

---

**Ein interdisziplinärer Ansatz zur  
dreidimensionalen numerischen Simulation  
von  
Strömung, Stofftransport und Wachstum  
in Biofilmsystemen auf der Mikroskala**

---

Miriam Mehl



Institut für Informatik  
der Technischen Universität München  
Lehrstuhl für numerische Programmierung  
und Ingenieur Anwendungen in der Informatik

**Ein interdisziplinärer Ansatz zur dreidimensionalen  
numerischen Simulation von Strömung, Stofftransport  
und Wachstum in Biofilmsystemen auf der Mikroskala**

Miriam Mehl

Vollständiger Abdruck der von der Fakultät für Informatik der Technischen Universität München zur Erlangung des akademischen Grades eines

Doktors der Naturwissenschaften (Dr. rer. nat.)

genehmigten Dissertation.

Vorsitzender: Univ.-Prof. Dr. Heinz-Gerhard Hegering

Prüfer der Dissertation: 1. Univ.-Prof. Dr. Christoph Zenger  
2. Univ.-Prof. Dr. Hans-Joachim Bungartz,  
Universität Augsburg

Die Dissertation wurde am 20.06.2001 bei der Technischen Universität München eingereicht und durch die Fakultät für Informatik am 07.08.2001 angenommen.



# Inhaltsverzeichnis

<b>Einleitung</b>	<b>1</b>
<b>1 Experimenteller Hintergrund</b>	<b>5</b>
1.1 Versuchsaufbau . . . . .	7
1.2 Confokale Laser-Scanning-Mikroskopie (CLSM) . . . . .	8
1.3 Fluoreszenzmarkierung des Biofilms im Fließkanal . . . . .	12
1.4 Automatisierte Bildakquisition . . . . .	13
1.5 Bildverarbeitung und Bildanalyse . . . . .	14
1.6 Auswahl eines repräsentativen Biofilmvolumens . . . . .	17
1.7 Partikelverfolgung . . . . .	20
1.8 Bestimmung der Diffusionskoeffizienten . . . . .	22
<b>2 Mathematische Modelle</b>	<b>23</b>
2.1 Eindimensionale Modelle . . . . .	26
2.1.1 Massenerhaltung . . . . .	27
2.1.2 Randbedingungen . . . . .	27
2.1.3 Basismodell für Biofilmsysteme . . . . .	28
2.1.4 Verwandte Modelle . . . . .	29
2.1.5 Simulation realer Biofilme . . . . .	31
2.2 Dreidimensionale Modelle . . . . .	32
2.2.1 Makroskopisches Modell . . . . .	33
2.2.2 Mikroskopisches Modell . . . . .	37

<b>3</b>	<b>Numerische Simulationswerkzeuge 1 – Nast++</b>	<b>49</b>
3.1	Leistungsumfang . . . . .	49
3.2	Diskretisierung . . . . .	51
3.2.1	Diskretisierung der Geometrie . . . . .	52
3.2.2	Diskretisierung der Funktionen . . . . .	52
3.2.3	Diskretisierung der Zeitableitung . . . . .	52
3.2.4	Diskretisierung der räumlichen Differential-Operatoren . . . . .	55
3.2.5	Diskretisierung der Randbedingungen . . . . .	60
3.2.6	Stabilitätsbedingungen . . . . .	65
3.3	Gleichungslöser . . . . .	66
3.3.1	Jacobi-Verfahren . . . . .	67
3.3.2	Gauß-Seidel- und SOR-Verfahren . . . . .	69
3.3.3	Mehrgitterverfahren . . . . .	70
3.3.4	Beispiel Nischen-Strömung . . . . .	77
3.3.5	Beispiel Biofilm . . . . .	84
3.4	Modularität . . . . .	89
3.4.1	Simulator . . . . .	90
3.4.2	Navier-Stokes-Löser . . . . .	91
3.4.3	Poissonlöser . . . . .	92
3.4.4	Transportlöser . . . . .	92
3.4.5	Differenzen-Quotienten . . . . .	93
3.4.6	Diskrete Geometrie . . . . .	94
3.4.7	Versetztes Gitter . . . . .	95
3.5	Äußere Schnittstellen . . . . .	96
3.5.1	Parametereingabe . . . . .	96
3.5.2	Geometrie- und Werteschnittstelle . . . . .	97
3.5.3	Randbedingungen . . . . .	97
3.6	Parallelisierung . . . . .	98
3.6.1	Implementierung . . . . .	98

3.6.2	Beispiel . . . . .	103
<b>4</b>	<b>Numerische Simulationswerkzeuge 2 – CAsim</b>	<b>105</b>
4.1	Leistungsumfang . . . . .	105
4.2	Implementierung . . . . .	105
4.3	Äußere Schnittstellen . . . . .	109
4.3.1	Konfigurationsdatei . . . . .	109
4.3.2	Initialisierungs- und Ausgabedatei . . . . .	109
4.4	Konfiguration zur Simulation des Biofilmwachstums . . . . .	110
<b>5</b>	<b>Einbettung der numerischen Simulation</b>	<b>113</b>
5.1	Kopplung von <i>Nast++</i> und <i>CAsim</i> . . . . .	113
5.1.1	Zeitdiskretisierung . . . . .	113
5.1.2	Verbindung der Schnittstellen . . . . .	115
5.2	Schnittstelle Experiment – Simulation . . . . .	116
5.3	Schnittstelle Simulation – Visualisierung . . . . .	117
<b>6</b>	<b>Visualisierung</b>	<b>119</b>
6.1	Geometrievisualisierung . . . . .	119
6.2	Visualisierung der Simulationsergebnisse . . . . .	121
6.2.1	Strömungsvisualisierung . . . . .	121
6.2.2	Visualisierung der Konzentration gelöster Stoffe . . . . .	122
6.2.3	Visualisierung des Wachstums . . . . .	122
6.3	Animierte und 3D-Visualisierung . . . . .	123
6.3.1	Animationen . . . . .	123
6.3.2	3D-Visualisierung . . . . .	125
	<b>Zusammenfassung</b>	<b>133</b>
	<b>Anhang A – Ergänzungen zu den Schnittstellen von <i>Nast++</i> und <i>CAsim</i></b>	<b>135</b>
A.1	Beschreibung der Parameterdatei von <i>Nast++</i> . . . . .	135

A.2	Beschreibung der <i>CAsim</i> -Konfigurationsdatei . . . . .	139
<b>Anhang B – Ausblick</b>		<b>143</b>
B.1	Modellerweiterungen . . . . .	143
B.2	Kopplung der Mikro- und Makroskala . . . . .	145
B.2.1	Strömung und Nährstofftransport . . . . .	145
B.2.2	Biofilmwachstum . . . . .	147
B.3	Verbesserungen der numerischen Algorithmen . . . . .	147
B.3.1	Adaptivität . . . . .	147
B.3.2	Weiterentwicklung des Mehrgitterverfahrens . . . . .	148
<b>Literaturverzeichnis</b>		<b>151</b>

# Einleitung

Während sich in vielen Bereichen der Naturwissenschaften, wie zum Beispiel der Physik und der Chemie, die Informatik im Allgemeinen und die Numerische Simulation im Speziellen bereits fest etabliert haben, ist der Forschungsbereich der Bioinformatik noch relativ jung. Typische Anwendungsgebiete sind die Bereitstellung und Analyse von Gen- und Proteindatenbanken, die Untersuchung von Gen-/DNA-Sequenzen, die Herstellung synthetischer Gene, das Arzneimitteldesign und die Molekularbiologie. Die Bioinformatik verwendet neben ausgeklügelten Datenbanksystemen vor allem Methoden der kombinatorischen Mustererkennung und der Fuzzy-Logik sowie Neuronale Netze. Im Gegensatz zu diesen Methoden werden die Werkzeuge der Numerischen Simulation noch relativ selten zur Untersuchung biologischer Prozesse genutzt. Dies mag zum einen in der hohen Komplexität biologischer Vorgänge und der damit einhergehenden Schwierigkeiten bei der mathematischen Modellierung der beobachteten Phänomene in Form von Gleichungssystemen begründet sein. Zum anderen erschwert aber auch der große fachliche Abstand der Biologie auf der einen von der Informatik/Mathematik auf der anderen Seite die interdisziplinäre Kommunikation und Zusammenarbeit.

Auf dem Gebiet der (industriellen und kommunalen) Abwasserreinigung beziehungsweise -aufbereitung jedoch, einem traditionellen Anwendungsgebiet der Ingenieurwissenschaften, wurden in den letzten Jahren zahlreiche Simulationsrechnungen durchgeführt – sowohl für chemisch-physikalische als auch für biologische Verfahren. Während für die bereits seit etwa zwanzig Jahren etablierten chemisch-physikalischen Verfahren im Wesentlichen alle ablaufenden Prozesse bekannt und entsprechend ihrer Relevanz in die mathematischen Modelle integriert sind, herrscht bei den biologischen Verfahren noch weitgehende Unklarheit über die zu verwendenden Modelle ([32]). Neben der oft komplexen und teilweise unklaren Zusammensetzung der Bakterienpopulation sind die genaue Wirkungsweise und das Zusammenspiel der einzelnen Anlagenkomponenten (Mikroben, Aufwuchskörper, Röhrelemente, etc.) nur unzureichend bekannt. Ferner ist auch die Frage nach dem notwendigen Detaillierungsgrad in Bezug auf die Modellierung einzelner Prozesse, die explizite Auflösung der räumlichen Dimensionen und die Raum- und Zeitskalen noch offen. Die Palette möglicher Modelle reicht von eindimensionalen Modellen für ein ausgewähltes Biofilmsystem ([18]), die den Transport beziehungsweise Abbau eines Stoffs oft über lange Zeit hinweg (Tage, Monate, Jahre) wiedergeben, bis zu dreidimensionalen Modellen für mikroskopisch kleine Bereiche (Durchmesser ca. 100 Mikrometer) beliebiger Biofilme auf der Zeitskala von wenigen Sekunden, die den Einfluss einzelner Mikroben detailliert beschreiben ([62, 11]).

Im Interesse der Weiterentwicklung und/oder Optimierung biologischer Abwasser-Aufberei-

tungsanlagen, deren Bedeutung als umweltgerechte Technologie ständig zunimmt, haben beide Ansätze ihre Berechtigung: Grobe Modelle bestehen aus einer überschaubaren Zahl von Eingabeparametern und vergleichsweise einfach zu lösenden Modellgleichungen. So lassen sich in kürzester Zeit Aussagen über die Wirkung einer Parameteränderung treffen und damit die optimalen Steuerungsparameter abschätzen. Wissenschaftliche Erkenntnisse über einzelne Prozesse und ihre Wechselwirkungen lassen sich dagegen nur mit sehr detaillierten (und damit für die Simulation rechenaufwändigen) Modellen gewinnen. Ferner haben solche Detailmodelle im Allgemeinen den Vorteil, unabhängig von der betrachteten Anlage gültig zu sein, während bei vereinfachenden Modellen üblicherweise für jedes betrachtete System eine langwierige Kalibrierung nötig ist.

Diese Arbeit befasst sich mit einem speziellen Typ von biologischen Abwasser-Aufbereitungsanlagen, den Biofilmreaktoren (zum Beispiel Tropfkörper als spezielle Form der Festbettreaktoren). Biofilme sind Ansammlungen von Mikroorganismen und polymeren, gelartigen Substanzen und finden sich auf fast allen natürlichen Grenzflächen zwischen Gasen und Flüssigkeiten (zum Beispiel Gewässeroberflächen), zwischen Festkörpern und Flüssigkeiten (beispielsweise Flussbetten, Schiffsrümpfe) und zwischen zwei Flüssigkeiten (zum Beispiel Öltropfen in Wasser). Abhängig von den Wachstumsbedingungen können Biofilme in sehr unterschiedlichen Formen auftreten – von fast homogenen, glatten Filmoberflächen bis zu komplizierten, stark verzweigten, mit Poren und Kanälen durchsetzten dreidimensionalen Strukturen. Neben wünschenswerten Wirkungen, wie zum Beispiel der natürlichen Reinigungswirkung in Gewässern oder Kläranlagen, können Biofilme in vielen Fällen auch unerwünschte Effekte haben, beispielsweise steigende Reibung durch Biofilme auf Schiffsrümpfen ([17]) oder erhöhte Kariesgefahr durch Zahnbelag.

Ein Festbettbiofilmreaktor besteht aus einer Schüttung von Aufwuchskörpern (zum Beispiel Kugeln), auf denen sich Biofilme bilden. Die Schüttung wird vom Abwasser mit relativ geringer Geschwindigkeit (weniger als 1 cm/sec) durchströmt. Die Klärwirkung beruht sowohl auf der metabolischen Aktivität der Mikroben als auch auf chemischer und physikalischer Absorption der Schadstoffe. Die Effizienz des Reaktors hängt wesentlich von der Form und Zusammensetzung des Biofilmbewuchses ab, welche wiederum stark von den herrschenden Strömungsverhältnissen und der Verteilung der Schad- beziehungsweise Nährstoffe im Reaktor beeinflusst werden.

Ziel dieser Arbeit war die Bereitstellung von flexiblen und leicht erweiterbaren Simulationswerkzeugen für die genaue Untersuchung einzelner relevanter Prozesse in Biofilmsystemen, *nicht* die Optimierung des gesamten Reaktors mit Hilfe eines speziell auf den betreffenden Reaktor zugeschnittenen Näherungsmodells. Um den dabei nötigen Detaillierungsgrad zu erreichen, werden sehr kleine Gebiete von wenigen hundert Mikrometern Durchmesser auf der Oberfläche eines Aufwuchskörpers betrachtet (Mikroskala), auf denen sich die Zusammensetzung des Biofilmsystems noch mit ausreichender Genauigkeit auflösen lässt. Eine Simulation des gesamten Reaktors ist mit dieser hohen Genauigkeit aufgrund der großen Bandbreite der auftretenden räumlichen Skalen – vom Durchmesser einer Mikrobe von wenigen Mikrometern bis zu den Abmessungen eines Reaktors von größenordnungsmäßig zehn Metern – und der begrenzten Zeit- und Speicherressourcen nicht möglich. In Anhang B.2 ist eine Möglichkeit beschrieben, die Ergebnisse der Rechnungen auf der Mikroskala dennoch in die Simulation des gesamten Reaktors zu integrieren. Im Einzelnen werden auf der Mikroskala die Strömung des Fluids, die sich in den letzten Jahren entgegen früherer Annahmen als

wichtiger Einflussparameter für das Wachstum und die Funktion des Biofilms herausgestellt hat ([3]), der Transport und Abbau von Nähr- beziehungsweise Schadstoffen im Fluid und im Mikrobengebiet und davon abhängig das Wachstum des Biofilms betrachtet. Alle Prozesse werden räumlich dreidimensional aufgelöst.

Um der Unvollständigkeit und ständigen Weiterentwicklung der Biofilmmodelle gerecht zu werden, wurden zur Simulation statt eines einzigen starren Programms flexible und effiziente Simulationswerkzeuge entwickelt und implementiert. Eine hohe Modularität und fest definierte Schnittstellen zu den experimentellen Eingabe- und Vergleichsdaten auf der einen und der Visualisierungssoftware auf der anderen Seite garantieren eine universelle Einsetzbarkeit, leichte Anpassbarkeit und Erweiterbarkeit der Simulationsprogramme.

Diese Arbeit wurde in enger Zusammenarbeit mit dem Lehrstuhl für Wassergüte und Abfallwirtschaft der TU München durchgeführt. Dort entstand parallel zur vorliegenden Arbeit eine weitere Dissertation, die die experimentellen Grundlagen zur Simulation dreidimensionaler Biofilmsysteme auf der Mikroskala bereitstellte ([45]). Die wichtigsten Methoden und Ergebnisse dieser experimentellen Arbeit sind in Kapitel 1 zusammengefasst. Verschiedene mathematische Modelle für Biofilmsysteme mit unterschiedlicher Zielsetzung, unterschiedlichen räumlichen und zeitlichen Skalen sowie unterschiedlicher Dimension und Detailliertheit werden in Kapitel 2 vorgestellt. Die Kapitel 3 und 4 behandeln den Kern dieser Arbeit, die numerischen Simulationswerkzeuge für das ausgewählte mikroskopische dreidimensionale Biofilmmodell. Die Einbettung dieser Werkzeuge in den Gesamtzusammenhang wird in Kapitel 5 dargestellt. Wichtig für die Verwert- und Interpretierbarkeit der experimentellen und numerischen Ergebnisse ist die anschließende Visualisierung (Kapitel 6). Im Anhang A werden einige Details zu den Schnittstellen der Simulationswerkzeuge nachgereicht, Anhang B beschreibt denkbare Erweiterungen des verwendeten Modells, Wege zur Kopplung der unterschiedlichen räumlichen Skalen sowie weitere Verbesserungsmöglichkeiten der numerischen Verfahren.

An dieser Stelle möchte ich mich bei allen bedanken, die zum Gelingen dieser Arbeit beigetragen haben. An erster Stelle steht hier Prof. C. Zenger, der als Themensteller durch zahlreiche kreative Anregungen die Entwicklung der numerischen Verfahren unterstützt hat. Prof. H.-J. Bungartz hat die Arbeit während ihrer gesamten Entstehung begleitet und an allen auftretenden Problemen, sowohl was inhaltliche Fragen als auch interdisziplinäre Verständigungsschwierigkeiten betraf, intensiv teilgenommen. PD Dr. Stefan Wuertz und Dipl.-Ing. Martin Kühn gilt mein Dank für die fruchtbare Zusammenarbeit im Rahmen des Sonderforschungsbereichs 411 (Grundlagen der aeroben biologischen Abwasserreinigung), ihre ständige Bereitschaft, alle „biologischen“ Fragen zu beantworten, und die Bereitstellung der experimentellen Daten. An dieser Stelle möchte ich mich ferner bei der Deutschen Forschungsgemeinschaft bedanken, die über den Sonderforschungsbereich 411 wesentlich zur Finanzierung dieser Arbeit beigetragen hat. Für das Korrekturlesen von Teilen der Arbeit bedanke ich mich bei Prof. H.-J. Bungartz, Dr. Michael Bader, Dipl.-Math. Stefan Achatz, Dipl.-Math. Frank Günther und meiner Schwester Judith.



# Kapitel 1

## Experimenteller Hintergrund

Die in dieser Arbeit durchgeführten numerischen Simulationen stützen sich vor allem bei der Erzeugung einer realitätsnahen Biofilmgeometrie stark auf experimentelle Ergebnisse<sup>1</sup>. Die Experimente müssen daher entsprechend der räumlichen Auflösung der Simulation ebenfalls auf der Mikroskala stattfinden und dürfen nicht destruktiv in Bezug auf die Biofilmstruktur sein, sondern müssen *in situ*, also ohne Entfernen des Biofilms von der Aufwuchsfläche, durchgeführt werden.

Biofilmreaktoren von Abwasser-Aufbereitungsanlagen (vgl. Abbildung 1.1) bestehen aus Millionen von kleinen (Durchmesser ca. 1 bis 3 cm) Aufwuchskörpern, die – abhängig von der Anfangspopulation und der Inhaltsstoffe des zugeführten Abwassers – von zahlreichen verschiedenen Mikroorganismen bewachsen sind. Die genaue Zusammensetzung der mikrobiellen Population einer Anlage kann sehr komplex sein und starke Schwankungen im zeitlichen und räumlichen Verlauf aufweisen, die beispielsweise durch jahreszeitliche Veränderungen der Wachstumsbedingungen oder spezielle Strömungsverhältnisse im Ein- und Ausflussbereich des Reaktors verursacht werden. Grundsätzlich ist nur ein kleiner Teil der beteiligten Organismenarten bekannt. Zur vollständigen Detektion und Lokalisierung aller Mikroben eines Biofilms – beispielsweise durch fluoreszierende In-situ-Hybridisierung (FISH) und confokale Laser-Scanning-Mikroskopie (CLSM) ([59, 84]) – ist jedoch eine genaue Kenntnis der qualitativen Zusammensetzung der Population erforderlich, da die hierbei verwendeten Sonden nur spezifisch eine bestimmte Bakteriengattung oder -spezies markieren. Arten, deren Vorkommen im Biofilm a priori nicht vermutet wird, können daher bisher experimentell nur durch richtiges Raten detektiert werden. Zusätzlich zu den zahlreichen Mikrobenarten bestehen Biofilme aus extrazellulären polymeren Substanzen, einer gelartigen Masse, die vermutlich aus Stoffwechselprodukten der Mikroben entsteht und bis zu neunzig Prozent der Trockenmasse eines Biofilms ausmachen kann, und aus Protozoen und Metazoen, mehrzelligen Organismen, die in einer Räuber-Beute-Beziehung zu den Mikroben stehen und somit die Funktion des Biofilms stören können. Insbesondere die extrazellulären polymeren Substanzen sind aufgrund ihres komplexen Aufbaus aus verschiedensten Polymeren und Proteinen durch Experimente nicht vollständig nachweisbar.

Viele Experimente, in denen der komplette Aufbau eines Biofilms von Interesse ist, werden

---

<sup>1</sup>Die experimentellen Arbeiten wurden im Rahmen einer eigenen Dissertation ([45]) durchgeführt



**Abbildung 1.1:** Tropfkörper-Berieselungsanlage der Kläranlage Oberwesel als Beispiel für den Praxisbetrieb eines aeroben Festbettverfahrens zum Ammoniumabbau (aus [98])

aufgrund der oben genannten Schwierigkeiten noch nicht an Biofilmen realer Abwasser-Aufbereitungsanlagen durchgeführt, sondern an Laborreaktoren (vgl. Abbildung 1.2), die unter definierten Bedingungen, bei aseptischer Arbeitsweise und unter Verwendung von Reinkulturen mit bekannter Zusammensetzung der Bakterienpopulation betrieben werden können.

Da in dieser Arbeit Biofilme auf der Mikroskala und somit nur ein sehr kleiner Ausschnitt aus der Oberfläche eines Aufwuchskörpers (siehe Abbildung 1.3) und nicht das Zusammenspiel der Komponenten des gesamten Reaktors (Aufwuchskörper, Reaktorwände, Ein- und Ausflüsse) betrachtet werden, können wir in der Vereinfachung sogar noch einen Schritt weiter gehen und Mono- oder 2-Kultur-Biofilme in einem gläsernen Fließkanal (siehe 1.1) untersuchen. Auf diese Weise kann der Biofilm ohne weitere Bearbeitungsschritte (*in situ*) – und damit ohne unnötige Zerstörung seiner Struktur – direkt mittels confokaler Laser-Scanning-Mikroskopie (CLSM, siehe 1.2) aufgenommen werden, nachdem seine Bestandteile durch geeignete Verfahren zur Fluoreszenz angeregt wurden (siehe 1.3). Makrogesteuerte automatisierte Bildakquisition (siehe 1.4) und -verarbeitung (siehe 1.5) ermöglichen eine schnelle Generierung der benötigten Bilder und Parameter eines repräsentativen Ausschnitts des Biofilms (siehe 1.6). Neben der reinen Geometrieerfassung werden lokale Strömungsgeschwindigkeiten im Fluid per Partikelverfolgung bestimmt (siehe 1.7) und anhand von Messungen der Stoffkonzentrationen und Konzentrationsgradienten die zur numerischen Simulation des Transportes gelöster Stoffe benötigten Diffusionskoeffizienten geschätzt (siehe 1.8).

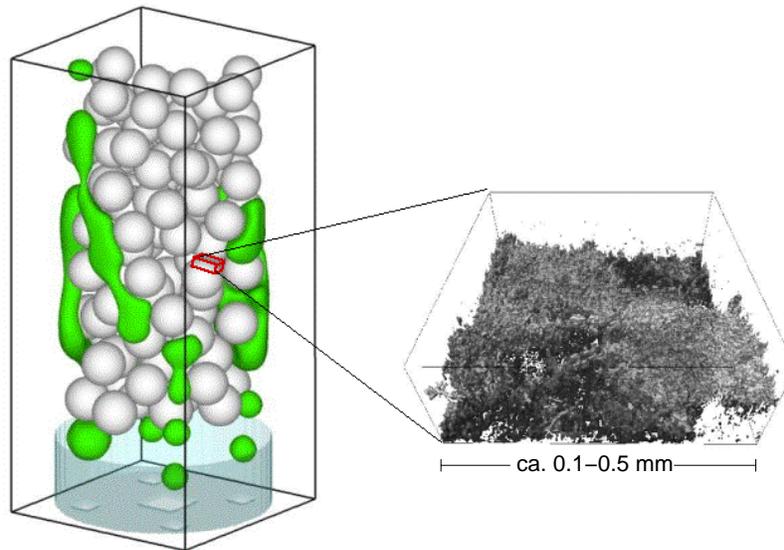


**Abbildung 1.2:** Anaerober Festbettreaktor im Labormaßstab (Fraunhofer Institut für Grenzflächen- und Bioverfahrenstechnik, aus [99])

## 1.1 Versuchsaufbau

Um einen Biofilm auf der Mikroskala dreidimensional zu erfassen, werden „einfache“ Biofilme (Monokulturen und 2-Kultur-Biofilme, zum Beispiel *Sphingomonas* und *E. coli*) in einem gläsernen Fließkanal kultiviert, der unter dem confokalen Laser-Scanning-Mikroskop fixiert ist. Der Fließkanal wird aus einem belüfteten Rührgefäß mit einer künstlichen Nährlösung versorgt. Für konstante und kontinuierliche Flussbedingungen innerhalb der Fließzelle sorgt eine geeignet justierte und über Ventile an den hydrodynamischen Kreislauf angeschlossene Druckpumpe. Um die Wachstumsbedingungen für den Biofilm zu überwachen, werden durch Probenentnahme regelmäßig der pH-Wert und der Sauerstoffgehalt des Fluids in der Fließzelle bestimmt. Abbildung 1.4 zeigt den Aufbau der Versuchsanlage, Abbildung 1.5 die gläserne Fließzelle. Die Anlage wird im „offenen hydraulischen Kreis“-Modus betrieben, das heißt, die nach dem Durchfließen des Kanals mit Bakterien belastete Nährlösung wird direkt in einen Auffangbehälter abgeleitet und nicht rückgeführt. Die Startbesiedelung des Fließkanals erfolgt durch Animpfung mit einer Bakteriensuspension aus einer wenige Tage oder Stunden alten Monokultur ([46]).

Um die gewünschten Daten zu erhalten, müssen die Biofilme im Fließkanal über mehrere Tage oder Wochen kultiviert werden. Die Versuchsreihen sind somit sehr zeitaufwändig. Daher wird statt eines einfachen Kanals häufig eine Mehrkanalfließzelle verwendet, die die parallele Untersuchung von vier oder mehr Biofilmen unter gleichen oder auch unterschiedlichen Wachstumsbedingungen ermöglicht. Der Einfluss äußerer Bedingungen wie Nährstoffangebot, Sauerstoffzufuhr, Strömungsgeschwindigkeit und Temperatur auf das Wachstum und die Funktion eines Biofilms lässt sich so in wesentlich kürzerer Zeit experimentell ermitteln. Ferner sind mit Hilfe der Mehrkanalfließzelle Zeitreihenuntersuchungen auch für solche experimentelle Methoden möglich, die den Biofilm irreversibel beeinträchtigen oder verändern. Dies ist beispielsweise bei der Detektion der extrazellulären polymeren Substanzen (EPS) durch Markierung mit Lektinen der Fall (siehe 1.3). Hier wird für jeden Zeitpunkt der Reihe ein neuer Kanal untersucht, der bisher nicht bearbeitet wurde ([47]). So lässt sich zwar nicht



**Abbildung 1.3:** Experimentell untersuchter Ausschnitt (rechts) eines Biofilmreaktors (links)

die Entwicklung eines bestimmten Biofilms verfolgen, aber zumindest eine Aussage über charakteristische Merkmale der verschiedenen Stadien des Wachstums treffen.

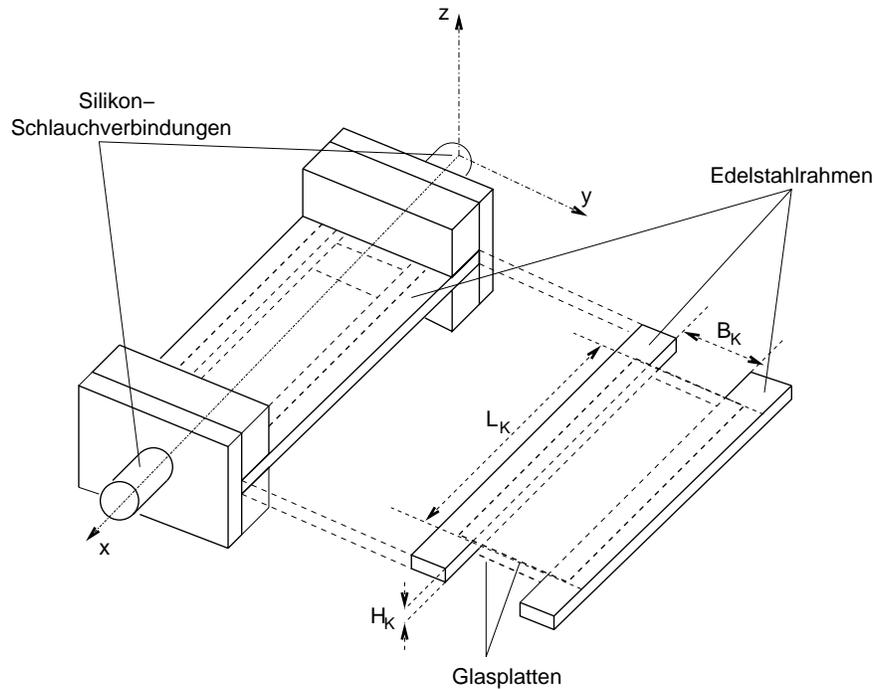
## 1.2 Confokale Laser-Scanning-Mikroskopie (CLSM)

Die experimentelle Untersuchung der dreidimensionalen Struktur von Biofilmen auf der Mikroskala stellt spezielle Anforderungen an das zu verwendende Mikroskop. Zum einen muss eine möglichst hohe Auflösung (unter einem Mikrometer) erreicht werden, um die Geometrie bis zur Größe einzelner Mikroben detailliert zu erfassen. Zum anderen muss die Dreidimensionalität des Biofilms berücksichtigt werden, das heißt, es sollte möglich sein, nicht nur die Oberfläche des Präparats sichtbar zu machen, sondern Aufnahmen in fest definierten Tiefen des Biofilms zu machen. Nur so können wichtige Merkmale des Biofilms wie Poren und Kanäle auch in tieferen Schichten sichtbar gemacht werden.

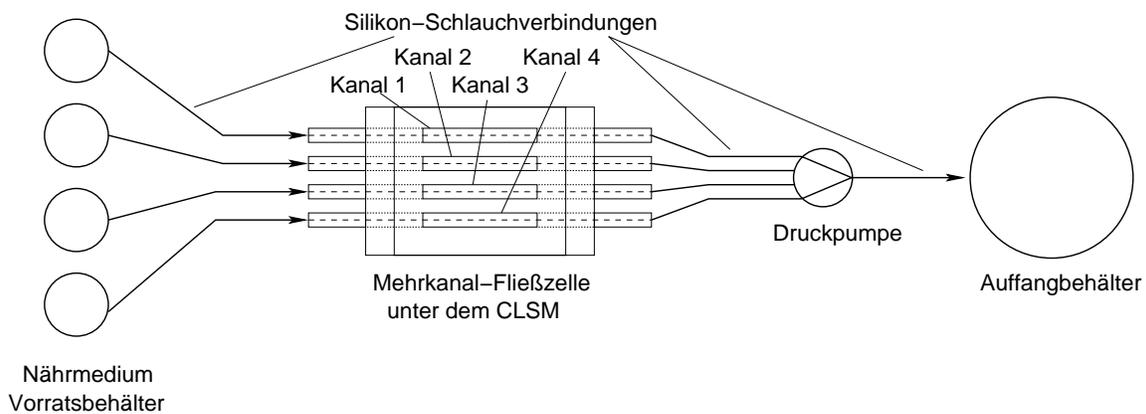
Confokale Laser-Scanning-Mikroskope erfüllen beide Anforderung optimal. Sie erzielen eine deutlich höhere Auflösung (bis  $0.25 \mu\text{m}$ ) als konventionelle Durchlicht- und insbesondere Epifluoreszenzmikroskope. Ferner bieten sie die Möglichkeit, Bilder von Schnittebenen in definierten Abständen mit einer Tiefenschärfe von ca.  $0.5 \mu\text{m}$  zu erzeugen. Die komplette dreidimensionale Geometrie entsteht durch Zusammenfügen der Schnittbilder unter Berücksichtigung ihres vertikalen Abstandes.

Die genannten Vorteile confokaler Laser-Scanning-Mikroskope beruhen im Wesentlichen auf drei Merkmalen des Mikroskops:





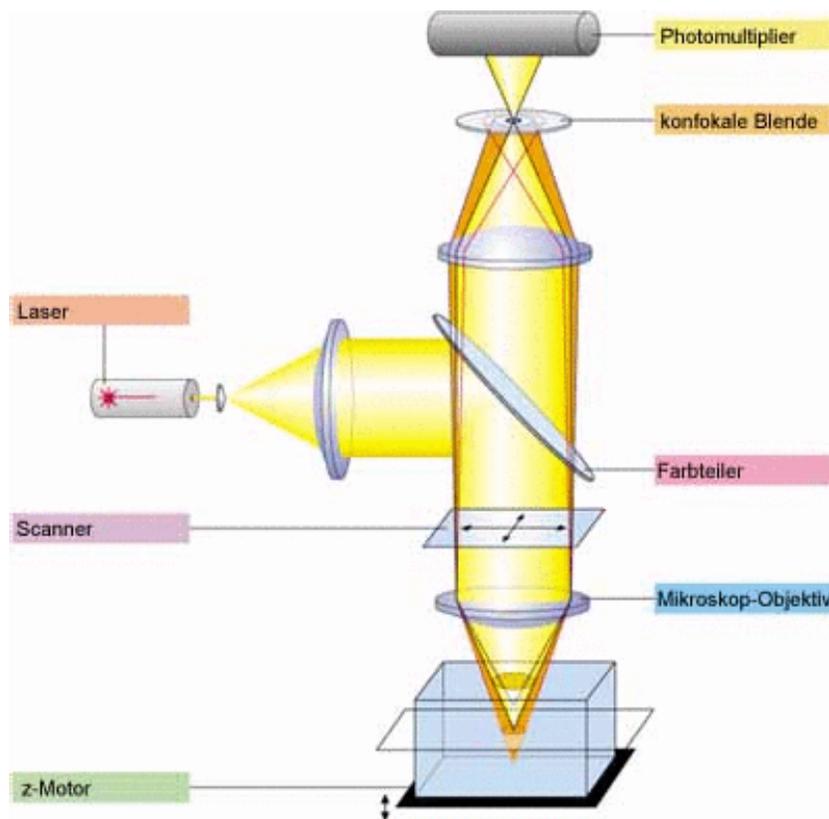
**Abbildung 1.5:** Detaillierte Ansicht einer gläsernen Fließzelle zur Beobachtung der Biofilmentwicklung mit dem confokalen Laser-Scanning-Mikroskop (Kanallänge  $L_K = 46$  mm, Kanalbreite  $B_K = 8$  mm, Kanalhöhe  $H_K = 2.7$  mm, aus [46])



**Abbildung 1.6:** Schematische Darstellung einer Mehrkanalfließzelle mit getrennten Nährstoffreservoirs (aus [47], Kanallänge jeweils 40 mm, Kanalbreite und -höhe 4 mm)

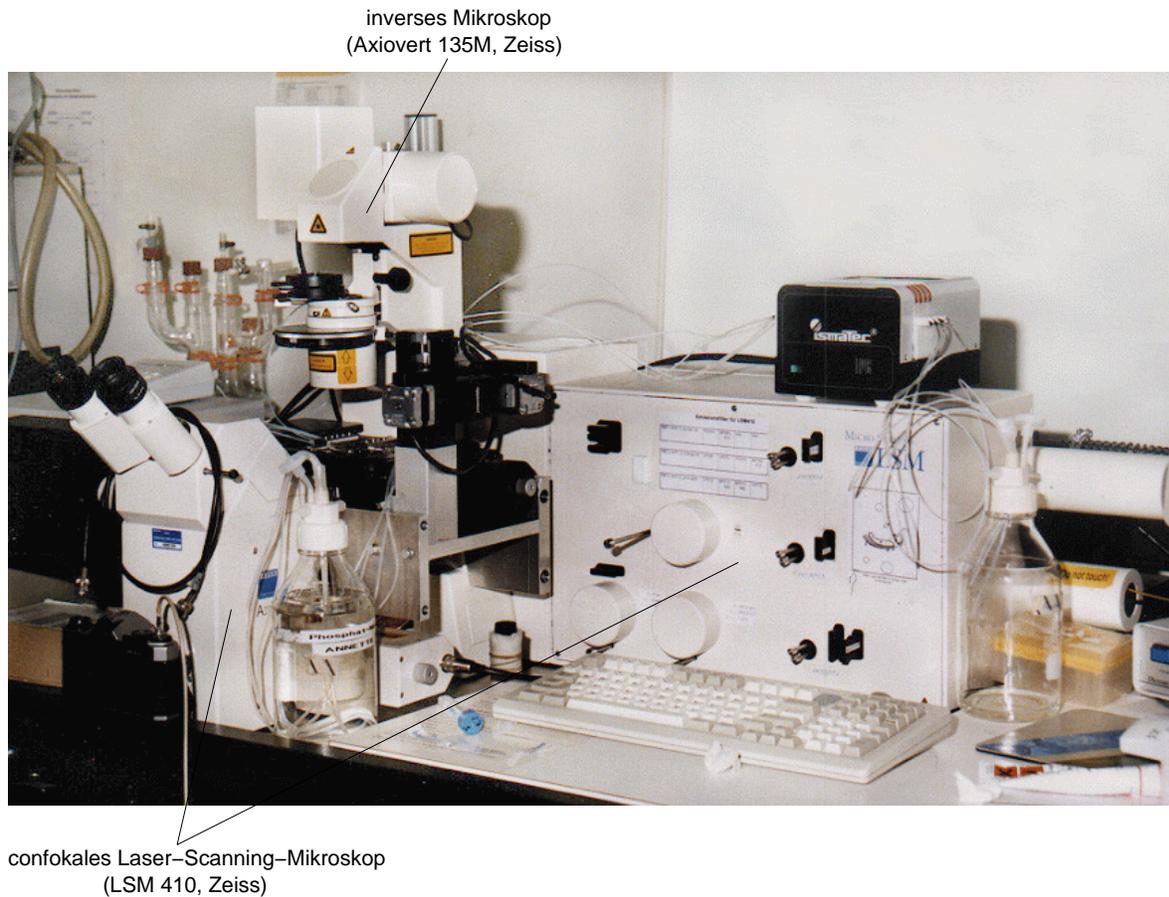
- 1) Laserlicht ist durch sein linienförmiges Spektrum besser zum Herausfiltern einer bestimmten, für das verwendete Fluorochrom benötigten Anregungswellenlänge geeignet als das Licht herkömmlicher Mikroskoplampen. Fluoreszenzmarkierte Bereiche des Präparats lassen sich so wesentlich schärfer und störungsfreier darstellen.
- 2) Streulicht aus entfernteren Bereichen des Objekts wird durch das Einbringen einer Illuminationslochblende reduziert. Der Objektbereich wird so Punkt für Punkt und Zeile für Zeile abgetastet und nicht zu einem Zeitpunkt komplett ausgeleuchtet.
- 3) Zusätzlich filtert eine zweite Lochblende mit variablem Durchmesser in der aufnehmenden Optik, die auf dieselbe Objektebene fokussiert wie die beleuchtende Optik (Prinzip der Konfokalität), Licht aus anderen Ebenen vor Erreichen des Aufnahmegerätes, zum Beispiel eines Photomultipliers oder einer Kamera, heraus (siehe Abbildung 1.7).

Mikroskope mit zwei oder drei Kanälen ermöglichen ferner simultane oder sequentielle Aufnahmen von zwei verschiedenen Fluoreszenzsignalen desselben Präparats ([92]).



**Abbildung 1.7:** Schematischer Aufbau eines konfokalen Laser-Scanning-Mikroskops (aus [101])

Die in dieser Arbeit verwendeten experimentellen Biofilmdaten wurden mit Hilfe eines an ein inverses Mikroskop (Axiovert 135M, Zeiss) gekoppelten konfokalen Laser-Scanning-Mikroskops (LSM 410, Zeiss) gewonnen (vgl. Abbildung 1.8 und [46, 101]).



**Abbildung 1.8:** Confokales Laser-Scanning-Mikroskop (CLSM) zur In-situ-Untersuchung von Biofilmen und Belebtschlammflocken

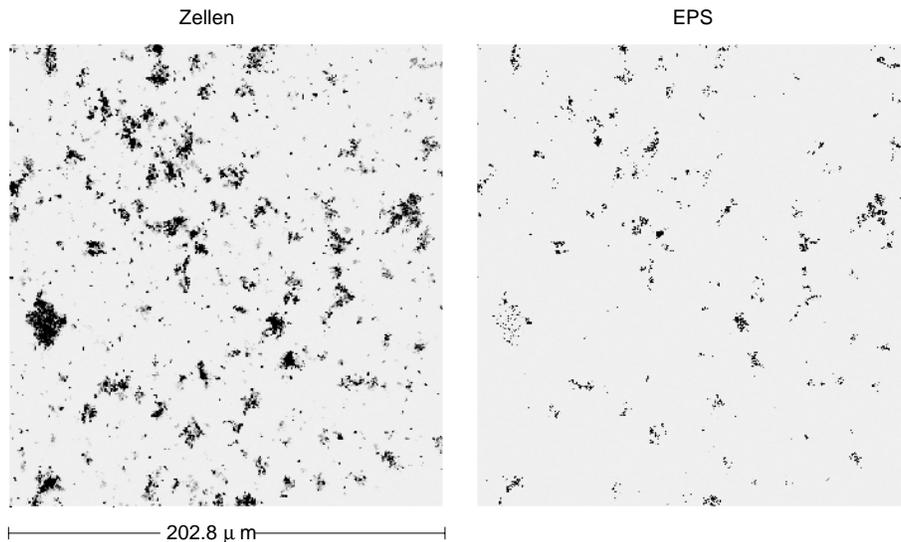
### 1.3 Fluoreszenzmarkierung des Biofilms im Fließkanal

Um einen Biofilm im Fließkanal mittels CLSM beobachtbar zu machen, müssen zunächst seine Bestandteile (im Wesentlichen Mikroben und extrazelluläre polymere Substanzen (EPS)) fluoreszenzmarkiert werden.

Die Mikroben werden dazu durch Einfügen des Gens für das grün fluoreszierende Protein (GFP, siehe [76]) genetisch verändert. Nach Anregung mit Licht der entsprechenden Wellenlänge beginnen die Bakterienzellen zu fluoreszieren und können so mikroskopisch detektiert werden. Für die dieser Arbeit zugrunde liegenden Experimente wurden beispielsweise der *E. coli* Stamm DH5 $\alpha$  ([46]) und eine gram-negative *Sphingomonas* sp. LB126 Monokultur ([47]) untersucht. Die genetische Markierung beeinflusst die weitere Entwicklung des Biofilms nicht. Daher können problemlos Zeitreihenuntersuchungen für einen Biofilm durchgeführt werden.

Die extrazellulären polymeren Substanzen sind aufgrund ihrer vielfältigen Bestandteile (Polysaccharide, Proteine u. ä.) nur sehr schwer vollständig zu markieren. In [47] gelang es, mit-

tels Bindung fluoreszenzmarkierter Lektine (TRITC-markiertes Concavalin A) an die *Sphingomonas* Exopolysaccharide (Sphingane), zumindest die bisher bekannten Bestandteile der EPS einer *Sphingomonas* sp. LB126 Monokultur zu detektieren (siehe Abbildung 1.9).

