den Aneurysma-Datensatz soll der Effekt der übrigen Simulationsparameter auf den Simulationsverlauf ermittelt werden. Dafür wurde ein Ensemble generiert, bei dem die Viskosität, Dichte und charakteristische Geschwindigkeit modifiziert wurde. Dabei wurde Blut als Fluid angenommen, um den Effekt von Dichte und Viskosität zu ermitteln. Dafür wurde der in der Literatur spezifizierte Wertebereich (vgl. Tabelle 4.2) durch drei Werte diskretisiert, wodurch der obere, untere und mittlere Wert verwendet wurde. Aus vorangehender Analyse konnte die Erkenntnis gewonnen werden, dass die charakteristische Geschwindigkeit höher sein muss als die maximale Magnitude im gemessenen Datensatz, um im späteren Verlauf entsprechende Magnituden zu generieren. Aus diesem Grund wurde die charakteristische Geschwindigkeit für das Ensemble auf zwei, bis dreimal höhere Werte gesetzt. Die Smagorinsky-Konstante wurde auf 0.1 festgelegt, da turbulente Strömungen im Inneren des Aneurysmas nicht ausgeschlossen werden

können.



**Abbildung 5.5:** Ähnlichkeitsdarstellung des Ensembles, welche mit dem Aneurysma-Datensatz erstellt wurde.



**Abbildung 5.6:** Visualisierung von Strömungslinien im Vektorfeld der gemessenen Daten und in dem entsprechend ähnlichsten Zeitschritt der Simulation.

Nach der Durchführung der Simulationen offenbart die Ähnlichkeitsdarstellung unter Verwendung von Metrik 4.12 erneut mehrere Gruppen, welche den Einfluss der Parameter zeigen. Abbildung 5.5(a) zeigt die gemessenen Daten (blau) am unteren Rand. Auf der linken Seite werden drei Gruppen sichtbar, welche sich über die Zeit auffächern. Dabei liegt bei der untersten Gruppe die höchste charakteristische Geschwindigkeit vor und bei der obersten entsprechend die niedrigste. Wird die Gruppe höchster Ähnlichkeit zu den gemessenen Daten nun separat betrachtet, wird die Aufspaltung deutlicher. Dabei gehören dieser Gruppe insgesamt neun Simulationsläufe an, von denen jedoch in Abbildung 5.5(b) nur drei zu erkennen sind. Alleine die Änderung der Viskosität ist dabei verantwortlich für die unterschiedliche Entwicklung der Kurven, denn

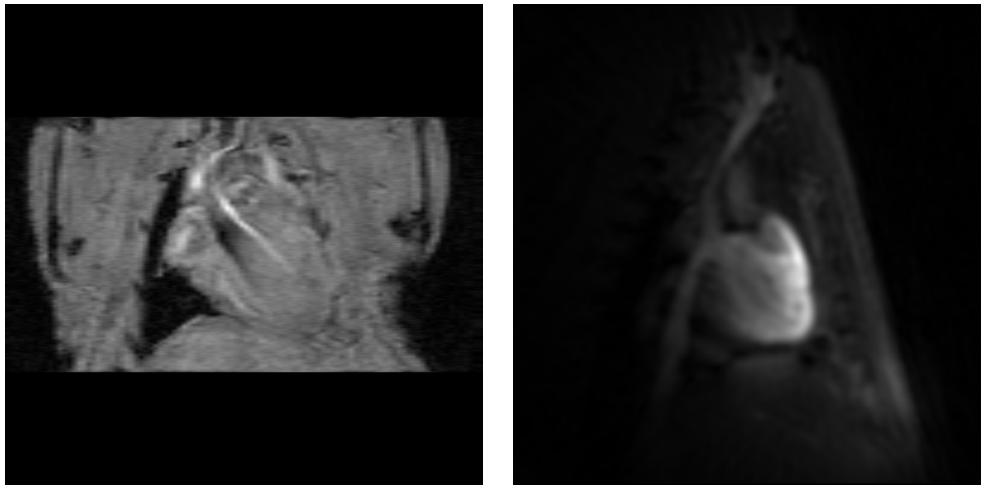
jene drei der unterschiedlichen Dichten je Wert für die Viskosität verlaufen exakt übereinander. Es ist nicht anzunehmen, dass das Variieren der Dichte keinen Effekt hat, jedoch ist dieser offenbar vernachlässigbar im Vergleich zu anderen Parametern. Für den Zeitschritt der Simulationsläufe, welcher den gemessenen Daten am ähnlichsten ist, wurden abschließend für einen Vergleich Strömungslinien berechnet und visualisiert. Das in den gemessenen Daten enthaltene Rauschen ist dabei der wesentliche Grund für das Auftreten unterbrochener Strömungslinien. Im Gegensatz dazu bieten die simulierten Daten unterbrechungsfreie Strömungslinien, welche auch im Aneurysma zirkulieren. Die dort vorherrschende Physik wird damit weitestgehend von der Simulation und der entsprechenden Parametrisierung modelliert.

### 5.3 Diskussion

Die vorgestellten Methoden wurden auf zwei Datensätze unterschiedlicher Komplexität angewendet. Dabei konnten Beobachtungen gemacht werden, die bei dem verwendeten Lattice-Boltzmann-Verfahren, bzw. der Implementierung mittels OpenLB, für die Parametrisierung und Analyse beachtet werden müssen. So darf die Region unmittelbar an den Ein- und Auslässen nicht in den Vergleich von Einsebleteilnehmern involviert werden, da dort unrealistische Werte vorliegen können. Ferner muss die charakteristische Geschwindigkeit stets deutlich höher gewählt werden, als die maximale Magnitude im gemessenen Datensatz. Eine konkrete Regel diesbezüglich konnte bislang nicht ausgemacht werden. Durch den Ensembleansatz konnte der Effekt der einzelnen Parameter auf den Verlauf der Simulation ermittelt werden. Die Dichte spielt dabei die geringste Rolle, gefolgt von der Viskosität. Den größten Einfluss hat die charakteristische Geschwindigkeit. Über den Effekt der Smagorinsky-Konstante lässt sich für beliebige Simulationsgeometrien keine allgemeine Aussage treffen, da Turbulenzen stark von deren Beschaffenheit abhängen. Die räumliche Simulations- und Ausgabeauflösung sollte ebenfalls für jeden Datensatz ermittelt werden, um eine physikalisch akkurate Simulation zu gewährleisten.

## 6 Fazit und Ausblick

Das Ziel dieser Arbeit war das Visualisieren von Ensembles bestehend aus mittels 4D-Phasenkontrast-MR akquirierten sowie durch Strömungssimulationen generierten Daten zum Zweck der Datenassimilation. Dafür wurde eine auf dem Voreen-Framework basierende Anwendung implementiert, welche die gemessenen Daten verarbeiten, Strömungssimulationen durchführen und die Ähnlichkeit der beiden visualisieren kann. Dies beinhaltet die Segmentierung der gemessenen Daten durch die Anwendung von Gauß- und Medianfiltern, einer Binarisierung, morphologischen Operation und Schwellenwertoperationen sowie der abschließenden Extraktion einer Isofläche. Letztere wird als Simulationsgeometrie verwendet und es lassen sich automatisiert Ein- und Auslässe anhand des Gefäßgraphen detektieren. Diese können für die Parametrisierung einer Lattice-Boltzmann-Simulation verwendet werden, deren Implementierung auf dem OpenLB-Framework basiert. Die Ausführung dieser kann sowohl lokal als auch auf einem Rechencluster wie dem PALMA-II erfolgen. Die Werte einiger weiterer Parameter lassen sich ebenfalls automatisiert anhand des Datensatzes festlegen, andere müssen jedoch experimentell ermittelt werden. Dafür wurde ein Ensembleansatz gewählt, bei dem zunächst die allgemeine Auswirkung der Parameter auf den Verlauf der Simulation und die Ähnlichkeit zu den gemessenen Daten bestimmt wird. Dadurch ließen sich die Dichte und Viskosität, sowie die zeitliche und räumliche Auflösung für weitere Simulationen festsetzen. Die maximale Geschwindigkeit des Poiseuille-Profiles an den Einlässen und der Wert der Smagorinsky-Konstante haben den größten Einfluss auf den Verlauf der Simulationen. Dies konnte anhand einer Ähnlichkeitsdarstellung unter Verwendung von klassischer Multidimensionaler Skalierung und verschiedenen Metriken zur Generierung von Distanzmatrizen festgestellt werden. Betrachtet wurden dafür die Vektorfelder der Geschwindigkeiten jedes Zeitschrittes, da diese sowohl für die gemessenen, als auch für die simulierten Daten vorliegen. Die Ähnlichkeiten von Magnitude und Richtung wurden zunächst getrennt ermittelt und anschließend zu einer neuen Distanzmatrix kombiniert, wobei jeder Zeitschritt jedes Ensembleteilnehmers mit jedem anderen verglichen wurde. Mit einem quasi-Monte Carlo-Ansatz wurden innerhalb der zu vergleichenden Domäne randomisierte Datenpunkte ermittelt. Auf deren lokale Magnituden wurde die verallgemeinerte Feldähnlichkeit von Fofonov *et al.* angewendet. Die Ähnlichkeit der Richtung zweier Vektorfelder wurde durch den Durchschnitt aller Winkel zwischen den lokalen Richtungen an den Datenpunkte definiert. Kombinationen der Distanzmatrizen beider Vektorfelder wurden durch das Wählen des Maximums der Unähnlichkeiten sowie das Multi-