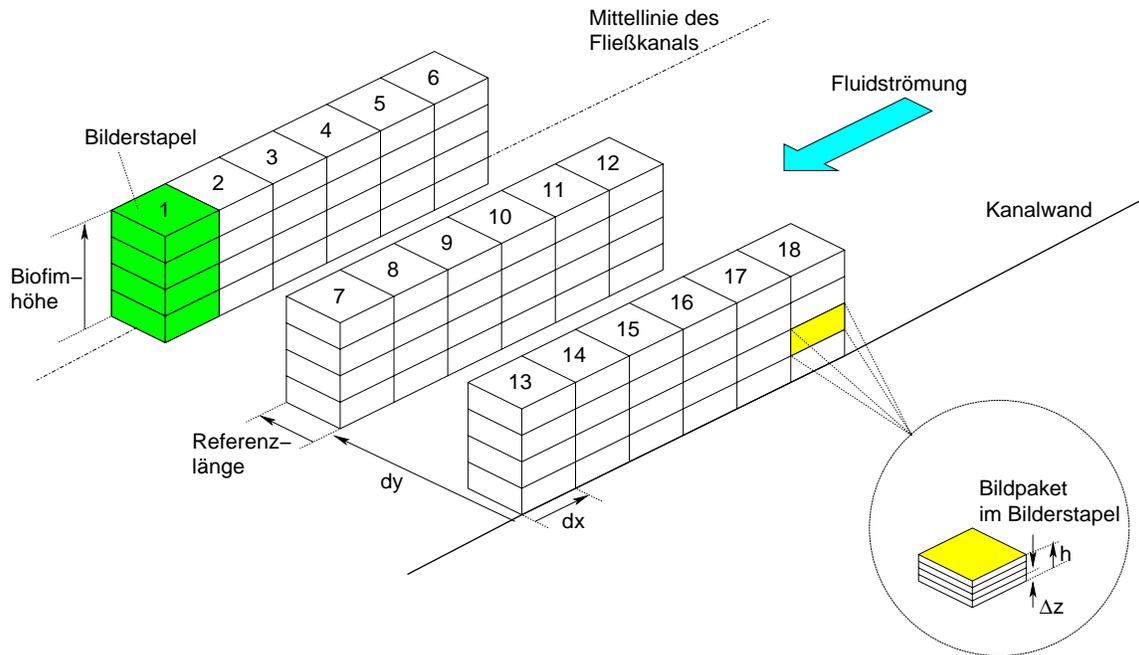


**Abbildung 1.9:** CLSM-Aufnahmen der Mikrobenzellen (links) und der extrazellulären polymeren Substanzen (rechts) einer *Sphingomonas* sp. LB126 Monokultur

## 1.4 Automatisierte Bildakquisition

Um ein zeit- und wegoptimiertes Scannen der einzelnen Bilder zu ermöglichen und einen einfachen Aufbau der Bilderstapel für die anschließende Bildverarbeitung zu gewährleisten, werden den aktuellen Versuchsbedingungen angepasste und auf die Erfassung von Zeitserien-Schnittbildern erweiterte, benutzerdefinierte Makro-Prozeduren zur automatisierten Bildakquisition eingesetzt ([46]). So können nach der manuellen Einstellung der Linsenvergrößerung, des Kontrastes, der Mikroskopverstärkung und der Parameter für die Positionen der gewünschten Schnitte die Bilddaten ohne weitere interaktive Eingriffe des Benutzers generiert werden. Es werden ausschließlich horizontale Schnittbilder erzeugt, um die bei Schnitten in der xz- oder yz-Ebene auftretenden axialen Verschiebungen aufeinanderfolgender Schnitte zu verhindern ([46]). Als Ergebnis der Bildakquisition erhält man eine a priori festzulegende Anzahl von Bilderstapeln, deren Abstände  $dx$  und  $dy$  in der horizontalen Ebene variabel und frei wählbar sind. Abbildung 1.10 zeigt ein Beispiel mit 18 Bilderstapeln in drei unterschiedlichen Positionen relativ zur Kanalwand, die die Untersuchung des Einflusses der Strömungsgeschwindigkeiten (gering in Wandnähe, hoch in der Kanalmitte) auf die Entwicklung des Biofilms ermöglichen. Die einzelnen Bilderstapel setzen sich wiederum aus mehreren Paketen mit jeweils einer festen Anzahl horizontaler Schnitte mit konstantem vertikalen Abstand  $\Delta z$  zusammen. Die Anzahl der Bilderpakete pro Stapel hängt von der Gesamthöhe des betrachteten Biofilms ab.

Alle in dieser Arbeit verwendeten Bilder wurden mit einer  $63 \times /1.2$  NA (c-Apochromat)



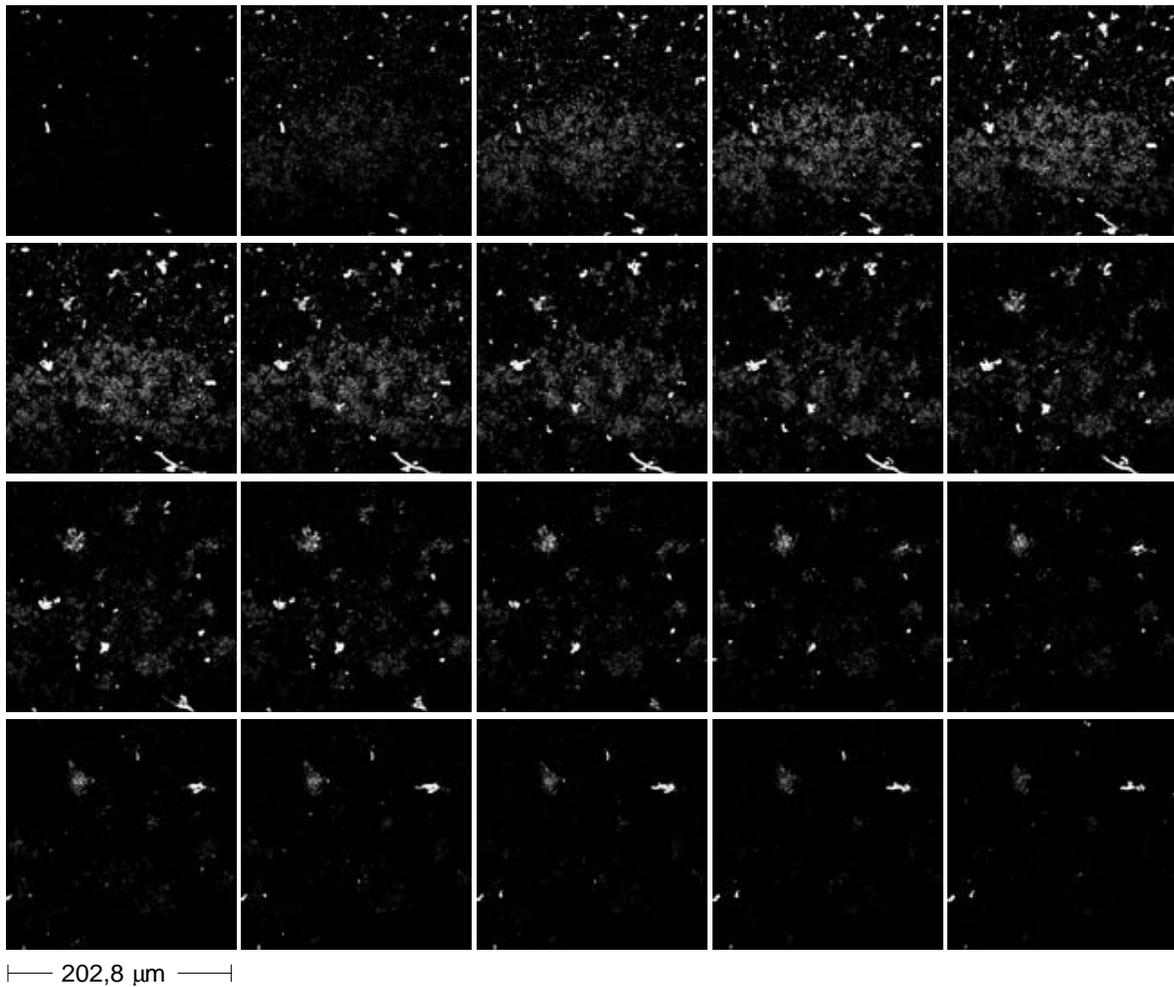
**Abbildung 1.10:** Beispiel für die Auswahl der Bilderpakete und -stapel durch die makrogesteuerte automatische Bildakquisition mit dem konfokalen Laser-Scanning-Mikroskop (CLSM)

Wassertauchlinse aufgenommen. Die GFP-Signale der Mikroben (vgl. 1.3) wurden mit Hilfe der 488 nm Laserlinie und einem 515-540 nm Bandpass-Emissions-Filter, die mit Concavalin A markierten EPS (vgl. 1.3) mit einem 543 nm Anregungslaser in Verbindung mit einem 590-610 nm Bandpass-Emissions-Filter ausgewertet. Abbildung 1.11 zeigt 20 übereinander liegende horizontale Schnittbilder der Mikrobenbesiedelung eines wenige Tage alten, etwa 20  $\mu\text{m}$  dicken *Sphingomonas* sp. LB126 Monokultur-Biofilms.

## 1.5 Bildverarbeitung und Bildanalyse

Im Anschluss an die automatisierte Bildakquisition werden die erzeugten zweidimensionalen Schnittbilder off-line mit Hilfe eines Quantimet 570 Computer-Systems (Leica, Cambridge) weiter bearbeitet. Ziel ist die Verbesserung der Bildqualität, die Beseitigung ungewollter Details wie zum Beispiel der Reflexionen des Kanalbodens in bodennahen konfokalen Ebenen und die Berechnung verschiedener Parameter wie beispielsweise des Volumenanteils der Mikroben im betrachteten Gebiet ([46]).

Vor Beginn der Bildbearbeitung werden zunächst manuell Größe und Position des ausgewählten Gebietes über Dialogfunktionen des Computer-Systems eingegeben. Die zu diesem Gebiet existierenden CLSM-Bilder werden automatisch von einem externen Speicher in den Quantimet Hauptspeicher geladen. Zur Verbesserung der Qualität der Graustufenbilder werden die mathematischen Filter WSharpen in Verbindung mit WTopHat oder ein Medianfilter angewandt. Anschließend werden zur Detektion bestimmter Bildbereiche Schwellenwerte als

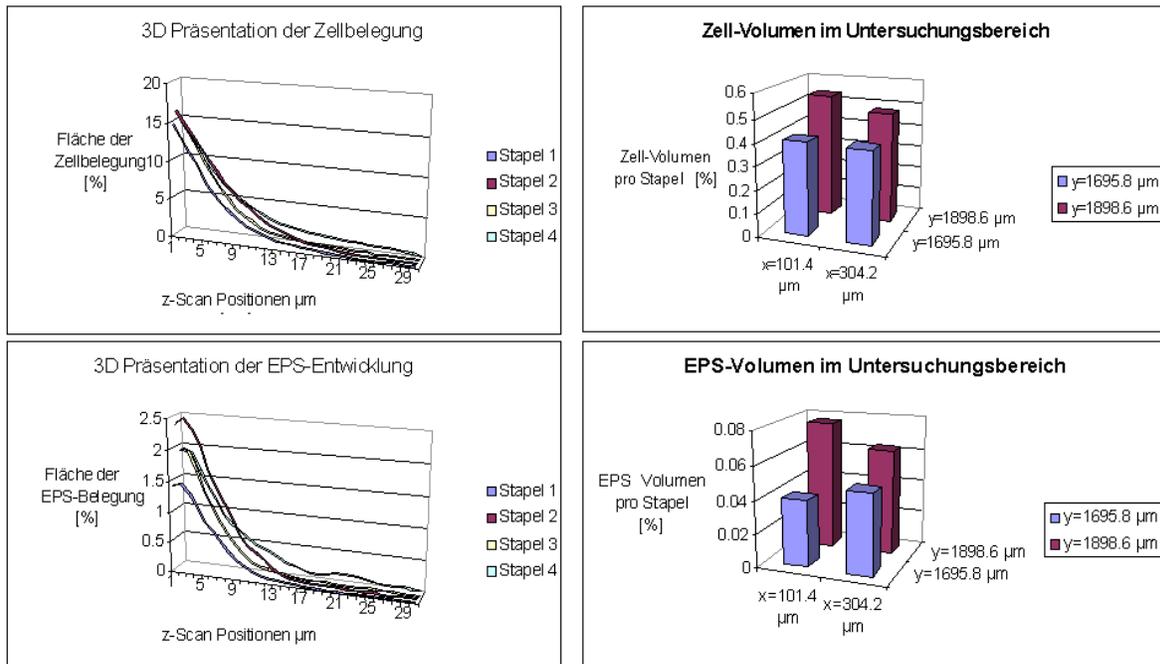


**Abbildung 1.11:** CLSM-Aufnahmen horizontaler Schnitte durch einen *Sphingomonas* sp. LB126 Biofilm (vertikaler Abstand der Ebenen 1.0 μm)

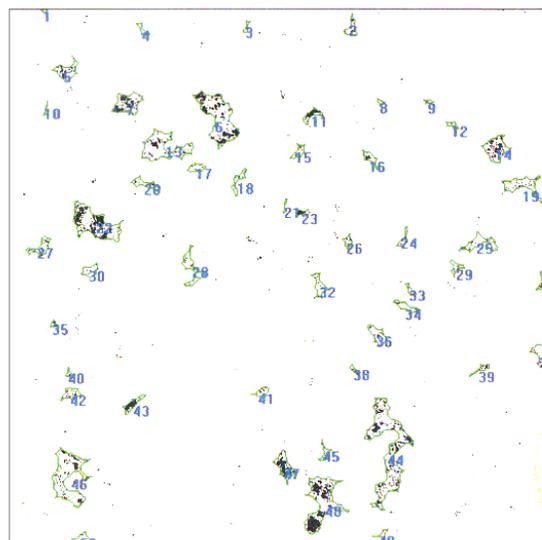
Grundlage für die Umwandlung der Graustufenbilder in binäre 0-1- oder Schwarzweiß-Bilder festgelegt. Mit Hilfe dieser Schwarzweiß-Bilder ist nun durch Abzählen der schwarzen oder weißen Bildpunkte die Messung des Flächenanteils der Mikroben oder der EPS an der Gesamtfläche des confokalen Schnitts möglich. Das Mikroben- oder EPS-Volumen eines Biofilmausschnitts wird durch numerische Integration über die Flächenanteile nach der Trapezregel berechnet. Abbildung 1.12 zeigt Ergebnisse der experimentellen Flächen- und Volumenbestimmung für einen wenige Tage alten *Sphingomonas* sp. LB126 Biofilm.

In Abhängigkeit von der untersuchten Spezies und der Zielrichtung des Experiments werden im Laufe der Bildanalyse weitere Prozeduren angewandt. So werden beispielsweise durch die Analyse zusammenhängender Bakteriencluster in Zeitreihenbildern dynamische Veränderungen der Biofilmstruktur sichtbar. Dazu werden zunächst die Cluster-Umrissformen durch Setzen eines „OPEN“-Filters (Erosion gefolgt von einer Dilation) bestimmt und mit dem Binärbild des Originalbildes überlagert (vgl. Abbildung 1.13). Nach Identifizierung und Abzählen der Zellen innerhalb und außerhalb der Cluster-Umrisslinien wird die Bakterien-

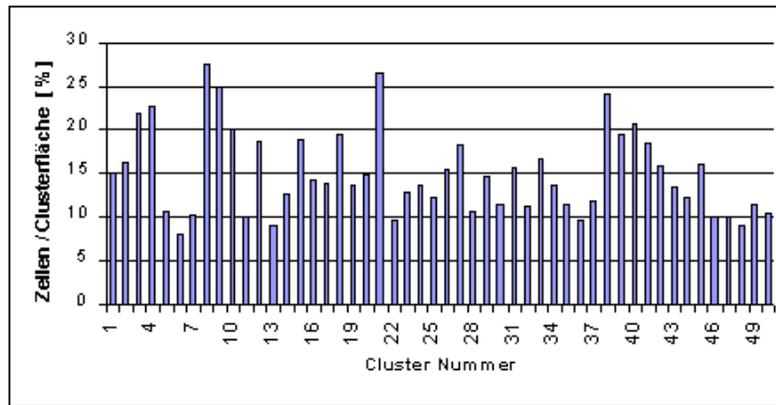
dichte der einzelnen Cluster (vgl. Abbildung 1.14) und der Anteil der nicht in Cluster eingebundenen Mikroben berechnet.



**Abbildung 1.12:** Volumen- und Flächenanteil der Mikrobenzellen und der EPS in verschiedenen Tiefen und zu verschiedenen  $x, y$ -Koordinaten in einer *Sphingomonas sp.* LB126 Monokultur (aus [11])



**Abbildung 1.13:** Clusteranalyse in der confokalen Ebene (*Sphingomonas sp.* LB126, aus [11])



**Abbildung 1.14:** Zelldichte der Cluster im horizontalen Beobachtungsfeld (*Sphingomonas* sp. LB126)

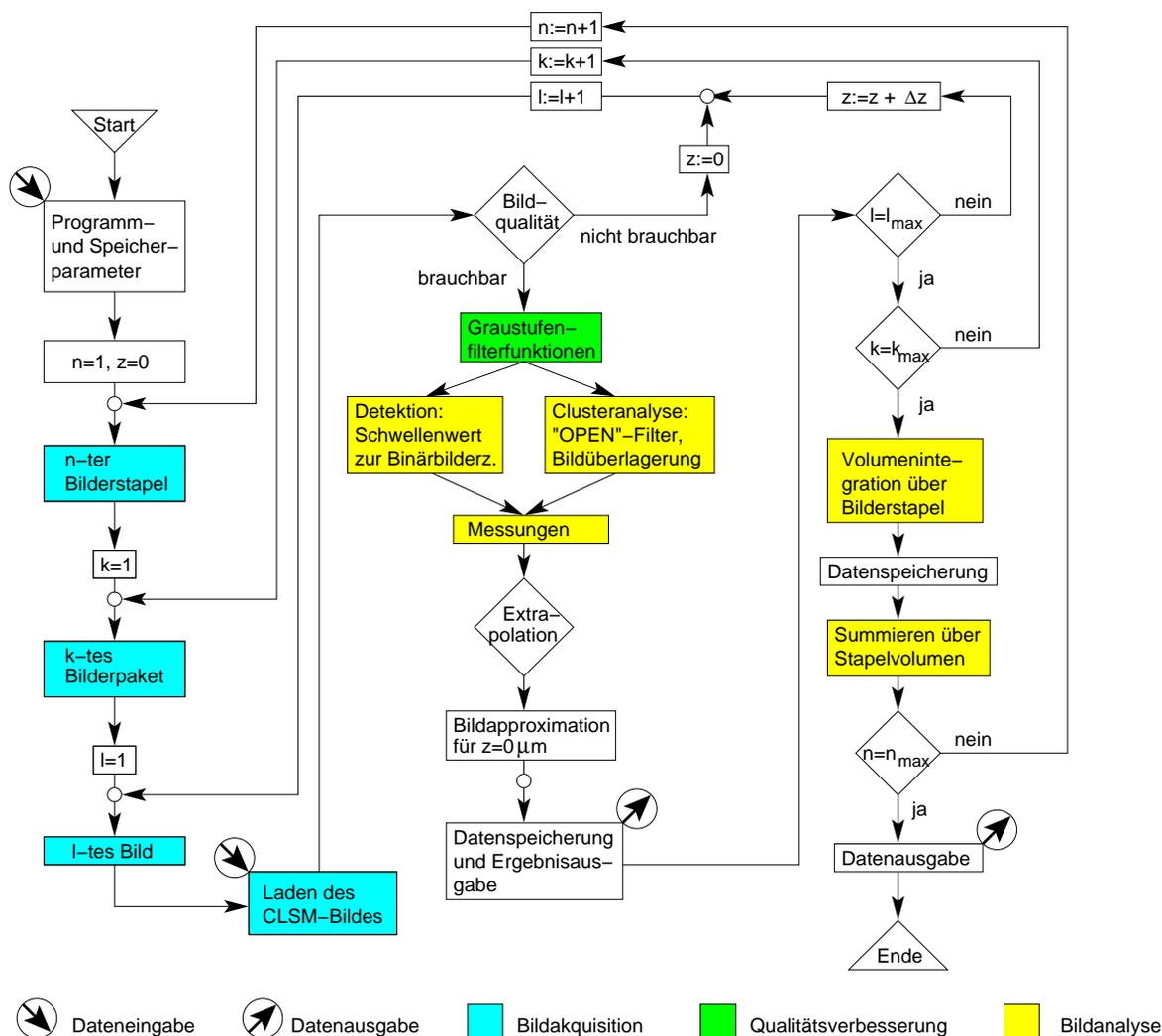
Zur schnellen Untersuchung der Bilddaten ist die Selektion von Datentabellen, Histogrammen und Graustufenprofilen auch online durchführbar. Die Bildanalyse mit Hilfe des Quantimet Computer-Systems erfolgt über eine graphische Benutzerschnittstelle mittels einer interaktiven Bildanalyse-Software. Wiederholbare Messroutinen wurden basierend auf der Bildverarbeitungs-Makro-Software QUIPS (Quantimet Image Processing System) entwickelt ([46]). Abbildung 1.15 zeigt das Flussdiagramm der Makroroutine zur Analyse der Biofilmschnittbilder mit Hilfe des Quantimet Computer Systems.

## 1.6 Auswahl eines repräsentativen Biofilmvolumens

Um die Abtastzeit, die zur Bilderfassung mit dem CLSM benötigt wird, zu minimieren, gleichzeitig aber noch allgemeingültige Aussagen über die Struktur und Entwicklung eines Biofilms aus den Bilddaten ableiten zu können, darf der per CLSM erfasste Ausschnitt des Biofilms weder zu groß noch zu klein gewählt werden. Ein in diesem Sinne optimal großer Bereich des Biofilms wird als „repräsentatives Biofilm-Volumen“ bezeichnet und nach einem statistischen Verfahren bestimmt, das auf der in [43] beschriebenen Methode basiert.

Es wird zunächst ein ausreichend großer Bereich eines Beispielbiofilms mittels CLSM gescannt. Anschließend werden unterschiedlich große Teilgebiete dieser Fläche auf ihre Repräsentativität getestet. Um dabei Gebietsgrößen in feinerer Abstufung untersuchen zu können, wird zunächst jedes CLSM Einzelbild in vier Quadranten unterteilt. Für jeden dieser Quadranten  $Q$  wird die biologische Flächenbelegung  $F_Q$  berechnet (siehe 1.5). Um die dem statistischen Verfahren zur Verfügung stehende Datenmenge zu erhöhen, werden zusätzlich weitere, künstliche Biofilmbereiche erzeugt, indem die Quadranten des gescannten Bereichs per Zufallsgenerator permutiert werden (vgl. Abbildung 1.16).

Nun werden für Unterbereiche  $U$  zunehmender Fläche (2, 4, 8, 16, ... Quadranten) die vor-



**Abbildung 1.15:** Flussdiagramm der Quantimet-Makroroutine zur Bildbearbeitung und Bildanalyse von der interaktiven Auswahl eines oder mehrerer Bilderpakete beziehungsweise Bilderstapel aus den vorhandenen CLSM-Daten über die Verbesserung der Bildqualität bis zur Berechnung charakteristischer Parameter wie des biologischen Volumenanteils (aus [46])

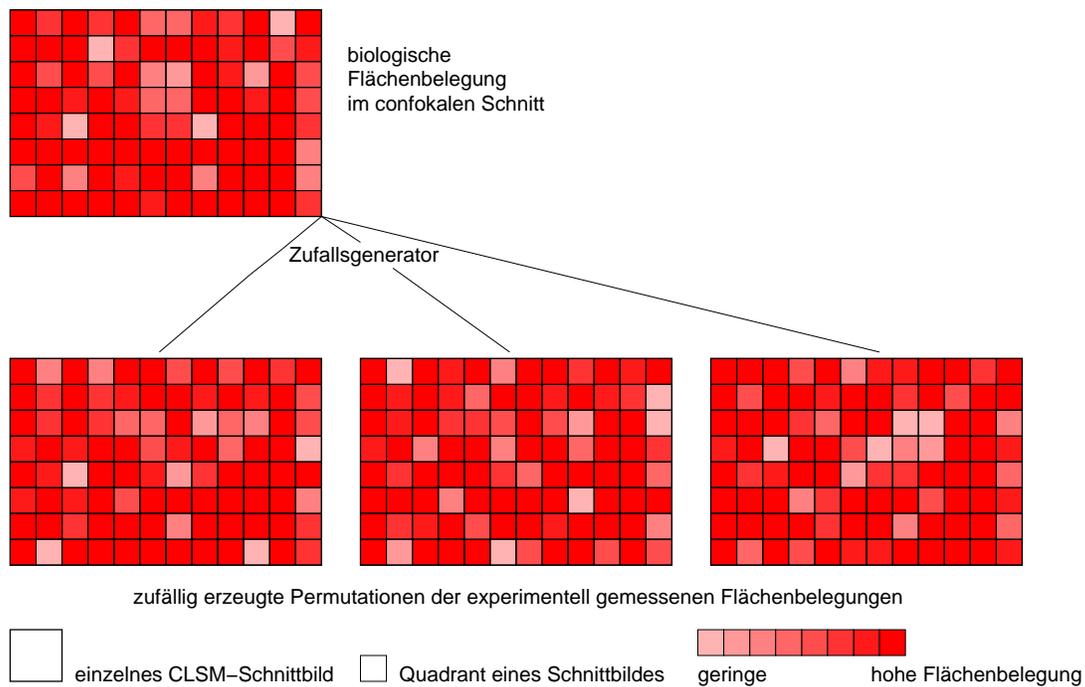
kommenden biologischen Flächenbelegungen

$$F_U = \sum_{Q \in U} F_Q$$

betrachtet und deren Variationskoeffizienten durch

$$V = \frac{\sqrt{\frac{1}{N-1} \sum_U (F_U - \bar{F}_U)^2}}{\bar{F}_U}$$

geschätzt, wobei  $N$  die Anzahl der unterschiedlichen Unterbereiche entsprechender Größe und  $\bar{F}_U$  das arithmetische Mittel  $\frac{1}{N} \sum_U F_U$  bezeichnen. Die repräsentative Flächengröße

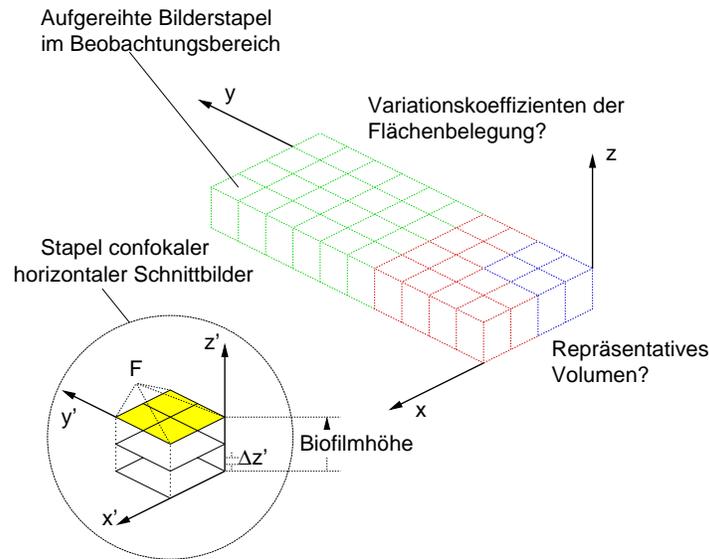


**Abbildung 1.16:** Randomisierte Erzeugung unterschiedlicher Permutationen der experimentell ermittelten biologischen Flächenbelegungen eines Beispielbiofilms über das gesamte Untersuchungsgebiet

ist erreicht, wenn die Variationskoeffizienten unterhalb einer bestimmten, vorher gewählten Grenze liegen (vgl. Abbildung 1.17). Die gesamte gescannte Fläche muss dabei so groß sein, dass  $N$  für alle betrachteten Unterbereichsgrößen ausreichend groß ist. Um dies genauer zu überprüfen, wird zusätzlich unter Annahme der Student-Verteilung für die Flächenbelegungen der Unterbereiche eine Konfidenzintervallanalyse für die Variationskoeffizienten durchgeführt.

Aufgrund der starken Abhängigkeit der Biofilmstruktur von der Höhe  $z$  über der Aufwuchsfläche kann die repräsentative Fläche je nach Tiefe des confokalen Schnitts im Biofilm variieren, müsste also für jede Schnittebene separat bestimmt werden. Aus Effizienzgründen beschränkt man sich hierbei jedoch auf einzelne, stichprobenartig ausgewählte Schnittebenen. Wegen der erwähnten Tiefenabhängigkeit der Biofilmstruktur muss der Biofilm in  $z$ -Richtung in seiner gesamten Höhe erfasst werden. Als repräsentatives Volumen nimmt man daher die maximale repräsentative Fläche der Schnittebenen multipliziert mit der gesamten Biofilmhöhe an.

Das beschriebene Verfahren verwendet die biologische Flächenbelegung als einziges Kriterium für die Repräsentativität eines Biofilmausschnitts. Man erhält daher nur in Bezug auf dieses Kriterium repräsentative Daten. Analog wären jedoch auch Verfahren mit anderen Kriterien denkbar, zum Beispiel der Porosität, der fraktalen Dimension, der Flächenbelegung durch die EPS oder Kombinationen mehrerer Parameter. [51] beschreibt eine Reihe von charakteristischen Parametern der Biofilmstruktur und deren Berechnung aus Bilddaten.



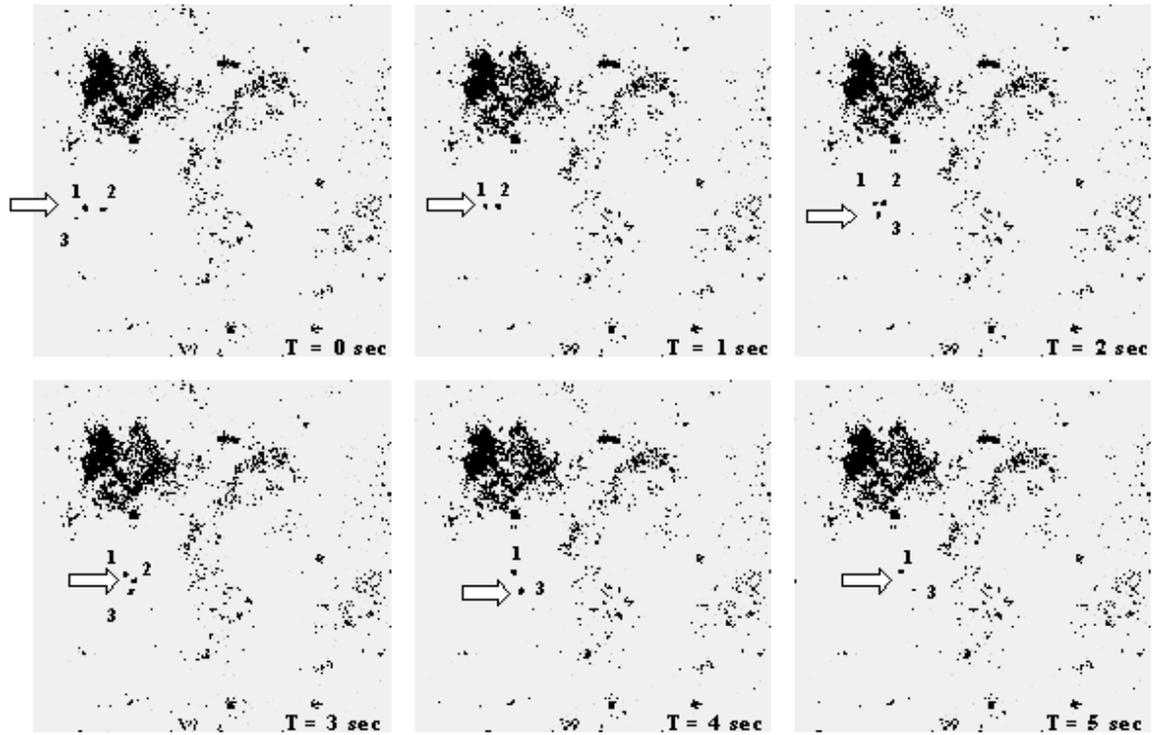
**Abbildung 1.17:** Statistische Untersuchung unterschiedlich großer Biofilmausschnitte zur Bestimmung des repräsentativen Biofilmvolumens

## 1.7 Partikelverfolgung

Mit Hilfe winziger Partikel, die der Nährlösung des Biofilms zugegeben werden, lassen sich lokale Strömungsgeschwindigkeiten experimentell bestimmen und das Verhalten einer nicht wachsenden kokkoiden Bakterienzelle oder kleiner anorganischer Partikel im Abwasser untersuchen.

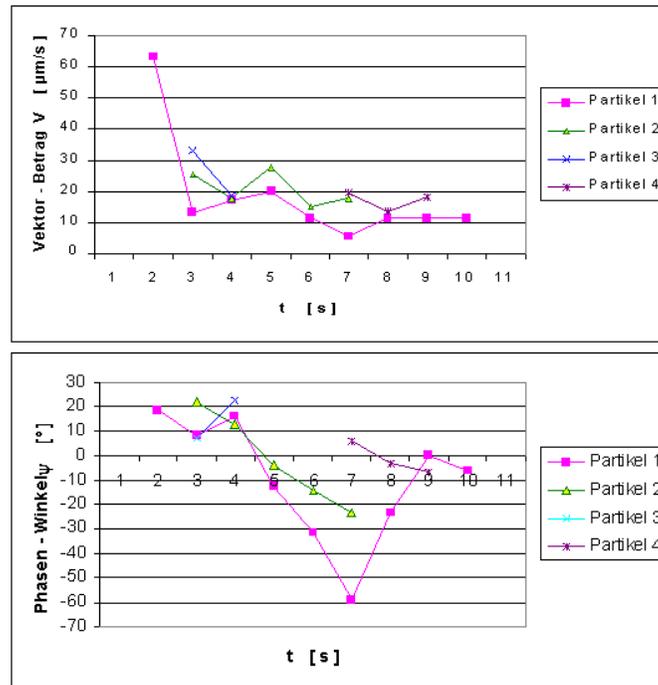
Zur Durchführung solcher Partikelverfolgungsversuche eignen sich zum Beispiel fluoreszierende, carboxylat-modifizierte Mikrokügelchen (Molecular Probes, Eugene Or., USA) mit einem Durchmesser von etwa einem Mikrometer. Diese Mikrokügelchen können mit dem CLSM mit Hilfe eines 633 nm Anregungslasers und eines 665 nm Langpass-Emissions-Filters beobachtet werden. Die Mikrokügelchen werden in einer Konzentration von ungefähr  $5 \cdot 10^8$  Partikeln pro Milliliter unter die Nährlösung gemischt, in der sie frei mitschwimmen, da sie in etwa die Dichte der Lösung besitzen. Durch eine Zeitserie von CLSM-Bilder in einer vorab gewählten confokalen Ebene mit einer Zeitdifferenz von circa einer Sekunde werden die Pfade der Partikel im Biofilm aufgenommen. Um aus diesem Bildmaterial lokale Geschwindigkeiten berechnen zu können, müssen zunächst markante Partikel manuell anhand einer Zeitlupenanimation der Zeitserien-Schnittbilder herausgefiltert werden. Die nicht benötigten Partikel werden in jedem Bild der Zeitserie ausgeblendet (siehe Abbildung 1.18). Die Geschwindigkeitsvektoren der verbleibenden Partikel werden nun automatisch mit Hilfe einer Makro-Routine im Rahmen der Bildanalyse (siehe 1.5) aus der Differenz ihrer  $x, y$ -Koordinaten und der Zeitdifferenz aufeinanderfolgender Bilder berechnet.

Abbildung 1.19 zeigt die Ergebnisse für vier ausgewählte Partikel in der confokalen Ebene. Da sich die Strömung im Biofilm vor allem in der Nähe von sehr unregelmäßigen Biofilmoberflächen und in Poren und Kanälen nicht nur in einer Ebene abspielt, sondern stark dreidimensionalen Charakter hat, ist die Geschwindigkeitsbestimmung einzelner Partikel oft



**Abbildung 1.18:** Positionen von drei ausgewählten Partikeln in einer Zeitserienstudie einer *Sphingomonas sp. LB126* Monokultur

nicht ohne weiteres möglich. Das ständige Verschwinden und Wiederauftauchen der Partikel in der fokussierten confokalen Ebene erschwert die Identifizierung der ausgewählten Partikel in aufeinanderfolgenden Zeitreihenbildern erheblich. Selbst wenn diese Identifizierung zweifelsfrei möglich wäre, könnte nur der zweidimensionale Anteil der Geschwindigkeiten berechnet werden. Vertikale Bewegungen werden nicht erfasst. Die numerische Simulation (siehe Kapitel 3) ist daher zum aktuellen Zeitpunkt die einzige Möglichkeit, dreidimensionale Geschwindigkeitsprofile zu erstellen.



**Abbildung 1.19:** Geschwindigkeitsvektoren (Absolutgeschwindigkeiten und Richtungswinkel) von vier ausgewählten Partikeln in der confokalen Ebene

## 1.8 Bestimmung der Diffusionskoeffizienten

Für die realistische numerische Simulation des Transports von Sauerstoff und Nähr- beziehungsweise Schadstoffen im Fluid, im Mikrobengebiet und in den EPS eines Biofilms müssen die Diffusionskoeffizienten der einzelnen Stoffe in allen drei Medien bekannt sein. Dazu werden Zeitreihenmessungen der Stoffkonzentrationen und -gradienten an festen Punkten im Beobachtungsgebiet benötigt. In [49] wird die Berechnung der Diffusionskoeffizienten für bestimmte Teststoffe (fluor-konjugierte Dextrane unterschiedlichen Molekulargewichts) aus CLSM-Zeitreihenbildern fester Orte im Biofilm detailliert beschrieben. Man nimmt dabei an, dass bei geeigneter Einstellung der Mikroskope die Graustufenwerte der CLSM-Bilder direkt proportional zur Konzentration der gelösten Stoffe sind. Alternativ lassen sich die Konzentrationsgradienten ausgewählter Substanzen auch über photoakustische Effekte ([41, 42]) messen.

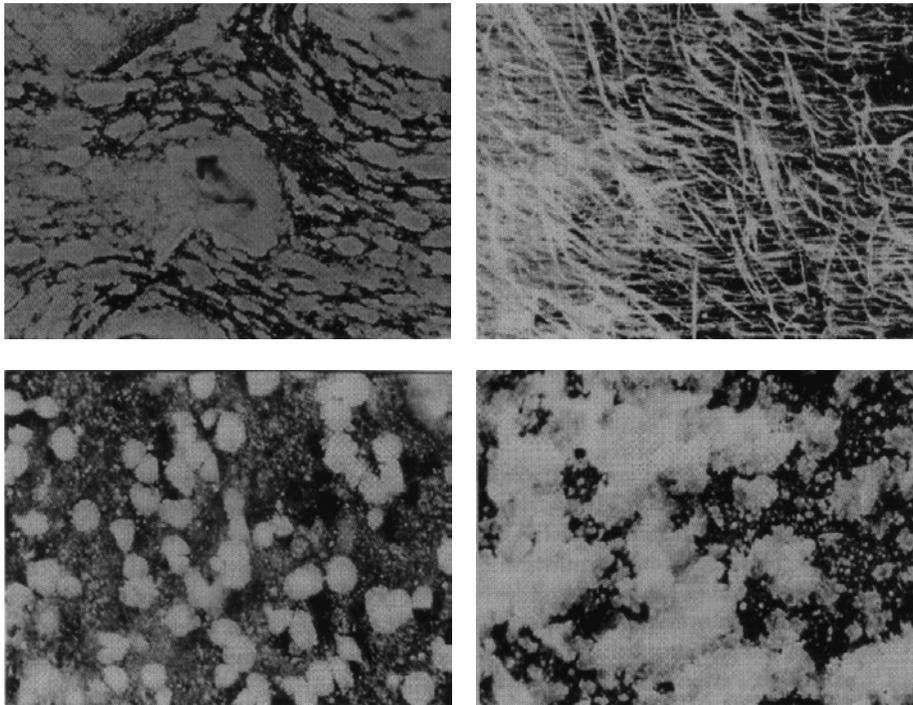
## Kapitel 2

# Mathematische Modelle

In der Literatur findet man zahlreiche, je nach Einsatzbereich (beispielsweise Betrieb und Steuerung eines Biofilmreaktors auf der einen oder Grundlagenforschung auf der anderen Seite) sehr unterschiedliche mathematische Modelle für Biofilmsysteme. Die besonderen Schwierigkeiten für die Modellierung von Biofilmen liegen zum einen in der Vielfalt der Erscheinungsformen – beispielsweise dichte, homogene Filme, poröse, mit Hohlräumen und Kanälen durchsetzte Strukturen (Abbildung 2.1) oder pilzartige Bakteriencluster (Abbildung 2.2) – zum anderen in der Komplexität der für das Wachstum und die Funktionalität des Biofilms verantwortlichen Mechanismen.

Bei der Auswahl oder Generierung eines im speziellen Fall geeigneten Modells sind zunächst mehrere grundlegende Fragen zu beantworten:

- Welche räumliche Skala wird betrachtet? Ist man am Verhalten des gesamten Reaktors (Größenordnung ein Meter) oder eher an der detaillierten Funktionsweise des Biofilms auf der Ebene einzelner Mikroben (Größenordnung ein Mikrometer) interessiert?
- Welche Zeitskala ist von Interesse? Sollen nur die „schnellen“ Prozesse wie Fluidströmung und konvektiver Stofftransport bei statischer Geometrie oder die Entwicklung einer Bakterienkultur über mehrere Wochen modelliert werden?
- Welche Prozesse werden als besonders relevant angesehen und welche können bei der gegebenen Zielsetzung vernachlässigt werden? Es wird beispielsweise häufig die Bedeutung biologischer Prozesse wie des Wettbewerbs und der Kommunikation zwischen den einzelnen Mikroben gegenüber den rein physikalisch/chemischen Prozessen wie Strömung, Konvektion, Diffusion und Reaktionen diskutiert. Der Einfluss, vor allem aber auch die genauen Mechanismen dieser biologischen Phänomene sind bisher sehr umstritten. Daher wird die „Biologie“ in den meisten Modellen zunächst vernachlässigt.
- Welche Dimensionalität soll das Modell haben? Soll es sich um ein stark vereinfachendes eindimensionales Modell handeln, das relevante Größen wie zum Beispiel Schadstoffkonzentrationen lediglich in Abhängigkeit von der Position auf einer eindimensionalen Ortsachse – im Allgemeinen der Höhe über der Aufwuchsfläche – wiedergibt? Oder sollen die dreidimensionale heterogene Struktur des Biofilms und alle durch sie

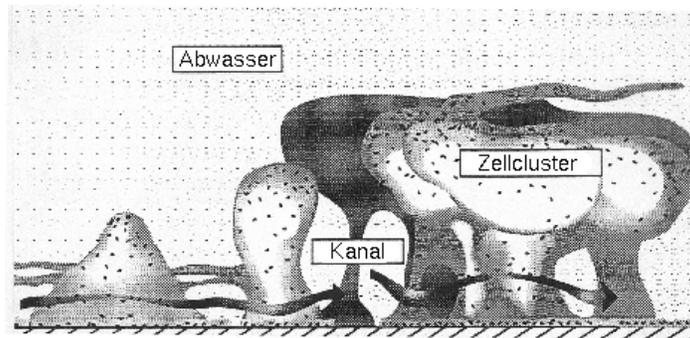


**Abbildung 2.1:** Unterschiedliche heterogene Biofilmstrukturen (aus [24], Größe der dargestellten Ausschnitte etwa  $0,5 \times 0,3 \text{ cm}^2$ )

verursachten Effekte auf Strömung und Stofftransport im Biofilmsystem vollständig aufgelöst werden?