(a) Aufnahme mit der FLASH-Sequenz (aktueller Goldstandard). (b) Aufnahme mittels UTE-Bildgebung.

**Abbildung 6.1:** Querschnitt durch das anatomische Bild eines arbiträren Zeitschritts der beiden vorliegenden Datensätze zweier Mäuse. Im Zentrum ist jeweils das Herz der Maus zu sehen. Die Daten wurden durch Verena Hoerr zur Verfügung gestellt.

plizieren der entsprechenden Ähnlichkeiten jedes Eintrags erzeugt. Damit lässt sich zudem eine beliebige Anzahl von Merkmalen kombinieren. Die Ergebnisse der händisch zu gewichtende Metrik von Jiang *et al.* waren dagegen auf synthetischen Daten nicht sinnvoll interpretierbar. Das Projizieren der Ähnlichkeiten erfolgte in bis zu drei Dimensionen, wobei die ersten drei Hauptkomponenten und, im eindimensionalen Fall, die Zeit als Achsen verwendet wurden. Die Methoden wurden auf einen Stabphantom- und einen Aneurysma-Datensatz der Arbeitsgruppe *Magnetresonanz* des Instituts für Medizinische Mikrobiologie der Universität Jena angewendet und mithilfe weiterer interaktiver Werkzeuge in mehreren koordinierten Ansichten ausgewertet. Dazu gehört sowohl die Volumenvisualisierung einzelner Zeitschritte, mittels DVR-Raycastings und einer beliebigen Schnittebene, als auch die Darstellung von Strömungslinien. Auch die Differenz zweier Vektorfelder kann berechnet und ebenfalls in Gänze oder im Schnitt betrachtet werden, um den genauen Ort der Unähnlichkeiten ausmachen zu können.

Es wurde gezeigt, dass es kein triviales Unterfangen oder gar unmöglich ist, eine Simulation zu generieren, die den gemessenen global sehr ähnlich ist. Da Letztere starkes, durch die Messtechnik bedingtes Rauschen mit sich führen, ist das Ziel mitnichten eine hochaufgelöste Reproduktion der gemessenen Daten mittels physikalisch akkuraten Simulationen anzufertigen. Stattdessen soll an dieser Stelle die Erkenntnis festgehalten werden, dass bestimmte lokale Merkmale, die auf das Einsetzen oder Vorhandensein einer Krankheit hindeuten, entweder in sowohl gemessenen als auch simulierten Daten vorkommen müssen