Bei der Beantwortung obiger Fragen muss im Hinblick auf die numerische Berechenbarkeit des Modells auf möglichst weitgehende Vereinfachungen, gleichzeitig aber auch auf ausreichende Genauigkeit geachtet werden, um den erwarteten Kenntnisgewinn zu erzielen oder die Steuerung bestimmter Prozesse verbessern zu können. Dazu muss die Zielsetzung des Modells genau definiert sein. Sind Zielsetzung und Inhalt festgelegt, muss eine geeignete Art von Modell gefunden werden. In der Literatur findet man vor allem drei Arten von Modellen: gleichungsbasierte oder physikalische Modelle, regelbasierte oder biologische Modelle und hybride Modelle, Mischformen aus gleichungs- und regelbasierten Modellen.

Gleichungsbasierte oder physikalische Modelle beschreiben durch algebraische Gleichungen oder Differentialgleichungen die Beziehung zwischen charakteristischen physikalischen Größen. Die Skala reicht dabei von eindimensionalen Schichtmodellen, die das Biofilmsystem vereinfachend als aus den drei Schichten Aufwuchskörper, Biofilm und freies Fluid bestehend modellieren ([85, 36]) bis zu dreidimensionalen räumlichen Modellen auf Grundlage der Grundgesetze der Kontinuumsmechanik (Navier-Stokes-, Konvektions-Diffusions-, Diffusions-Reaktions-Gleichungen, [12, 24]). Gleichungsbasierte Modelle eignen sich besonders für die „schnellen“ Prozesse innerhalb des Biofilmsystems wie beispielsweise die Fluidströmung, den Transport gelöster Stoffe und die Reaktionen im Biofilm. Da sie meist auf physikalischen und chemischen Grundgesetzen beruhen, besitzen gleichungsbasierte Modelle in der Regel große Allgemeingültigkeit. Lediglich die Modellparameter, nicht aber die Modellgleichungen selbst,



**Abbildung 2.2:** Mushroom-Modell für die Architektur der Biofilmcluster ([16])

müssen durch eine – mit zunehmendem Vereinfachungsgrad des Modells immer aufwändigere – Kalibrierung für jedes einzelne betrachtete System neu bestimmt werden. Da Wachstum und Ablösung von Teilen des Biofilms durch eine Kombination physikalischer, chemischer und biologischer Prozesse verursacht werden, ist ihre Beschreibung mit Hilfe gleichungsbasierter Modelle schwieriger. Regelbasierte oder biologische Modelle, die auf – meistens heuristisch motivierten – Verhaltensregeln für Teile des Systems (zum Beispiel die Mikroben) beruhen, lassen sich dagegen vergleichsweise einfach und schnell erstellen, sind in ihrer Allgemeingültigkeit jedoch oft eingeschränkt, da die zugrunde liegenden Regeln oft lediglich auf Erfahrungswerten und intuitiven Vorstellungen beruhen. Zur Modellierung von Biofilmen werden regelbasierte Modelle beispielsweise in Form von zellulären Automaten ([33, 87]) oder fraktalen Modellen ([34, 51]) verwendet. Hybride Modelle versuchen, beide Ansätze und damit die gesamte Bandbreite der Prozesse von der Fluidströmung bis zum Wachstum des Biofilms in sich zu vereinen. Besondere Schwierigkeiten bereitet hierbei die Kopplung der unterschiedlichen auftretenden Zeitskalen ([24], [12] und 2.2.2).

Für die numerische Simulation von Biofilmsystemen wird in dieser Arbeit ein solcher hybrider Ansatz verwendet, bestehend aus einem kontinuumsmechanischen Modell für Strömung, Transport und Reaktion und einem zellulären Automaten für das Wachstum des Biofilms (siehe 2.2.2). Weitere Prozesse, insbesondere biologische Aspekte wie der bereits erwähnte Wettbewerb und die Kommunikation zwischen den Bakterienzellen werden ebenso wie die Ablösung von Teilen des Biofilms nicht berücksichtigt – zum Teil aus Mangel an genauen Kenntnissen über die zugrunde liegenden Mechanismen, zum Teil weil die Bedeutung insbesondere der biologischen Vorgänge umstritten ist. Aus Gründen der Einfachheit werden zunächst nur Monokultur-Biofilmsysteme, bestehend aus Zellen einer einzigen Bakterienart und dem Nährfluid, betrachtet. Für mögliche Erweiterungen dieses Modells siehe Anhang B.1. Um den Einfluss der Heterogenität des Biofilms voll zu erfassen, wird der Biofilm dreidimensional auf einer räumlichen Skala von wenigen Mikrometern modelliert. Hinsichtlich der zeitlichen Skala wird durch den hybriden Ansatz ein breites Spektrum abgedeckt – von der Strömung mit einer charakteristischen Zeitskala im Bereich von Millisekunden bis zum Wachstum des Biofilms, das sich über Stunden oder sogar Tage erstreckt.

Einen kurzen Überblick über andere allgemein gebräuchliche Modelle geben 2.1 und 2.2.1.

## 2.1 Eindimensionale Modelle

Eindimensionale Modelle werden aufgrund ihrer Einfachheit – sowohl die Anzahl und Komplexität der Modellgleichungen als auch die Anforderungen an ein numerisches Simulationsprogramm betreffend – im Vergleich zu raumauflösenden, dreidimensionalen Modellen bereits häufig verwendet. Für die Simulation eindimensionaler Modelle sind kommerzielle Programme wie RIOVAL ([37]) oder AQUASIM ([67]) erhältlich.

Eindimensionale Modelle gehen von der Annahme aus, dass Biofilme in Schichten parallel zur Aufwuchsfläche weitgehend homogen sind in Bezug auf Charakteristika wie metabolische Leistungsfähigkeit, Mikrobendichte und Anteil der Konvektion am Stofftransport. Nach Mittelung der entsprechenden Werte über die Fläche eines repräsentativen Ausschnitts einer solchen Schicht, hängen die zu berechnenden Daten nur noch vom Abstand von der Aufwuchsfläche und nicht mehr vom exakten Ort im Biofilm ab. Die Gleichungen sind somit eindimensional bezüglich der Raumkoordinaten.

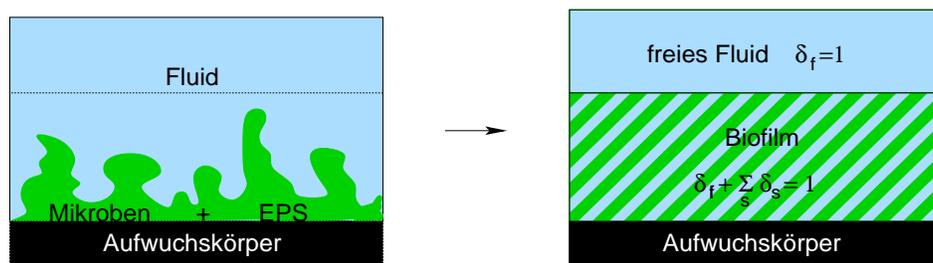
Das gesamte Biofilmsystem wird als dreiteiliger Komplex, bestehend aus den parallelen Schichten Aufwuchskörper, Biofilm und freies Fluid, modelliert (vgl. Abbildung 2.3). Der Biofilmbereich besteht dabei wiederum aus verschiedenen „Phasen“ wie beispielsweise enthaltenes Fluid einschließlich gelöster und suspendierter Partikel, extrazelluläre polymere Substanzen (EPS) und verschiedene feste Phasen für die beteiligten Mikrobenarten. Im Unterschied zu dreidimensionalen Modellen sind diese Phasen jedoch nicht lokalisierbar. Für jede Phase ist nur der Volumenanteil  $\delta_s$  am Gesamtvolumen zu bestimmen. Die Volumenanteile der einzelnen Phasen des Bereichs summieren sich zu eins:

$$\delta_f + \sum_s \delta_s = 1, \quad (2.1)$$

wobei  $\delta_f$  den Volumenanteil der Fluidphase bezeichnet. Im Bereich des freien Fluids nimmt man im Allgemeinen an, dass gilt

$$\delta_f = 1. \quad (2.2)$$

Mitgeführte Mikrobencluster oder anorganische Mikropartikel werden somit vernachlässigt.



**Abbildung 2.3:** Schematische Darstellung des Übergangs vom realen Biofilmsystem (links) zum Schichtmodell (rechts)

Jede Phase  $s$  des Biofilms enthält bestimmte Komponenten  $i$  (z. B. gelöste Schad- oder Nährstoffe, Sauerstoff oder gebundene Partikel). Ziel des Modells ist die Vorhersage der Konzentrationen der Komponenten in den verschiedenen Phasen des Biofilms in Abhängigkeit von der Zeit  $t$  und dem Abstand  $z$  von der Aufwuchsfläche. Eindimensionale Modelle beschreiben

ein Biofilmsystem somit auf einer relativ kleinen räumlichen Skala (Dicke des Biofilms als charakteristisches Längenmaß). Substituiert man allerdings die Zeitachse durch eine weitere räumliche Dimension wie beispielsweise die vom Abwasser zurückgelegte Strecke in einem Fluss ([36]), so können mit Hilfe dieser Modelle auch wesentlich größere Systeme beschrieben werden. Im Folgenden wird das Modell von Gujer und Wanner ([19]) beschrieben, das bereits mehrfach erweitert ([68] und [85]) und auf reale Biofilmsysteme angewandt wurde ([36]).

### 2.1.1 Massenerhaltung

Die Konzentration  $c_{si}$  der Komponente  $i$  in der Phase  $s$  erfüllt die Massenerhaltungsgleichung

$$\frac{\partial c_{si} \delta_s}{\partial t} = -\frac{\partial J_{si}}{\partial z} + r_{si} \quad \forall s, i, \quad (2.3)$$

wobei  $J_{si}$  den Fluss der Komponente  $i$  in der Phase  $s$  über die gesamte horizontale Schnittfläche des Biofilms,  $r_{si}$  die Produktionsrate für die Komponente  $i$  in der Phase  $s$  über das gesamte Biofilmvolumen ist. Diese physikalischen Grundgleichungen gelten immer und überall unabhängig vom betrachteten Biofilm und enthalten in dieser Form keinerlei Annahmen über die spezifische Kinetik oder Stöchiometrie im Biofilm. Die unbekanntenen Größen sind die Konzentrationen  $c_{si}$  und die Volumenanteile  $\delta_s$ .  $J_{si}$  und  $r_{si}$  werden als bekannte Funktionen betrachtet, die je nach spezieller Wahl des Modells unterschiedlich – im Allgemeinen unter Zuhilfenahme weiterer, zum Teil heuristischer Annahmen über den Biofilm – festgelegt werden.  $J_{si}$  beschreibt dabei alle relevanten Transportprozesse wie zum Beispiel Advektion oder Diffusion,  $r_{si}$  vor allem Wachstum und metabolische Aktivität der Mikrobenzellen.

### 2.1.2 Randbedingungen

Da die Volumenanteile und die Konzentrationen im freien Fluid und im Biofilmbereich nicht voneinander unabhängig sind, müssen an der Grenzfläche zwischen freiem Fluid und Biofilmbereich bestimmte Randbedingungen gelten, die für eine Kopplung von  $\delta_s$  und  $c_{si}$  über die Grenzfläche hinweg sorgen. Diese ergeben sich wiederum aus der Massenerhaltung, aus der die Identität der Flüsse auf beiden Seiten der Grenzfläche folgt. Die Grenzfläche wird dabei nicht als statisch angenommen, sondern kann sich mit der Geschwindigkeit  $v_G$  senkrecht zur Aufwuchsfläche bewegen ( $v_G > 0$  entspricht dem Wachstum des Biofilms,  $v_G < 0$  dem Schrumpfen des Biofilms). Damit ergibt sich an der Grenzfläche die Kontinuitätsgleichung

$$v_G \left( \delta_s^{(f)} c_{si}^{(f)} - \delta_s^{(b)} c_{si}^{(b)} \right) = J_{si}^{(f)} - J_{si}^{(b)} - r_{si}^* \quad \forall s, i, \quad (2.4)$$

wobei die oberen Indizes  $(f)$  und  $(b)$  die beiden Seiten der Grenzfläche (freies Fluid und Biofilm) bezeichnen. Die Grenzflächentransferprozesse  $r_{si}^*$  beschreiben Prozesse wie die Bindung im freien Fluid enthaltener Partikel oder Mikroben an den Biofilm oder umgekehrt die Ablösung von Teilen des Biofilms.

### 2.1.3 Basismodell für Biofilmsysteme

Die Gleichungen (2.1), (2.3) und (2.4) sind die Hauptgleichungen des Modells von Gujer und Wanner ([19]). Konkrete Formulierungen des Modells unterscheiden sich lediglich in der Wahl

- der Flussfunktionen  $J_{si}$ ,
- der Produktionsraten  $r_{si}$ ,
- der Grenzflächentransferprozesse  $r_{si}^*$  und
- der Anfangsbedingungen  $\delta_s(z, t_0)$  und  $c_{si}(z, t_0)$ , die einigen grundlegenden physikalischen Beschränkungen genügen müssen.

Das Basismodell von Gujer und Wanner ([19]) für weitgehend homogene Biofilme stützt sich bei dieser Wahl im Wesentlichen auf folgende Annahmen ([19, 35]):

- (A1) Der Volumenanteil  $\delta_f$  der flüssigen Phase im Biofilmbereich ist konstant.
- (A2) Die flüssige Phase enthält nur gelöste Komponenten.
- (A3) In der flüssigen Phase findet nur diffusiver Transport statt.
- (A4) Die gelösten Stoffe in der flüssigen Phase sind elektrisch neutral und erfüllen das Ficksche Gesetz der Diffusion.
- (A5) Das Verhältnis  $D_i^{(f)}/D_i^{(b)}$  der Diffusionskoeffizienten im freien Fluid und im Biofilmbereich ist konstant.
- (A6) Die Dichte  $\rho_s$  der strukturbildenden Komponente in der Phase  $s$  ist konstant.
- (A7) Advektiver Transport der festen Phase resultiert aus Volumenänderungen der strukturbildenden Komponenten.
- (A8) Die advektiven Geschwindigkeiten  $v_s$  sind für alle festen Phasen gleich ( $v_s = v \forall s$ ).

Berechnet werden also neben den Volumenanteilen  $\delta_s$  einer flüssigen und mehrerer fester Phasen die Konzentrationen  $c_{fi}$  verschiedener gelöster Stoffe in der liquiden Phase. Als strukturbildende Komponenten einer festen Phase bezeichnet man hier die Komponenten, die zum Aufbau der Strukturmatrix beitragen. Je nach der Skala des Modells können dies beispielsweise einzelne Mikroben oder ganze Bakteriencluster sein. Mit den Annahmen (A6), (A7) und (A8) erhält man aus der Massenerhaltungsgleichung (2.3)

$$\rho_s \frac{\partial \delta_s}{\partial t} = - \frac{\partial v \rho_s \delta_s}{\partial z} \quad \forall s, \quad (2.5)$$

wobei die advektive Geschwindigkeit  $v$  durch die gewöhnliche Differentialgleichung

$$\frac{\partial v}{\partial z} = \frac{1}{1 - \delta_f} \sum_s \frac{r_s}{\rho_s} \quad (2.6)$$

mit der Produktionsrate  $r_s$  der strukturbildenden Komponente der Phase  $s$  gegeben ist. Da nach Annahme (A1)  $\delta_f$  konstant und damit durch die Anfangswerte  $\delta_f(z, t_0)$  gegeben ist, kann  $v$  ohne vorheriges Lösen einer Massenerhaltungsgleichung sofort aus dieser Gleichung berechnet werden ([19]).

Für die Konzentrationen  $c_{fi}$  gelöster Stoffe in der flüssigen Phase ergibt sich aus den Annahmen (A3) und (A4)

$$\frac{\partial c_{fi}}{\partial t} = f \cdot D_i \frac{\partial^2 c_{fi}}{\partial z^2}. \quad (2.7)$$

Der empirische Faktor  $f$  beschreibt hierbei den Einfluss der Komplexität der Biofilmgeometrie. Da eine stark verzweigte Geometrie die freie Diffusion behindert, ist  $f$  typischerweise kleiner als eins, falls keine turbulenten Strömungen auftreten.

Die Änderungen in der fluiden Phase (Gleichung (2.7)) weisen eine wesentlich kürzere charakteristische Zeit auf als die Änderungen in den festen Phase (Gleichung (2.5)). Prinzipiell entsteht dadurch ein steifes Gleichungssystem. Da die Produktionsrate  $r_s$  unabhängig von den Stoffkonzentrationen  $c_{fi}$  in der flüssigen Phase ist und die beiden Anteile des gesamten Gleichungssystems somit getrennt gelöst werden können, stellt dies jedoch kein Problem für die numerische Simulation der Modellgleichungen dar.

Der konstante Volumenanteil  $\delta_f$  der flüssigen Phase am Biofilmbereich, die Dichte  $\rho_s$  und die Produktionsraten  $r_s$  der strukturbildenden Komponenten der festen Phasen, die Grenzflächentransferprozesse  $r_{si}^*$ , die Geschwindigkeit  $v_G$  der Grenzfläche zwischen freiem Fluid und Biofilmbereich und der Faktor  $f$  für die Diffusion gelöster Stoffe in der flüssigen Phase werden für jeden einzelnen Biofilm empirisch oder experimentell bestimmt. Um eine numerische Simulation der Modellgleichungen durchführen zu können, müssen ferner neben geeigneten Randbedingungen an der Aufwuchsfläche und am oberen Rand des freien Fluids Anfangswerte  $c_{fi}(z, t_0)$  für die Konzentrationen der gelösten Stoffe in der flüssigen Phase und  $\delta_s(z, t_0)$  für die Volumenanteile der festen Phasen im Biofilmbereich aus experimentellen Beobachtungen abgeleitet werden.

Erweiterungen des Basismodells basierend auf neuen experimentellen Erkenntnissen werden in [68] und [85] beschrieben. Insbesondere wurden die Annahmen (A1), (A2), (A3) und (A7) fallengelassen: Der Volumenanteil der fluiden Phase wird nicht mehr als konstant angenommen, in den Poren und Kanälen suspendierte Partikel werden als zusätzliche Komponente der flüssigen Phase in das Modell integriert, der experimentell nachgewiesene advective Transport gelöster Stoffe in der flüssigen Phase – verursacht durch Ortsänderungen partikelartiger Komponenten – berücksichtigt und ein diffusiver Transport strukturbildender und anderer Teilchen eingeführt, der die Bewegung der Partikel in Richtung der Aufwuchsfläche und damit eine Mischung der Zellen im Biofilm und simultane Ablagerung und Ablösung von Teilen des Films ermöglicht ([35]).

#### 2.1.4 Verwandte Modelle

Im oben beschriebenen Modell sind die Gleichungen (2.5) und (2.7) nicht miteinander gekoppelt. Insbesondere ist also das Wachstum des Biofilms, das heißt der strukturbildenden Komponenten der festen Phasen des Biofilms, nicht beziehungsweise nur indirekt über die expe-



**Abbildung 2.4:** Schematische Darstellung des Schichtmodells mit Grenzschicht für den Transport gelöster Stoffe

rimentell für den einzelnen Biofilm geschätzte Produktionsrate  $r_s$  vom Nährstofftransport in der liquiden Phase abhängig. Aus zahlreichen Experimenten ist jedoch bekannt, dass Wachstum und Struktur des Biofilms wesentlich durch das Nährstoffangebot beeinflusst werden.

Um diesen Zusammenhang zu modellieren, wird eine weitere Schicht eingeführt, die sogenannte Grenzschicht zwischen Biofilmbereich und freiem Fluid (Abbildung 2.4). Die Grenzschicht bildet einen Widerstand für den advektiven Transport gelöster Stoffe im freien Fluid. Die Dicke  $h$  dieser Schicht hängt von den Strömungsgeschwindigkeiten im freien Fluid ab. Hohe Geschwindigkeiten implizieren eine dünne Grenzschicht, das heißt, die gelösten Stoffe werden advektiv sehr nahe an die Oberfläche des Biofilmbereichs transportiert. Bei geringen Geschwindigkeiten ist die Grenzschicht deutlich dicker, der Nährstofftransport zum Biofilm also erschwert ([4]). Innerhalb des Biofilmbereichs selbst findet ausschließlich diffusiver Transport der gelösten Stoffe statt. Der Biofilmbereich wird dabei nicht mehr wie oben beschrieben in einzelne Phasen unterteilt, sondern als homogene Gesamtmasse betrachtet.

Die Konzentration  $c^{(f)}$  des Nährstoffs im freien Fluid kann aufgrund der starken Advektion durch die Fluidströmung als konstant angenommen werden. Im Biofilmbereich erfüllt die Konzentration  $c$  des gelösten Nährstoffes nach Chang ([18]) – unter Annahme von Monod-Kinetik ([53]) für den Reaktionsterm – die Diffusions-Reaktions-Gleichung

$$\frac{\partial c}{\partial t} = D \frac{\partial^2 c}{\partial z^2} - \frac{qc}{K+c} \rho_b, \quad (2.8)$$

wobei  $D$  den Diffusionskoeffizienten,  $q$  die maximale Nährstoffverbrauchsrate, also die Verbrauchsrate bei unbegrenzter Verfügbarkeit des Nährstoffes ( $c = \infty$ ),  $\rho_b$  die Dichte der aktiven Biomasse und  $K$  die maximale Sättigungskonzentration des Stoffs bezeichnen.  $q$  und  $K$  sind bei gleichbleibenden Umweltbedingungen (Temperatur, pH-Wert etc.) organismusspezifische Konstanten.

In der Grenzschicht wird der Anteil der Advektion am Transport des gelösten Stoffes vernachlässigt. Nimmt man ferner an, dass die Nährstoffkonzentration im freien Fluid und damit auch in der Grenzschicht nicht zeitabhängig ist, so gilt

$$0 = \frac{\partial c}{\partial t} = \frac{\partial^2 c}{\partial z^2}. \quad (2.9)$$

Die Konzentration verläuft daher innerhalb der Grenzschicht linear in  $z$ -Richtung.

An der Grenzfläche zwischen Fluid und Biofilm gilt nach dem Fickschen Gesetz die Kontinuitätsgleichung

$$D \frac{\partial c}{\partial z} \Big|_{z=L, \text{Biofilmseite}} = D^{(f)} \frac{\partial c}{\partial z} \Big|_{z=L, \text{Fluidseite}},$$

wobei  $D^{(f)}$  der Diffusionskoeffizient des gelösten Stoffes im Fluid und  $L$  die Dicke der Biofilmschicht ist. Da aufgrund des linearen Verlaufs der Konzentration in der Grenzschicht

$$\frac{\partial c}{\partial z} \Big|_{z=L, \text{Fluidseite}} = \frac{c^{(f)} - c|_{z=L}}{h}$$

gilt, erhält man schließlich als Randbedingung an der Fluid-Biofilm-Grenzfläche

$$D \frac{\partial c}{\partial z} \Big|_{z=L} = \frac{D^{(f)}}{h} (c^{(f)} - c|_{z=L}). \quad (2.10)$$

Es findet kein Nährstofftransport durch die Oberfläche des Aufwuchskörpers statt. Daher gelten dort homogene Neumann-Randbedingungen für die Stoffkonzentration:

$$\frac{\partial c}{\partial z} \Big|_{z=0} = 0. \quad (2.11)$$

Abhängig von der vorhandenen Nährstoffmenge nimmt die Dicke  $L$  des Biofilms zu, das heißt die obere Grenzfläche des Films bewegt sich in  $z$ -Richtung mit positiver Geschwindigkeit  $v_G = \frac{\partial L}{\partial t}$ , die durch den ersten Monod-Kinetik-Integranden bestimmt ist:

$$v_G = \frac{\partial L}{\partial t} = \int_0^L \left( \frac{Ykc}{K+c} - b_t - b_a \right) dz. \quad (2.12)$$

$Y$  bezeichnet dabei den Ertragskoeffizienten,  $k$  die Rate für den Nährstoffverbrauch durch die aktive Biomasse und  $b_t$  und  $b_a$  den Verlust an aktiver Biomasse durch Zelltod beziehungsweise Ablösung aufgrund von Scherkräften.

Die Lösung von Gleichung (2.8) für die Konzentration  $c$  des gelösten Stoffes mit den Randbedingungen (2.10) und (2.11) lässt sich mit Hilfe der Legendre-Polynome sechsten Grades approximieren ([18]).

Ein eindimensionales Biofilmmodell, das außerdem den Einfluss von Zellclustern auf den Transport gelöster Stoffe und die ökologische Selektion einzelner Mikrobenarten wiedergibt, wird in [70] vorgestellt. Die Struktur und Funktion der EPS wird in [38] modelliert.

### 2.1.5 Simulation realer Biofilme

In [36] wird die Simulation realer Biofilme mit eindimensionalen Modellen am Beispiel eines autotrophen Biofilmsystems gezeigt. Die meist verbreiteten Softwarepakete zur Durchführung solcher Rechnungen sind AQUASIM ([67]) und RIOVAL ([37]).

AQUASIM bietet neben Routinen zur numerischen Lösung eines konkreten Biofilmmodells der in 2.1.1 und 2.1.2 beschriebenen Form Funktionen zur automatischen Parameterkalibrierung und zur Unsicherheitsanalyse bezüglich der Parameterwahl.

Im Allgemeinen lassen sich die Lösungen der in 2.1.1 und 2.1.2 beschriebenen Modellgleichungen nicht mehr wie im Spezialfall in 2.1.4 mit Hilfe bekannter Funktionen angeben, sondern nur numerisch approximieren. Ein sehr effizienter numerischer Algorithmus („*h-p*-adaptives Verfahren“) zur numerischen Simulation bestimmter Biofilmtypen mit hoher Genauigkeit ist im Paket RIOVAL implementiert.

## 2.2 Dreidimensionale Modelle

Die Lösungen eindimensionaler Modellgleichungen sind relativ einfach und schnell zu berechnen. Dies gilt allerdings erst dann, wenn alle benötigten Parameter und Funktionen wie Volumenanteil der flüssigen Phase, Dichte der strukturbildenden Komponenten, Produktionsraten, Grenzflächentransferprozesse oder Geschwindigkeit der Fluid-Biofilm-Grenzfläche bekannt sind. Die experimentelle Bestimmung dieser Größen für jeden einzelnen Biofilm kann sehr zeitaufwändig (bis zu mehrere Monate) sein. Ferner wird der Einfluss der räumlichen Heterogenität der Biofilmstruktur weitgehend vernachlässigt. Es ist nicht klar, ob dieser beispielsweise im Hinblick auf die Diffusion gelöster Stoffe in der fluiden Phase (Gleichung 2.7) durch den Faktor  $f$  ausreichend beschrieben werden kann.

Dreidimensionale Modelle sind deutlich komplexer im Hinblick auf die numerische Berechenbarkeit der Gleichungen, können aber – insbesondere im Fall der mikroskopischen Modelle – die genannten Nachteile eindimensionaler Modelle weitgehend vermeiden. Bei der Wahl eines geeigneten dreidimensionalen Modells muss man sich allerdings von vornherein auf eine bestimmte räumliche Skala festlegen, da aufgrund des ohnehin hohen Rechen- und Speicheraufwands bei der numerischen Lösung dreidimensionaler partieller Differentialgleichungen nicht gleichzeitig ein ganzer Reaktor mit Ausmaßen in der Größenordnung von einigen Metern und der Einfluss einer einzelnen Mikrobe mit einem Durchmesser von einem Mikrometer simuliert werden können. Als Beispiel betrachte man einen Reaktor der Größe  $10\text{ m} \times 5\text{ m} \times 3\text{ m}$ . Will man eine einzelne Mikrobe von einem Mikrometer Durchmesser gerade noch darstellen, benötigt man  $15 \cdot 10^{19}$  Gitterpunkte, für jede zu berechnende physikalische Größe (vom Typ `double`) also  $1,2 \cdot 10^{12}$  Gigabyte Speicherplatz. Nimmt man an, dass der Wert der Unbekannten an jedem Gitterpunkt mit nur einer elementaren Rechenoperation bestimmt werden kann und eine Rechenleistung von 1,3 Teraflops (Maximalleistung des Bundeshöchstleistungsrechners Hitachi *SR8000-F1* des Leibniz-Rechenzentrums München) erreicht wird, so beträgt die Rechenzeit etwa 3,6 Jahre. Daher werden getrennte Modelle für die Makroskala (siehe 2.2.1) und die Mikroskala (siehe 2.2.2) verwendet. Um auch auf der Makroskala von den Ergebnissen auf der Mikroskala profitieren zu können, müssen beide Ansätze durch eine geeignete externe Kopplung miteinander verbunden werden (siehe Anhang B.2).

Neben echt dreidimensionalen Modellen werden auch zweidimensionale Modelle (siehe zum Beispiel [33]) verwendet, die von der Annahme der Rotationssymmetrie des Biofilms beziehungsweise des Reaktors um eine bestimmte Achse ausgehen. Die teilweise erhebliche

räumliche Heterogenität der Biofilmgeometrie sowie dreidimensionale Strömungs- und Transporteffekte (wie zum Beispiel Turbulenz) können mit diesen Modellen jedoch nicht erfasst werden.

### 2.2.1 Makroskopisches Modell

Makroskopische Modelle beschreiben Prozesse auf der Ebene eines größeren abgeschlossenen Systems, zum Beispiel eines natürlichen Gewässers oder eines Biofilmreaktors. Hier sollen nur Modelle zur Abbildung der Vorgänge in Festbettreaktoren, gefüllt mit polydispersen, kugelförmigen Trägerpartikeln (Abbildungen 1.1 und 1.2) betrachtet werden, wie sie bevorzugt in der Abwasserreinigung eingesetzt werden. Die Größe solcher Anlagen variiert stark, je nachdem, ob es sich um einen Laborreaktor im kleinen Maßstab (Durchmesser ca. 0,1 m, größenordnungsmäßig 10.000 Trägerpartikel) oder um eine reale Abwasser-Aufbereitungsanlage (Durchmesser mehrere Meter, Millionen Trägerpartikel) handelt. Die Frage der Übertragbarkeit von Untersuchungen am Laborreaktor auf Praxisanlagen kann experimentell aufgrund der schwierigen Messbedingungen an Anlagen im Praxiseinsatz nicht befriedigend beantwortet werden. Numerische Simulationen der vorgestellten Modelle können hier wertvolle Hinweise auf elementare Unterschiede zwischen Reaktoren unterschiedlicher Abmessungen liefern.

Nach [25] ist bekannt, dass die Strömungsverhältnisse im Festbettreaktor den Nährstofftransport, das Biofilmwachstum, die hydraulische Verweilzeit und damit die Abbauleistung des Reaktors wesentlich beeinflussen. Innerhalb des Reaktors weist die Geschwindigkeitsverteilung starke lokale Inhomogenitäten auf, die experimentell nur sehr schwer erfassbar sind, da Messinstrumente kaum so platziert werden können, dass die Strömung nicht gestört wird, und gleichzeitig die interessanten Messpunkte im Reaktor empirisch nicht leicht zu bestimmen sind. Nur mit Hilfe der numerischen Simulation eines geeigneten Modells ist eine störungsfreie und vollständige Abbildung der Strömungsgeschwindigkeiten im Reaktor möglich. Die Fluidströmung ist daher zentraler Bestandteil der meisten Modelle. Nimmt man an, dass der Transport gelöster Stoffe oder kleiner suspendierter Partikel im Fluid schlupffrei ist, das heißt keine Trägheits- und Sedimentationseffekte auftreten, so ist er durch die Strömungsgeschwindigkeiten festgelegt ([21]). Auf eine explizite Modellierung der Transportvorgänge wird daher oft verzichtet. Ebenso werden chemische und biologische Vorgänge im Biofilm, die sich auf einer wesentlich kleineren räumlichen Skala abspielen, sowie Ortsänderungen der Aufwuchskörper nicht modelliert. Das Wachstum des Biofilms wird lediglich über eine Variation des Durchmessers der als sphärisch angenommenen Aufwuchskörper berücksichtigt. Diese Vereinfachung lässt sich dadurch rechtfertigen, dass die innere Zeit des Impulsaustausches wesentlich kleiner ist als die typische Zeit für das Biofilmwachstum und die Geometrie somit bezüglich der Zeitskala der Strömung als quasistationär betrachtet werden kann. Aufgrund der sehr geringen mittleren Strömungsgeschwindigkeiten im Reaktor (wenige Millimeter pro Sekunde) treten nur laminare und stabile Strömungen auf ([22]).

## Modellierung der Geometrie

Da die Geometrie eines realen Festbettreaktors je nach seinen Ausmaßen und seiner Anwendung nicht oder nur mit nicht vertretbarem Aufwand vermessen werden kann und die so definierte Geometrie nicht als repräsentativ für beliebige Reaktoren angenommen werden kann, wird die Geometrie computergestützt mit Hilfe eines Modells für Kugelschüttungen ([55, 69]) erzeugt. Durch dieses Vorgehen können beliebige Schüttungen mit unterschiedlichen Kugeldurchmessern erzeugt werden.

Ein vorgegebener Raum wird nach und nach mit einzelnen Partikeln gefüllt. Dazu wird für ein „fallendes“ Partikel nach jeder inkrementellen Änderung der axialen Position der Abstand zu allen Nachbarpartikeln und den Gefäßwänden abgefragt. Sobald ein Partikel drei Berührungspunkte mit Nachbarpartikeln oder Wänden oder den Boden des Gefäßes erreicht, ist seine Position stabil und wird ab diesem Zeitpunkt nicht mehr geändert. Der Durchmesser und die Startposition der einzelnen Partikel werden durch einen Zufallsgenerator bestimmt. Je nach Wahl der Verteilung für die Durchmesser können so weitgehend reale Schüttungsgeometrien mit beliebiger Partikelgrößenverteilung und Partikelzahl erzeugt werden. Je unterschiedlicher die Partikeldurchmesser sind, desto besser können Zwischenräume ausgefüllt werden, die bei konstanter Partikelgröße vor allem an den Wänden des Reaktors entstehen. Der Einfluss der Reaktorwände auf die lokale Porosität wird somit deutlich reduziert.

Insgesamt weisen die künstlich erzeugten Kugelschüttungen jedoch immer eine im Vergleich zu realen Reaktoren zu hohe Porosität auf, da das Setzen der Schüttung durch mechanische Belastungen wie beispielsweise Erschütterungen nicht berücksichtigt wird.

## Strömungsmodell

Zur Modellierung der Fluidströmung werden zwei grundsätzliche Ansätze verfolgt: Die Navier-Stokes-Gleichungen (2.13) und (2.14) und die Lattice-Boltzmann-Gleichung (2.17). Unter gewissen Voraussetzungen liefern beide Ansätze äquivalente Ergebnisse.

Die inkompressiblen Navier-Stokes-Gleichungen

$$\frac{\partial \vec{u}}{\partial t} + (\vec{u} \cdot \nabla) \vec{u} = \frac{1}{Re} \Delta \vec{u} - \nabla p + \vec{g}, \quad (2.13)$$

$$0 = \nabla^T \vec{u} \quad (2.14)$$

für die Fließgeschwindigkeit  $\vec{u}$  und den Druck  $p$  ergeben sich aus den Erhaltungsgleichungen der Kontinuumsmechanik.  $\vec{g}$  bezeichnet äußere Kräfte (wie zum Beispiel die Schwerkraft) und  $Re$  die Reynoldszahl. Gleichung (2.13) beinhaltet Konvektion  $((\vec{u} \cdot \nabla) \vec{u})$  und Diffusion  $(\frac{1}{Re} \Delta \vec{u})$  und folgt aus der Erhaltung des Impulses, Gleichung (2.14) stellt die Divergenzfreiheit des Geschwindigkeitsfelds und damit die Massenerhaltung sicher.

Der Lattice-Boltzmann-Ansatz hat seine Wurzeln in der statistischen Physik und ist aus der Theorie der idealen Gase entstanden. Die zugrunde liegende Boltzmann-Gleichung

$$\frac{\partial f(t, \xi, x)}{\partial t} + \xi \frac{\partial f(t, \xi, x)}{\partial x} = \Omega(t, \xi, x) \quad (2.15)$$

beschreibt das Verhalten eines Fluids auf der Ebene ideeller mikroskopischer Fluid-Partikel, wobei die kontinuierliche Partikelverteilungsfunktion  $f(t, \xi, x)$  die Wahrscheinlichkeit angibt, dass sich zum Zeitpunkt  $t$  ein Partikel mit der Geschwindigkeit  $\xi$  am Ort  $x$  befindet. Die linke Seite der Gleichung beschreibt die Advektion der Teilchen mit der Geschwindigkeit  $\xi$ , die rechte Seite mit Hilfe des Kollisionsoperators  $\Omega$  die Interaktion der Partikel.

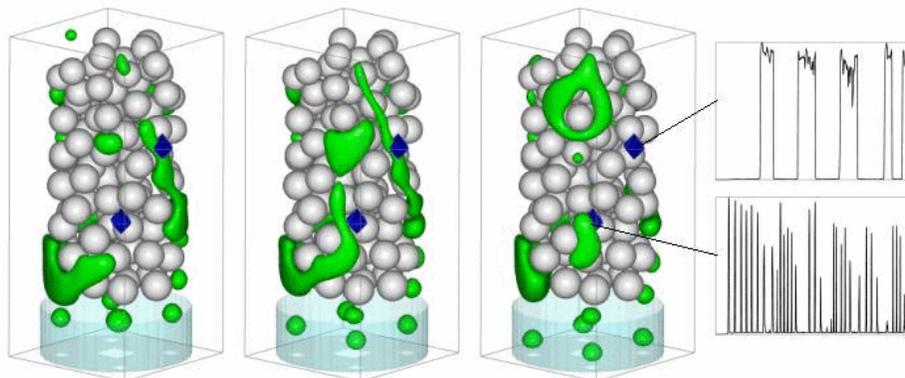
Bei geeigneter Wahl des Kollisionsoperators  $\Omega$  ergeben sich Dichte und Impuls als Momente der Teilchenaufenthaltswahrscheinlichkeiten  $f$  und sind identisch mit der zugehörigen Lösung der inkompressiblen Navier-Stokes-Gleichungen ([14]):

$$\rho(t, x) = \int f(t, \xi, x) d\xi, \quad \rho \vec{u} = \int \xi f(t, \xi, x) d\xi. \quad (2.16)$$

Die Lattice-Boltzmann-Gleichung (2.17) entsteht aus der Diskretisierung von Gleichung (2.15). Der Geschwindigkeitsraum wird dabei durch eine Menge  $\{\xi_i, i = 0, \dots, a\}$  von diskreten Geschwindigkeitsvektoren ersetzt. In Raum und Zeit wird die Gleichung durch ein explizites Finite-Differenzen-Schema auf dem kartesischen Gitter diskretisiert, das von  $\{dt \cdot \xi_i, i = 0, \dots, a\}$  aufgespannt wird, wobei  $dt$  die gewählte Zeitschrittweite ist.

$$f_i(t + dt, x + \xi dt) = f_i(t, x) + \Omega_i(t, x), \quad i = 1, \dots, a. \quad (2.17)$$

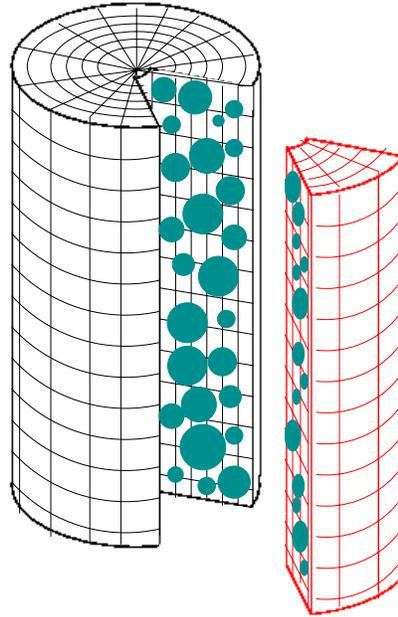
Eine Zweiphasenströmung zur Modellierung der Durchlüftung des Reaktors lässt sich analog zu (2.15) und (2.17) durch zwei getrennte Partikelverteilungsfunktionen  $f^1$  und  $f^2$  für die beiden Phasen und eine geeignete Erweiterung des Kollisionsoperators  $\Omega$  ([30, 66]) realisieren. Abbildung 2.5 zeigt Ergebnisse einer numerischen Simulation des Zweiphasenmodells für einen Modellreaktor.



**Abbildung 2.5:** Zwei-Phasen-Strömung zu drei Zeitpunkten und zeitabhängige Sauerstoffkonzentration an zwei verschiedenen Stellen eines Modellreaktors (oben: in Wandnähe, unten: im Inneren des Reaktors, Quelle: J. Tölke und M. Krafczyk, Lehrstuhl für Bauinformatik, Fakultät für Bauingenieur- und Vermessungswesen, TU München, 2000)

Betrachtet man sowohl reale als auch simulierte Kugelschüttungen, so lässt sich in beiden Fällen ein rotationssymmetrischer Aufbau der Schüttung im zylinderförmigen Reaktor feststellen. Der Reaktor kann daher in repräsentative Segmente zerlegt werden (Abbildung 2.6). Bei geeigneter Wahl des Ausschnittswinkels kann eines dieser Segmente als exemplarisches

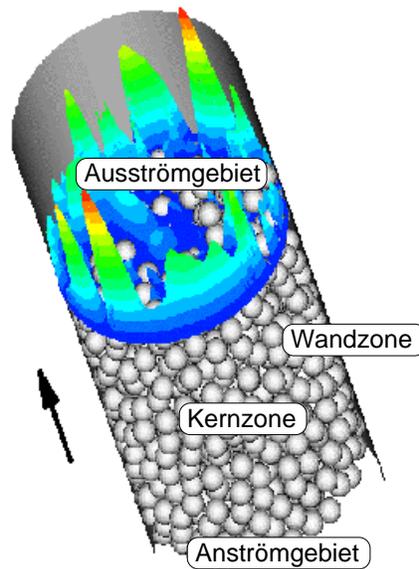
Modell für jedes beliebige Segment gleicher Größe verwendet werden. An der Außenwand des Reaktors werden dabei Haftbedingungen, an den Segmentschnittflächen Rutschbedingungen angenommen. Bei Partikeln, die sich nur teilweise im Segment befinden, wird auch nur dieser Teil berücksichtigt ([21]). Durch die resultierende Verkleinerung des modellierten Gebiets verringert sich die Rechenzeit numerischer Simulationen erheblich.



**Abbildung 2.6:** Segmentierung des zylinderförmigen Festbettreaktors zur Reduktion des Rechenaufwands bei der numerischen Simulation (aus [21])

Ferner zerfällt der gesamte Reaktor bezüglich Geschwindigkeitsverteilung, Nährstofftransport, hydraulischer Verweilzeit, mechanischer Belastung der Biomaterie und Biofilmwachstum in vier charakteristische Zonen: Anström- und Abströmgebiet sowie Wand- und Kernzone (vgl. Abbildung 2.7 und [21, 20, 72]). Das Anströmgebiet hat eine Dicke von zwei bis drei Partikeldurchmessern und ist durch den Transport des Fluids in den Wandbereich mit seiner im Vergleich zum Kern des Reaktors höheren Porosität gekennzeichnet. Es treten hohe radiale Geschwindigkeiten, ein starker Druckabfall, höhere Porosität und weniger oder gar keine Toträume auf. Die Biofilme sind daher in diesem Bereich relativ dünn. Im Ausströmgebiet wird das Fluid umgekehrt vom Wandbereich wieder in die Kernzone des Reaktors transportiert. Da die Nährstoffmenge hier bereits stark reduziert ist, sind die Biofilme noch dünner als im Anströmgebiet. Die Wandzone besitzt eine Ausdehnung von etwa drei Partikeldurchmessern und weist eine im Vergleich zur Kernzone deutlich erhöhte Porosität auf, wodurch die Geschwindigkeiten im Vergleich zur mittleren Geschwindigkeit im Reaktor um das bis zu Zehnfache erhöht sind ([21]). Die Wandzone hat aufgrund des unterschiedlichen Anteils am Gesamtvolumen beim Laborreaktor eine wesentlich größere Bedeutung als bei Praxisanlagen. Auf der Grundlage dieser Zonierung kann der gesamte Reaktor bis auf den Anström- und Ausströmbereich als Summe gleichartiger, repräsentativer Schichten angenommen werden. Dadurch lassen sich der Zeit- und Speicheraufwand für die numerische Simulation weiter

reduzieren.



**Abbildung 2.7:** Zonierung eines Modellreaktors mit axialer Geschwindigkeitsverteilung (deutlich zu erkennen: höhere Porosität in Wandnähe, aus [21])

## 2.2.2 Mikroskopisches Modell

Auf der mikroskopischen Ebene – typischerweise werden Gebiete mit wenigen hundert Mikrometern Durchmesser betrachtet – können im Unterschied zur makroskopischen Ebene auch Effekte auf der räumlichen Skala der einzelnen Mikroben berücksichtigt werden. Neben der dreidimensionalen Strömung werden der Transport gelöster Stoffe im Fluid und im Biofilm selbst (konvektiv-diffusiv im Fluid, diffusiv-reaktiv im Biofilm) abgebildet. Ferner wird auch das an das lokale Nährstoffangebot gekoppelte Wachstum des Biofilms explizit modelliert.

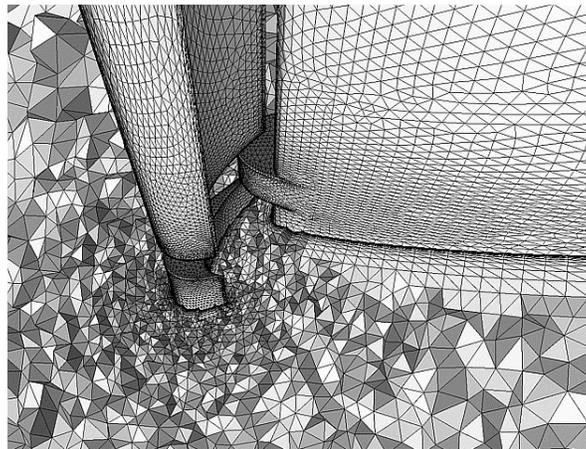
### Modellierung der Geometrie

Anders als im Fall makroskopischer Modelle kann die mikroskopische Geometrie mit Hilfe der confokalen Laser-Scanning-Mikroskopie (CLSM) experimentell bestimmt werden (vgl. Kapitel 1). Die Oberflächenkrümmung der sphärischen Aufwuchskörper eines Festbettreaktors kann auf dieser Skala vernachlässigt werden. Daher können ohne Einschränkung der Relevanz der Ergebnisse Biofilme auf ebenen Aufwuchskörpern im Fließkanal (siehe 1.1) untersucht werden. Es werden ausschließlich Mono- oder Zwei-Kultur-Biofilme betrachtet. Komplexere oder gar reale Biofilme aus Reaktoren sind experimentell nur schwer vollständig zu erfassen (vgl. Kapitel 1) und aufgrund mangelnden Wissens über einen Großteil der auftretenden Prozesse und ihrer Wechselwirkungen untereinander zum aktuellen Zeitpunkt noch nicht auf einer sicheren Grundlage modellierbar. Dies schränkt die derzeitigen Anwendungsmöglichkeiten der

dreidimensionalen Simulation von Biofilmen auf der Mikroskala ein. Verschiedene qualitative Aussagen zur Vertiefung des Verständnisses der Vorgänge in Biofilmsystemen können dennoch bereits aus den Simulationsergebnissen abgeleitet werden (siehe die Zusammenfassung).

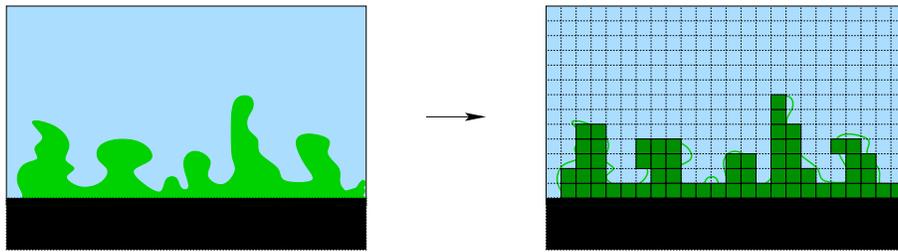
Zur Erzeugung einer Modellgeometrie müssen die CLSM-Schnittbilder in eine diskrete Geometrie übersetzt werden. Die Art der Beschreibung der Modellgeometrie muss dabei bereits der Modellierung weiterer Prozesse und den numerischen Lösungsverfahren angepasst sein. Da Biofilmgeometrien sehr kompliziert sein können, ist eine Beschreibung der Oberfläche als zweidimensionale Funktion über der  $x, y$ -Ebene kaum möglich. Diskrete Geometriebeschreibungen eignen sich in diesem Fall wesentlich besser. Dabei stehen zwei grundsätzliche Ansätze zur Auswahl:

- Unstrukturierte Gitter approximieren die Geometrie mit Hilfe kleiner Elemente eines speziellen Typs (beispielsweise Tetraeder oder Quader), die jedoch unterschiedlich groß und von unterschiedlichem Seitenverhältnis sein können (Abbildung 2.8). Solche Gitter lassen sich sehr flexibel zur genauen Beschreibung beliebiger Geometrien einsetzen. Ihr Nachteil besteht jedoch im hohen Speicher- und Organisationsaufwand, der sich aus dem Fehlen einer systematischen Ordnung in der Struktur des Gitters ergibt.
- Strukturierte Gitter bestehen dagegen aus Elementen eines bestimmten Typs und fester Form (Abbildung 2.9). Aufgrund der festen Struktur des Gitters ist die Handhabung – beispielsweise mit Hilfe von Matrizen oder Oktalbäumen ([9, 73]) – sehr effizient. Insbesondere sind Änderungen der Geometrie, wie sie durch das Wachstum des Biofilms verursacht werden, einfach zu realisieren. Strukturiertheit des Gitters bedeutet jedoch nicht, dass alle Gitterzellen gleich groß sein müssen. Lokale Verfeinerungen, zum Beispiel durch Unterteilung eines Quaders in acht gleichgroße Unterquader, sind problemlos möglich, ohne die Struktur des Gitters zu zerstören (Abbildung 2.10).

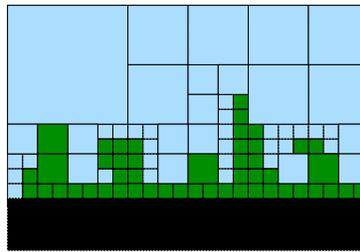


**Abbildung 2.8:** Unstrukturiertes Volumen- und Oberflächengitter für einen Ausschnitt einer Flugzeugtragfläche (aus [91])

Die CLSM-Daten beschreiben die Biofilmgeometrie bereits in diskreter Form (Graustufen-Schnittbilder von diskreten confokalen Ebenen mit diskreten Bildpunkten). Aus der enthaltenen Information lässt sich daher sehr leicht ein strukturiertes Fluid-Biofilm-Quadergitter generieren. Daher und um Geometrieänderungen durch Biofilmwachstum einfach und effizient



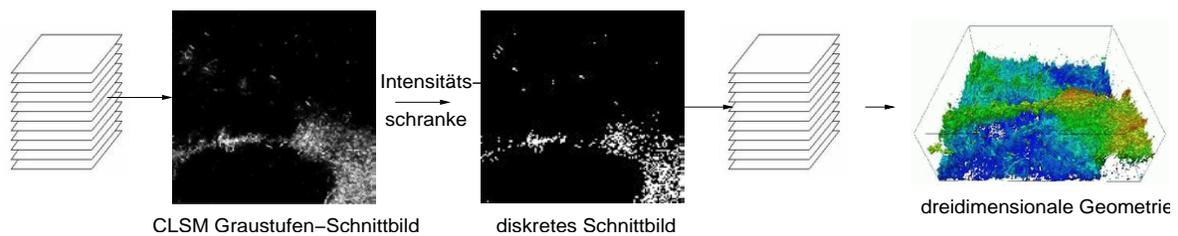
**Abbildung 2.9:** Approximation eines zweidimensionalen Biofilm-Fluid-Gebiets mit Hilfe eines regulären strukturierten Quadratgitters



**Abbildung 2.10:** Approximation eines zweidimensionalen Biofilm-Fluid-Gebiets mit Hilfe eines adaptiven strukturierten Quadratgitters

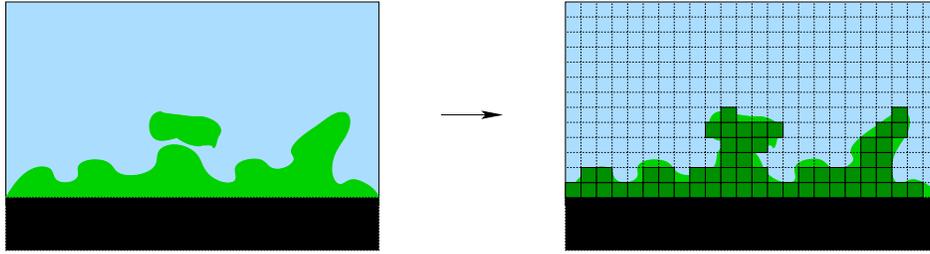
realisieren zu können, werden in dieser Arbeit strukturierte Gitter zur Geometriebeschreibung verwendet. Diese eignen sich auch ideal zur numerischen Simulation der Modellgleichungen des mikroskopischen Modells. Entspricht die minimale Quadergröße gerade der Bildauflösung des CLSM, verliert man bei der Modellierung der Biofilmgeometrie mit strukturierten Quadergittern keine weiteren Informationen.

Um eine dreidimensionale Quadergeometrie aus den Graustufen-Schnittbildern des CLSM zu konstruieren, wird eine empirische Intensitätsschranke festgelegt. Je nachdem, ob ein Bildpunkt unter- oder oberhalb dieser Intensitätsschranke liegt, wird er dem Fluid oder dem Biofilm zugeordnet. Die entstehenden diskreten Schnittbilder werden unter Berücksichtigung ihres vertikalen Abstands zu einem dreidimensionalen Modellbiofilm zusammengesetzt, wobei aus jedem Bildpunkt der diskreten Schnittbilder ein Quader der resultierenden Geometrie entsteht (Abbildung 2.11). Obwohl die Auflösung des CLSM im Bereich des Durchmessers



**Abbildung 2.11:** Erzeugung einer dreidimensionalen Biofilmgeometrie aus zweidimensionalen Graustufen-Schnittbildern

der einzelnen Mikroben liegt und somit nahezu alle Mikroben auf dem CLSM-Bild erfasst werden, kann es bei dieser Art der Definition einer experimentell bestimmten Biofilmgeometrie zu Verfälschungen der Topologie des tatsächlichen Fluid-Hindernis-Gebiets kommen. Kanäle mit geringem Durchmesser werden bei einem hohen Flächenanteil der Mikroben an den beteiligten Bildpunkten nicht erkannt (siehe Abbildung 2.12). Da die Bildauflösung in  $z$ -Richtung im Allgemeinen geringer ist als in  $x$ - und  $y$ -Richtung, gilt dies vor allem für horizontal verlaufende Poren und Kanäle. Die durch das 0-1-Feld gegebene Geometrie darf daher nicht als topologisch genaue Abbildung der tatsächlichen Geometrie des betrachteten Biofilms interpretiert werden. Sie stellt lediglich eine im statistischen Sinne einer natürlichen Biofilmgeometrie ähnliche Struktur dar.



**Abbildung 2.12:** Schematisches Beispiel für eine topologische Verfälschung der tatsächlichen Biofilmgeometrie bei der Erzeugung eines geometriebeschreibenden 0-1-Feldes

### Kontinuierliches Strömungs- und Transportmodell

Die Fluidströmung in der Modellgeometrie wird wie im makroskopischen Modell (siehe 2.2.1) beschrieben durch die dreidimensionalen inkompressiblen Navier-Stokes-Gleichungen

$$\frac{\partial \vec{u}}{\partial t} + (\vec{u} \cdot \nabla) \vec{u} = \frac{1}{Re} \Delta \vec{u} - \nabla p + \vec{g}, \quad (2.18)$$

$$0 = \nabla^T \vec{u} \quad (2.19)$$

für die Fließgeschwindigkeit  $\vec{u}$  und den Druck  $p$ . Die Reynoldszahl  $Re$  ist typischerweise sehr klein (ca. 0,1 bis 5,0), die Strömung somit immer laminar und stabil. An der Grenzfläche zwischen Biofilm und Fluid werden Haftbedingungen ( $\vec{u} = 0$ ) angenommen.

Alternativ könnte auch hier die Lattice-Boltzmann-Gleichung (2.17) verwendet werden ([52, 61, 24]).

Die Konzentration  $c$  eines Nähr- oder Schadstoffs wird im Fluidgebiet  $\Omega^{(f)}$  durch eine Konvektions-Diffusions-Gleichung beschrieben. Der Biofilmbereich selbst wurde mittels CLSM mit ausreichend hoher Auflösung bestimmt, um annehmen zu können, dass dieser Bereich für die Fluidströmung ein im Wesentlichen dichtes Hindernis darstellt. Die Stoffkonzentration  $c$  wird daher dort durch eine Diffusions-Reaktions-Gleichung festgelegt. An der Grenzfläche  $\Gamma$  von  $\Omega^{(f)}$  und  $\Omega^{(b)}$  gelten Randbedingungen gemäß dem Fickschen Gesetz ([24]):

$$\frac{\partial c}{\partial t} + (\vec{u} \cdot \nabla) c = D^{(f)} \Delta c \quad \text{in } \Omega^{(f)}, \quad (2.20)$$

$$\frac{\partial c}{\partial t} = D^{(b)} \Delta c - r(c) \quad \text{in } \Omega^{(b)}, \quad (2.21)$$

$$D^{(f)} \frac{\partial c}{\partial \vec{n}} \Big|_{\Omega^{(f)}} = D^{(b)} \frac{\partial c}{\partial \vec{n}} \Big|_{\Omega^{(b)}} \quad \text{auf } \Gamma. \quad (2.22)$$

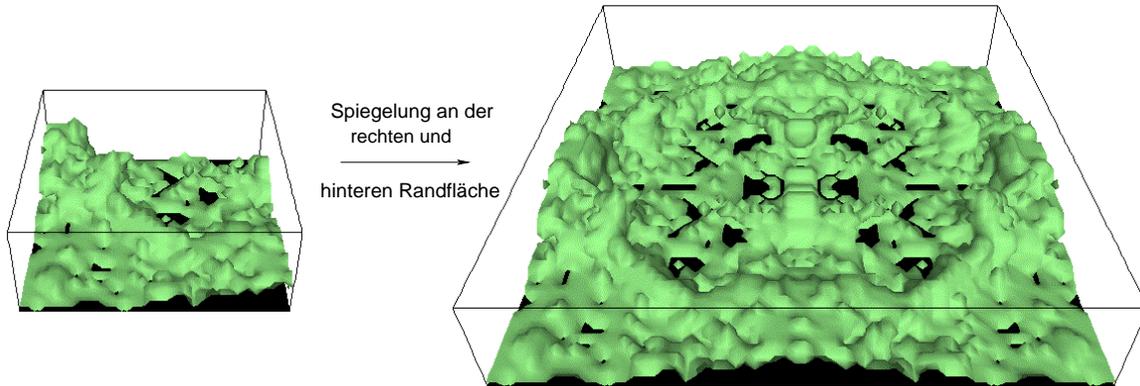
In Gleichung (2.22) ist  $\frac{\partial c}{\partial \vec{n}} \Big|_{\Omega^{(f)}}$  die erste Normalableitung der Stoffkonzentration  $c$  bezüglich der Grenzfläche  $\Gamma$  auf der Fluid-Seite von  $\Gamma$ .  $\frac{\partial c}{\partial \vec{n}} \Big|_{\Omega^{(b)}}$  ist analog dazu die erste Normalableitung von  $c$  auf der Biofilm-Seite von  $\Gamma$ .  $D^{(f)}$  und  $D^{(b)}$  sind die Diffusionskoeffizienten im Fluid und im Biofilm.  $r$  bezeichnet den Reaktionsterm innerhalb des Biofilms. Gleichung (2.22) impliziert eine Unstetigkeit der Normalableitung von  $c$  an der Grenzfläche  $\Gamma$  zwischen Biofilm und Fluid falls  $D^{(f)} \neq D^{(b)}$ .

Für den Reaktionsterm  $r(c)$  wird wie im eindimensionalen Modell von Chang (siehe Gleichung (2.8) und [18]) Monod-Kinetik angenommen:

$$r(c) = \frac{qc}{K + c} \rho_b, \quad (2.23)$$

mit der maximalen Nährstoffverbrauchsrate  $q$  und der maximalen Sättigungskonzentration  $K$ . Die Dichte  $\rho_b$  der aktiven Biomasse wird als konstant im gesamten Biofilm angesehen. Diese Annahme wird dadurch gerechtfertigt, dass die Auflösung des CLSM hoch genug ist, um die wesentlichen Hohlräume und Kanäle im Biofilm, die die Dichte der Biomasse reduzieren könnten, zu erfassen und damit bei der Geometriegenerierung explizit dem Fluidgebiet zuordnen zu können.

Bildet die verwendete Biofilmgeometrie ein repräsentatives Elementarvolumen (siehe 1.6), so kann man für Strömung und Transport an den Grenzen des Gebiets in  $x$ - und  $y$ -Richtung periodische Randbedingungen annehmen. Hierfür muss jedoch auch die Geometrie periodisch sein. Dies wird durch eine Spiegelung der Geometrie in  $x$ - und  $y$ -Richtung erreicht (siehe Abbildung 2.13).



**Abbildung 2.13:** Spiegelung eines Modellbiofilms in  $x$ - und  $y$ -Richtung zur Erzeugung einer periodischen Geometrie

Die Stärke der Fluidströmung wird entweder durch die Vorgabe einer Geschwindigkeits-Randbedingung  $\vec{u} \equiv u_\infty$  am oberen Rand des Rechengebietes oder durch eine Druckdifferenz in Hauptströmungsrichtung (zum Beispiel  $p \equiv p_w$  am linken Gebietsrand und  $p \equiv p_o$  am rechten Gebietsrand mit  $p_w > p_o$ ) bestimmt. Die erste Bedingung ist realistisch, falls die

obere Gebietsgrenze ausreichend weit vom Biofilm entfernt ist und damit außerhalb des Einflussbereichs der Biofilmheterogenität auf die Strömung liegt. Dasselbe gilt für die konstante Dirichlet-Randbedingung  $c \equiv c_\infty$  für den Stofftransport am oberen Gebietsrand.

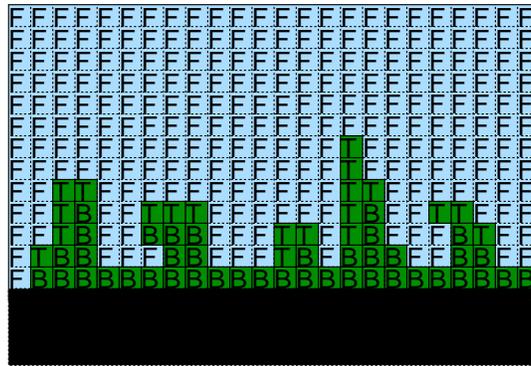
### Diskretes Wachstumsmodell

Das Wachstum des Biofilms auf der Mikroskala wird im Gegensatz zu Strömung und Stofftransport nicht auf der Grundlage kontinuierlicher Erhaltungsgleichungen, sondern mit Hilfe eines stochastischen zellulären Automaten modelliert. In ein solches regelbasiertes Modell lassen sich Erfahrungswerte und intuitive Vorstellungen von den biologischen Vorgängen, die nicht wie Strömung und Stofftransport durch bekannte physikalische und chemische Gesetzmäßigkeiten bestimmt sind, sehr leicht integrieren. Ferner können die Wachstumsgesetze, die noch nicht vollständig bekannt sind, jederzeit geändert werden, ohne dass wie bei gleichungsbasierten Modellen neue Simulationswerkzeuge erforderlich werden. In jüngster Zeit wurden jedoch auch erste kontinuierliche Modelle (nichtlineare Massendiffusion) entwickelt ([23]). Zelluläre Automaten sind im Gegensatz dazu zeitlich und räumlich diskrete Modelle, da sie zu diskreten Zeitpunkten auf einer endlichen Anzahl von Zellen operieren.

Ein zellulärer Automat ist definiert durch das Gitter, auf dem er arbeitet, eine endliche Menge von möglichen Zuständen der einzelnen Gitterzellen und Regeln zum Zustandswechsel der Zellen in Abhängigkeit von den Zuständen der Nachbarzellen. Zur Modellierung biologischer Systeme werden zelluläre Automaten gerne verwendet, da sie eine intuitive Modellierung biologischer Vorgänge ermöglichen: Biologische Systeme lassen sich häufig auf einfache Weise auf ein Gitter abbilden (zum Beispiel durch die Zuordnung einer Gitterzelle zu jedem beteiligten Organismus). Die Eigenschaften des Gesamtsystems ergeben sich meist aus lokalen Wechselwirkungen der beteiligten Organismen. Diese Wechselwirkungen lassen sich oft sehr einfach in empirisch definierte Regeln des Automaten umsetzen.

In dieser Arbeit basiert der zelluläre Automat auf dem bei der Modellierung der Biofilmgeometrie erzeugten regelmäßigen strukturierten Gitter. Da die Auflösung des CLSM etwa im Bereich des Durchmessers einer einzelnen Bakterienzelle liegt, orientiert sich die Definition möglicher Zustände und der Regeln stark am angenommenen Verhalten einer einzelnen Mikrobe: Die Zellen des Automaten können die drei Grundzustände „Fluid“ (F), „nicht teilungsbereite Biofilmzelle“ (B) und „teilungsbereite Biofilmzelle“ (T) annehmen (Abbildung 2.14). T-Zellen teilen sich in einem Automatenlauf in zwei B-Zellen. Die dabei neu entstehende Zelle besetzt entweder eine bisherige Fluidzelle oder verschiebt eine benachbarte Biofilmzelle. Die Teilungsrichtung wird stochastisch bestimmt, wobei im Falle von Zellen mit unmittelbar benachbarten Fluidzellen immer eine dieser Zellen gewählt wird, im Falle innerer Zellen die Richtung, in der sich die nächste Fluidzelle befindet (Weg des geringsten Widerstands) bevorzugt wird.

Die Regeln eines zellulären Automaten sind streng lokal, das heißt bei Besuch einer Zelle kann durch Anwendung einer Regel nur der Zustand dieser Zelle, nicht der der Nachbarzellen, geändert werden. Durch diese Einschränkung ist das Ergebnis eines Automatenlaufs über alle Zellen unabhängig von der gewählten Durchlaufreihenfolge. Die Zellteilung kann daher nicht in einem Schritt, sondern nur iterativ über geeignete Übergangszustände realisiert werden:



**Abbildung 2.14:** Zweidimensionales Automatengitter für einen Biofilm mit Berechnungszellen in den Grundzuständen F („Fluid“), B („nicht teilungsbereite Biofilmzelle“) und T („teilungsbereite Biofilmzelle“)

- 1) Im ersten Schritt wird die gewünschte Teilungsrichtung jeder T-Zelle festgelegt. Mögliche Richtungen sind dabei Rechts, Links, Oben und Unten im zweidimensionalen beziehungsweise Rechts, Links, Vorne, Hinten, Oben und Unten im dreidimensionalen Fall. Je nachdem, ob die T-Zelle in einer dieser Richtungen eine Fluid-Nachbarzelle besitzt oder nicht, wird die Teilungsrichtung auf unterschiedliche Weise gewählt. Gibt es eine oder mehrere Fluid-Nachbarzellen, so wird eine von diesen zufällig ausgewählt (gleichverteilt).

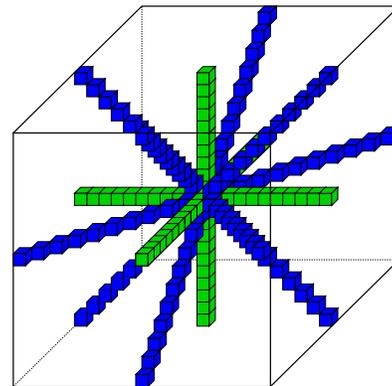
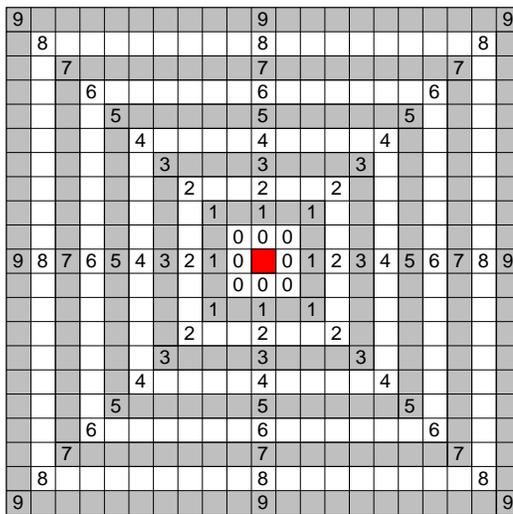
Gibt es keine benachbarte Fluidzelle, so wird eine der direkt benachbarten B-Zellen ausgewählt. Da sich T-Zellen im Inneren des Biofilms in Richtung des geringsten Widerstands teilen, wird diese Zelle jedoch nicht zufällig aus den möglichen Kandidaten festgelegt, sondern erst nach einer Suche der nächstgelegenen Fluid-Zelle in einer erweiterten Nachbarschaft der T-Zelle bestimmt. Diese Suche läuft über zehn Schichten von der aktuellen T-Zelle aus nach außen, wobei aus Effizienzgründen im Hinblick auf die Simulation des Automatenmodells nur Nachbarzellen in den Richtungen Rechts, Links, Vorne, Hinten, Oben, Unten (bzw. Rechts, Links, Oben, Unten im zweidimensionalen Fall) und in zwölf (bzw. vier im Zweidimensionalen) Diagonalen berücksichtigt werden (siehe Abbildung 2.15), in denen direkt neben der teilungsbereiten Zelle eine B-Zelle liegt. Es ist also keine Teilung in Richtung einer weiteren teilungsbereiten Zelle möglich. Befindet sich die nächstgelegene Fluidzelle in einer der sechs (vier) räumlichen Hauptrichtungen, so wird als Zielzelle die in dieser Richtung liegende B-Nachbarzelle festgesetzt. Liegt die nächste Fluidzelle auf einer der Diagonalen, so wählt man als Zielzelle mit jeweils 50 Prozent Wahrscheinlichkeit eine der beiden in den benachbarten Hauptrichtungen gelegenen B-Zellen zufällig aus (Abbildung 2.16). Wird in keiner Richtung eine Fluidzelle gefunden, wird eine der unmittelbar benachbarten B-Zellen zufällig ausgewählt.

Existiert in der unmittelbaren Nachbarschaft der T-Zelle weder eine Fluid- noch eine B-Zelle, wird keine Richtung gewählt. Die entsprechende Zelle wird in späteren Iterationen des Automaten erneut besucht.

Jede T-Zelle, für die eine Teilungsrichtung gefunden werden konnte, wird in einen

Übergangszustand versetzt, der die jeweils gewünschte Teilungsrichtung angibt.

- 2) Nach Schritt 1 kann der Fall eintreten, dass zwei T-Zellen dieselbe Fluid- oder B-Zelle als Teilungsziel ausgewählt haben. Solche Kollisionsfälle müssen daher vor der endgültigen Teilung der Zellen in einem weiteren Zwischenschritt beseitigt werden. Hierbei ändern sich jedoch nicht mehr die Zustände der zu teilenden Zellen, sondern die der Zielzellen der teilungsbereiten Zellen. In jeder Fluid- oder B-Zelle, in deren unmittelbarer Nachbarschaft sich eine oder mehrere teilungsbereite Zellen befinden, die sich in Richtung der aktuellen Zelle teilen wollen, wird in einen Übergangszustand versetzt, der angibt, welcher der entsprechenden T-Zellen die gewünschte Teilung erlaubt wird. Die Unterscheidung zwischen Fluid- und B-Zellen wird dabei aufrechterhalten. Konkurrieren mehrere T-Zellen um die aktuelle Zelle als Teilungsziel, so wird auch hier wieder eine dieser Zellen zufällig und gleichverteilt ausgewählt.
- 3) Im letzten Schritt wird die eigentliche Zellteilung realisiert. Teilungsbereite Zellen, deren gewünschte Zielzelle die Teilung tatsächlich gestattet, gehen in den Zustand B über. Ist die entsprechende Zielzelle eine Fluidzelle, nimmt sie ebenfalls den Zustand B an. Liegt in der Teilungsrichtung eine ehemalige B-Zelle, so geht diese in den Zustand T über, wird also ihrerseits eine teilungsbereite Zelle. Durch die Berücksichtigung der Richtung der nächstgelegenen Fluidzelle in Schritt 1 verschieben sich die T-Zellen also im Laufe der Iteration immer weiter in Richtung der Biofilmoberfläche bis sie einen direkten Fluid-Nachbarn besitzen. Existieren nach diesem dritten Schritt noch T-Zellen, so startet man erneut bei Schritt 1, ansonsten ist die Zellteilung erfolgreich abgeschlossen.



**Abbildung 2.15:** Berücksichtigte Nachbarschaft einer T-Zelle (rot) bei der Suche nach der nächstgelegenen freien Fluidzelle im zwei- (links) und dreidimensionalen (rechts) Fall (vgl. [31])

Man erhält folgenden schematischen Algorithmus für die Durchführung eines Wachstumsschritts:

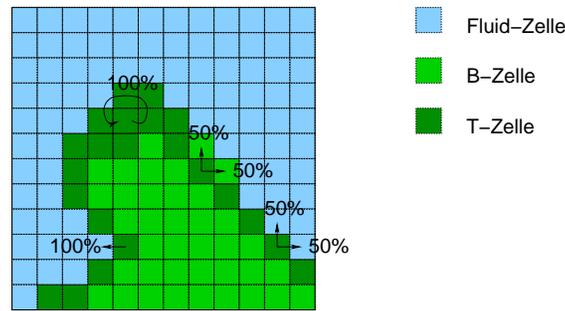


Abbildung 2.16: Beispiele für die stochastische Auswahl der Teilungsrichtung von T-Zellen

```

bis Anzahl der T-Zellen = 0
  bestimme Teilungsrichtungen der T-Zellen (Schritt 1)
  beseitige Konflikte bei der Wahl der Teilungsrichtung (Schritt 2)
  Teilungen durchführen, falls möglich (Schritt 3)
    
```

Abbildung 2.17 zeigt eine komplette Iteration des zellulären Automaten an zwei einfachen Beispielen.

Ähnliche Ansätze zur Modellierung des Wachstums auf der Basis zellulärer Automaten werden in [24], [33] und [87] beschrieben.

### Hybrides Gesamtmodell

Das Wachstum des Biofilms hängt wesentlich von der lokal verfügbaren Nährstoffmenge ab. Um die Entwicklung des Biofilms vorhersagen zu können, muss daher das oben beschriebene Wachstumsmodell mit dem Strömungs- und Transportmodell gekoppelt werden. Die Startzustände für den zellulären Automaten müssen also in Abhängigkeit von der mit Hilfe des kontinuierlichen Modells bestimmten Nährstoffkonzentration  $c$  (Gleichung (2.21)) festgelegt werden. Man geht dabei von der (vereinfachten) Annahme aus, dass die Teilungsbereitschaft einer Automatenzelle von keinen weiteren Faktoren beeinflusst wird. Die Konzentration gelöster Stoffe nimmt mit zunehmender Tiefe im Biofilm ab. Daher werden bei diesem Vorgehen fast ausschließlich Zellen in der Nähe der Biofilmoberfläche teilungsbereit. Da im Unterschied zu [24] auf einer feineren Raumskala gearbeitet wird, auf der jede Zelle des Automatengitters größenordnungsmäßig einer Mikrobenezelle entspricht, kann auf die Einführung der zusätzlichen Größe „Biomasse pro Automatenzelle“ verzichtet werden.

Die Teilungsbereitschaft bestimmt sich allein aufgrund der zum Zeitpunkt  $t_1$  des Starts eines Automaten durchlaufs seit dem Abschluss der letzten Zellteilung ( $t_0$ ) kumulierten Nährstoffmenge  $C_a$ , die von den Mikroben in der jeweiligen Automatenzelle absorbiert wurde (vgl. Gleichung (2.23)).  $C_a$  ist also definiert durch

$$C_a(t) = \int_{t_0}^{t_1} \left( \int_{\text{Zellvolumen}} \frac{qc}{K+c} \rho_b d\vec{x} \right) dt \quad \text{oder} \quad (2.24)$$

$$\frac{\partial C_a}{\partial t} = \int_{\text{Zellvolumen}} \frac{qc}{K+c} \rho_b d\vec{x}, \quad C_a(t_0) = 0. \quad (2.25)$$

Eine Automatenzelle wird als teilungsbereit angenommen, falls  $C_a$  größer ist als ein vom speziellen Biofilm und vom Nährstoff abhängiger Grenzwert  $C_{a,G}$ .

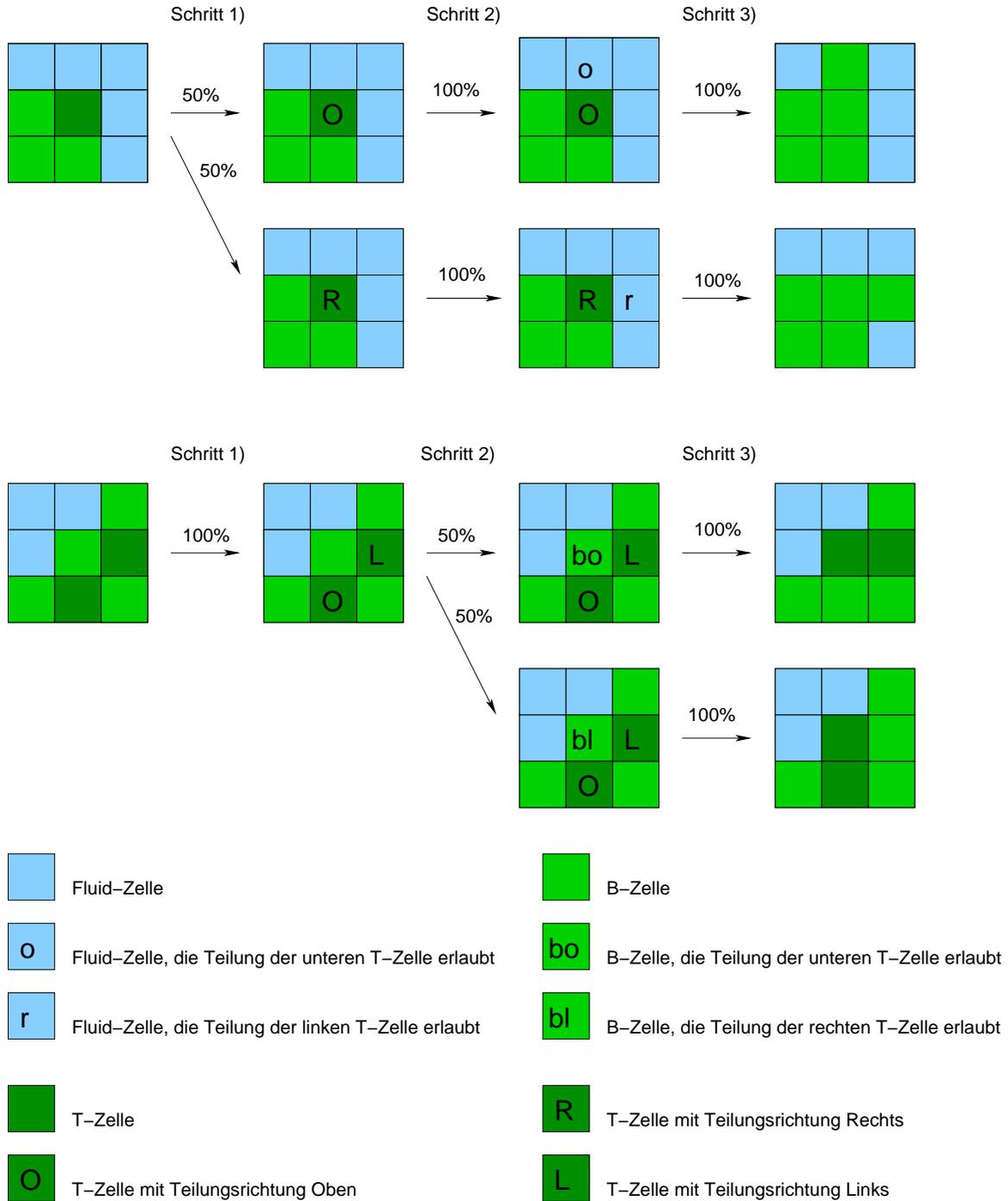


Abbildung 2.17: Ablauf einer Automateniteration an zwei einfachen Beispielen

modellierter Prozess	eindimensionale Modellgleichungen	dreidimensionale Modellgleichungen
Fluidströmung	Dicke $h$ der Grenzschicht freies Fluid – Biofilm	$\frac{\partial \vec{u}}{\partial t} + (\vec{u} \cdot \nabla) \vec{u} = \frac{1}{Re} \Delta \vec{u} - \nabla p + \vec{g}$ , $0 = \nabla^T \vec{u}$ ,
Stofftransport	$c \equiv c^{(f)}$ in $\Omega^{(f)}$ , $\frac{\partial c}{\partial t} = D \frac{\partial^2 c}{\partial z^2} - \frac{qc}{K+c} \rho_b$ in $\Omega^{(b)}$ , $D \frac{\partial c}{\partial z} \Big _{z=L} = \frac{D^{(f)}}{h} (c^{(f)} - c _{z=L})$	$\frac{\partial c}{\partial t} + (\vec{u} \cdot \nabla) c = D^{(f)} \Delta c$ in $\Omega^{(f)}$ , $\frac{\partial c}{\partial t} = D^{(b)} \Delta c - \frac{qc}{K+c} \rho_b$ in $\Omega^{(b)}$ , $D^{(f)} \frac{\partial c}{\partial \vec{n}} \Big _{\Omega^{(f)}} = D^{(b)} \frac{\partial c}{\partial \vec{n}} \Big _{\Omega^{(b)}}$
Wachstum	$v_G = \frac{\partial L}{\partial t} =$ $\int_0^L \left( \frac{Ykc}{K+c} - b_t - b_a \right) dz$	$C_a(t) =$ $\int_{t_0}^{t_0+\Delta t} \left( \int_{Zellvolumen} \frac{qc}{K+c} \rho_b d\vec{x} \right) dt$ zellulärer Automat zur Teilung der Automatenzellen mit $C_a > \text{Grenzwert}$

**Tabelle 2.1:** Gegenüberstellung der Modellierung von Strömung, Nähr- beziehungsweise Schadstofftransport und Wachstum in Biofilmsystemen im eindimensionalen Modell von Chang ([18]) und im in dieser Arbeit verwendeten dreidimensionalen Modell

Das hier beschriebene Modell weist in einigen Punkten starke Ähnlichkeiten mit dem eindimensionalen Modell von Chang ([18] und 2.1.4) auf. Tabelle 2.1 zeigt eine Gegenüberstellung dieser beiden Modelle. Der Nährstofftransport im Biofilmgebiet  $\Omega^{(b)}$  wird in beiden Fällen durch eine Diffusions-Reaktions-Gleichung modelliert, die Wachstumsgeschwindigkeit jeweils über den Integranden des Monod-Kinetik-Terms bestimmt. Die Fluidströmung und ihr Einfluss auf den Nährstofftransport können jedoch im eindimensionalen Modell nicht explizit berücksichtigt werden, sondern gehen nur indirekt über die Dicke der sogenannten Grenzschicht zwischen dem freien Fluid und dem Biofilm in die Modellgleichungen ein. Man beachte ferner, dass sich die Bezeichnungen  $\Omega^{(f)}$  und  $\Omega^{(b)}$  je nach Modell auf qualitativ unterschiedliche Bereiche beziehen. Beim eindimensionalen Modell ist der Biofilmbereich  $\Omega^{(b)}$  der gesamte Bereich bis zur maximalen Höhe  $L$ , in dem sich Biomasse befindet. Er ist also nicht dicht von Mikroben besetzt sondern umfasst einen wesentlichen Fluidanteil. Beim dreidimensionalen Modell dagegen wird nur das Volumen als Biofilmbereich bezeichnet, das ausschließlich Mikroben und kein Fluid enthält. Praktisch alle Poren und Kanäle des Films, in denen sich Fluid befindet, werden in der dreidimensionalen Geometrie explizit dem Fluidgebiet zugeordnet. Dies ist bei eindimensionalen Modellen nicht möglich.



# Kapitel 3

## Numerische Simulationswerkzeuge 1

### – *Nast++*

Zur Lösung der zeitabhängigen Strömungs- und Transportgleichungen des dreidimensionalen mikroskopischen Modells wird die objektorientierte Finite-Differenzen-Navier-Stokes-Software *Nast++* verwendet, die für die Simulation von Biofilmsystemen um Module für den konvektiv-diffusiven Transport gelöster Stoffe im Fluid und den diffusiv-reaktiven Stofftransport im Hindernis erweitert wurde (siehe 3.1). Klar definierte Schnittstellen nach außen (siehe 3.5) sowie eine hohe innere Modularität (siehe 3.4) garantieren die Flexibilität der Software sowohl im Hinblick auf die Kopplung mit Experiment (Kapitel 1), Visualisierung (Kapitel 6) oder weiteren Programmen – wie zum Beispiel dem Zelluläre-Automaten-Simulator *CAsim* (Kapitel 4) – als auch im Hinblick auf Änderungen und Erweiterungen des kontinuumsmechanischen Modells selbst (siehe B.1). Da die Gleichungen aufgrund der teilweise sehr komplexen Biofilmgeometrie auf sehr feinen Gittern gelöst werden müssen, wurde zur Effizienzsteigerung ein Mehrgitterlöser (siehe 3.3) entwickelt und implementiert und das gesamte Programmpaket zur Beschleunigung der Simulation auf der Basis von MPI (Message Passing Interface, [100]) parallelisiert (siehe 3.6).