oder aber in keiner der beiden. Dafür reicht die Verwendung der Geschwindigkeit in Form von Magnitude und Richtung in den meisten Fällen nicht aus. Es müssen also weitere Merkmale involviert werden, wie der Druck, die Scherung an der Wand, die Vortizität und weitere. Es sollte somit vielmehr eine physikalisch plausible Simulation durchgeführt werden, welche einen Teil der wesentlichen Eigenschaften der gemessenen Daten widerspiegelt, wie Turbulenzen, laminare Strömungen oder Druckgradienten. Dann ist möglicherweise anzunehmen, dass auch andere Bereiche der Simulation der zugrunde liegenden Physik entspricht. Beispielsweise ist die Strömung unmittelbar entlang der Gefäßwand bei gemessenen Daten oftmals technisch bedingt nicht akkurat messbar. Teilweise haben abzweigende Gefäße einen derart geringen Durchmesser, das keine Messtechnik mehr sinnvolle Daten liefern kann. Diese Datenarmut und das allgemeine Problem der geringen räumlichen und zeitlichen Auflösung der gemessenen Daten kann durch Simulationen kompensiert werden, sofern an anderer Stelle eine ausreichende Übereinstimmung ermittelt werden konnte. Damit stellt diese Arbeit eine Grundlage für fortgeschrittene Datenassimilation dar, die Simulationen können beispielsweise durch eine Kopplung an die gemessenen Daten gesteuert werden. Dafür bedarf es zudem der Betrachtung von Zeitreihen gemessener Daten, welche in dieser Arbeit nicht erfolgte. Es liegen bereits Datensätzen vor, welche den Herzzyklus zweier Mäuse mit großen Teilen der Brustregion mit bis zu 50 Zeitschritten aufzeichnen, zu sehen in Abbildung 6.1. Dabei wird stets an neuen Sequenzen für eine Verbesserung der Messergebnisse geforscht. Aktuell etabliert sind unter anderem die sogenannte FLASH<sup>1</sup>-Sequenz, sowie die UTE-Bildgebung<sup>2</sup>. Die vorgestellten Methoden können somit künftig auch verwendet werden, um verschiedene Sequenzen, bzw. Bildgebungsverfahren miteinander zu vergleichen. Problematisch ist bei den genannten Daten jedoch das starke Pulsieren der Aorta und die Atembewegungen der Maus. Dies führt zu einer Änderung der Simulationsgeometrie über die Zeit. Bisher wurde in der Literatur primär die menschliche Aorta untersucht, deren Bewegung in Relation zur räumlichen Auflösung vernachlässigt wird. Die Segmentierung dieser komplexeren Daten kann zudem beispielsweise mit dem halb automatischen *Random-Walker*-Verfahren erfolgen, bei dem manuell einige Labels für das Gefäßinnere und -äußere festgelegt werden müssen. Die Transition von einer Geometrie zu der des nächsten Zeitschrittes ist jedoch in der Simulation zu berücksichtigen, was einen ganz eigenen Themenbereich definiert. Ferner kann die implementierte Anwendung um weitere Werkzeuge erweitert werden, wie beispielsweise um Pfadlinien, welche sich als Merkmal zum Vergleich von Zeitreihen heranziehen lassen.