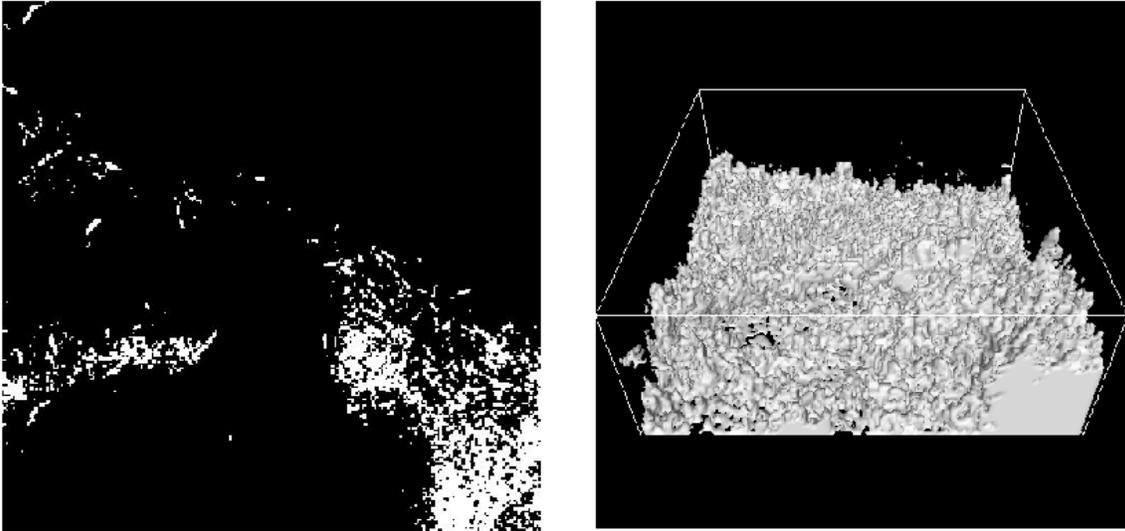
### 3.1 Leistungsumfang

*Nast++* löst die zeitabhängigen Navier-Stokes-Gleichungen

$$\begin{aligned}\frac{\partial \vec{u}}{\partial t} + (\vec{u} \cdot \nabla) \vec{u} &= \frac{1}{Re} \Delta \vec{u} - \nabla p + \vec{g}, \\ 0 &= \nabla^T \vec{u}\end{aligned}$$

für laminare Strömungen viskoser inkompressibler Fluide in beliebigen, durch ein Quadergitter darstellbaren Fluid-Hindernis-Geometrien (vgl. Abbildung 3.1). Die äußere Kraft  $\vec{g}$ , die im Allgemeinen die Gravitationskraft beschreibt, ist dabei orts- und zeitunabhängig.

An den äußeren Rändern des Rechengebietes sowie an der Oberfläche der Hindernisse können die Randbedingungen beliebig aus folgenden Typen gewählt werden:



**Abbildung 3.1:** Beispiele für mit Hilfe von Rechteckgittern (zweidimensional, links) beziehungsweise Quadrigittern (dreidimensional, rechts) darstellbare komplexe Geometrien

- NOSLIP oder Haftbedingung: Das Fluid „haftet“ am Rand, die Strömungsgeschwindigkeit ist somit null ( $\vec{u} = 0$ ).
- SLIP oder Rutschbedingung: Das Fluid erfährt am Rand keinen Reibungswiderstand, kann das Gebiet über diesen Rand aber nicht verlassen. Die zur Wand senkrechte Geschwindigkeitskomponente  $u_{\perp}$  verschwindet daher ebenso wie die Normalableitung der zur Wand parallelen Geschwindigkeitskomponenten  $\vec{u}_{\parallel}$  ( $\frac{\partial \vec{u}_{\parallel}}{\partial \vec{n}} = 0, u_{\perp} = 0$ ).
- OUTFLOW oder Ausströmbedingung: Das Fluid setzt seinen Weg über den Rand hinaus ohne Änderung der Geschwindigkeit fort ( $\frac{\partial \vec{u}}{\partial \vec{n}} = 0$ ).
- INFLOW oder Einströmbedingung: Die Strömungsgeschwindigkeit wird an diesem Rand fest vorgegeben ( $\vec{u} = \vec{u}_{in}$ ).
- PERIODIC oder periodische Randbedingung: Die Geschwindigkeitswerte an gegenüberliegenden Rändern stimmen überein. Man erhält so ein in senkrechter Richtung zur entsprechenden Grenzfläche periodisches Strömungsfeld.

Alternativ zur Vorgabe einer Randbedingung für die Geschwindigkeit kann ein Dirichlet-Randwert für den Druck (CONSTPRESS) festgesetzt werden ( $p = p_{rb}$ ). Eine gleichzeitige Vorgabe des Drucks und einer der oben genannten Geschwindigkeitsrandbedingungen ist mit Ausnahme der periodischen Randbedingungen nicht möglich (siehe 3.2.4 und [29]).

Die berechnete Fluidströmung kann mit einer Konvektions-Diffusions-Gleichung

$$\frac{\partial c}{\partial t} + (\vec{u} \cdot \nabla) c = D^{(f)} \Delta c$$

im Fluidgebiet  $\Omega^{(f)}$  gekoppelt werden. Mit Hilfe dieser Gleichung können sowohl die Temperaturverteilung als auch der Transport gelöster Stoffe im Fluid beschrieben werden ([29]). Im

ersten Fall entsteht durch die temperaturabhängige Dichteänderung des Fluids eine Rückkopplung an die Navier-Stokes-Gleichung, die sich bei Anwendung der Boussinesq-Approximation (siehe [29]) jedoch nur in einer Modifikation der äußeren Kraft  $\vec{g}$  niederschlägt:

$$\frac{\partial \vec{u}}{\partial t} + (\vec{u} \cdot \nabla) \vec{u} = \frac{1}{Re} \Delta \vec{u} - \nabla p + \vec{g}(1 - \beta c),$$

wobei  $\beta$  der sogenannte Volumenexpansionskoeffizient des betrachteten Fluids ist. Die im Fluid eines Biofilmsystems gelösten Stoffe haben dagegen annähernd dieselbe Dichte wie das „unverschmutzte“ Fluid. Auf eine Kopplung der Gleichungen für die Strömungsgeschwindigkeiten an die berechnete Konzentration kann daher verzichtet werden.

Speziell für die Simulation von Biofilmsystemen wurde der Leistungsumfang von *Nast++* um die Lösung einer Diffusions-Reaktions-Gleichung

$$\frac{\partial c}{\partial t} = D^{(b)} \Delta c - \frac{k_1 c}{k_2 + c}$$

im Hindernisgebiet  $\Omega^{(b)}$  erweitert.

Als Randbedingung gilt an der Grenzfläche  $\Gamma$  zwischen Fluid und Hindernis die Kontinuitätsbedingung (DIFFUSION, vgl. Gleichung (2.22))

$$D^{(f)} \frac{\partial c}{\partial \vec{n}} \Big|_{\Omega^{(f)}} = D^{(b)} \frac{\partial c}{\partial \vec{n}} \Big|_{\Omega^{(b)}}.$$

An den übrigen Rändern sind für die Temperatur beziehungsweise Konzentration  $c$  folgende Randbedingungen implementiert:

- ISOLATE oder Isolierrandbedingung:  $c$  ändert sich senkrecht zum Rand nicht, das heißt es geht weder Wärme beziehungsweise gelöster Stoff verloren noch wird  $c$  erhöht ( $\frac{\partial c}{\partial \vec{n}} = 0$ ).
- CONSTTRANS:  $c$  wird am Rand fest vorgegeben ( $c = c_{rb}$ ).

Die Randbedingungen für Strömung und Transport können beliebig miteinander kombiniert werden (zum Beispiel INFLOW&CONSTTRANS, OUTFLOW&ISOLATE, SLIP&ISOLATE, etc.).

## 3.2 Diskretisierung

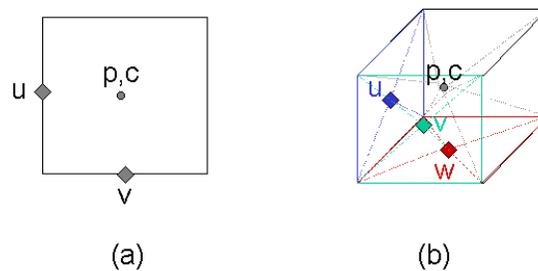
Bei der numerischen Simulation kontinuierlicher Modelle steht vor der eigentlichen Lösung der Gleichungen die Diskretisierung der Geometrie des Simulationsgebiets, der zu berechnenden Funktionen und der in den Modellgleichungen vorkommenden Differential-Operatoren, da im Rechner nur endlich viele – und somit notwendigerweise diskrete – Daten gespeichert und berechnet werden können. *Nast++* verwendet hierzu eine Finite-Differenzen-Diskretisierung auf der Basis regulärer strukturierter Quadrigitter. Die einzelnen Bestandteile dieser Diskretisierung werden im Folgenden genauer dargestellt.

### 3.2.1 Diskretisierung der Geometrie

Die Geometrie des Berechnungsgebietes wird – wie in 2.2.2 und 5.2 für den Anwendungsfall Biofilm beschrieben – durch ein reguläres strukturiertes Quadrigitter approximiert. Jede Zelle (Quader) des Gitters ist dabei entweder dem Fluid- oder dem Hindernis-Bereich zugeordnet (siehe auch [29]).

### 3.2.2 Diskretisierung der Funktionen

Die gesuchten Funktionen (Geschwindigkeit, Druck und gegebenenfalls Temperatur beziehungsweise Konzentration) werden auf demselben Quadrigitter diskretisiert wie die Geometrie. Jeder Gitterzelle ist dabei je ein Wert der Geschwindigkeit  $u$  in  $x$ -Richtung, der Geschwindigkeit  $v$  in  $y$ -Richtung, der Geschwindigkeit  $w$  in  $z$ -Richtung, des Drucks  $p$  und eventuell der Temperatur oder Konzentration  $c$  zugeordnet. Die Werte der einzelnen Funktionen sind an unterschiedlichen Punkten der Zelle definiert. So gehören Druck und Temperatur/Konzentration zum Mittelpunkt der Zelle, die  $x$ -Geschwindigkeit zum Mittelpunkt der linken Seitenfläche des Quaders, die  $y$ -Geschwindigkeit zum Mittelpunkt der vorderen Grenzfläche des Quaders und die  $z$ -Geschwindigkeit zum Mittelpunkt der Bodenfläche des Quaders (Abbildung 3.2). Durch diese spezielle Anordnung der Werte der physikalischen Größen, die als versetztes Gitter bezeichnet wird, wird eine stabile Diskretisierung der räumlichen Differential-Operatoren möglich (siehe 3.2.4). Die Werte der Unbekannten sind bei Finite-Differenzen-Verfahren nur an den vorgegebenen diskreten Punkten definiert. Durch das Verfahren selbst ist keine Interpolationsregel für die Berechnung von Werten an anderen Punkten des Gebiets gegeben.



**Abbildung 3.2:** Diskretisierung der Geschwindigkeiten, des Drucks und der Temperatur/Konzentration ((a): zweidimensionaler Fall, (b): dreidimensionaler Fall)

### 3.2.3 Diskretisierung der Zeitableitung

Sowohl für die Navier-Stokes- als auch für die Transportgleichungen sind in *Nast++* eine explizite und eine semiimplizite Zeitdiskretisierung implementiert (vgl. 3.4). In dieser Arbeit wird jedoch nur auf die explizite Version eingegangen, da im semiimpliziten Fall die Frage der Zeitschrittweitensteuerung nicht ausreichend beantwortet ist.

Der Zeitschritt der Transportgleichungen wird jeweils vor dem Zeitschritt der Navier-Stokes-

Gleichungen ausgeführt (siehe 3.4) und daher auch hier vor den Navier-Stokes-Gleichungen behandelt. Unter Verwendung der Geschwindigkeit  $\vec{u}^{(n)}$  zum alten Zeitpunkt  $t^{(n)}$  lautet die explizite Diskretisierung der Transportgleichungen

$$\frac{c^{(n+1)} - c^{(n)}}{dt} + \left( \vec{u}^{(n)} \cdot \nabla \right) c = D^{(f)} \Delta c^{(n)} \quad \text{in } \Omega^{(f)}, \quad (3.1)$$

$$\frac{c^{(n+1)} - c^{(n)}}{dt} = D^{(b)} \Delta c^{(n)} - \frac{k_1 c^{(n)}}{k_2 + c^{(n)}} \quad \text{in } \Omega^{(b)}. \quad (3.2)$$

Die neuen Werte  $c^{(n+1)}$  werden also berechnet als

$$c^{(n+1)} = c^{(n)} + dt \left( D^{(f)} \Delta c^{(n)} - \left( \vec{u}^{(n)} \cdot \nabla \right) c \right) \quad \text{in } \Omega^{(f)}, \quad (3.3)$$

$$c^{(n+1)} = c^{(n)} + dt \left( D^{(b)} \Delta c^{(n)} - \frac{k_1 c^{(n)}}{k_2 + c^{(n)}} \right) \quad \text{in } \Omega^{(b)}. \quad (3.4)$$

Im Gegensatz dazu ist die Diskretisierung der Navier-Stokes-Gleichungen komplizierter, da neben dem Geschwindigkeitsvektor  $\vec{u}$  auch der Druck  $p$  in jedem Zeitschritt angepasst werden muss und die Kontinuitätsgleichung (2.19) als Nebenbedingung der Impulsgleichung (2.18) erfüllt werden muss. Zunächst wird die Impulsgleichung explizit bezüglich der Geschwindigkeiten diskretisiert. Die auf diese Weise neu berechnete Geschwindigkeit  $\vec{u}^{(n+1)}$  muss die Kontinuitätsgleichung erfüllen:

$$\frac{\vec{u}^{(n+1)} - \vec{u}^{(n)}}{dt} + \left( \vec{u}^{(n)} \cdot \nabla \right) \vec{u}^{(n)} = \frac{1}{Re} \Delta \vec{u}^{(n)} - \nabla p + \vec{g} \left( 1 - \beta c^{(n+1)} \right), \quad (3.5)$$

$$0 = \nabla^T \vec{u}^{(n+1)}. \quad (3.6)$$

Löst man (3.5) nach  $\vec{u}^{(n+1)}$  auf und setzt das Ergebnis in (3.6) ein, so erhält man

$$\vec{u}^{(n+1)} = \underbrace{\vec{u}^{(n)} + dt \left( \frac{1}{Re} \Delta \vec{u}^{(n)} - \left( \vec{u}^{(n)} \cdot \nabla \right) \vec{u}^{(n)} + \vec{g} \left( 1 - \beta c^{(n+1)} \right) \right)}_{=: \vec{U}^{(n+1)}} - dt \nabla p, \quad (3.7)$$

$$0 = \nabla^T \vec{u}^{(n+1)} = \nabla^T \vec{U}^{(n+1)} - dt \underbrace{\nabla^T \nabla p}_{= \Delta p}. \quad (3.8)$$

Die Geschwindigkeit  $\vec{u}^{(n+1)}$  zum Zeitpunkt  $t^{(n+1)}$  lässt sich somit in drei Schritten berechnen:

- Berechnung des „vorläufigen“ Geschwindigkeitsfeldes

$$\vec{U}^{(n+1)} = \vec{u}^{(n)} + dt \left( \frac{1}{Re} \Delta \vec{u}^{(n)} - \left( \vec{u}^{(n)} \cdot \nabla \right) \vec{u}^{(n)} + \vec{g} \left( 1 - \beta c^{(n+1)} \right) \right) \quad (3.9)$$

ohne Berücksichtigung des Einflusses des herrschenden Drucks  $p$ ,

- Lösung der der Kontinuitätsgleichung entsprechenden Poissongleichung

$$\Delta p = \frac{1}{dt} \nabla^T \vec{U}^{(n+1)} \quad (3.10)$$

für den Druck  $p$ ,

- Berechnung der endgültigen Geschwindigkeit

$$\vec{u}^{(n+1)} = \vec{U}^{(n+1)} - dt \nabla p, \quad (3.11)$$

die die Kontinuitätsgleichung erfüllt.

Bei der Diskretisierung der Navier-Stokes-Gleichungen wird also die Kontinuitätsgleichung (2.19) in eine Poissongleichung für den Druck überführt.

Der durch Gleichung (3.10) bestimmte Druck entspricht nur dann dem physikalischen Druck, wenn das Geschwindigkeitsfeld  $\vec{u}^{(n)}$  die Kontinuitätsgleichung erfüllt. Startet man die Simulation mit einem (physikalisch nicht möglichen) Strömungsfeld, das die Kontinuitätsgleichung nicht erfüllt, so setzt sich der Druck aus dem physikalischen Druck  $p_{phys}$  und einem Anteil  $p_{div}$  zur Erzwingung der Divergenzfreiheit von  $\vec{u}^{(n+1)}$  zusammen:

$$p = p_{phys} + p_{div}, \quad (3.12)$$

wobei  $p_{phys}$  durch die Gleichung

$$\Delta p_{phys} = \nabla^T \left( \vec{g} \left( 1 - \beta c^{(n+1)} \right) - \left( \vec{u}^{(n)} \cdot \nabla \right) \vec{u}^{(n)} \right) \quad (3.13)$$

und  $p_{div}$  durch

$$\Delta p_{div} = \nabla^T \left( \frac{1}{dt} \vec{u}^{(n)} + \frac{1}{Re} \Delta \vec{u}^{(n)} \right) \quad (3.14)$$

gegeben ist. Die rechte Seite der Poissongleichung (3.14) enthält dabei alle Terme der gesamten rechten Seite der Poissongleichung (3.10), die nur dann nicht verschwinden, wenn  $\vec{u}^{(n)}$  die Kontinuitätsgleichung nicht erfüllt. Bereits im nächsten Zeitschritt ist  $p_{div} \equiv 0$ , da aus Gleichung (3.10) und (3.11) für  $n = 1$  die Divergenzfreiheit von  $\vec{u}^{(n+1)}$  folgt.

Die Randbedingungen zur Poissongleichung (3.10) werden entsprechend der Chorinschen Projektionsmethode ([29]) durch das Skalarprodukt von Gleichung (3.7) mit dem Normalenvektor  $\vec{n}$  zum Gebiets- oder Hindernisrand bestimmt:

$$\nabla p \cdot \vec{n} = \frac{1}{dt} \left( \vec{U}^{(n+1)} - \vec{u}^{(n+1)} \right) \cdot \vec{n}. \quad (3.15)$$

Bei geeigneter Wahl von  $\vec{U}^{(n+1)}$  am Rand des Gebietes ist die räumlich diskretisierte Version dieser Randbedingung unabhängig von der zum Zeitpunkt der Lösung der Poissongleichung noch unbekanntem Geschwindigkeit  $\vec{u}^{(n+1)}$  (siehe 3.2.5).

### 3.2.4 Diskretisierung der räumlichen Differential-Operatoren

Die räumlichen Differential-Operatoren werden nach einem Finite-Differenzen-Schema diskretisiert. Die berechneten approximativen Operatorauswertungen sind dabei ebenso wie die Funktionswerte unterschiedlichen Punkten in der Gitterzelle zugeordnet: Die Operatoren der Impulsgleichungen werden an den Gitterpunkten zu den Geschwindigkeitswerten berechnet, die Operatoren der Poissongleichung (Gleichung (3.10)) und der Transportgleichungen jeweils im Mittelpunkt der Zelle, wo auch die Druck- und Temperatur-/Konzentrationswerte angesiedelt sind. Für den konvektiven Term  $(\vec{u} \cdot \nabla) \vec{u}$  der Impulsgleichungen wird die Gleichung

$$(\vec{u} \cdot \nabla) \vec{u} = \frac{\partial \vec{u} u}{\partial x} + \frac{\partial \vec{u} v}{\partial y} + \frac{\partial \vec{u} w}{\partial z}$$

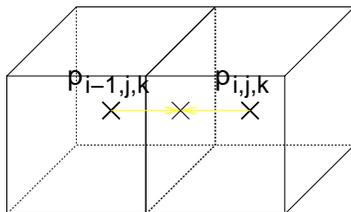
verwendet, die aus der Produktregel

$$\begin{aligned} \frac{\partial \vec{u} u}{\partial x} + \frac{\partial \vec{u} v}{\partial y} + \frac{\partial \vec{u} w}{\partial z} &= \underbrace{u \frac{\partial \vec{u}}{\partial x} + v \frac{\partial \vec{u}}{\partial y} + w \frac{\partial \vec{u}}{\partial z}}_{= (\vec{u} \cdot \nabla) \vec{u}} + \underbrace{\vec{u} \frac{\partial u}{\partial x} + \vec{u} \frac{\partial v}{\partial y} + \vec{u} \frac{\partial w}{\partial z}}_{= \vec{u} \nabla^T \vec{u}} \end{aligned}$$

und  $\nabla^T \vec{u} = 0$  folgt. Es werden daher statt  $(\vec{u} \cdot \nabla) \vec{u}$  die Terme  $\frac{\partial \vec{u} u}{\partial x}$ ,  $\frac{\partial \vec{u} v}{\partial y}$  und  $\frac{\partial \vec{u} w}{\partial z}$  diskretisiert. Analog werden für den konvektiven Term  $(\vec{u} \cdot \nabla) c$  der Transportgleichung  $\frac{\partial c u}{\partial x}$ ,  $\frac{\partial c v}{\partial y}$  und  $\frac{\partial c w}{\partial z}$  approximiert. Tabelle 3.1 zeigt alle zu diskretisierenden räumlichen Differential-Operatoren der Navier-Stokes- und Transportgleichungen sowie schematisch die Zuordnung zu den entsprechenden Gitterpunkten des versetzten Gitters.

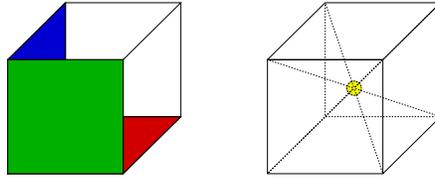
Für sämtliche Ortsableitungen wird nach Möglichkeit eine Diskretisierung zweiter Ordnung (das heißt mit einem Fehler der Größenordnung  $\mathcal{O}(h^2)$  bei Verwendung eines Gitters mit Maschenweite  $h$ ) verwendet. Lediglich bei den konvektiven Termen der Impuls- und der Transportgleichung muss zugunsten der Stabilität der diskreten Gleichungen eine Mischform aus Diskretisierungen erster und zweiter Ordnung in Kauf genommen werden (siehe unten und [29]).  $dx$ ,  $dy$  und  $dz$  bezeichnen im Folgenden die Gitterweiten in  $x$ -,  $y$ - und  $z$ -Richtung.

Der Druckgradient  $\nabla p$  kann mit Genauigkeit  $\mathcal{O}(h^2)$  durch zentrale Differenzen approximiert werden, da im versetzten Gitter die den Komponenten des Druckgradienten zugeordneten Punkte in jeder Raumrichtung gerade zwischen zwei Druckwerten liegen (siehe Abbildung 3.3).



$$\frac{\partial p}{\partial x} = \frac{p_{i,j,k} - p_{i-1,j,k}}{dx} + \mathcal{O}(dx^2)$$

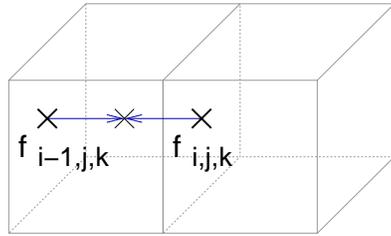
**Abbildung 3.3:** Zentrale-Differenzen-Diskretisierung von  $\frac{\partial p}{\partial x}$  in der Zelle  $(i, j, k)$



				
Impulsgleichungen (3.9), (3.11)	$\frac{\partial u^2}{\partial x}$ $\frac{\partial uv}{\partial y}$ $\frac{\partial uw}{\partial z}$  $\Delta u$	$\frac{\partial uv}{\partial x}$ $\frac{\partial v^2}{\partial y}$ $\frac{\partial vw}{\partial z}$  $\Delta v$	$\frac{\partial uw}{\partial x}$ $\frac{\partial vw}{\partial y}$ $\frac{\partial w^2}{\partial z}$  $\Delta w$	
Poissongleichung (3.10)	$\frac{\partial p}{\partial x}$	$\frac{\partial p}{\partial y}$	$\frac{\partial p}{\partial z}$	$\Delta p$ $\frac{\partial f}{\partial x}$ $\frac{\partial g}{\partial y}$ $\frac{\partial h}{\partial z}$
Transportgleichungen (3.3), (3.4)				$\frac{\partial uc}{\partial x}$ $\frac{\partial vc}{\partial y}$ $\frac{\partial wc}{\partial z}$ $\Delta c$

**Tabelle 3.1:** Zu diskretisierende Differential-Operatoren der Navier-Stokes- und Transportgleichungen und zugehörige Punkte im versetzten Gitter

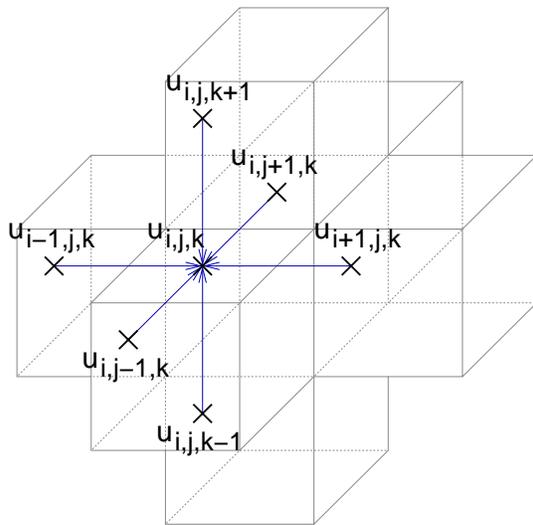
Analog wird die Divergenz von  $\vec{U} =: \begin{pmatrix} f \\ g \\ h \end{pmatrix}$  mit Hilfe zentraler Differenzen berechnet (vgl. Abbildung 3.4). Die Komponenten des Vektors  $\vec{U}$  sind dabei denselben Gitterpunkten wie die Geschwindigkeiten  $u, v$  und  $w$  zugeordnet.



$$\frac{\partial f}{\partial x} = \frac{f_{i+1,j,k} - f_{i,j,k}}{dx} + \mathcal{O}(dx^2)$$

**Abbildung 3.4:** Zentrale-Differenzen-Diskretisierung von  $\frac{\partial f}{\partial x}$  in der Zelle  $(i, j, k)$

Die Laplace-Operatoren  $\Delta u, \Delta v, \Delta w, \Delta p$  und  $\Delta c$  werden mit Hilfe des Standard-7-Punkte-Sterns diskretisiert. Dieser Quotient liefert eine Approximation zweiter Ordnung für den Laplace-Operator am der entsprechenden Funktion zugeordneten Ort in der Zelle  $(i, j, k)$  (vgl. Abbildung 3.2 und Tabelle 3.1). Abbildung (3.5) gibt beispielhaft den Differenzen-Quotienten für  $\Delta u$  an.



$$\begin{aligned} \Delta u &= \frac{\partial^2 u}{\partial x^2} + \frac{\partial^2 u}{\partial y^2} + \frac{\partial^2 u}{\partial z^2} = \\ &= \frac{u_{i+1,j,k} - 2u_{i,j,k} + u_{i-1,j,k}}{dx^2} + \\ &+ \frac{u_{i,j+1,k} - 2u_{i,j,k} + u_{i,j-1,k}}{dy^2} + \\ &+ \frac{u_{i,j,k+1} - 2u_{i,j,k} + u_{i,j,k-1}}{dz^2} + \\ &\mathcal{O}(dx^2 + dy^2 + dz^2) \end{aligned}$$

**Abbildung 3.5:** 7-Punkte-Diskretisierung des Laplace-Operators der  $x$ -Geschwindigkeit  $u$  in der Zelle  $(i, j, k)$

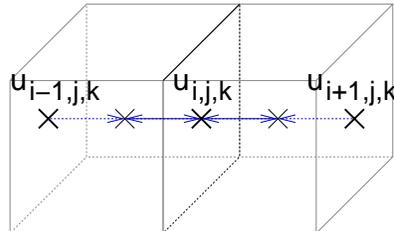
Die Operatoren der konvektiven Terme  $(\vec{u} \cdot \nabla) \vec{u}$  und  $(\vec{u} \cdot \nabla) c$  der Impuls- und der Transportgleichung sind wesentlich schwieriger zu behandeln, da die Verwendung der entsprechenden zentralen Differenzen bei starker Konvektion, das heißt hohen Geschwindigkeiten oder hoher Reynoldszahl (betrifft die Impulsgleichung (2.18)) beziehungsweise niedriger Schmidtzahl (betrifft die Transportgleichung (2.20)), zu einer instabilen diskreten Gleichung führt. Diese Instabilität wird in *Nast++* durch die zusätzliche Verwendung einer Diskretisierung nach dem Donor-Cell-Schema (siehe [29]) vermieden. Die nach diesem Schema entstehenden

Differenzen-Quotienten besitzen im Gegensatz zu den zentralen Differenzen allerdings nur noch die Approximationsordnung  $\mathcal{O}(h)$ . Die endgültigen Differenzen-Quotienten werden als gewichtetes Mittel

$$\gamma \cdot \text{Donor-Cell-Differenzen} + (1 - \gamma) \cdot \text{zentrale Differenzen} \quad (3.16)$$

gebildet. Der Gewichtungsfaktor  $\gamma \in [0; 1]$  wird dabei umso größer gewählt, je stärker die Konvektion ist. Bei geeigneter Wahl von  $\gamma$  können so für die konkrete Strömungs- (und Transport-) Gleichung sowohl die Stabilität des Donor-Cell-Schemas als auch die höhere Diskretisierungsordnung der zentralen Differenzen ausgenutzt werden ([29]).

Die Abbildungen 3.6 und 3.7 zeigen die Differenzen-Quotienten für  $\frac{\partial u^2}{\partial x}$  und  $\frac{\partial uv}{\partial y}$ . Die übrigen Differential-Operatoren der konvektiven Terme der Impulsgleichung werden analog approximiert.



$$\frac{\partial u^2}{\partial x} = \frac{1}{dx} \left( \left( \frac{u_{i+1,j,k} + u_{i,j,k}}{2} \right)^2 - \left( \frac{u_{i,j,k} + u_{i-1,j,k}}{2} \right)^2 \right) + \mathcal{O}(dx^2),$$

$$\frac{\partial u^2}{\partial x} = \frac{1}{dx} \left( \frac{u_{i+1,j,k} + u_{i,j,k}}{2} u_{(o)} - \frac{u_{i,j,k} + u_{i-1,j,k}}{2} u_{(w)} \right) + \mathcal{O}(dx), \text{ wobei}$$

$$u_{(o)} = \begin{cases} u_{i+1,j,k} & \text{falls } \frac{u_{i+1,j,k} + u_{i,j,k}}{2} > 0, \\ u_{i,j,k} & \text{falls } \frac{u_{i+1,j,k} + u_{i,j,k}}{2} \leq 0, \end{cases}$$

$$u_{(w)} = \begin{cases} u_{i,j,k} & \text{falls } \frac{u_{i,j,k} + u_{i-1,j,k}}{2} > 0, \\ u_{i-1,j,k} & \text{falls } \frac{u_{i,j,k} + u_{i-1,j,k}}{2} \leq 0 \end{cases}$$

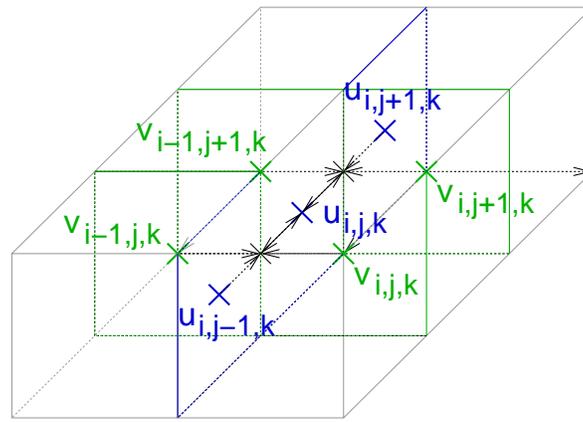
**Abbildung 3.6:** Zentrale-Differenzen- (oben) und Donor-Cell-Diskretisierung (unten) von  $\frac{\partial u^2}{\partial x}$  in der Zelle  $(i, j, k)$

Die Differential-Operatoren des konvektiven Terms  $(\vec{u} \cdot \nabla) c$  der Transportgleichung (2.20) werden analog als Mittelung zentraler Differenzen und Donor-Cell-Differenzen (Abbildung 3.8) gebildet.

Die Verwendung eines versetzten Geschwindigkeits-Druck-Gitters sorgt für die Stabilität der stationären diskreten Navier-Stokes-Gleichungen. Zentrale Differenzen auf einem nicht-versetzten Gitter erzeugen ein indefinites Gleichungssystem. Betrachte dazu die stationären Navier-Stokes-Gleichungen

$$\begin{aligned} (\vec{u} \cdot \nabla) \vec{u} &= \frac{1}{Re} \Delta \vec{u} - \nabla p + \vec{g} \\ \nabla^T \vec{u} &= 0 \end{aligned}$$

mit NOSLIP-Randbedingungen an allen Rändern des Gebietes. Approximiert man den Druckgradienten in dieser Gleichung bei einem nicht-versetzten Gitter mit zentralen Differenzen, so ist



$$\frac{\partial uv}{\partial y} = \frac{1}{dy} \left( \frac{v_{i,j+1,k} + v_{i-1,j+1,k}}{2} \frac{u_{i,j,k} + u_{i,j+1,k}}{2} - \frac{v_{i,j,k} + v_{i-1,j,k}}{2} \frac{u_{i,j-1,k} + u_{i,j,k}}{2} \right) + \mathcal{O}(dy^2),$$

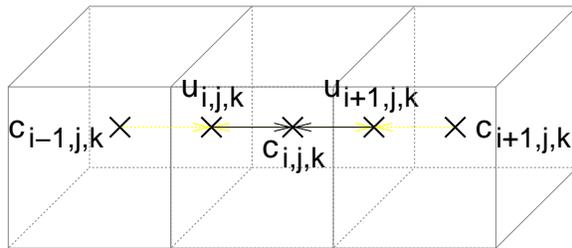
$$\frac{\partial uv}{\partial y} = \frac{1}{dy} \left( \frac{v_{i,j+1,k} + v_{i-1,j+1,k}}{2} u^{(n)} - \frac{v_{i,j,k} + v_{i-1,j,k}}{2} u^{(s)} \right) + \mathcal{O}(dy), \text{ wobei}$$

$$u^{(n)} = \begin{cases} u_{i,j+1,k} & \text{falls } \frac{v_{i,j+1,k} + v_{i-1,j+1,k}}{2} > 0, \\ u_{i,j,k} & \text{falls } \frac{v_{i,j+1,k} + v_{i-1,j+1,k}}{2} \leq 0, \end{cases}$$

$$u^{(s)} = \begin{cases} u_{i,j,k} & \text{falls } \frac{v_{i,j,k} + v_{i-1,j,k}}{2} > 0, \\ u_{i,j-1,k} & \text{falls } \frac{v_{i,j,k} + v_{i-1,j,k}}{2} \leq 0 \end{cases}$$

Abbildung 3.7: Zentrale-Differenzen- (oben) und Donor-Cell-Diskretisierung (unten) von  $\frac{\partial uv}{\partial y}$  in der Zelle  $(i, j, k)$

$$\frac{\partial uc}{\partial x} = \frac{1}{dx} \left( u_{i+1,j,k} \frac{c_{i,j,k} + c_{i+1,j,k}}{2} - u_{i,j,k} \frac{c_{i-1,j,k} + c_{i,j,k}}{2} \right) + \mathcal{O}(dx^2),$$



$$\frac{\partial uc}{\partial x} = \frac{u_{i+1,j,k} c^{(o)} - u_{i,j,k} c^{(w)}}{dx} + \mathcal{O}(dx),$$

wobei

$$c^{(o)} = \begin{cases} c_{i+1,j,k} & \text{falls } u_{i+1,j,k} > 0, \\ c_{i,j,k} & \text{falls } u_{i+1,j,k} \leq 0, \end{cases}$$

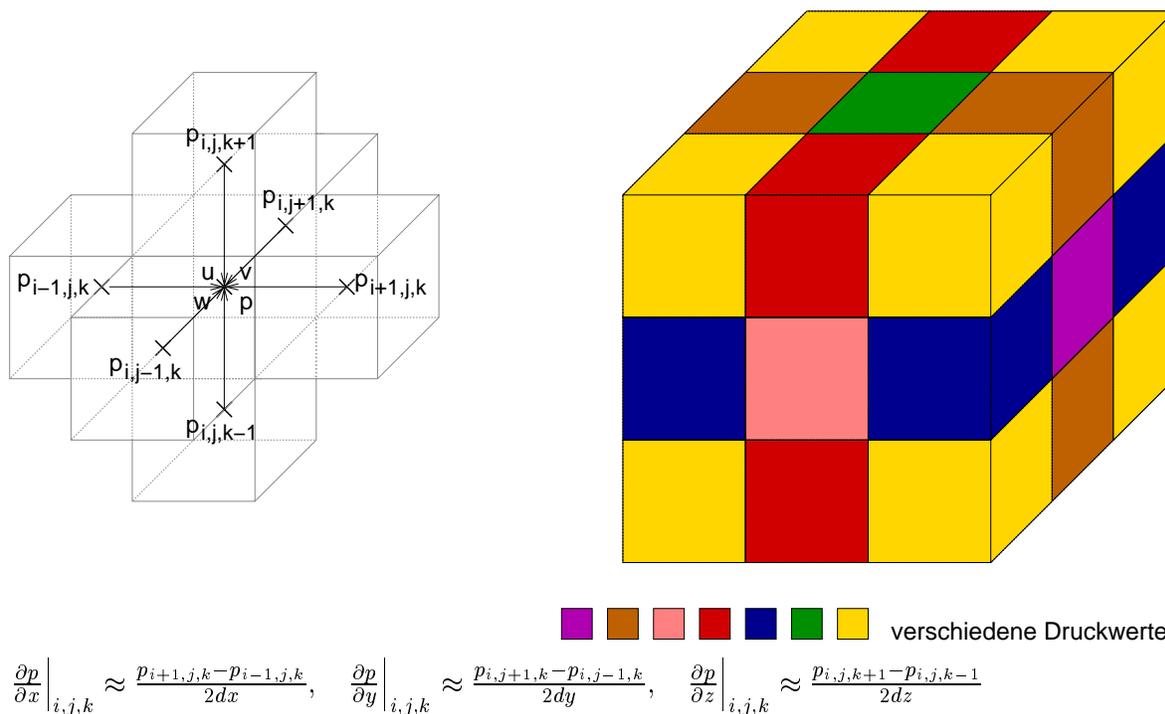
$$c^{(w)} = \begin{cases} c_{i,j,k} & \text{falls } u_{i,j,k} > 0, \\ c_{i-1,j,k} & \text{falls } u_{i,j,k} \leq 0 \end{cases}$$

Abbildung 3.8: Zentrale-Differenzen- (oben) und Donor-Cell-Diskretisierung (unten) von  $\frac{\partial uc}{\partial y}$  in der Zelle  $(i, j, k)$

$$u_{i,j,k} = 0, v_{i,j,k} = 0, w_{i,j,k} = 0 \quad \forall(i, j, k),$$

$$p_{i,j,k} = \begin{cases} p_1 & \text{falls } i \text{ gerade, } j \text{ gerade, } k \text{ gerade} \\ p_2 & \text{falls } i \text{ gerade, } j \text{ gerade, } k \text{ ungerade} \\ p_3 & \text{falls } i \text{ gerade, } j \text{ ungerade, } k \text{ gerade} \\ p_4 & \text{falls } i \text{ gerade, } j \text{ ungerade, } k \text{ ungerade} \\ p_5 & \text{falls } i \text{ ungerade, } j \text{ gerade, } k \text{ gerade} \\ p_6 & \text{falls } i \text{ ungerade, } j \text{ gerade, } k \text{ ungerade} \\ p_7 & \text{falls } i \text{ ungerade, } j \text{ ungerade, } k \text{ gerade} \\ p_8 & \text{falls } i \text{ ungerade, } j \text{ ungerade, } k \text{ ungerade} \end{cases}$$

mit beliebigen Werten  $p_1, p_2, p_3, p_4, p_5, p_6, p_7$  und  $p_8$  eine Lösung des Problems, da die Differenzen-Quotienten für den Druckgradienten jeweils eine Reichweite von zwei Gitterweiten besitzen und daher Druck-Oszillationen mit höherer Frequenz nicht erfassen (siehe Abbildung 3.9 und [29]).

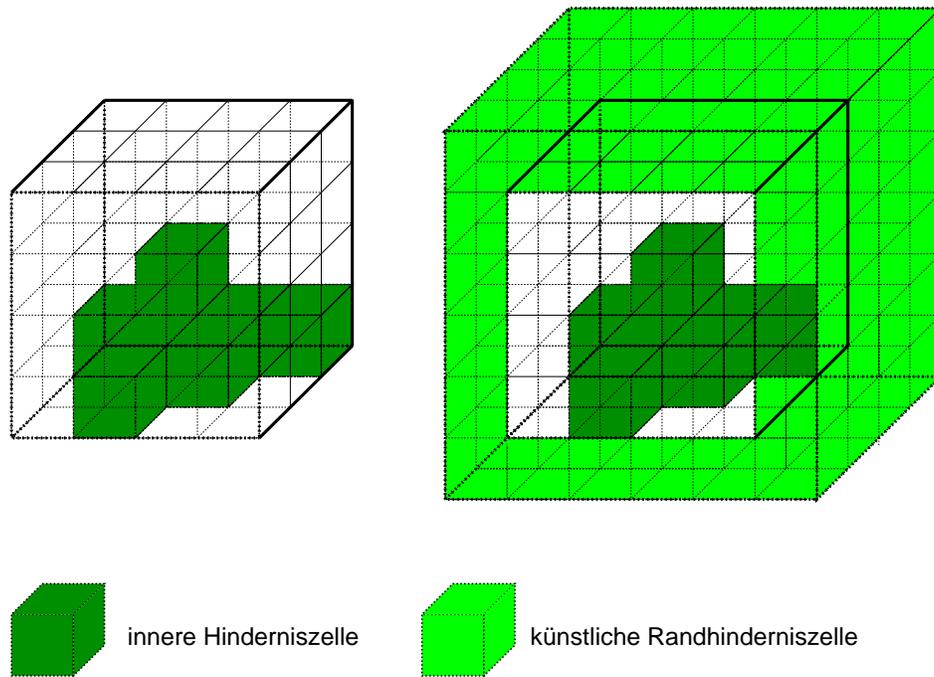


**Abbildung 3.9:** Zentrale Differenzen für den Druckgradienten auf einem nicht-versetzten Gitter und nicht erfassbare Oszillationen des Drucks  $p$  (rechts)

### 3.2.5 Diskretisierung der Randbedingungen

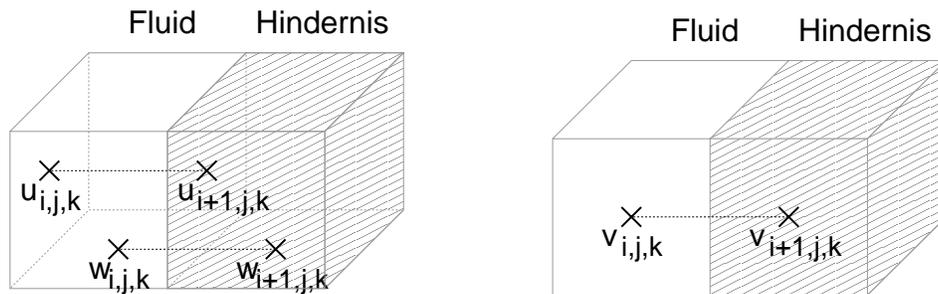
Die Randbedingungen für Strömung und Transport werden in *Nast++* ebenfalls möglichst mit der Genauigkeit  $\mathcal{O}(h^2)$  diskretisiert und über entsprechende Implementierungen der Zugriffsfunktionen auf Werte der Nachbarzellen der aktuellen Gitterzelle realisiert. Randbedingungen sind an allen Grenzflächen zwischen Hindernis- und Fluidquadrern des Berechnungsgitters definiert (siehe 3.4). Um Randbedingungen nicht nur an den eigentlichen Fluid-Hindernis-Grenzen, sondern auch am äußeren Rand des Gebietes definieren zu können,

wird daher das gesamte Berechnungsgebiet mit einer zusätzlichen, künstlichen Hinderniszellschicht umgeben (siehe Abbildung 3.10).