---

<sup>1</sup>FLASH: Fast low angle shot

<sup>2</sup>UTE: Ultrashort echo time



# Literaturverzeichnis

- [1] BOUZIDI, M'HAMED, MOUAOUIA FIRDAOUSS und PIERRE LALLEMAND: *Momentum transfer of a Boltzmann-lattice fluid with boundaries*. Physics of fluids, 13(11):3452–3459, 2001.
- [2] BOVENKAMP, PHILIPP RENE, TOBIAS BRIX, FLORIAN LINDEMANN, RICHARD HOLTMEIER, DESIREE ABDURRACHIM, MICHAEL T KUHLMANN, GUSTAV J STRIJKERS, JÖRG STYPMANN, KLAUS H HINRICHS und VERENA HOERR: *Velocity mapping of the aortic flow at 9.4 T in healthy mice and mice with induced heart failure using time-resolved three-dimensional phase-contrast MRI (4D PC MRI)*. Magnetic Resonance Materials in Physics, Biology and Medicine, 28(4):315–327, 2015.
- [3] DREES, DOMINIK, AARON SCHERZINGER, RENÉ HÄGERLING, FIEDEMANN KIEFER und XIAOYI JIANG: *Scalable Robust Graph and Feature Extraction for Arbitrary Vessel Networks in Volumetric Datasets*. (in preparation).
- [4] FOFONOV, ALEXEY und LARS LINSEN: *A Top-Down Interactive Visual Analysis Approach for Physical Simulation Ensembles at Different Aggregation Levels*. Information, 9(7):163, 2018.
- [5] FOFONOV, ALEXEY und LARS LINSEN: *Projected Field Similarity for Comparative Visualization of Multi-Run Multi-Field Time-Varying Spatial Data*. In: *Computer Graphics Forum*, Band 38, Seiten 286–299. Wiley Online Library, 2019.
- [6] FOFONOV, ALEXEY, VLADIMIR MOLCHANOV und LARS LINSEN: *Visual analysis of multi-run spatio-temporal simulations using isocontour similarity for projected views*. IEEE transactions on visualization and computer graphics, 22(8):2037–2050, 2015.
- [7] FUNKE, SIMON WOLFGANG, MAGNE NORDAAS, ØYVIND EVJU, MARTIN SANDVE ALNÆS und KENT ANDRE MARDAL: *Variational data assimilation for transient blood flow simulations: Cerebral aneurysms as an illustrative example*. International journal for numerical methods in biomedical engineering, 35(1):e3152, 2019.
- [8] GARYFALLIDIS, ELEFThERIOS, MATTHEW BRETT, MARTA MORGADO CORREIA, GUY B WILLIAMS und IAN NIMMO-SMITH: *Quickbundles, a*

- method for tractography simplification*. *Frontiers in neuroscience*, 6:175, 2012.
- [9] GROEN, DEREK, ROBIN A RICHARDSON, RACHEL COY, ULF D SCHILLER, HOSKOTE CHANDRASHEKAR, FERGUS ROBERTSON und PETER V COVENEY: *Validation of patient-specific cerebral blood flow simulation using transcranial Doppler measurements*. *Frontiers in physiology*, 9, 2018.
- [10] HENN, THOMAS, VINCENT HEUVELINE, MATHIAS J KRAUSE und SEBASTIAN RITTERBUSCH: *Aortic coarctation simulation based on the lattice Boltzmann method: benchmark results*. In: *International Workshop on Statistical Atlases and Computational Models of the Heart*, Seiten 34–43. Springer, 2012.
- [11] HOON, NHLC DE, ROY VAN PELT, A JALBA und ANNA VILANOVA: *4D MRI flow coupled to physics-based fluid simulation for blood-flow visualization*. In: *Computer Graphics Forum*, Band 33, Seiten 121–130. Wiley Online Library, 2014.
- [12] HOON, NIELS HLC DE, ANDREI C JALBA, ELMAR EISEMANN und ANNA VILANOVA: *Temporal Interpolation of 4D PC-MRI Blood-flow Measurements Using Bidirectional Physics-based Fluid Simulation*. In: *VCBM*, Seiten 59–68, 2016.
- [13] JASAK, HRVOJE, ALEKSANDAR JEMCOV, ZELJKO TUKOVIC et al.: *OpenFOAM: A C++ library for complex physics simulations*. In: *International workshop on coupled methods in numerical dynamics*, Band 1000, Seiten 1–20. IUC Dubrovnik Croatia, 2007.
- [14] JIANG, JINGFENG, KEVIN JOHNSON, KRISTIAN VALEN-SENDSTAD, KENT-ANDRE MARDAL, OLIVER WIEBEN und CHARLES STROTHER: *Flow characteristics in a canine aneurysm model: A comparison of 4D accelerated phase-contrast MR measurements and computational fluid dynamics simulations*. *Medical physics*, 38(11):6300–6312, 2011.
- [15] KÖHLER, BENJAMIN, SILVIA BORN, ROY FP VAN PELT, ANJA HENNEMUTH, UTA PREIM und BERNHARD PREIM: *A survey of cardiac 4D PC-MRI data processing*. In: *Computer Graphics Forum*, Band 36, Seiten 5–35. Wiley Online Library, 2017.
- [16] KRAUSE, MATHIAS J. und JONAS LATT: *OpenLB User Guide*. [https://www.openlb.net/wp-content/uploads/2019/05/olb\\_ug-1.3r0.pdf](https://www.openlb.net/wp-content/uploads/2019/05/olb_ug-1.3r0.pdf), 2019. [Online; accessed 2-June-2019].
- [17] KRAUSE, MATHIAS J., A. MINK, R. TRUNK, F. KLEMENS, M.-L. MAIER, M. MOHRHARD, A. CLARO BARETTO, M. HAUSSMANN, M. GAEDTKE und J. ROSS-JONES: *OpenLB Release 1.2: Open Source Lattice Boltzmann Code*. <https://www.openlb.net/download>, 2018.