**Abbildung 3.10:** Künstliche Hindernisrandschicht um das Berechnungsgebiet zur Definition der Randbedingungen am äußeren Gebietsrand

Neumann-Randbedingungen werden bei Werten, die nicht direkt auf dem Rand liegen, mit zentralen Differenzen, andernfalls mit einseitigen Differenzen (auf der Seite des Fluids) approximiert. Bei Dirichlet-Randbedingungen wird der Funktionswert gleich dem vorgegebenen Randwert gesetzt, falls der entsprechende Gitterpunkt auf dem Rand liegt. Andernfalls wird der Wert in der anschließenden Hinderniszelle durch lineare Extrapolation mit Hilfe des Randwertes und des benachbarten Wertes im Fluid bestimmt. Abbildung 3.11 zeigt als Beispiel für diese Vorgehensweise die Diskretisierung der IN- und OUTFLOW-Randbedingung.

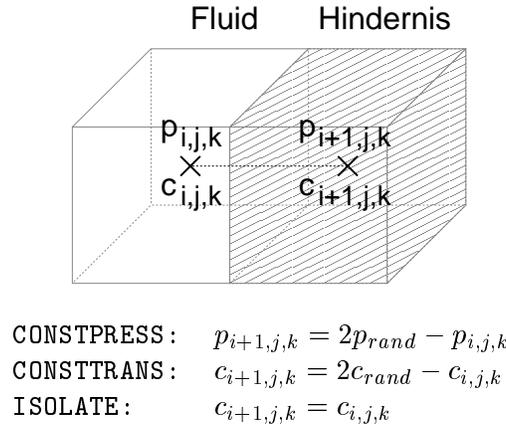


$$\begin{aligned} \text{INFLOW:} \quad & u_{i+1,j,k} = u_{in}, \quad v_{i+1,j,k} = 2v_{in} - v_{i,j,k}, \quad w_{i+1,j,k} = 2w_{in} - w_{i,j,k} \\ \text{OUTFLOW:} \quad & u_{i+1,j,k} = u_{i,j,k}, \quad v_{i+1,j,k} = v_{i,j,k}, \quad w_{i+1,j,k} = w_{i,j,k} \end{aligned}$$

**Abbildung 3.11:** Diskretisierung der IN- und OUTFLOW-Randbedingungen (vgl. 3.1)

Die NOSLIP-Randbedingung entspricht gerade einer INFLOW-Randbedingung mit  $\vec{u}_{in} = 0$ , die SLIP-Randbedingung setzt sich aus einem Dirichlet-Anteil (in senkrechter Richtung zum Rand) und einem Neumann-Anteil (parallel zum Rand) zusammen. Der Dirichlet-Anteil wird analog zur INFLOW-, der Neumann-Anteil analog zur OUTFLOW-Randbedingung diskretisiert.

Die Neumann-Randbedingungen für den Druck und die Temperatur/Konzentration werden immer durch zentrale Differenzen angenähert, da diese Werte immer im Zellmittelpunkt liegen. Aus demselben Grund werden Dirichlet-Randbedingungen immer durch lineare Extrapolation realisiert (siehe Abbildung 3.12).



**Abbildung 3.12:** Diskretisierung der CONSTPRESS-, CONSTTRANS und ISOLATE-Randbedingungen (vgl. 3.1)

Als Diskretisierung der Randbedingung (3.15) erhält man so für einen Rand in  $x$ -Richtung

$$\frac{p_{i+1,j,k} - p_{i,j,k}}{dx} = \frac{1}{dt} \left( f_{i+1,j,k}^{(n+1)} - u_{i+1,j,k}^{(n+1)} \right). \quad (3.17)$$

Im Allgemeinen ist bei der Lösung der Poissongleichung für den Druck jedoch weder  $u_{i+1,j,k}^{(n+1)}$  bekannt (Ausnahme: Dirichlet-Randbedingung für  $u$ ), noch lässt sich  $f_{i+1,j,k}^{(n+1)}$  am Rand berechnen, da die dafür benötigten Differenzen-Quotienten (vgl. Gleichung 3.10)) nur im Inneren des Gebietes berechnet werden können (Ausnahme: periodische Randbedingungen). Setzt man jedoch (3.17) in die räumlich diskretisierte Version

$$\frac{p_{i+1,j,k} - 2p_{i,j,k} + p_{i-1,j,k}}{dx^2} + \frac{p_{i,j+1,k} - 2p_{i,j,k} + p_{i,j-1,k}}{dy^2} + \frac{p_{i,j,k+1} - 2p_{i,j,k} + p_{i,j,k-1}}{dz^2} = \frac{1}{dt} \left( \frac{f_{i+1,j,k}^{(n+1)} - f_{i,j,k}^{(n+1)}}{dx} + \frac{g_{i,j+1,k}^{(n+1)} - g_{i,j,k}^{(n+1)}}{dy} + \frac{h_{i,j,k+1}^{(n+1)} - h_{i,j,k}^{(n+1)}}{dz} \right)$$

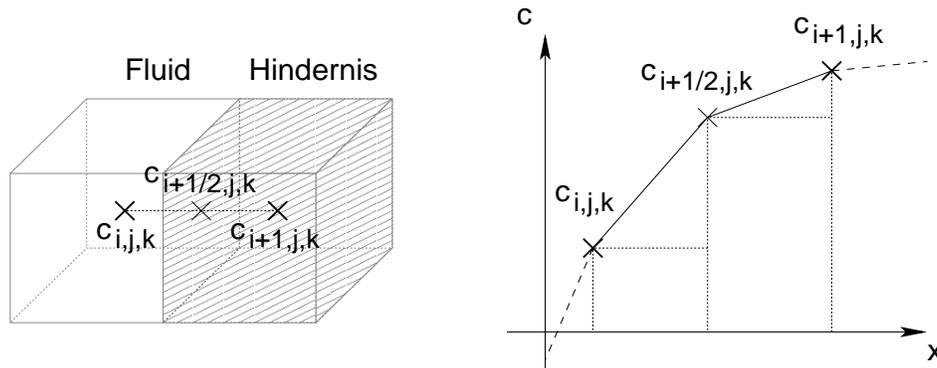
der Poissongleichung (3.10) ein, so ist die resultierende Gleichung

$$\frac{-p_{i,j,k} + p_{i-1,j,k}}{dx^2} + \frac{p_{i,j+1,k} - 2p_{i,j,k} + p_{i,j-1,k}}{dy^2} + \frac{p_{i,j,k+1} - 2p_{i,j,k} + p_{i,j,k-1}}{dz^2} = \frac{1}{dt} \left( \frac{u_{i+1,j,k}^{(n+1)} - f_{i,j,k}^{(n+1)}}{dx} + \frac{g_{i,j+1,k}^{(n+1)} - g_{i,j,k}^{(n+1)}}{dy} + \frac{h_{i,j,k+1}^{(n+1)} - h_{i,j,k}^{(n+1)}}{dz} \right)$$

unabhängig von der Wahl von  $f_{i+1,j,k}^{(n+1)}$  am Rand ([29]). Es kann also  $f_{i+1,j,k}^{(n+1)} = u_{i+1,j,k}^{(n+1)}$  gewählt werden. Somit ergeben sich als endgültige Randbedingungen für die Poissongleichung homogene Neumann-Randbedingungen

$$p_{i+1,j,k} = p_{i,j,k}. \quad (3.18)$$

An der Grenzfläche zwischen Fluid und Hindernis müssen auch im Falle der Simulation der vollständigen Transportgleichungen (2.20) und (2.21) diskretisierte Randbedingungen beachtet werden, obwohl hier jeweils alle Nachbarwerte der Funktion  $c$  mit tatsächlich berechneten Werten belegt sind, die Differenzen-Quotienten am Rand also theoretisch analog zu den Quotienten im Inneren des Fluids oder des Hindernisses gebildet werden könnten. Während der Lösung der Transportgleichungen wird beispielsweise von der Seite des Fluids auf den Wert  $c_{i+1,j,k}$ , von der Seite des Hindernisses auf den Wert  $c_{i,j,k}$  zugegriffen (siehe Abbildung 3.13). Die Werte werden in den entsprechenden Differenzen-Quotienten jeweils zur Approximation der Ableitung  $\frac{\partial c}{\partial x}$  benötigt. Wegen der nach (2.22) bei unterschiedlichen Diffusionskoeffizienten ( $D^{(f)} \neq D^{(b)}$ ) entstehenden Unstetigkeit der Ableitung ist die naheliegende Approximation dieser Ableitung durch  $\frac{c_{i+1,j,k} - c_{i,j,k}}{dx}$  jedoch nicht zulässig. Daher wird die DIFFUSION-Randbedingung durch Identifizierung der beiden einseitigen Differenzen-Quotienten mit halber Schrittweite am Rand diskretisiert (siehe Abbildung 3.13).



$$D^{(f)} \frac{c_{i+1,j,k} - c_{i+\frac{1}{2},j,k}}{dx} = D^{(b)} \frac{c_{i+\frac{1}{2},j,k} - c_{i,j,k}}{dx}$$

$$\Rightarrow c_{i+\frac{1}{2},j,k} = \frac{D^{(f)} c_{i,j,k} + D^{(b)} c_{i+1,j,k}}{D^{(f)} + D^{(b)}}.$$

**Abbildung 3.13:** Diskretisierung der DIFFUSION-Randbedingung für die Transportgleichung an der Grenzfläche zwischen Fluid und Hindernis

Die Ableitung  $\frac{\partial c}{\partial x}$  lässt sich mit dem so berechneten Wert  $c_{i+\frac{1}{2},j,k}$  durch

$$\frac{c_{i+\frac{1}{2},j,k} - c_{i,j,k}}{\frac{1}{2}dx} = \frac{\left(2c_{i+\frac{1}{2},j,k} - c_{i,j,k}\right) - c_{i,j,k}}{dx}$$

auf der Seite des Fluids und analog durch

$$\frac{c_{i+1,j,k} - c_{i+\frac{1}{2},j,k}}{\frac{1}{2}dx} = \frac{c_{i+1,j,k} - \left(2c_{i+\frac{1}{2},j,k} - c_{i+1,j,k}\right)}{dx}$$

auf der Seite des Hindernisses approximieren, es muss also von der Seite des Fluids auf den Wert

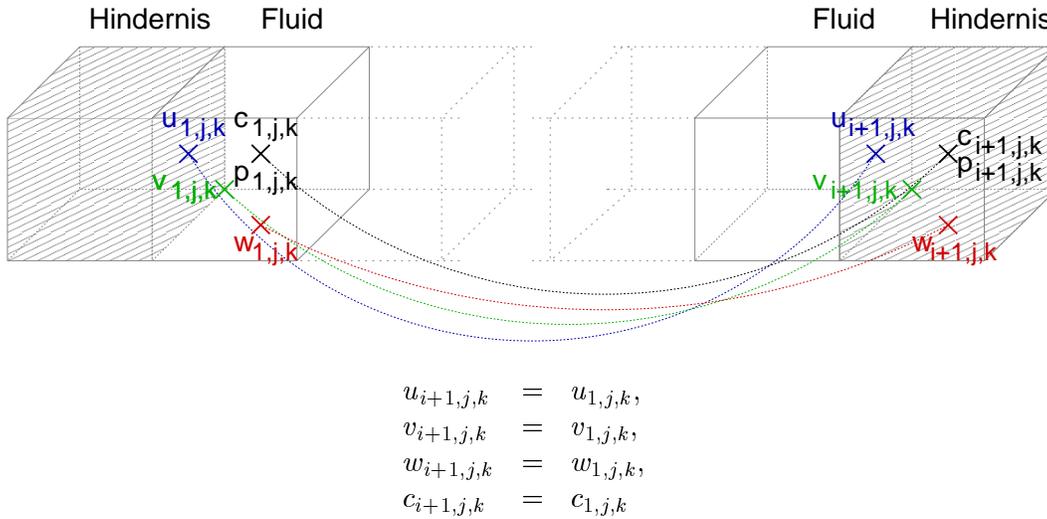
$$\tilde{c}_{i+1,j,k} := 2c_{i+\frac{1}{2},j,k} - c_{i,j,k} = \frac{2D^{(f)}c_{i+1,j,k} + (D^{(b)} - D^{(f)})c_{i,j,k}}{D^{(f)} + D^{(b)}}$$

anstelle von  $c_{i+1,j,k}$  und von der Seite des Hindernisses auf

$$\tilde{c}_{i,j,k} := 2c_{i+\frac{1}{2},j,k} - c_{i+1,j,k} = \frac{2D^{(b)}c_{i,j,k} + (D^{(f)} - D^{(b)})c_{i+1,j,k}}{D^{(f)} + D^{(b)}}$$

anstelle von  $c_{i,j,k}$  zugegriffen werden.

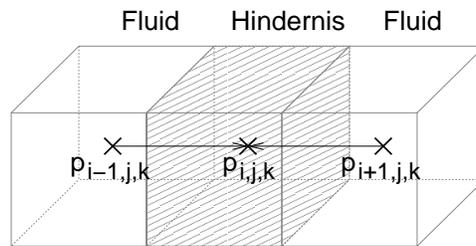
Periodische Randbedingungen für die Geschwindigkeiten  $\vec{u}$  und die Temperatur beziehungsweise Konzentration  $c$  können nur an den äußeren Rändern des Gebiets verwendet werden. Dabei wird die an die Randfluidzelle anschließende künstliche Hinderniszelle mit der Randfluidzelle am gegenüberliegenden Ende des Gebietes identifiziert (Abbildung 3.14). Die Druckwerte dürfen jedoch nicht identifiziert werden, da die Strömung eine Druckdifferenz in Strömungsrichtung bedingt und umgekehrt die Strömung durch die Druckdifferenz ermöglicht wird. Für den Druck gelten daher auch bei periodischen Randbedingungen homogene Neumann-Randbedingungen (falls an mindestens einem der anderen Gebietsränder Dirichlet-Randbedingungen für die Geschwindigkeiten oder den Druck vorgegeben sind) oder Dirichlet-Bedingungen (zur Vorgabe einer Druckdifferenz zwischen den gegenüberliegenden Rändern, falls an allen anderen Rändern nur Neumann-Bedingungen definiert sind).



**Abbildung 3.14:** Diskretisierung der periodischen Randbedingungen für Strömung und Transport an äußeren Gebietsrändern

Die Diskretisierung der räumlichen Differential-Operatoren und der Randbedingungen ist damit vollständig beschrieben. Die Wahl konkreter Randbedingungen für die Biofilmsimulation wird wie in 2.2.2 beschrieben durchgeführt. Es folgt eine Bemerkung zur Implementierung der Randbedingungen. Diese wird in *Nast++* nicht wie in [29] beschrieben durch explizites Schreiben und Lesen der entsprechenden Werte im Hindernis auf den/von dem der betreffenden Hinderniszelle zugeordneten Speicherplatz realisiert, sondern durch explizite Berechnung der benötigten Werte innerhalb der lesenden Zugriffsfunktionen des Gitterobjekts (vgl.

3.4 und [10]). Dadurch fällt die Anforderung an die Geometrie des Simulationsgebietes weg, dass Hindernisse zwischen zwei Fluidzellen immer mindestens zwei Zellschichten dick sein müssen. Abbildung 3.15 verdeutlicht die entstehende Problematik beim Schreiben und Lesen der Werte über den Speicherplatz des Gitters.



$$\begin{aligned} \text{von links:} & \quad p_{i,j,k} = p_{i-1,j,k} \\ \text{von rechts:} & \quad p_{i,j,k} = p_{i+1,j,k} \end{aligned}$$

**Abbildung 3.15:** Konflikt beim Schreiben und Lesen von Randwerten über den Speicherplatz des Berechnungsgitters am Beispiel homogener Neumann-Randbedingungen für den Druck

### 3.2.6 Stabilitätsbedingungen

Die Stabilität der stationären diskreten Gleichungen wurde durch versetzte Geschwindigkeits- und Druckgitter erreicht (siehe 3.2.4). Für die zeitabhängigen diskreten Gleichungen müssen darüber hinaus noch weitere Bedingungen erfüllt sein, um eine oszillationsfreie Lösung der diskreten Navier-Stokes-Gleichungen zu erhalten ([29]):

- Der konvektive Term  $(\vec{u} \cdot \nabla) \vec{u}$  begrenzt die Zeitschrittweite gemäß der sogenannten Courant-Friedrichs-Levi-(CFL) Bedingungen

$$|u_{max}| dt < dx, \quad |v_{max}| dt < dy, \quad |w_{max}| dt < dz. \quad (3.19)$$

Durch diese Bedingungen wird sichergestellt, dass kein ideelles Teilchen pro Zeitschritt mehr als eine räumliche Gitterweite zurücklegt.

- Zusätzlich muss die Bedingung

$$\frac{2dt}{Re} < \left( \frac{1}{dx^2} + \frac{1}{dy^2} + \frac{1}{dz^2} \right)^{-1} \quad (3.20)$$

erfüllt sein, die aus der Steifigkeit des diffusiven Anteils  $\frac{1}{Re} \Delta \vec{u}$  der Impulsgleichung resultiert.

Bei konvektionsdominierten Strömungen wird die Zeitschrittweite  $dt$  also linear durch die räumliche Gitterweite beschränkt, andernfalls, das heißt bei kleiner Reynoldszahl, hängt die Zeitschrittweite quadratisch von der räumlichen Gitterweite ab.

Für die Transportgleichungen ist neben den Courant-Friedrichs-Levi-Bedingungen analog zu (3.20) die Bedingung

$$2dtD^{(f)} < \left( \frac{1}{dx^2} + \frac{1}{dy^2} + \frac{1}{dz^2} \right)^{-1} \quad (3.21)$$

zu erfüllen ([29]).

### 3.3 Gleichungslöser

Verwendet man die in 3.2 beschriebene explizite Zeitdiskretisierung für die Navier-Stokes- und Transportgleichungen, so ist in jedem Zeitschritt nur ein lineares Gleichungssystem zu lösen, das aus der Poissongleichung (3.10)

$$\Delta p = \frac{1}{dt} \nabla^T \vec{U}^{(n+1)}$$

entsteht. Zur Lösung dieses Gleichungssystems wurden zwei verschiedene iterative Löser, ein SOR(successive overrelaxation)-Verfahren sowie ein geometrisches Mehrgitterverfahren, implementiert (vgl. 3.4). Zur besseren Übersicht wird im Folgenden neben dem SOR-Verfahren ein weiteres grundlegendes Relaxationsverfahren, das sogenannte Jacobi-Verfahren, kurz vorgestellt. Relaxationsverfahren durchlaufen in jeder Iteration alle Zellen des Berechnungsgitters und aktualisieren jeweils die entsprechenden Werte mit Hilfe des Residuums der entsprechenden Zeile des diskreten Gleichungssystems. Mehrgitterverfahren arbeiten dagegen nicht nur auf dem Berechnungsgitter selbst, sondern verwenden zur Fehlerreduktion zusätzliche gröbere Gitter mit eigenen diskreten Gleichungssystemen.

Im Folgenden werden zunächst einige Bezeichnungen eingeführt, die zur Definition der Verfahren benötigt werden:

- $\Delta_{i,j,k}^m, \Delta_{i,j,k}^e, \Delta_{i,j,k}^w, \Delta_{i,j,k}^n, \Delta_{i,j,k}^s, \Delta_{i,j,k}^u$  und  $\Delta_{i,j,k}^d$  bezeichnen die Koeffizienten der 7-Punkte-Stern-Diskretisierung  $[\Delta p]_{i,j,k}$  des Laplace-Operators des Drucks in der Gitterzelle  $(i, j, k)$ :

$$\begin{aligned} [\Delta p]_{i,j,k} &= \Delta_{i,j,k}^m p_{i,j,k} + \\ &\quad \Delta_{i,j,k}^e p_{i+1,j,k} + \Delta_{i,j,k}^w p_{i-1,j,k} + \\ &\quad \Delta_{i,j,k}^n p_{i,j+1,k} + \Delta_{i,j,k}^s p_{i,j-1,k} + \\ &\quad \Delta_{i,j,k}^u p_{i,j,k+1} + \Delta_{i,j,k}^d p_{i,j,k-1}. \end{aligned}$$

- $rhs_{i,j,k}$  sei die diskretisierte rechte Seite  $\frac{1}{dt} \left[ \nabla^T \vec{U}^{(n+1)} \right]_{i,j,k}$  der Poissongleichung.

Die Abhängigkeit der Koeffizienten des diskreten Laplace-Operators und der Diskretisierung der rechten Seite wird in obigen Bezeichnungen nicht explizit berücksichtigt.

### 3.3.1 Jacobi-Verfahren

Beim Jacobi-Verfahren werden pro Iteration alle Gitterpunkte des Berechnungsgitters besucht und dabei der zugehörige Wert der Unbekannten so gewählt, dass das lokale Residuum, das heißt das Residuum der entsprechenden Zeile des Gleichungssystems, verschwindet. Zur Auswertung des Residuums der Zeile werden dabei an allen Gitterpunkten außer dem aktuellen die Druckwerte der vorhergehenden Iteration verwendet. Damit ergibt sich folgender Algorithmus:

$$\begin{array}{l} \text{für } it = 0, 1, 2, \dots \\ \quad \text{für alle Zellen} \\ \quad \quad p_{i,j,k}^{(it+1)} := p_{i,j,k}^{(it)} - \frac{1}{\Delta_{i,j,k}^m} \left( [\Delta p^{(it)}]_{i,j,k} - rhs_{i,j,k} \right); \end{array}$$

Da zur Bestimmung der neuen Werte ( $p^{(it+1)}$ ) aller Zellen jeweils nur die alten Werte ( $p^{(it)}$ ) verwendet werden, ist das Ergebnis einer Iteration unabhängig von der Durchlaufreihenfolge über die Zellen und die Werte von  $p^{(it+1)}$  können simultan berechnet werden. Das Verfahren ist also trivial parallelisierbar.

Für quaderförmige Gebiete ohne innere Hindernisse und mit Dirichlet-Randbedingungen lassen sich die Konvergenzeigenschaften des Verfahrens genau bestimmen. Bezeichnet man mit  $p^*$  die exakte Lösung der diskreten Poissongleichung

$$[\Delta p]_{i,j,k} = rhs_{i,j,k} \quad \forall i, j, k, \quad (3.22)$$

so ergibt sich aus obigem Algorithmus für die Fehler  $e^{(it)} = p^* - p^{(it)}$ ,  $it = 0, 1, 2, \dots$ , die Iteration

$$\begin{array}{l} \text{für } it = 0, 1, 2, \dots \\ \quad \text{für alle Zellen} \\ \quad \quad e_{i,j,k}^{(it+1)} := - \frac{[\Delta e^{(it)}]_{i,j,k} - \Delta_{i,j,k}^m e_{i,j,k}^{(it)}}{\Delta_{i,j,k}^m}; \end{array}$$

Für den von *Nast++* verwendeten 7-Punkte-Stern erhält man

$$\begin{aligned} e_{i,j,k}^{(it+1)} := & - \frac{1}{2(dx^{-2} + dy^{-2} + dz^{-2})} \left( dx^{-2} \cdot \left( e_{i-1,j,k}^{(it)} + e_{i+1,j,k}^{(it)} \right) + \right. \\ & \left. dy^{-2} \cdot \left( e_{i,j-1,k}^{(it)} + e_{i,j+1,k}^{(it)} \right) + dz^{-2} \cdot \left( e_{i,j,k-1}^{(it)} + e_{i,j,k+1}^{(it)} \right) \right) \quad \forall i, j, k. \quad (3.23) \end{aligned}$$

Die zugehörige Iterationsmatrix  $M_e$  besitzt für quaderförmige Gebiete ohne innere Hindernisse und bei Dirichlet-Randbedingungen die Eigenvektoren  $q^{(l,m,n)}$  mit

$$q_{i,j,k}^{(l,m,n)} = \sin\left(\frac{l\pi i \cdot dx}{L_x}\right) \sin\left(\frac{m\pi j \cdot dy}{L_y}\right) \sin\left(\frac{n\pi k \cdot dz}{L_z}\right) \quad (3.24)$$

und die Eigenwerte

$$\begin{aligned} \lambda^{(l,m,n)} = & - \frac{1}{dx^{-2} + dy^{-2} + dz^{-2}} \\ & \left( dx^{-2} \cos\left(\frac{l\pi \cdot dx}{L_x}\right) + dy^{-2} \cos\left(\frac{m\pi \cdot dy}{L_y}\right) + dz^{-2} \cos\left(\frac{n\pi \cdot dz}{L_z}\right) \right), \quad (3.25) \end{aligned}$$

wobei  $L_x, L_y$  und  $L_z$  die Abmessungen des Berechnungsgebietes in  $x$ -,  $y$ - und  $z$ -Richtung sind.

**Begründung** (siehe auch [78] für den zweidimensionalen Fall mit  $dx = dy$ ):

$$\begin{aligned} [M_e q^{(l,m,n)}]_{i,j,k} &= -\frac{1}{2(dx^{-2} + dy^{-2} + dz^{-2})} \\ &\left( dx^{-2} \left( \sin\left(\frac{l\pi(i-1) \cdot dx}{L_x}\right) \sin\left(\frac{m\pi j \cdot dy}{L_y}\right) \sin\left(\frac{n\pi k \cdot dz}{L_z}\right) + \right. \right. \\ &\quad \left. \left. \sin\left(\frac{l\pi(i+1) \cdot dx}{L_x}\right) \sin\left(\frac{m\pi j \cdot dy}{L_y}\right) \sin\left(\frac{n\pi k \cdot dz}{L_z}\right) \right) + \right. \\ &\quad dy^{-2} \left( \sin\left(\frac{l\pi i \cdot dx}{L_x}\right) \sin\left(\frac{m\pi(j-1) \cdot dy}{L_y}\right) \sin\left(\frac{n\pi k \cdot dz}{L_z}\right) + \right. \\ &\quad \left. \sin\left(\frac{l\pi i \cdot dx}{L_x}\right) \sin\left(\frac{m\pi(j+1) \cdot dy}{L_y}\right) \sin\left(\frac{n\pi k \cdot dz}{L_z}\right) \right) + \\ &\quad \left. dz^{-2} \left( \sin\left(\frac{l\pi i \cdot dx}{L_x}\right) \sin\left(\frac{m\pi j \cdot dy}{L_y}\right) \sin\left(\frac{n\pi(k-1) \cdot dz}{L_z}\right) + \right. \right. \\ &\quad \left. \left. \sin\left(\frac{l\pi i \cdot dx}{L_x}\right) \sin\left(\frac{m\pi j \cdot dy}{L_y}\right) \sin\left(\frac{n\pi(k+1) \cdot dz}{L_z}\right) \right) \right). \end{aligned}$$

Es gilt

$$\begin{aligned} \sin\left(\frac{l\pi(i \pm 1) \cdot dx}{L_x}\right) &= \cos\left(\frac{l\pi \cdot dx}{L_x}\right) \sin\left(\frac{l\pi i \cdot dx}{L_x}\right) \pm \sin\left(\frac{l\pi \cdot dx}{L_x}\right) \cos\left(\frac{l\pi i \cdot dx}{L_x}\right), \\ \sin\left(\frac{m\pi(j \pm 1) \cdot dy}{L_y}\right) &= \cos\left(\frac{m\pi \cdot dy}{L_y}\right) \sin\left(\frac{m\pi j \cdot dy}{L_y}\right) \pm \sin\left(\frac{m\pi \cdot dy}{L_y}\right) \cos\left(\frac{m\pi j \cdot dy}{L_y}\right) \quad \text{und} \\ \sin\left(\frac{n\pi(k \pm 1) \cdot dz}{L_z}\right) &= \cos\left(\frac{n\pi \cdot dz}{L_z}\right) \sin\left(\frac{n\pi k \cdot dz}{L_z}\right) \pm \sin\left(\frac{n\pi \cdot dz}{L_z}\right) \cos\left(\frac{n\pi k \cdot dz}{L_z}\right). \end{aligned}$$

$$\begin{aligned} \Rightarrow [IT_e q^{(l,m,n)}]_{i,j,k} &= -\frac{dx^{-2} \cos\left(\frac{l\pi \cdot dx}{L_x}\right) + dy^{-2} \cos\left(\frac{m\pi \cdot dy}{L_y}\right) + dz^{-2} \cos\left(\frac{n\pi \cdot dz}{L_z}\right)}{dx^{-2} + dy^{-2} + dz^{-2}} \\ &\quad \underbrace{\sin\left(\frac{l\pi i \cdot dx}{L_x}\right) \sin\left(\frac{m\pi j \cdot dy}{L_y}\right) \sin\left(\frac{n\pi k \cdot dz}{L_z}\right)}_{= q_{i,j,k}^{(l,m,n)}} \quad \square \end{aligned}$$

Die asymptotische Konvergenzrate  $\rho_J := \max_{l,m,n} |\lambda^{(l,m,n)}|$  des Jacobi-Verfahrens ist also

$$\rho_J = \frac{dx^{-2} \cos\left(\frac{\pi \cdot dx}{L_x}\right) + dy^{-2} \cos\left(\frac{\pi \cdot dy}{L_y}\right) + dz^{-2} \cos\left(\frac{\pi \cdot dz}{L_z}\right)}{dx^{-2} + dy^{-2} + dz^{-2}} \quad (3.26)$$

und somit wegen  $\cos(a) = 1 - \mathcal{O}(a^2)$  von der Ordnung  $1 - \mathcal{O}(dx^2 + dy^2 + dz^2)$ . Mit jeder Halbierung der Gitterweiten  $dx, dy$  und  $dz$  werden also etwa viermal so viele Iterationen benötigt, um den Fehler der Anfangsschätzung um einen festen Faktor zu reduzieren.

### 3.3.2 Gauß-Seidel- und SOR-Verfahren

Beim Gauß-Seidel-Verfahren werden die Gitterpunkte des Berechnungsgitters in einer *festen* Reihenfolge besucht und dabei – ähnlich wie beim Jacobi-Verfahren – der zugehörige Wert der Unbekannten so gewählt, dass das lokale Residuum, das heißt das Residuum der entsprechenden Zeile des Gleichungssystems, verschwindet. Anders als beim Jacobi-Verfahren werden dabei jedoch an allen bereits besuchten Gitterpunkten die neu berechneten Werte von  $p$  verwendet. Somit entfällt die beim Jacobi-Verfahren notwendige Speicherung unterschiedlicher Druckfelder  $p^{(it+1)}$  und  $p^{(it)}$  für die neuen und alten Werte. Es ergibt sich folgender Algorithmus:

$$\begin{array}{l}
 \text{für } it = 0, 1, 2, \dots \\
 \quad \text{für } k = 1 \dots k_{max} \\
 \quad \quad \text{für } j = 1 \dots j_{max} \\
 \quad \quad \quad \text{für } i = 1 \dots i_{max} \\
 \quad \quad \quad \quad p_{i,j,k} := p_{i,j,k} - \frac{1}{\Delta_{i,j,k}^m} \left( [\Delta p]_{i,j,k} - rhs_{i,j,k} \right);
 \end{array}$$

Für die asymptotische Konvergenzrate  $\rho_{GS}$  des Gauß-Seidel-Verfahrens auf quaderförmigen Gebieten ohne innere Hindernisse und mit Dirichlet-Randbedingungen gilt nach [78]:

$$\rho_{GS} = \rho_J^2. \quad (3.27)$$

Das Gauß-Seidel-Verfahren benötigt somit asymptotisch betrachtet nur halb so viele Iterationen wie das Jacobi-Verfahren.

Die Konvergenz lässt sich durch Multiplikation des Druckkorrekturterms

$$\frac{1}{\Delta_{i,j,k}^m} \left( [\Delta p]_{i,j,k} - rhs_{i,j,k} \right)$$

mit einem Überrelaxationsparameter  $\omega \in [1; 2[$  weiter verbessern. Das resultierende Verfahren bezeichnet man als SOR-Verfahren:

$$\begin{array}{l}
 \text{für } it = 0, 1, 2, \dots \\
 \quad \text{für } k = 1 \dots k_{max} \\
 \quad \quad \text{für } j = 1 \dots j_{max} \\
 \quad \quad \quad \text{für } i = 1 \dots i_{max} \\
 \quad \quad \quad \quad p_{i,j,k} := p_{i,j,k} - \omega \frac{1}{\Delta_{i,j,k}^m} \left( [\Delta p]_{i,j,k} - rhs_{i,j,k} \right);
 \end{array}$$

Die minimale asymptotische Konvergenzrate  $\rho_{SOR,opt}$  erhält man – wie in [78] gezeigt – für quaderförmige, hindernisfreie Gebiete mit Dirichlet-Randbedingungen mit

$$\omega := \omega_{opt} = \frac{2}{1 + \sqrt{1 - \rho_{GS}}}. \quad (3.28)$$

$\rho_{SOR,opt}$  ergibt sich dann ebenfalls nach [78] als

$$\rho_{SOR,opt} = \omega_{opt} - 1 = \frac{1 - \sqrt{1 - \rho_{GS}}}{1 + \sqrt{1 - \rho_{GS}}}. \quad (3.29)$$

Wegen  $\rho_{GS} = 1 - \mathcal{O}(dx^2 + dy^2 + dz^2)$  folgt

$$\rho_{SOR,opt} = \frac{1 - \sqrt{\mathcal{O}(dx^2 + dy^2 + dz^2)}}{1 + \sqrt{\mathcal{O}(dx^2 + dy^2 + dz^2)}} = 1 - \mathcal{O}\left(\sqrt{dx^2 + dy^2 + dz^2}\right).$$

Die Anzahl der benötigten Iterationen des SOR-Verfahrens verdoppelt sich also bei Halbierung der Gitterweiten. Insbesondere für sehr feine Gitter ist das SOR-Verfahren mit optimalem Relaxationsparameter daher wesentlich schneller als das Jacobi- und das Gauß-Seidel-Verfahren, bei denen sich die Iterationszahl bei jeder Halbierung der Gitterweiten vervierfacht.

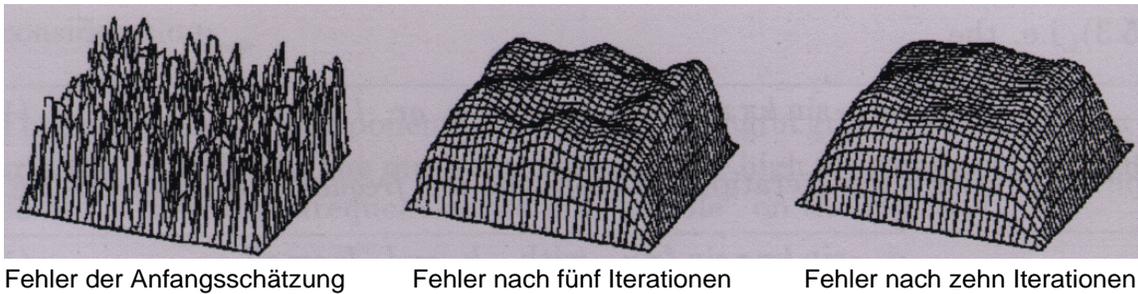
### 3.3.3 Mehrgitterverfahren

Die Konvergenz aller beschriebenen Relaxationsverfahren verschlechtert sich bei zunehmender Gitterauflösung erheblich. Da sich bei dreidimensionaler Rechnung außerdem die Anzahl der Gitterpunkte und damit der zu berechnenden Werte bei Halbierung von  $h$  verachtfacht, wächst der Gesamtaufwand für die Lösung der Poissongleichung beim Jacobi- und Gauß-Seidel-Verfahren insgesamt proportional zu  $h^5$ . Da beim SOR-Verfahren mit optimalem Relaxationsparameter eine Halbierung der Gitterweite lediglich eine Verdopplung der Iterationen nach sich zieht, ist der Rechenaufwand für dieses Verfahren nur noch proportional zu  $h^4$ . Optimal wäre dagegen ein Lösungsverfahren, das mit einem Aufwand auskommt, der proportional zur Anzahl der Unbekannten, das heißt zu  $h^3$ , ist. Da Relaxationsverfahren diese Forderung nicht erfüllen, sind sie insbesondere für hoch auflösende Berechnungsgitter, wie sie bei der Biofilmsimulation aufgrund der hohen Komplexität der Biofilmgeometrie erforderlich sind, nur begrenzt einsetzbar.

Von der Gitterauflösung unabhängige Konvergenzraten und damit ein Aufwand von der Ordnung  $\mathcal{O}(h^3)$  können mit den sogenannten Mehrgitterverfahren erzielt werden. In dieser Arbeit wurde ein solches Verfahren zur Verbesserung der numerischen Effizienz von Nast++ speziell für komplexe Hindernisgeometrien, bei denen herkömmliche Mehrgitterverfahren meist schnell an ihre Grenzen stoßen ([82]), neu entwickelt und implementiert. Die Idee der Mehrgitterverfahren beruht auf der Feststellung, dass Relaxationsverfahren zur Lösung linearer Gleichungssysteme, die beim Durchlaufen aller Zellen des Berechnungsgitters den Fehler jeweils mit Hilfe des lokalen Residuums (das heißt des Residuums der entsprechenden Zeile des diskreten Gleichungssystems) reduzieren, sehr bald zu einem „glatten“ Fehlerverlauf führen, den Fehler insgesamt jedoch mit sinkender Gitterweite  $h$  immer langsamer verringern (Abbildung 3.16 und [82]).

Für das Jacobi-Verfahren lässt sich dieses Verhalten sofort an den Eigenvektoren und -werten der Iterationsmatrix  $M_e$  für den Fehler ablesen: Der Betrag des Eigenwertes

$$\lambda^{(l,m,n)} = - \frac{dx^{-2} \cos\left(\frac{l\pi \cdot dx}{L_x}\right) + dy^{-2} \cos\left(\frac{m\pi \cdot dy}{L_y}\right) + dz^{-2} \cos\left(\frac{n\pi \cdot dz}{L_z}\right)}{dx^{-2} + dy^{-2} + dz^{-2}}$$



**Abbildung 3.16:** Entwicklung des Fehlers über mehrere Gauß-Seidel Iterationen bei der Lösung einer zweidimensionalen Poissongleichung auf dem Einheitsquadrat (ohne Hindernisse) mit der in Nast++ verwendeten Finite-Differenzen-Diskretisierung (Bild aus [82])

ist umso kleiner, je höher die Frequenzen  $l, m$  beziehungsweise  $n$  des zugehörigen Eigenvektors

$$q^{(l,m,n)} = \left( \sin \left( \frac{l\pi i \cdot dx}{L_x} \right) \sin \left( \frac{m\pi j \cdot dy}{L_y} \right) \sin \left( \frac{n\pi k \cdot dz}{L_z} \right) \right)_{i=1,\dots,i_{max}, j=1,\dots,j_{max}, k=1,\dots,k_{max}}$$

sind. Fehleranteile im durch die Eigenvektoren zu großen Werten von  $l, m$  und  $n$  aufgespannten Unterraum werden daher durch Multiplikation mit  $M_e$  wesentlich stärker reduziert als Fehleranteile im Unterraum, der durch die Eigenvektoren zu kleinen Werten von  $l, m$  oder  $n$  aufgespannt wird. Es gilt sogar

$$\min_{l,m,n} \left| \lambda^{(l,m,n)} \right| = O(dx^2 + dy^2 + dz^2).$$

Die hochfrequentesten Fehleranteile werden somit sogar umso schneller reduziert, je feiner das Gitter ist.

Der „glatte“ Restfehler nach einigen Iterationen eines Relaxationsverfahrens lässt sich auch mit größerer Gitterweite  $H > h$  gut approximieren. Ferner kann die Diskretisierung des ursprünglichen Gleichungssystems auf dem groben Gitter mit Gitterweite  $H$  deutlich schneller (mit weniger Iterationen, siehe oben) gelöst werden. Die Diskretisierung des Gleichungssystems auf unterschiedlich groben Gittern (Gitterebenen) und eine geeignete Kombination der Lösungsapproximationen dieser Gitter sollte daher eine deutlich schnellere Lösung des Gleichungssystems ermöglichen. Um ein Verfahren zu definieren, das auf dieser Idee beruht, sind folgende Komponenten festzulegen:

### Gitterfolge

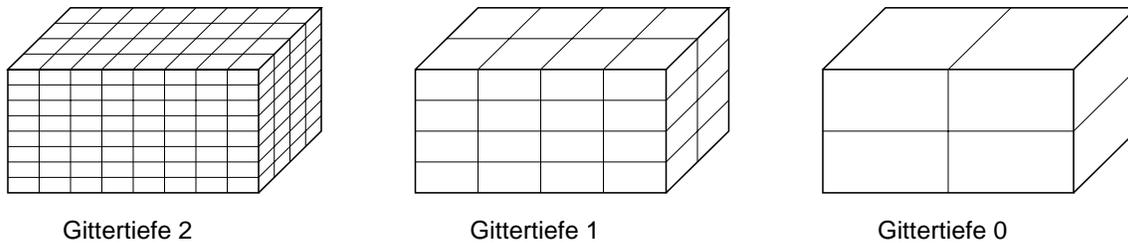
Zunächst ist eine Folge von Gittern zu wählen, die zur Lösung des Gleichungssystems verwendet werden sollen. Die einfachste Folge erhält man durch sukzessive Verdopplung der Gitterweiten  $dx, dy$  und  $dz$  (Abbildung 3.17). Diese Gittervergrößerung eignet sich nur für Gleichungssysteme, in denen die Werte der Unbekannten in alle drei Raumrichtungen in etwa gleich stark gekoppelt sind ([80]). Bei der Diskretisierung der Poissongleichung mit Hilfe der

7-Punkte-Sterne lautet die Zeile zur Zelle  $(i, j, k)$  der diskreten Poissongleichung (vgl. 3.2)

$$\frac{p_{i+1,j,k} - 2p_{i,j,k} + p_{i-1,j,k}}{dx^2} + \frac{p_{i,j+1,k} - 2p_{i,j,k} + p_{i,j-1,k}}{dy^2} + \frac{p_{i,j,k+1} - 2p_{i,j,k} + p_{i,j,k-1}}{dz^2} =$$

$$\frac{1}{dt} \left( \frac{f_{i+1,j,k}^{(n+1)} - f_{i,j,k}^{(n+1)}}{dx} + \frac{g_{i,j+1,k}^{(n+1)} - g_{i,j,k}^{(n+1)}}{dy} + \frac{h_{i,j,k+1}^{(n+1)} - h_{i,j,k}^{(n+1)}}{dz} \right).$$

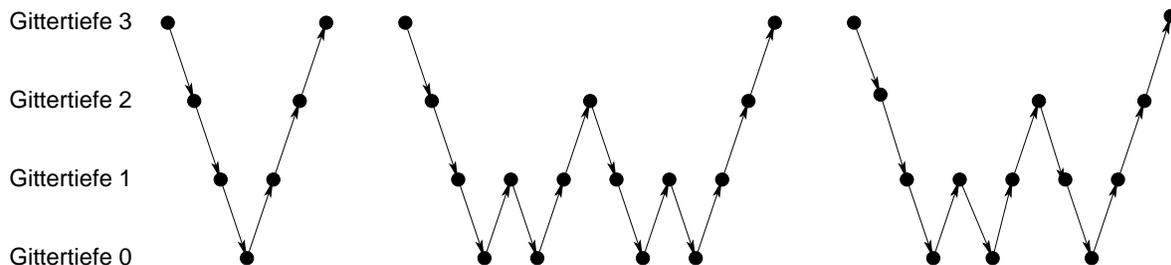
Der Wert  $p_{i,j,k}$  ist also mit  $p_{i\pm 1,j,k}$  jeweils mit dem Faktor  $\frac{1}{dx^2}$ , mit  $p_{i,j\pm 1,k}$  mit dem Faktor  $\frac{1}{dy^2}$  und mit  $p_{i,j,k\pm 1}$  mit dem Faktor  $\frac{1}{dz^2}$  gekoppelt. Sind die Gitterweiten  $dx, dy$  und  $dz$  von der gleichen Größenordnung, so ist also der Wert  $p_{i,j,k}$  mit allen sechs Nachbarwerten etwa gleich stark gekoppelt und die schrittweise Verdopplung der Gitterweite eine geeignete Strategie zur Generierung der Gitterhierarchie.



**Abbildung 3.17:** Gitterhierarchie bei schrittweiser Verdopplung der Gitterweiten  $dx, dy$  und  $dz$

### Durchlaufreihenfolge für die Gitterebenen

Zur effizienten Berechnung der Lösung auf dem feinsten Gitter können die so definierten Gitterebenen in unterschiedlicher Reihenfolge besucht werden. Am bekanntesten sind V-, W- und F-Zyklen (Abbildung 3.18). Bei geeigneter Wahl aller übrigen Verfahrenskomponenten



**Abbildung 3.18:** V-, W- und F-Zyklus (von links nach rechts) zur Besuchsreihenfolge der Gitterebenen am Beispiel von vier Gitterebenen

liefert der V-Zyklus, der auch im Programmpaket *Nast++* verwendet wird, bereits gitterweitenunabhängige Konvergenzraten ([82]).

## Interpolation und Restriktion

Um Informationen, das heißt Ergebnisse der Rechnungen von feinen zu groben Gittern und umgekehrt von groben zu feinen Gittern transportieren zu können, müssen Restriktions- und Interpolationsoperatoren festgelegt werden. Einfache geometrische Mehrgitterverfahren verwenden meist konstante oder lineare Restriktionen und Interpolationen. Im Programm *Nast++* sind die Werte des Drucks den Mittelpunkten der Gitterzellen zugeordnet. Die Grobgitterpunkte sind daher *keine* Teilmenge der Feingitterpunkte. Die Daten werden mittels konstanter Interpolation vom groben auf das feine Gitter transportiert. Bezeichnet man mit  $n_{f,i}$  die Anzahl der Fluidzellen (des feinsten Gitters), die die Zelle  $i$  (siehe Abbildung 3.19) enthält, so erhält man also damit im zweidimensionalen Fall den Interpolations-Stern

$$I = \begin{bmatrix} \text{sign}(n_{f,1}) & \text{sign}(n_{f,2}) \\ \text{sign}(n_{f,3}) & \text{sign}(n_{f,4}) \end{bmatrix}. \quad (3.30)$$

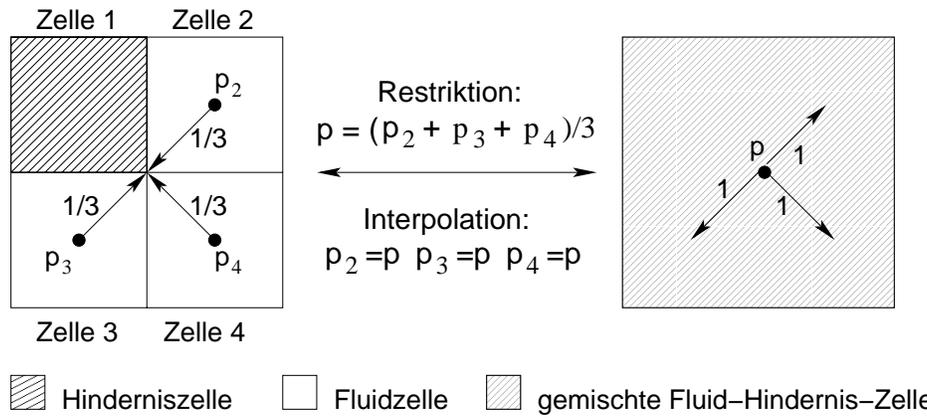
Der entsprechende Wert des Grobgitters wird also nur auf solche Punkte des nächstfeineren Gitters übertragen, bei denen die zugehörigen Zellen nicht nur aus Hinderniszellen des feinsten Gitters bestehen ( $n_{f,i} = 0$ ). Umgekehrt werden die Grobgitterwerte durch lineare Restriktion aus den Feingitterwerten abgeleitet. Ist eine Grobgitterzelle keine reine Fluidzelle, das heißt setzt sie sich aus Feingitterzellen zusammen, von denen mindestens eine eine Hinderniszelle ist, so kann der Grobgitterwert nicht mehr mittels linearer Restriktion aus den vier zugehörigen Feingitterwerten berechnet werden. In diesem Fall wird die lineare Restriktion durch die Mittelwertbildung über die Druckwerte der Feingitterfluidzellen ersetzt (Abbildung 3.19). Sind alle Feingitterzellen Fluidzellen, so entspricht die Mittelwertbildung gerade der linearen Restriktion. Verallgemeinert man dieses Vorgehen auf die nachfolgenden Vergrößerungsschritte, so erhält man im zweidimensionalen Fall den Restriktions-Stern

$$R = \begin{bmatrix} \frac{n_{f,1}}{n_f} & \frac{n_{f,2}}{n_f} \\ \frac{n_{f,3}}{n_f} & \frac{n_{f,4}}{n_f} \end{bmatrix} \quad (3.31)$$

wobei  $n_f$  ist die Gesamtzahl der Fluidzellen in der Grobgitterzelle ist (das heißt  $n_f = n_{f,1} + n_{f,2} + n_{f,3} + n_{f,4}$ ). Die dreidimensionale Restriktion und Interpolation sind analog festgelegt.

## Grobgitterdiskretisierung der Operatoren

Für jede Gitterebene ist eine eigene Diskretisierung der Gleichungen anzugeben. Die ersten geometrischen Mehrgitterverfahren verwendeten zur Feingitterdiskretisierung analoge Differenzen-Quotienten auf den groben Gittern. Dieses Vorgehen ist intuitiv leicht verständlich, die resultierenden Verfahren sind jedoch im Allgemeinen nur mit Einschränkungen zur Lösung von Gleichungssystemen auf Gebieten mit komplizierten Hindernisstrukturen geeignet ([82]), da die Geometrie des Gebietes auf den groben Gitterebenen nicht ausreichend genau berücksichtigt werden kann. Algebraische Mehrgitterverfahren (AMG) dagegen beruhen auf einer rein operatorabhängigen Wahl von Restriktion, Interpolation und Grobgitterdiskretisierung. Die Grobgitter-Differenzen-Quotienten werden mit Hilfe der Restriktion und Interpolation automatisch aus der Feingitterdiskretisierung berechnet. AMG-Verfahren lassen



**Abbildung 3.19:** Konstante Interpolation und Restriktion durch Mittelwertbildung für die Druckwerte im Zellmittelpunkt im zweidimensionalen Fall

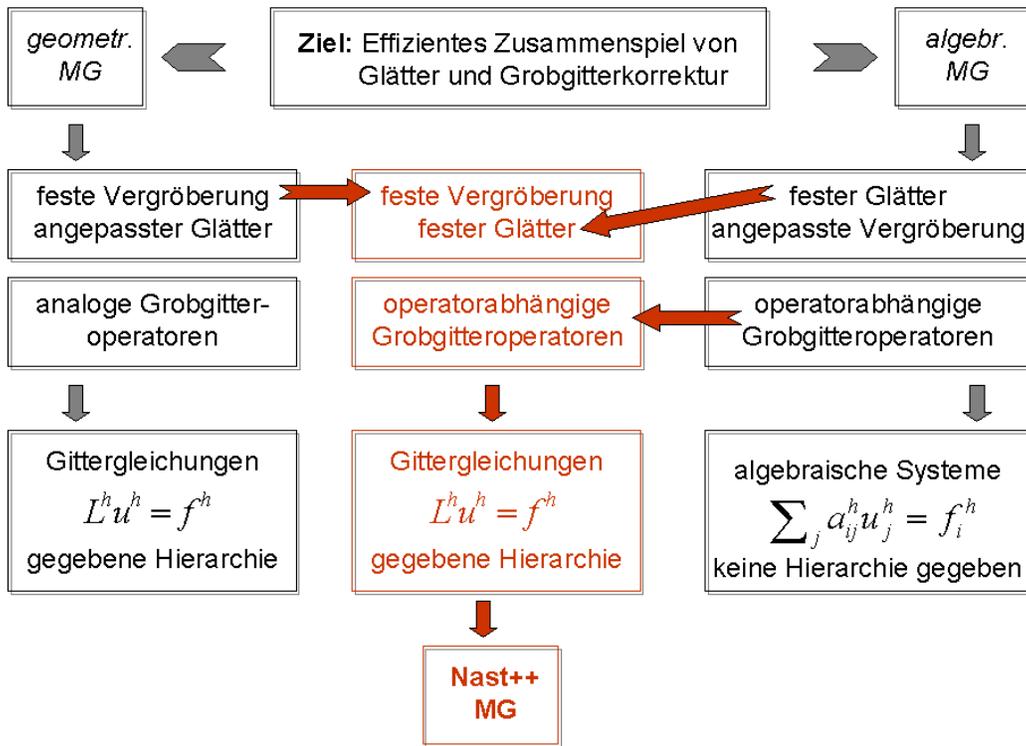
sich daher auf beliebige Gleichungssysteme anwenden, erreichen aber häufig kein optimales Konvergenzverhalten, da räumliche Symmetrien in den Konstruktionsregeln der Verfahren nicht berücksichtigt werden. So besteht beispielsweise nach den Kriterien für die Definition der Interpolation bei AMG-Verfahren oft kein qualitativer Unterschied zwischen Interpolationsformeln erster und solchen zweiter Ordnung ([80]). In dieser Arbeit wurde daher ein geometrisches Mehrgitterverfahren entwickelt, das jedoch die Komplexität der Geometrie wie bei algebraischen Mehrgitterverfahren über eine operatorabhängige Berechnung der Grobgitter-Differenzen-Quotienten berücksichtigt. Abbildung 3.20 zeigt einen Vergleich dieses Ansatzes mit geometrischen und algebraischen Mehrgitterverfahren. Die Differenzen-Quotienten auf den Grobgittern werden dabei wie beim Galerkin-Verfahren ([1]) durch Anwendung der Restriktions- und Interpolationsoperatoren auf die Feingitter-Differenzen-Quotienten bestimmt. Der Grobgitter-Differenzen-Stern  $L_G$  ist danach in Abhängigkeit vom Feingitter-Stern  $L$  definiert als

$$L_G = RLI. \quad (3.32)$$

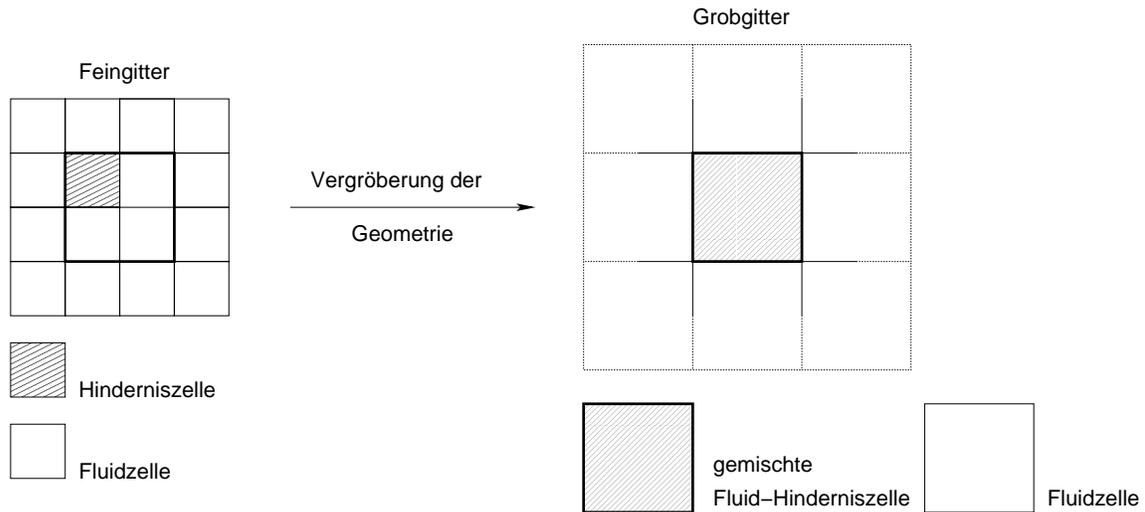
Abbildung 3.22 zeigt die Bildung des Differenzen-Sterns an einem zweidimensionalen Beispiel (zugehörige Geometrievergrößerung siehe Abbildung 3.21). Für reine Fluidzellen erhält man auf diese Weise den Grobgitter-Differenzen-Stern

$$\frac{1}{4h^2} \begin{bmatrix} & 2 & \\ 2 & -8 & 2 \\ & 2 & \end{bmatrix},$$

also genau das Zweifache des Analogons zum Feingitter-Differenzen-Stern. Die auf dem groben Gitter berechnete Lösung ist daher größenordnungsmäßig nur halb so groß wie die tatsächliche Lösung des ursprünglichen Gleichungssystems. Das resultierende Mehrgitterverfahren würde daher keine von der Auflösung des feinsten Gitters unabhängige Konvergenzrate erreichen ([80]). Aus diesem Grund werden die Differenzen-Sterne nach jedem Vergrößerungsschritt mit dem Faktor  $\frac{1}{2}$  multipliziert.



**Abbildung 3.20:** Stellung des in Nast++ verwendeten Mehrgitterverfahrens zwischen geometrischen und algebraischen Mehrgitterverfahren



**Abbildung 3.21:** Zweidimensionales Beispiel für die Vergrößerung der Geometrie

Anwendung des Restriktionsoperators (RL):

$$\left( \frac{1}{h^2} \begin{array}{c} 1 \\ \begin{array}{|c|c|c|} \hline \text{shaded} & -3 & 1 \\ \hline \end{array} \\ 1 \end{array} + \frac{1}{h^2} \begin{array}{c} \begin{array}{|c|c|} \hline \text{shaded} & \\ \hline \end{array} \\ \begin{array}{|c|c|c|} \hline 1 & -3 & 1 \\ \hline \end{array} \\ 1 \end{array} + \frac{1}{h^2} \begin{array}{c} \begin{array}{|c|c|} \hline \text{shaded} & 1 \\ \hline \end{array} \\ \begin{array}{|c|c|c|} \hline 1 & -4 & 1 \\ \hline \end{array} \\ 1 \end{array} \right) \cdot 3 = \\
 \frac{1}{3h^2} \begin{array}{c} 1 \\ \begin{array}{|c|c|c|} \hline \text{shaded} & -2 & 1 \\ \hline \end{array} \\ \begin{array}{|c|c|c|} \hline 1 & -2 & -2 & 1 \\ \hline \end{array} \\ \begin{array}{|c|c|} \hline 1 & 1 \\ \hline \end{array} \end{array}$$

Anwendung des Interpolationsoperators ((RL)I):

$$\frac{1}{3h^2} \begin{array}{c} 1 \\ \begin{array}{|c|c|c|} \hline 1 & \begin{array}{|c|} \hline (-2)+(-2) \\ +(-2) \\ \hline \end{array} & 1+1 \\ \hline \end{array} \\ 1+1 \end{array} = \frac{1}{3h^2} \begin{array}{c} 1 \\ \begin{array}{|c|c|c|} \hline 1 & \begin{array}{|c|} \hline -6 \\ \hline \end{array} & 2 \\ \hline \end{array} \\ 2 \end{array}$$

**Abbildung 3.22:** Zweidimensionales Beispiel für die Berechnung des Grobgitter-Differenzensterns für den Geometrieausschnitt aus Abbildung 3.21

### Glätter

Um auf jedem Gitter eine Approximation der zugehörigen Lösung berechnen zu können, muss ein geeignetes iteratives Verfahren – in diesem Zusammenhang als Glätter bezeichnet – gewählt werden. *Nast++* verwendet dazu das Gauß-Seidel-Verfahren, da die Wahl eines Überrelaxationsfaktors  $\omega > 1$  beim SOR-Verfahren im Hinblick auf die Glättereigenschaften,

das heißt den Reduktionsfaktor der hochfrequenten Fehleranteile, nicht optimal ist ([82]).

Zusätzlich zum Glätter sind die Anzahl  $m_1$  der Glätteriterationen vor dem Übergang auf das nächstgrößere Gitter (Vorglättung) sowie die Anzahl  $m_2$  der Glätteriterationen vor dem Übergang auf das nächstfeinere Gitter (Nachglättung) festzulegen. Die Nachglättung ist nötig, da die Korrektur der Feingitterlösung durch die Lösung des Grobgittersystems auf dem feinen Gitter zwar den „glatten“ Fehleranteil reduziert, gleichzeitig aber neue hochfrequente Fehleranteile entstehen können.  $m_1$  und  $m_2$  sind üblicherweise sehr klein, da die Lösung auf jedem Gitter nach wenigen Gauß-Seidel-Iterationen bereits „glatt“ wird und solche glatten Fehler auf dem nächstgrößeren Gitter behandelt werden können.

Eine diskrete Form der Ausgangsgleichung selbst wird nur auf dem feinsten Gitter gelöst, auf allen größeren Gittern betrachtet man lediglich eine Gleichung für den Fehler des nächstfeineren Gitters. Es entsteht folgender rekursiver Algorithmus für eine Iteration des Mehrgitterverfahrens (V-Zyklus):

- 1) führe  $m_1$  Gauß-Seidel-Iterationen für die Gleichung  $[\Delta p] = rhs$  durch (Vorglättung);
- 2) berechne die Restriktion  $res_G$  des Residuums  $[\Delta p] - rhs$  auf dem nächstgrößeren Gitter;
- 3) führe einen V-Zyklus für die Gleichung  $[\Delta e]_G = res_G$  durch;
- 4) berechne die Interpolation  $e$  der berechneten Grobgitterapproximation  $e_G$  des Fehlers auf dem aktuellen Gitter;
- 5) berechne eine verbesserte Lösung auf dem aktuellen Gitter als  $p := p - e$ ;
- 6) führe  $m_2$  Gauß-Seidel-Iterationen für die Gleichung  $[\Delta p] = rhs$  durch (Nachglättung);

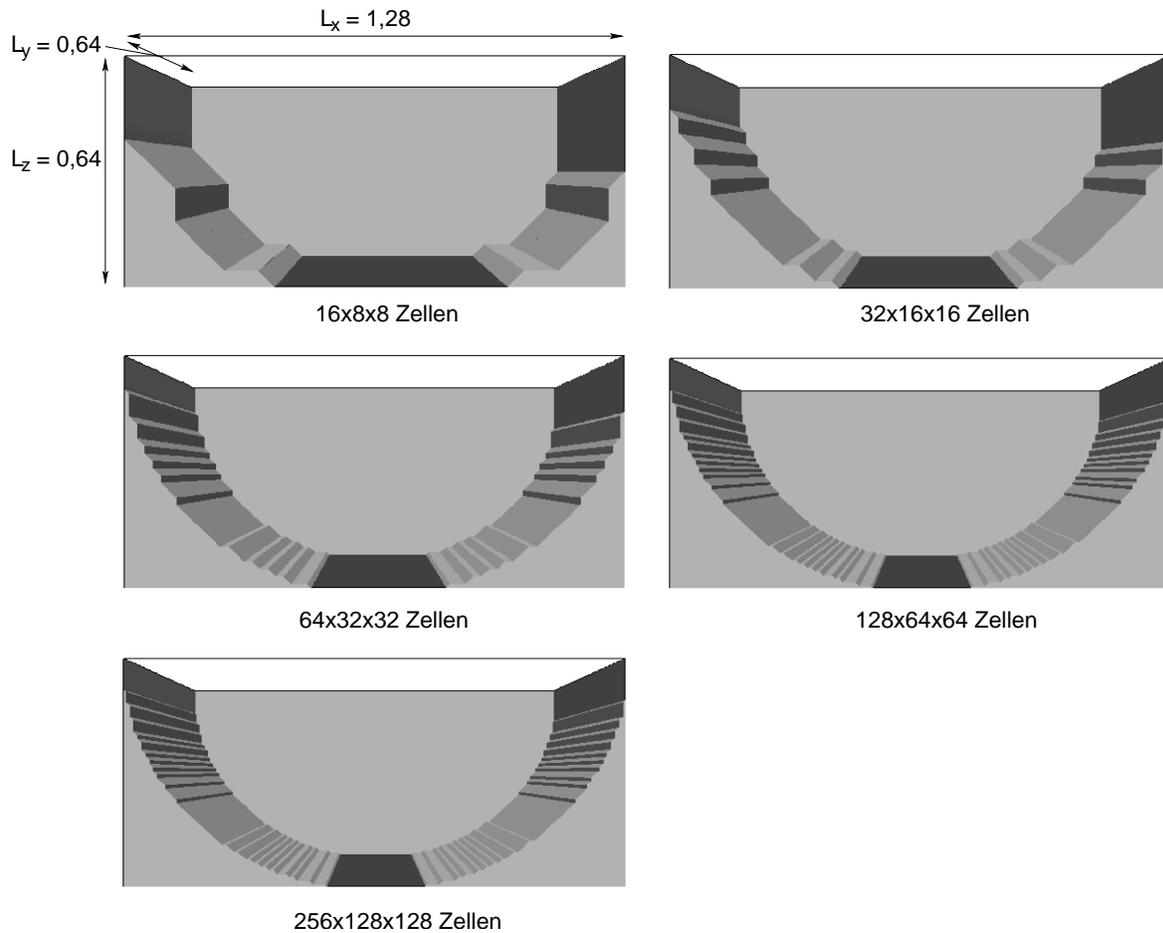
Dieser V-Zyklus wird so lange wiederholt, bis das Residuum auf dem feinsten Gitter unterhalb der geforderten Schranke liegt.

### 3.3.4 Beispiel Nischen-Strömung

Um die Konvergenzeigenschaften des Mehrgitterverfahrens zu testen und die Rechenzeiten von  $SOR$ - und Mehrgitterverfahren zu vergleichen, wurde zunächst die zeitabhängige Navier-Stokes-Gleichung für die Nischen-Strömung in einer quaderförmigen und einer abgerundeten Nische (siehe Abbildung 3.23) gelöst. Dabei wurde die Gitterauflösung schrittweise von  $16 \times 8 \times 8$  auf  $256 \times 128 \times 128$  Gitterzellen erhöht.

Es wurde eine Strömung mit Reynoldszahl zehn und folgenden Randbedingungen berechnet:

- obere Grenzfläche des Gebiets: INFLOW mit  $u \equiv 1$  und  $v \equiv w \equiv 0$ ,



**Abbildung 3.23:** Geometrie für die runde Nische in den verschiedenen Gitterauflösungen

- vordere, hintere, rechte und linke und untere Grenzfläche des Gebiets sowie Nischenbegrenzung: NOSLIP

Als Genauigkeitsschranke für die diskrete  $L_2$ -Norm

$$\|res\| := \sqrt{\frac{1}{\text{Anzahl Fluidzellen}} \sum_{i,j,k} \left( [\Delta p^{(it)}]_{i,j,k} - rhs_{i,j,k} \right)^2} \quad (3.33)$$

des Residuum der Poissongleichung wurde  $\epsilon_{poisson} = 10^{-8}$  festgelegt. Zu Beginn der Rechnung ( $t = 0$ ) wurden alle Geschwindigkeiten in den Fluidzellen auf null gesetzt (außer am oberen Rand). Es wurden die Rechenzeiten, die Anzahlen der nötigen Iterationen beziehungsweise V-Zyklen und die Konvergenzraten bei der Lösung der Poissongleichung des ersten Zeitschrittes verglichen.

Aus den Gleichungen (3.26) bis (3.28) erhält man für die verschiedenen Gitterauflösungen die optimalen Relaxationsparameter für hindernisfreie, quaderförmige Gebiete mit Dirichlet-Randbedingungen:

Gitterauflösung	$\omega_{opt}$
$16 \times 8 \times 8$	1,5002
$32 \times 16 \times 16$	1,9444
$64 \times 32 \times 32$	1,9857
$128 \times 64 \times 64$	1,9964
$256 \times 128 \times 128$	1,9990

Man beachte, dass obige Parameter für die beiden Beispielprobleme nur eine Näherung der tatsächlich optimalen Relaxationsparameter sind, da das Berechnungsgebiet im Fall der runden Nische nicht quaderförmig ist und in beiden Beispielen für den Druck homogene Neumann-Randbedingungen statt Dirichlet-Randbedingungen gelten. Für hindernisfreie quaderförmige Gebiete entsprechender Gitterauflösung mit Dirichlet-Randbedingungen ergeben sich nach (3.26), (3.27) und (3.29) folgende asymptotische Konvergenzraten der drei in 3.3.1 und 3.3.2 beschriebenen Relaxationsverfahren:

Verfahren	Gitterauflösung	Konvergenzrate
Jacobi-Verfahren	$16 \times 8 \times 8$	0,9429
	$32 \times 16 \times 16$	0,9856
	$64 \times 32 \times 32$	0,9964
	$128 \times 64 \times 64$	0,9991
	$256 \times 128 \times 128$	0,9998
Gauß-Seidel-Verfahren	$16 \times 8 \times 8$	0,8890
	$32 \times 16 \times 16$	0,9714
	$64 \times 32 \times 32$	0,9928
	$128 \times 64 \times 64$	0,9982
	$256 \times 128 \times 128$	0,9995
SOR-Verfahren	$16 \times 8 \times 8$	0,5002
	$32 \times 16 \times 16$	0,9444
	$64 \times 32 \times 32$	0,9857
	$128 \times 64 \times 64$	0,9964
	$256 \times 128 \times 128$	0,9990

Für jede Gitterauflösung wurde die Poissongleichung des ersten Zeitschrittes der Navier-Stokes-Gleichungen zunächst mit dem Gauß-Seidel- und dem SOR-Verfahren mit *Nast++* auf einem Pentium-III-(800-MHz)-Prozessor gelöst und die Zahl  $I$  der Iterationen, die benötigte Rechenzeit sowie die Konvergenzrate

$$\rho := \sqrt{\|res\|^{(I)} / \|res\|^{(I-1)}} * \sqrt{\|res\|^{(I-1)} / \|res\|^{(I-2)}} * \sqrt{\|res\|^{(I-2)} / \|res\|^{(I-3)}} * \dots,$$

protokolliert, die für  $I$  gegen unendlich gegen die asymptotische Konvergenzrate konvergiert.  $\|res\|^{(i)}$  bezeichnet dabei die diskrete  $L_2$ -Norm (siehe (3.33)) des Residuums der Poissongleichung nach der  $i$ -ten Iteration des Relaxationsverfahrens. Die Ergebnisse dieser Testrechnungen sind in den folgenden Tabellen zusammengefasst:

quaderförmige Nische				
Löser	Gitterauflösung	Iterationen	Rechenzeit	Konvergenzrate
Gauß-Seidel-Verfahren	$16 \times 8 \times 8$	1.808	$7,4 \cdot 10^{+0}$ sec	0,9774
	$32 \times 16 \times 16$	7.046	$1,4 \cdot 10^{+2}$ sec	0,9940
	$64 \times 32 \times 32$	18.222	$2,9 \cdot 10^{+3}$ sec	0,9984
	$128 \times 64 \times 64$	114.634	$7,4 \cdot 10^{+4}$ sec	0,9996
	$256 \times 128 \times 128$	—	—	—
SOR-Verfahren	$16 \times 8 \times 8$	736	$3,0 \cdot 10^{+0}$ sec	0,9451
	$32 \times 16 \times 16$	598	$1,1 \cdot 10^{+1}$ sec	0,9274
	$64 \times 32 \times 32$	1.040	$1,1 \cdot 10^{+2}$ sec	0,9557
	$128 \times 64 \times 64$	3.678	$2,4 \cdot 10^{+3}$ sec	0,9911
	$256 \times 128 \times 128$	14.864	$6,8 \cdot 10^{+4}$ sec	0,9979

runde Nische				
Löser	Gitterauflösung	Iterationen	Rechenzeit	Konvergenzrate
Gauß-Seidel-Verfahren	$16 \times 8 \times 8$	1.486	$5,7 \cdot 10^{+0}$ sec	0,9721
	$32 \times 16 \times 16$	5.490	$8,7 \cdot 10^{+1}$ sec	0,9921
	$64 \times 32 \times 32$	21.504	$2,0 \cdot 10^{+3}$ sec	0,9979
	$128 \times 64 \times 64$	86.356	$5,2 \cdot 10^{+4}$ sec	0,9995
	$256 \times 128 \times 128$	—	—	—
SOR-Verfahren	$16 \times 8 \times 8$	624	$2,1 \cdot 10^{+0}$ sec	0,9343
	$32 \times 16 \times 16$	532	$8,4 \cdot 10^{+0}$ sec	0,9173
	$64 \times 32 \times 32$	960	$8,3 \cdot 10^{+1}$ sec	0,9605
	$128 \times 64 \times 64$	3.620	$2,1 \cdot 10^{+3}$ sec	0,9913
	$256 \times 128 \times 128$	15.356	$5,9 \cdot 10^{+4}$ sec	0,9994

Für die Gitterauflösung  $128 \times 64 \times 64$  Zellen ist also das SOR-Verfahren bereits 31-mal beziehungsweise 24-mal schneller als das Gauß-Seidel-Verfahren. Vergleicht man die erzielten Konvergenzraten mit den theoretisch für quaderförmige hindernisfreie Gebiete mit Dirichlet-Randbedingungen errechneten Raten, so lässt sich sowohl für das Gauß-Seidel- als auch für das SOR-Verfahren bei der niedrigsten Gitterauflösung eine deutlich schlechtere Konvergenz als prognostiziert erkennen, während die Konvergenzraten bei höherer Gitterauflösung in etwa den berechneten Raten entsprechen. Dieser Effekt kann dadurch erklärt werden, dass der Einflussbereich der Randbedingungen an den äußeren Rändern bei geringer Gitterauflösung wesentlich größer ist als bei sehr feinen Gittern und damit weder die Konvergenzraten noch der optimale Relaxationsparameter genau berechnet werden können.

Dieselben Gleichungen wurden nun auch mit dem Mehrgitterverfahren gelöst. Statt der Iterationen des Relaxationsverfahrens werden hier die nötigen V-Zyklen bis zum Unterschreiten der vorgegebenen Genauigkeitsschranke  $\epsilon_{poisson}$  gezählt. Die Grobgitterapproximationen der Differential-Operatoren sind aufgrund der konstanten Interpolation höchstens erster Ordnung genau. Dies lässt sich sehr leicht daran ablesen, dass durch die Anwendung von Restriktions- und Interpolationsoperatoren auf die Feingitterdiskretisierung für bestimmte Geometrien, die sich in der Hindernisstruktur um  $\mathcal{O}(h)$  unterscheiden (siehe Abbildung 3.24), bis auf einen konstanten Faktor dieselben Grobgitter-Differenzen-Sterne entstehen (siehe Abbildung 3.25).

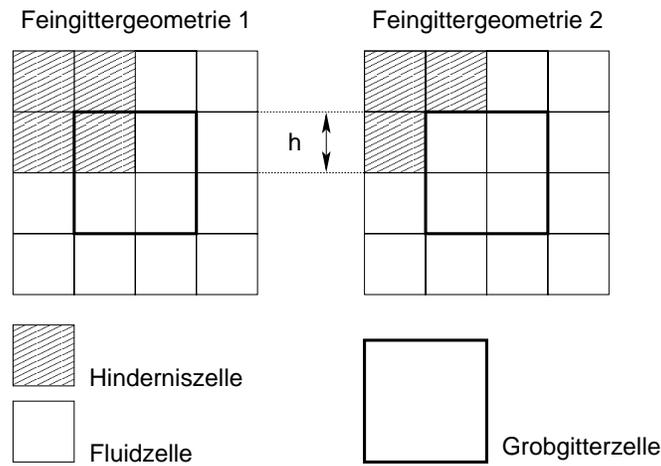


Abbildung 3.24: Beispiel für zwei zweidimensionale Fluid-Hinderniss-Geometrien, die sich um  $\mathcal{O}(h)$  unterscheiden

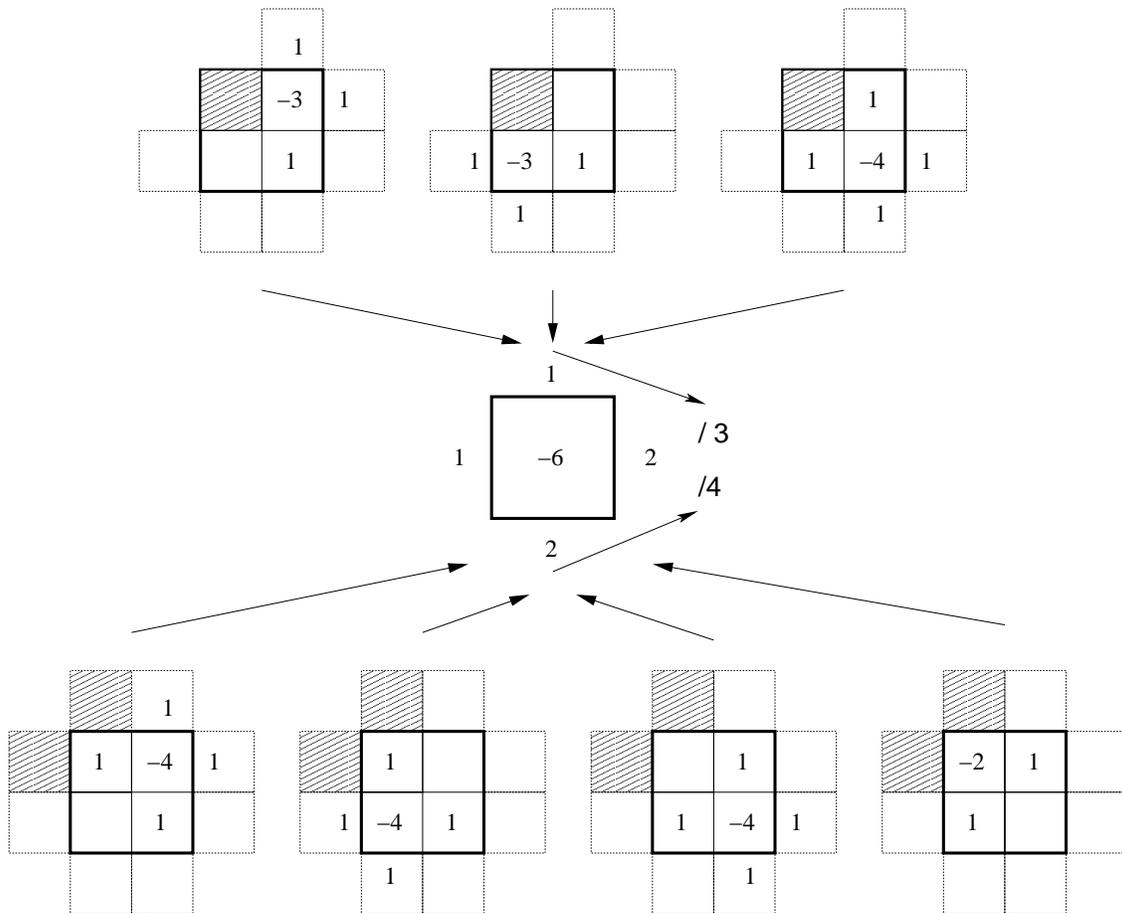


Abbildung 3.25: Entstehung gleicher Differenzen-Sterne trotz unterschiedlicher Geometrien (Abbildung 3.24) bei Verwendung des 5-Punkte-Sterns und der Restriktion durch Mittelwertbildung sowie konstanter Interpolation für die zweidimensionale Poissongleichung

Um die schlechtere Approximationsordnung auf den groben Gittern der runden Nische auszugleichen, wurde neben einer Testrechnung mit  $m_1 = m_2 = 2$  eine Simulation mit doppelt so vielen Nachglätter- wie Vorglätteriterationen durchgeführt:

quaderförmige Nische – Mehrgitterverfahren					
$m_1$	$m_2$	Gitterauflösung	V-Zyklen	Rechenzeit	Konvergenzrate
2	2	$16 \times 8 \times 8$	13	$3,5 \cdot 10^{-1}$ sec	0,3147
		$32 \times 16 \times 16$	13	$1,8 \cdot 10^0$ sec	0,2396
		$64 \times 32 \times 32$	14	$1,1 \cdot 10^1$ sec	0,3453
		$128 \times 64 \times 64$	14	$8,8 \cdot 10^1$ sec	0,2063
		$256 \times 128 \times 128$	15	$6,7 \cdot 10^2$ sec	0,3191
2	4	$16 \times 8 \times 8$	8	$3,1 \cdot 10^{-1}$ sec	0,0743
		$32 \times 16 \times 16$	9	$1,6 \cdot 10^0$ sec	0,2097
		$64 \times 32 \times 32$	9	$1,0 \cdot 10^1$ sec	0,1158
		$128 \times 64 \times 64$	10	$7,5 \cdot 10^1$ sec	0,2592
		$256 \times 128 \times 128$	10	$5,8 \cdot 10^2$ sec	0,1628

runde Nische – Mehrgitterverfahren					
$m_1$	$m_2$	Gitterauflösung	V-Zyklen	Rechenzeit	Konvergenzrate
2	2	$16 \times 8 \times 8$	16	$4,2 \cdot 10^{-1}$ sec	0,3487
		$32 \times 16 \times 16$	17	$1,9 \cdot 10^0$ sec	0,4105
		$64 \times 32 \times 32$	19	$1,4 \cdot 10^1$ sec	0,4394
		$128 \times 64 \times 64$	19	$1,0 \cdot 10^2$ sec	0,3897
		$256 \times 128 \times 128$	19	$7,7 \cdot 10^2$ sec	0,3102
2	4	$16 \times 8 \times 8$	12	$3,9 \cdot 10^{-1}$ sec	0,1825
		$32 \times 16 \times 16$	12	$1,9 \cdot 10^0$ sec	0,2838
		$64 \times 32 \times 32$	14	$1,3 \cdot 10^1$ sec	0,3191
		$128 \times 64 \times 64$	14	$9,6 \cdot 10^1$ sec	0,3036
		$256 \times 128 \times 128$	14	$7,1 \cdot 10^2$ sec	0,2261

Bei einer Auflösung von  $256 \times 128 \times 128$  Gitterzellen ist das Mehrgitterverfahren also 117-mal (quaderförmige Nische) beziehungsweise 83-mal (runde Nische) schneller als das SOR-Verfahren mit approximativ optimalem Relaxationsparameter. Ferner zeigt die obige Rechnung, dass eine Verdopplung der Nachglätteriterationen insbesondere für die runde Nische kaum eine Verbesserung der Konvergenz und der Rechenzeiten bringt. Die Anzahl der benötigten V-Zyklen ist bei beiden Varianten des Mehrgitterverfahrens annähernd unabhängig von der Gitterauflösung – sowohl für das hindernisfreie quaderförmige Gebiet als auch für die runde Nische, bei der auf den groben Gittern gemischte Fluid-Hindernis-Zellen auftreten und die Differenzen-Quotienten somit von den Operatoren auf dem feinsten Gitter abweichen.

Der Speicheraufwand des Mehrgitterverfahrens ist im dreidimensionalen Fall nicht wesentlich größer als der des SOR-Verfahrens. Es werden lediglich folgende Daten zusätzlich benötigt:

- Felder zur Speicherung der Werte von Geschwindigkeiten, Druck und Konzentration auf den groben Gitterebenen:

Da sich die Zahl der Unbekannten bei der gewählten Vergrößerungsstrategie (Halbierung der Gitterweite in jede Raumrichtung, siehe 3.3.3) beim Übergang zum nächstgrößeren Gitter jeweils um den Faktor  $1/8$  reduziert, ist dieser Zusatzaufwand nach oben beschränkt durch

$$S \cdot \sum_{m=1}^{\infty} \frac{1}{8^m} = \frac{1}{7}S,$$

falls  $S$  der Speicherbedarf des feinsten Gitters ist.

- Speicherung der modifizierten Differenzen-Quotienten auf den groben Gittern:

Da die Differenzen-Quotienten auf jeder Gitterebene durch 5-Punkt-Sterne dargestellt werden können und die Summe aller Elemente eines solchen Sterns jeweils verschwindet, müssen pro Zelle mit modifiziertem Quotienten nur vier Fließkommazahlen gespeichert werden. Ferner treten solche Quotienten nur an Hindernisgrenzen auf, das heißt der erforderliche Speicheraufwand ist bei glattem Grenzverlauf nicht proportional zur Zahl  $N$  der Gitterpunkte des feinsten Gitters, sondern nur zu  $N^{2/3}$ .