- [18] KRÜGER, TIMM, HALIM KUSUMAATMAJA, ALEXANDR KUZMIN, OREST SHARDT, GONCALO SILVA und ERLEND MAGNUS VIGGEN: *The lattice Boltzmann method*. Springer International Publishing, 10:978–3, 2017.
- [19] LEISTIKOW, SIMON, KARIM HUESMANN, ALEXEY FOFONOV und LARS LINSSEN: *Aggregated Ensemble Views for Deep Water Asteroid Impacts Simulations*. IEEE computer graphics and applications, 2019.
- [20] LI, HONGYU und I-FAN SHEN: *Similarity measure for vector field learning*. In: *International Symposium on Neural Networks*, Seiten 436–441. Springer, 2006.
- [21] MAZZEO, MARCO D und PETER V COVENEY: *HemeLB: A high performance parallel lattice-Boltzmann code for large scale fluid flow in complex geometries*. Computer Physics Communications, 178(12):894–914, 2008.
- [22] MEYER-SPRADOW, JENNIS, TIMO ROPINSKI, JÖRG MENSMANN und KLAUS HINRICH: *Voreen: A rapid-prototyping environment for ray-casting-based volume visualizations*. IEEE Computer Graphics and Applications, 29(6):6–13, 2009.
- [23] MIRZAEI, HANIEH, THOMAS HENN, MATHIAS J KRAUSE, LEONID GOUBERGRITS, CHRISTIAN SCHUMANN, MATHIAS NEUGEBAUER, TITUS KUEHNE, TOBIAS PREUSSER und ANJA HENNEMUTH: *MRI-based computational hemodynamics in patients with aortic coarctation using the lattice Boltzmann methods: Clinical validation study*. Journal of Magnetic Resonance Imaging, 45(1):139–146, 2017.
- [24] MOHAMAD, ABDULMAJEED A: *Lattice Boltzmann method: fundamentals and engineering applications with computer codes*. Springer Science & Business Media, 2011.
- [25] ROLOFF, CHRISTOPH, DANIEL STUCHT, OLIVER BEUNG und PHILIPP BERG: *Comparison of intracranial aneurysm flow quantification techniques: standard PIV vs stereoscopic PIV vs tomographic PIV vs phase-contrast MRI vs CFD*. Journal of neurointerventional surgery, 11(3):275–282, 2019.
- [26] SMAGORINSKY, JOSEPH: *General circulation experiments with the primitive equations: I. The basic experiment*. Monthly weather review, 91(3):99–164, 1963.
- [27] TU, JIYUAN, GUAN HENG YEOH und CHAOQUN LIU: *Computational fluid dynamics: a practical approach*. Butterworth-Heinemann, 2018.
- [28] WICKELMAIER, FLORIAN: *An introduction to MDS*. Sound Quality Research Unit, Aalborg University, Denmark, 46(5):1–26, 2003.

## **Eidesstattliche Erklärung**

Hiermit versichere ich, dass die vorstehende Arbeit selbstständig verfasst worden ist, dass keine anderen Quellen und Hilfsmittel als die angegebenen benutzt worden sind und dass die Stellen der Arbeit, die anderen Werken – auch elektronischen Medien – dem Wortlaut oder Sinn nach entnommen wurden, auf jeden Fall unter Angabe der Quelle als Entlehnung kenntlich gemacht worden sind.

---

(Datum, Unterschrift)

Ich erkläre mich mit einem Abgleich der Arbeit mit anderen Texten zwecks Auffindung von Übereinstimmungen sowie mit einer zu diesem Zweck vorzunehmenden Speicherung der Arbeit in eine Datenbank einverstanden.

---

(Datum, Unterschrift)