Folgende Tabelle für den Speicheraufwand bei der Lösung der Navier-Stokes-Gleichungen mit den verschiedenen Poissonlösern zeigt, dass der Speicheraufwand auf einem Gebiet ohne inneren Hindernisse (quaderförmige Nische) tatsächlich beim Mehrgitterverfahren höchstens um ein Siebtel höher liegt als beim SOR-Verfahren.

quaderförmige Nische		
Löser	Gitterauflösung	Speicherbedarf
SOR-Verfahren	$16 \times 8 \times 8$	1,1 MByte
	$32 \times 16 \times 16$	2,1 MByte
	$64 \times 32 \times 32$	9,0 MByte
	$128 \times 64 \times 64$	61,0 MByte
	$256 \times 128 \times 128$	451,0 MByte
Mehrgitterverfahren	$16 \times 8 \times 8$	1,1 MByte
	$32 \times 16 \times 16$	2,3 MByte
	$64 \times 32 \times 32$	10,0 MByte
	$128 \times 64 \times 64$	69,2 MByte
	$256 \times 128 \times 128$	508,0 MByte

Da der Rand der runden Nische sehr glatt ist, steigt der Speicherbedarf auch durch die in diesem Fall notwendige Speicherung zusätzlicher Differenzen-Quotienten kaum an:

runde Nische		
Löser	Gitterauflösung	Speicherbedarf
SOR-Verfahren	$16 \times 8 \times 8$	1,1 MByte
	$32 \times 16 \times 16$	2,1 MByte
	$64 \times 32 \times 32$	9,1 MByte
	$128 \times 64 \times 64$	61,0 MByte
	$256 \times 128 \times 128$	451,0 MByte
Mehrgitterverfahren	$16 \times 8 \times 8$	1,2 MByte
	$32 \times 16 \times 16$	2,4 MByte
	$64 \times 32 \times 32$	10,4 MByte
	$128 \times 64 \times 64$	69,5 MByte
	$256 \times 128 \times 128$	509,0 MByte

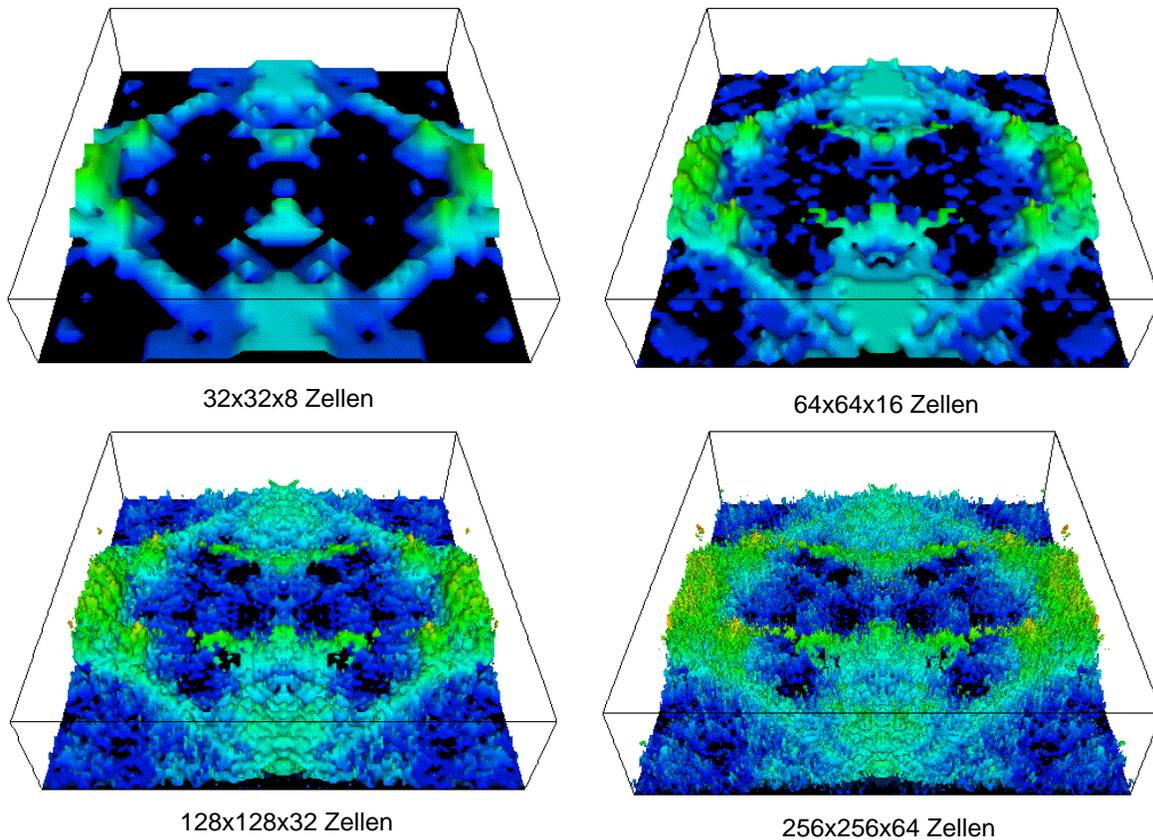
Das entwickelte Mehrgitterverfahren ermöglicht somit bei diesen Beispielen die Lösung der Poissongleichung mit nahezu auflösungsunabhängiger Konvergenzrate und bereits auf Gittern mit geringer Zellenzahl gegenüber den Relaxationsverfahren erheblich reduzierter Rechenzeit. Die Berücksichtigung der genauen Fluid-Hindernis-Geometrie erfolgt dabei automatisch über die operatorabhängige Restriktion der Differenzen-Quotienten für Grobgitterzellen mit Fluid- und Hindernis-Anteil und ihre Nachbarzellen. Da die Differenzen-Quotienten nur für diese Zellen explizit berechnet und gespeichert werden und ferner eine sehr einfache Gitter-Vergrößerungs-Strategie (Verdopplung der Gitterweiten in allen drei Raumrichtungen) verwendet werden konnte, fallen auch die Setup-Phase und die Speicherung der Differenzen-Quotienten kaum ins Gewicht (siehe obige Tabellen). Dies zeichnet das entwickelte Verfahren gegenüber algebraischen Mehrgitterverfahren aus, die für krummlinige Geometrien ähnlich robust sind und ebenfalls automatisch die Gebietsgeometrie berücksichtigen.

### 3.3.5 Beispiel Biofilm

Als typisches Beispiel für eine Biofilmgeometrie wurde ein experimentell definierter, sechs Tage alter *Sphingomonas* sp. LB126 Monokulturbiofilm betrachtet. Abbildung 3.26 zeigt die Geometrie in den vier betrachteten Gitterauflösungen mit  $32 \times 32 \times 8$ ,  $64 \times 64 \times 16$ ,  $128 \times 128 \times 32$  und  $256 \times 256 \times 64$  Gitterzellen.

Die simulierte Strömung hat die Reynoldszahl zehn und folgende Randbedingungen:

- obere Grenzfläche des Gebiets: SLIP,
- linke Grenzfläche des Gebiets: periodische Randbedingungen für die Geschwindigkeiten und konstanter Druck  $p \equiv 1$ ,
- rechte Grenzfläche des Gebiets: periodische Randbedingungen für die Geschwindigkeiten und konstanter Druck  $p \equiv 0$ ,
- vordere und hintere Grenzfläche des Gebiets: periodische Randbedingungen für die Geschwindigkeiten und Neumann-Randbedingungen für den Druck,
- untere Grenzfläche des Gebiets und Biofilmoberfläche: NOSLIP



**Abbildung 3.26:** Geometrie eines *Sphingomonas sp. LB126* Monokulturbiofilms in verschiedenen Gitterauflösungen (Gebietsgröße ca.  $250 \times 250 \times 60 \mu\text{m}^3$ )

Als Genauigkeitsschranke für das Residuum der Poissongleichung wurde wie bei der Nischenströmung  $\epsilon_{\text{poisson}} = 10^{-8}$  festgelegt. Zu Beginn der Berechnung ( $t = 0$ ) wurden wieder alle Geschwindigkeiten in den Fluidzellen auf null gesetzt.

Auch für dieses Beispiel wurden die nach den Formeln (3.26) bis (3.28) für hindernisfreie, quaderförmige Gebiete mit Dirichlet-Randbedingungen optimalen Relaxationsparameter als Approximation der tatsächlich optimalen Parameter berechnet:

Gitterauflösung	$\omega_{\text{opt}}$
$32 \times 32 \times 8$	1,6162
$64 \times 64 \times 16$	1,7864
$128 \times 128 \times 32$	1,8868
$256 \times 256 \times 64$	1,9386

Für hindernisfreie quaderförmige Gebiete entsprechender Gitterauflösung mit Dirichlet-Randbedingungen ergeben sich damit nach (3.26), (3.27) und (3.29) folgende asymptotische Konvergenzraten der Relaxationsverfahren:

Verfahren	Gitterauflösung	Konvergenzrate
Jacobi-Verfahren	$32 \times 32 \times 8$	0,9714
	$64 \times 64 \times 16$	0,9928
	$128 \times 128 \times 32$	0,9982
	$256 \times 256 \times 64$	0,9995
Gauß-Seidel-Verfahren	$32 \times 32 \times 8$	0,9436
	$64 \times 64 \times 16$	0,9857
	$128 \times 128 \times 32$	0,9964
	$256 \times 256 \times 64$	0,9990
SOR-Verfahren	$32 \times 32 \times 8$	0,6162
	$64 \times 64 \times 16$	0,7864
	$128 \times 128 \times 32$	0,8868
	$256 \times 256 \times 64$	0,9386

Wie schon bei der Nischenströmung wurden für jede Gitterauflösung die Rechenzeit, die Anzahl der nötigen Iterationen sowie die Konvergenzrate (siehe 3.3.4) bei der Lösung der Poissongleichung des ersten Zeitschrittes verglichen:

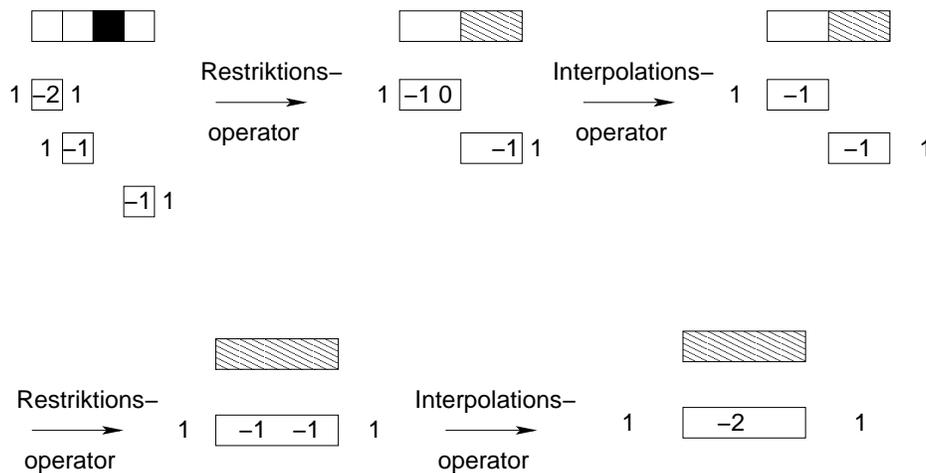
Relaxationsverfahren				
Löser	Gitterauflösung	Iterationen	Rechenzeit	Konvergenzrate
Gauß-Seidel-Verfahren	$32 \times 32 \times 8$	8.040	$2,3 \cdot 10^{+2}$ sec	0,9955
Gauß-Seidel-Verfahren	$64 \times 64 \times 16$	31.054	$5,0 \cdot 10^{+3}$ sec	0,9988
Gauß-Seidel-Verfahren	$128 \times 128 \times 32$	123.952	$1,0 \cdot 10^{+5}$ sec	0,9997
Gauß-Seidel-Verfahren	$256 \times 256 \times 64$	—	—	—
SOR-Verfahren	$32 \times 32 \times 8$	2.402	$6,9 \cdot 10^{+1}$ sec	0,9851
SOR-Verfahren	$64 \times 64 \times 16$	5.056	$8,0 \cdot 10^{+2}$ sec	0,9928
SOR-Verfahren	$128 \times 128 \times 32$	11.640	$1,2 \cdot 10^{+4}$ sec	0,9968
SOR-Verfahren	$256 \times 256 \times 64$	30.872	$2,4 \cdot 10^{+5}$ sec	0,9988

Bereits für die Gitterauflösung  $128 \times 128 \times 32$  Zellen) benötigt also das Gauß-Seidel-Verfahren fast elfmal so viele Iterationen wie das SOR-Verfahren. Die Anzahl der Iterationen vervierfacht sich bei Anwendung des Gauß-Seidel-Verfahrens bei jeder Halbierung der Gitterweite, während sie sich beim SOR-Verfahren nur etwas mehr als verdoppelt. Da der Anteil der Randzellen am Gesamtgebiet in diesem Beispiel wesentlich höher ist als bei den Nischenströmungen und somit die tatsächlich optimalen Relaxationsparameter durch die entsprechenden Formeln für hindernisfreie Gebiete mit Dirichlet-Randbedingungen weniger genau approximiert werden, ist die Konvergenz des SOR-Verfahrens jedoch schlechter als theoretisch für hindernisfreie Gebiete und Dirichlet-Randbedingungen zu erwarten.

Für das Mehrgitterverfahren mit zwei Vorglätter- und zwei beziehungsweise vier Nachglätteriterationen ergeben sich folgende Rechenzeiten, Anzahlen an V-Zyklen und Konvergenzraten:

Mehrgitterverfahren					
$m_1$	$m_2$	Gitterauflösung	V-Zyklen	Rechenzeit	Konvergenzrate
2	2	$32 \times 32 \times 8$	43	$8,0 \cdot 10^{+0}$ sec	0,7079
2	2	$64 \times 64 \times 16$	65	$7,0 \cdot 10^{+1}$ sec	0,8245
2	2	$128 \times 128 \times 32$	165	$1,3 \cdot 10^{+3}$ sec	0,9021
2	2	$256 \times 256 \times 64$	497	$2,8 \cdot 10^{+4}$ sec	0,9267
2	4	$32 \times 32 \times 8$	31	$7,9 \cdot 10^{+0}$ sec	0,6379
2	4	$64 \times 64 \times 16$	46	$6,6 \cdot 10^{+1}$ sec	0,7971
2	4	$128 \times 128 \times 32$	115	$1,2 \cdot 10^{+3}$ sec	0,8586
2	4	$256 \times 256 \times 64$	347	$1,7 \cdot 10^{+4}$ sec	0,9299

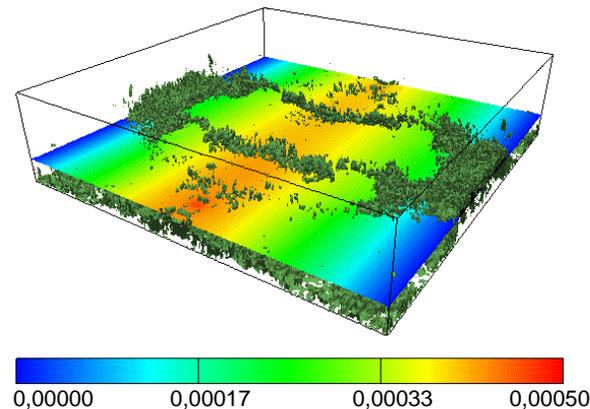
Bei einer Auflösung von  $256 \times 256 \times 64$  Gitterzellen ist das Mehrgitterverfahren also 14-mal schneller als das SOR-Verfahren mit approximativ optimalem Relaxationsparameter. Anders als noch bei der Nischenströmung in der runden Nische sind hier jedoch die Anzahl der V-Zyklen und die Konvergenzraten nicht mehr unabhängig von der Auflösung des Rechengitters. Dieser neue Effekt lässt sich dadurch erklären, dass aufgrund der einfachen konstanten Interpolation des Mehrgitterverfahrens auf dem feinen Gitter ursprünglich durch Hindernisse getrennte Fluidbereiche im Laufe der Vergrößerung der Differenzenoperatoren „zusammenwachsen“ können, das heißt die Trennung auf dem groben Gitter an den Differenzenquotienten nicht mehr zu erkennen ist (siehe Abbildung 3.27). Der Effekt dieser unphysikalischen Differenzen-Quotienten muss auf dem nächstfeineren Gitter im Laufe der Nachglättung wieder korrigiert werden, man benötigt daher mit steigender Gitterauflösung insgesamt mehr Iterationen. Daher lassen sich im obigen Beispiel die Konvergenzraten und Rechenzeiten durch eine höhere Anzahl an Nachglätteriterationen deutlich verbessern. Solange diese „falschen“ Differenzen-Quotienten nicht sehr häufig auftreten, ist das Mehrgitterverfahren immernoch wesentlich schneller als das SOR-Verfahren (siehe obige Tabellen) oder ein Mehrgitterverfahren, bei dem die Gittervergrößerung abgebrochen würde, sobald das erste Mal getrennte Fluid-Gebiete „zusammenwachsen“ würden.



**Abbildung 3.27:** Zusammenwachsen getrennter Gebiete im Laufe der Vergrößerung bei Verwendung des 3-Punkte-Sterns und zellzentrierter Gitter für die eindimensionale Poissongleichung

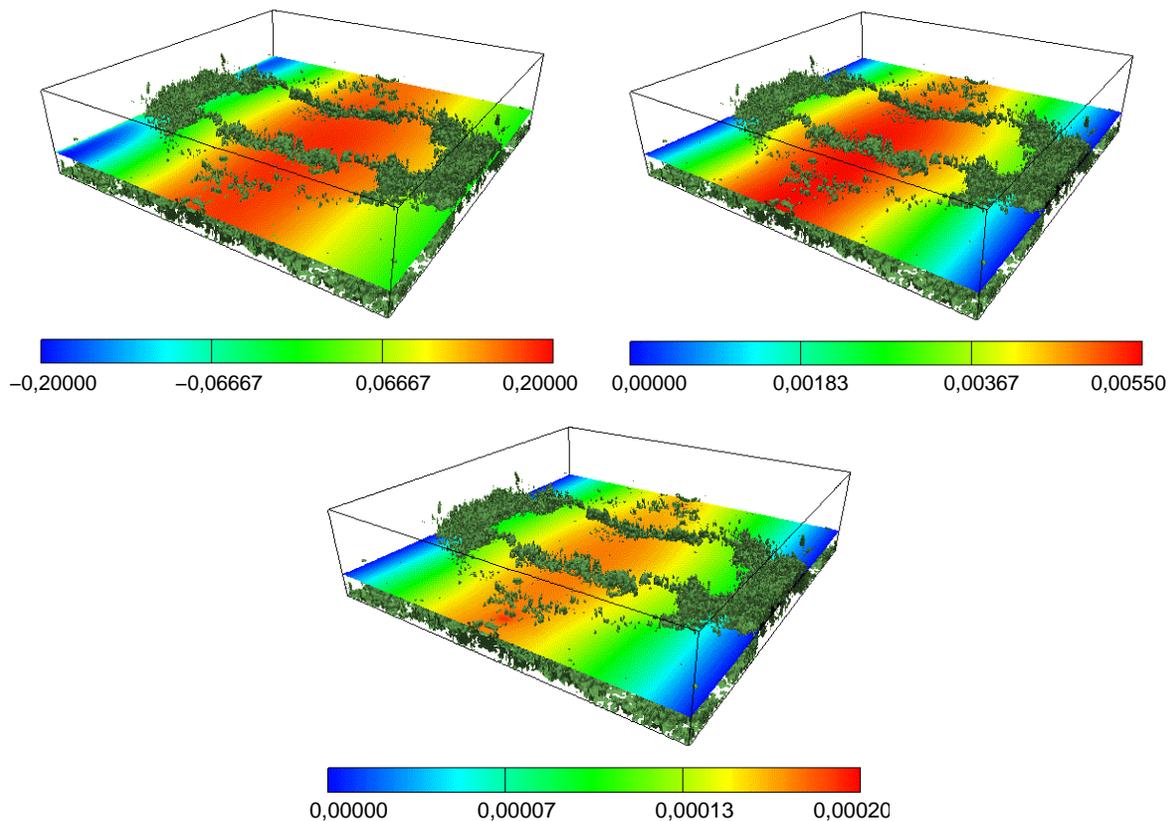
Mögliche Lösungen dieses Problems werden im Anhang B.3 diskutiert.

In obigen Tabellen wurden lediglich die Anzahlen der Iterationen beziehungsweise V-Zyklen bis zum Unterschreiten einer vorgegebenen Genauigkeitsschranke  $\epsilon_{poisson}$  für die Norm des Residuums der Poissongleichung verglichen. Da der Fehler der berechneten Lösungsapproximation jedoch noch relativ groß sein kann, obwohl das lokale Residuum überall klein ist, muss zusätzlich der Fehler betrachtet werden. Dazu wurde eine sehr genaue Lösung der Poissongleichung des ersten Zeitschrittes (berechnet mit dem Mehrgitterverfahren und  $\epsilon_{poisson} = 10^{-8}$ ) als Approximation der exakten Lösung verwendet und die Differenzen zu den Lösungen des SOR-Verfahrens und des Mehrgitterverfahrens ( $m_1 = m_2 = 2$ ) mit  $\epsilon_{poisson} = 10^{-3}$  gebildet. Die Abbildungen 3.28 und 3.29 zeigen den Fehlerverlauf für die beiden Verfahren in einer horizontalen Schnittebene bei  $z = 16 \mu\text{m}$ . Obwohl mit beiden Verfahren bis zur selben Schranke für die Norm des Residuums iteriert wurde, ist der Fehler bei Verwendung des Mehrgitterverfahrens bereits um den Faktor 2,5 kleiner als beim SOR-Verfahren. Der Restfehler zeigt bei beiden Verfahren einen glatten Verlauf. Der glatte Fehleranteil konnte also mit dem Mehrgitterverfahren zwar besser als mit dem SOR-Verfahren, jedoch noch nicht optimal reduziert werden. Dieses Verhalten lässt sich wiederum auf das oben erwähnte Problem „zusammenwachsender“ Teilgebiete zurückführen. Abbildung 3.29 zeigt außerdem, dass bereits nach drei V-Zyklen des Mehrgitterverfahrens nur noch die maximal glatten Fehleranteile nicht verschwunden sind. Dieselbe Betrachtung konnte für das einfachere Beispiel der Nischenströmung nicht durchgeführt werden, da in diesem Fall die Poissongleichung mit homogenen Neumann-Randbedingungen an allen Ränder gelöst wurde und daher die Lösung nur bis auf eine Konstante bestimmt ist.



**Abbildung 3.28:** Fehler der Lösungsapproximation der diskreten Poissongleichung des ersten Zeitschrittes mit dem SOR-Verfahren mit Genauigkeitsschranke  $\epsilon_{poisson} = 10^{-3}$  (11.808 Iterationen) für die diskrete  $L_2$ -Norm des Residuums

Das Hindernisgebiet besitzt in diesem Beispiel einen stark zerfaserten Rand. Der Aufwand zur Speicherung zusätzlicher Grobgitterdifferenzenquotienten für das Mehrgitterverfahren liegt trotzdem nur geringfügig höher als bei der Strömung in der runden Nische. Dies zeigt ein Vergleich der Differenz des in der folgenden Tabelle angegebenen Speicheraufwandes des SOR- und des Mehrgitterverfahrens mit einem Siebtel des Aufwandes des SOR-Verfahrens.



**Abbildung 3.29:** Fehler der Lösungsapproximation der diskreten Poissongleichung des ersten Zeitschrittes mit dem Mehrgitterverfahren nach einem V-Zyklus (oben links), nach 3 V-Zyklen (oben rechts) und mit Genauigkeitsschranke  $\epsilon_{poisson} = 10^{-3}$  für die diskrete  $L_2$ -Norm des Residuums (54 V-Zyklen, unten)

Löser	Gitterauflösung	Speicherbedarf
SOR-Verfahren	$32 \times 32 \times 8$	2,2 MByte
	$64 \times 64 \times 16$	9,3 MByte
	$128 \times 128 \times 32$	61,9 MByte
	$128 \times 128 \times 32$	520,0 MByte
Mehrgitterverfahren	$32 \times 32 \times 8$	2,5 MByte
	$64 \times 64 \times 16$	10,8 MByte
	$128 \times 128 \times 32$	71,2 MByte
	$128 \times 128 \times 32$	584,0 MByte

### 3.4 Modularität

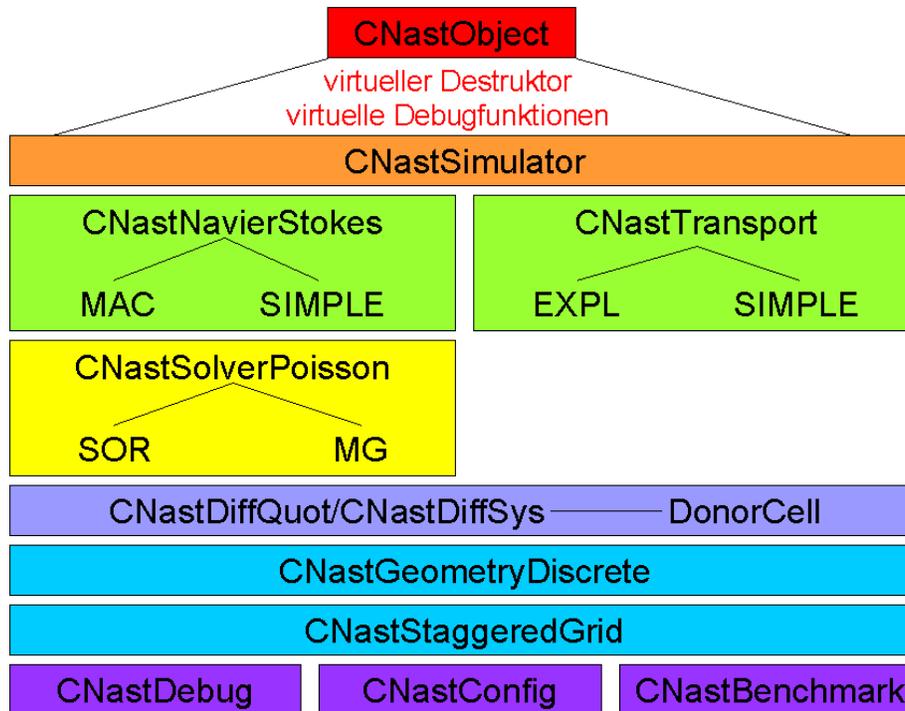
Während eindeutig definierte und möglichst allgemeine Schnittstellen nach außen (siehe 3.5) die nötige Vielseitigkeit und Flexibilität des Programmes in Bezug auf die Einbindung in eine weitere Simulationsumgebung garantieren, werden Flexibilität und Überschaubarkeit in Bezug auf den Programmcode selbst durch den modularen objektorientierten Aufbau von

*Nast++* gewährleistet. Sind die einzelnen Module ausreichend unabhängig voneinander, so können beispielsweise Gleichungslöser, Geometriebeschreibung und verwendete Differenzenquotienten jederzeit ausgetauscht oder in mehreren Varianten zur Auswahl gestellt werden. Ebenso können weitere Module wie Löser für zusätzliche Gleichungen im Zuge von Modellerweiterungen ergänzt werden, ohne Änderungen in allen Programmteilen durchführen zu müssen.

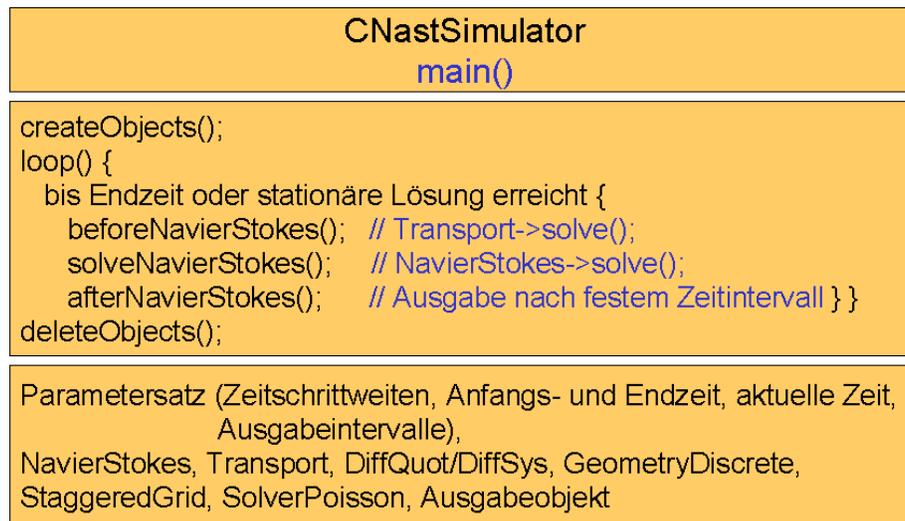
Die Module von *Nast++* sind hierarchisch auf unterschiedlichen Abstraktionsebenen angeordnet. Die oberste Ebene bildet der Simulator (`CNastSimulator`), der den gesamten Programmablauf steuert. Darunter befinden sich unmittelbar die Module für die Lösung der unterschiedlichen am simulierten Modell beteiligten Differentialgleichungen (bisher Navier-Stokes- und Transportgleichungen), gefolgt von weiteren Modulen, die zur Lösung einzelner Teile dieser Gleichungen (bisher der Poissongleichung (3.10) für den Druck als Teil der Navier-Stokes-Gleichungen) benötigt werden. Die weiteren Klassen betreffen die Diskretisierung der kontinuierlichen Differentialgleichungen (`CNastDiffQuot/CNastDiffSys`), die Geometriebeschreibung (`CNastGeometryDiscrete`) und das verwendete Gitter (`CNastStaggeredGrid`) für die zu berechnenden Werte (bisher Geschwindigkeit  $\vec{u}$ , Druck  $p$  und Temperatur beziehungsweise Konzentration  $c$ ). Die unterste Ebene bilden `CNastDebug` (Definition von Ausgaben und Programmabbruch zur Fehlersuche), `CNastConfig` (Einstellungen zur bedingten Compilierung von Programmteilen wie zum Beispiel der Debuggingfunktionen) und `CNastBenchmark` (Funktionen zur Messung der Rechenzeit einzelner Programmteile). Alle Klassen leiten sich von der gemeinsamen Basisklasse `CNastObject` ab, die einen virtuellen Destruktor und virtuelle Debugfunktionen zur Überprüfung eines Objekts auf Gültigkeit seines Zustandes und zur Ausgabe wichtiger Debugginginformationen des Objekts definiert (siehe [10]). Abbildung 3.30 zeigt die wichtigsten Module von *Nast++* und ihre Hierarchie. Nicht explizit erwähnt wurden hier grundlegende Klassen zur Speicherung und zum Zugriff auf Objekte (Arrays, Listen, etc.) sowie Ein- und Ausgabeklassen (Näheres dazu in [10]).

### 3.4.1 Simulator

Der Simulator (`CNastSimulator`) erzeugt die Objekte für den Navier-Stokes- und eventuell den Transportlöser, die Differenzenquotienten, die diskrete Geometrie, das versetzte Gitter für Geschwindigkeiten, Druck und gegebenenfalls Temperatur oder Konzentration, den Poissonlöser und die Ergebnisausgabe. Die zum Aufruf der entsprechenden Konstruktoren benötigten Parameter werden über ein eigenes Einleseobjekt aus einer Parameterdatei (`npa-Datei`, siehe 3.5) gelesen. Anschließend wird die Zeitschleife für die zeitabhängigen Navier-Stokes- (und Transport-) Gleichungen gestartet. Die hierzu notwendige Schrittweitensteuerung (siehe [29]) erfolgt ebenfalls direkt im Simulator. Innerhalb der Zeitschleife werden die drei Funktionen `beforeNavierStokes`, `solveNavierStokes` und `afterNavierStokes` aufgerufen. Durch diese allgemeine Formulierung lassen sich von der implementierten Simulatorklasse weitere Klassen mit unterschiedlichen Erweiterungen des Umfangs der simulierten Modellgleichungen aber derselben Schnittstelle zum Start der Zeitschleife ableiten. In dieser Arbeit werden der Löser der Transportgleichungen in `beforeNavierStokes`, der Löser der Navier-Stokes-Gleichungen in `solveNavierStokes` und die Ergebnisausgabe in `afterNavierStokes` aufgerufen. Zum Abschluss der Simulation werden alle Objekte wieder gelöscht (Abbildung 3.31).



**Abbildung 3.30:** Die wichtigsten Bausteine des objektorientierten Strömungs- und Transportcodes Nast++

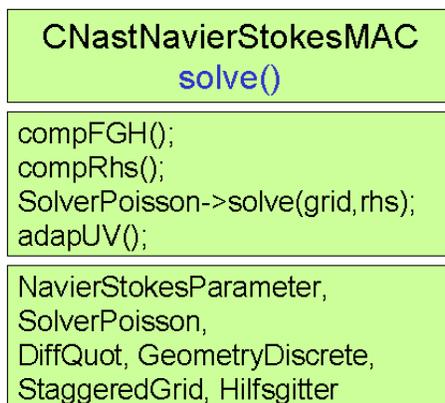


**Abbildung 3.31:** Die Klasse **CNastSimulator** zur Steuerung des gesamten Simulationslaufs mit der Schnittstelle **main**, einer Ablaufskizze der Funktion **main** und den wichtigsten Klassenmembern

### 3.4.2 Navier-Stokes-Löser

Zur Ausführung eines Zeitschritts der Navier-Stokes-Gleichungen stehen die beiden Söhne **CNastNavierStokesMAC** (explizite Zeitdiskretisierung) und **CNastNavierStokesSIMPLE** (se-

miimplizite Zeitdiskretisierung) der abstrakten virtuellen Basisklasse `CNastNavierStokes` zur Verfügung. Die Basisklasse definiert die Schnittstelle `solve` des Löser. `solve` führt einen einzelnen Zeitschritt durch. Die nötigen Information zur Zeitsteuerung sind im Parameter-Memberobjekt enthalten. Im Folgenden wird nur der Löser für die explizite Zeitdiskretisierung beschrieben. Für jeden Zeitschritt wird hier zunächst mit Hilfe der Impulsgleichung (2.18) das vorläufige Geschwindigkeitsfeld  $\vec{U}$  berechnet, aus diesem die rechte Seite  $rhs$  der Poissongleichung für den Druck ermittelt, der Löser der Poissongleichung aufgerufen und zum Schluss das endgültige Geschwindigkeitsfeld  $\vec{u}$  bestimmt (Abbildung 3.32). Die Herleitung dieses Algorithmus' wurde in 3.2.3 dargestellt.



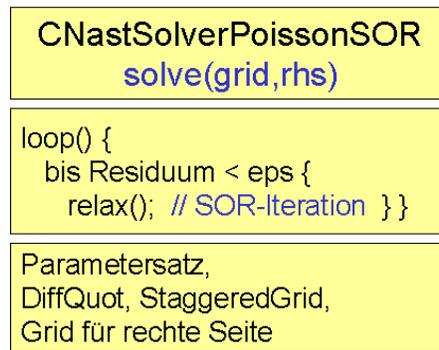
**Abbildung 3.32:** Die Klasse `CNastNavierStokesMAC` für explizite Zeitschritte der Navier-Stokes-Gleichungen mit der Schnittstelle `solve`, einer Ablaufskizze der Funktion `solve` und den wichtigsten Klassenmitgliedern

### 3.4.3 Poissonlöser

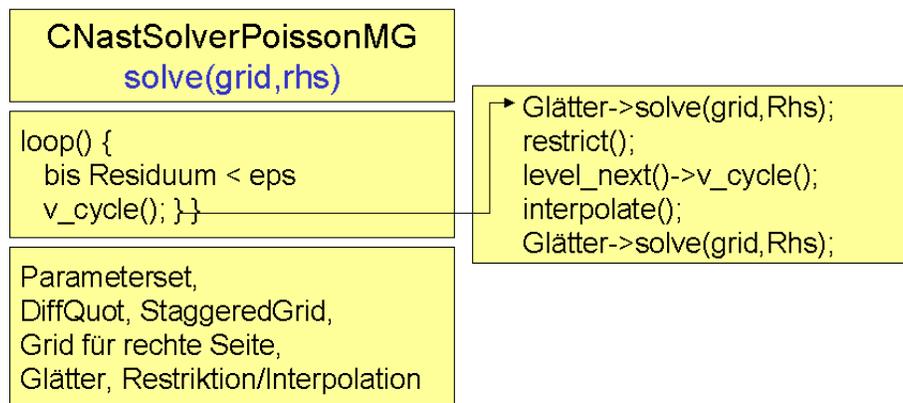
Die Schnittstelle zum Löser der Poissongleichung wird durch die abstrakte virtuelle Basisklasse `CNastSolverPoisson` beschrieben. Der Löser wird über die Funktion `solve` mit dem Druckgitter und dem Gitter der rechten Seite als Parameter aufgerufen. In `Nast++` sind zwei mögliche iterative Löser implementiert (siehe 3.3): Der SOR-Löser (`CNastSolverPoissonSOR`, Abbildung 3.33) und ein Mehrgitterlöser (`CNastSolverPoissonMG`, Abbildung 3.34). Die Iteration wird jeweils so lange fortgesetzt, bis die diskrete  $L_2$ -Norm (siehe (3.33)) des Residuums der diskretisierten Poissongleichung unter eine vorgegebene Schranke  $eps_{poisson}$  fällt.

### 3.4.4 Transportlöser

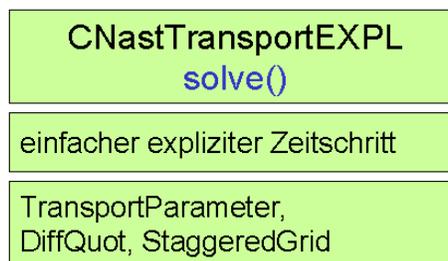
Analog zu den Navier-Stokes-Gleichungen stehen auch für die Transportgleichungen (2.20) und (2.21) sowohl eine Klasse für explizite Zeitschritte (`CNastTransportEXPL`) als auch für semiimplizite Zeitschritte (`CNastTransportSIMPLE`) zur Verfügung. Beide Klassen leiten sich von der abstrakten virtuellen Basisklasse `CNastTransport` ab. Da die Transportgleichungen kein Gleichungssystem sondern eine skalare Gleichung – mit je nach Zuordnung des aktuellen Ortes zum Fluid oder zum Hindernis unterschiedlichen Koeffizienten – bilden, ist die Ausführung des Zeitschrittes (zumindest im expliziten Fall) trivial (Abbildung 3.35).



**Abbildung 3.33:** Die Klasse `CNastSolverPoissonSOR` für den SOR-Löser der Druck-Poissongleichung (3.10) mit der Schnittstelle `solve`, einer Ablaufskizze der Funktion `solve` und den wichtigsten Klassenmitgliedern



**Abbildung 3.34:** Die Klasse `CNastSolverPoissonMG` für den Mehrgitter-Löser der Druck-Poissongleichung (3.10) mit der Schnittstelle `solve`, einer Ablaufskizze der Funktion `solve` sowie des V-Zyklus' `v_cycle` und den wichtigsten Klassenmitgliedern



**Abbildung 3.35:** Die Klasse `CNastTransportEXPL` für explizite Zeitschritte der Transportgleichungen mit der Schnittstelle `solve` und den wichtigsten Klassenmitgliedern

### 3.4.5 Differenzen-Quotienten

Die Schnittstelle zu den Differenzen-Quotienten ist in den abstrakten virtuellen Basisklassen `CNastDiffQuot` beziehungsweise `CNastDiffSys` definiert. `CNastDiffQuot` stellt Funktionen

zur Approximation sämtlicher in den Navier-Stokes- ((2.18) und (2.19)) und den Transportgleichungen ((2.20) und (2.21)) vorkommenden Differential-Operatoren (siehe 3.2.4) zur Verfügung. `CNastDiffSys` wird für die semiimpliziten Navier-Stokes- und Transportlöser zur Diskretisierung der nichtlinearen Terme  $(\vec{u} \cdot \nabla) \vec{u}$  und  $(\vec{u} \cdot \nabla) c$  benötigt, die in diesem Fall mit Hilfe der Werte des aktuellen *und* des nächsten Zeitpunktes berechnet werden.

Zu beiden Basisklassen sind in den abgeleiteten Klassen `CNastDiffQuotDonorCell` und `CNastDiffSysDonorCell` die aktuell ausschließlich verwendeten Quotienten definiert (siehe 3.2.4). Für den Mehrgitterlöser der Poissongleichung wird die um modifizierte Differenzen-Quotienten für grobe Gitterebenen erweiterte Klasse `CNastDiffQuotDonorCellMG` benötigt. Bei der konkreten Auswertung der Differenzen-Quotienten wird auf die Daten zu Geschwindigkeiten, Druck und Temperatur oder Konzentration nicht direkt sondern über die in der Klasse für das versetzte Gitter (siehe 3.4.7) definierten Lesefunktionen zugegriffen. Durch geeignete Wahl dieser Funktionen (siehe 3.2.4) können am Rand des Berechnungsgebiets und an allen Fluid-Hinderniss-Grenzen dieselben Differenzenquotienten verwendet werden wie im Inneren des Gebiets. Die Unterscheidung zwischen Rand- und inneren Zellen findet somit erst auf der untersten Ebene der Klassenhierarchie statt.

### 3.4.6 Diskrete Geometrie

Die diskrete Geometrie (`CNastGeometryDiscrete`) wird in *Nast++* im Wesentlichen durch die Belegung der Quaderzellen des verwendeten Gitters mit Fluid oder Hindernis und die Art der Ränder, das heißt die Randbedingungen festgelegt. Die Struktur des Fluid-Hinderniss-Gebiets wird durch ein dreidimensionales 0-1-Array beschrieben und im Klassenmember Zellenfeld gespeichert. Für die Randbedingungen sind die Member `Randbedingungstabelle` und `Randzellentabelle` zuständig (Abbildung 3.36). Die `Randbedingungstabelle` enthält alle in der aktuellen Simulationsrechnung vorkommenden Randbedingungstypen und die diesen gegebenenfalls zugeordneten Werte (zum Beispiel Geschwindigkeiten für `INFLOW`, Druck für `CONSTPRESS`). Die `Randzellentabelle` enthält für jede Zelle, die an eine Fluid-Hinderniss-Grenze oder einen äußeren Rand grenzt, und für jede Raumrichtung (Nord, Süd, Ost, West, Oben und Unten) einen Zeiger auf den Eintrag der `Randbedingungstabelle`, der an der betreffenden Quaderfläche gelten soll. Der Zugriff auf den einer Zelle zugeordneten Eintrag der `Randzellentabelle` erfolgt über die Indizes  $(i, j, k)$  der Zelle und eine Hashfunktion (siehe [63] und Abbildung 3.37).



**Abbildung 3.36:** Die Klasse `CNastGeometryDiscrete` für Geometrie und Randwerte und die wichtigsten Klassenmember

Das Zellenfeld ist dabei nicht mit der Randbedingungs- und Randzellentabelle verknüpft, das heißt, die Objekte der Klasse `Zelle`, aus denen sich das Zellenfeld zusammensetzt, können

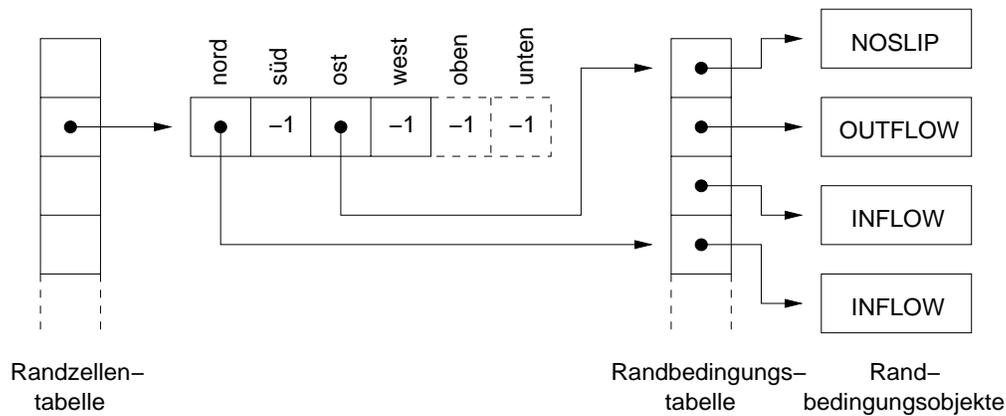


Abbildung 3.37: Datenstruktur der Randbedingungen (aus [63])

nicht direkt auf die Einträge der Randwerttabelle zu ihren sechs Grenzflächen zugreifen. Die reine Geometriebeschreibung ist somit unabhängig von der Randwertdefinition.

### 3.4.7 Versetztes Gitter

Das versetzte Gitter (CNastStaggeredGrid) ist die grundlegende Datenstruktur für die berechneten Geschwindigkeits-, Druck- und Temperatur- oder Konzentrationswerte. Es enthält sämtliche lesende und schreibende Zugriffsfunktionen auf die Werte einer aktuellen Gitterzelle und ihrer unmittelbaren Nachbarn (Abbildungen 3.38 und 3.39). Die Zugriffsfunktionen auf die Nachbarn einer Zelle dienen zur Kapselung der Randbedingungen und müssen im Falle einer Randzelle je nach geltender Randbedingung und Randwerten geeignet implementiert sein. Entsprechende Definitionen des Zugriffs existieren für alle in *Nast++* zulässigen Randbedingungen (siehe Tabelle 3.2). Die genaue Definition für die unterschiedlichen Randbedingungen wird in 3.2.4 angegeben.

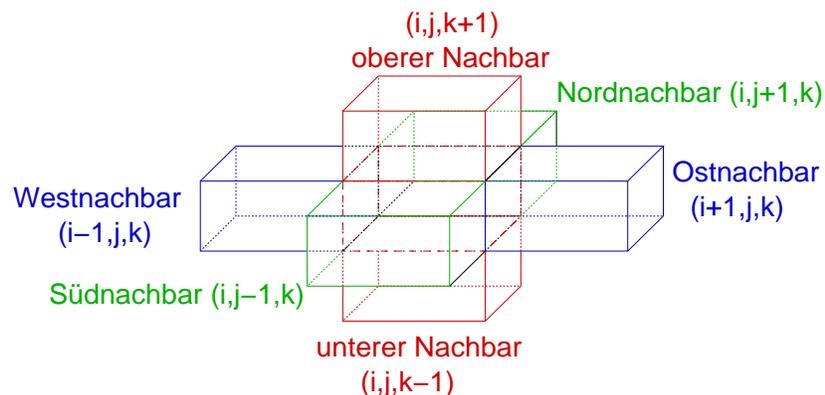
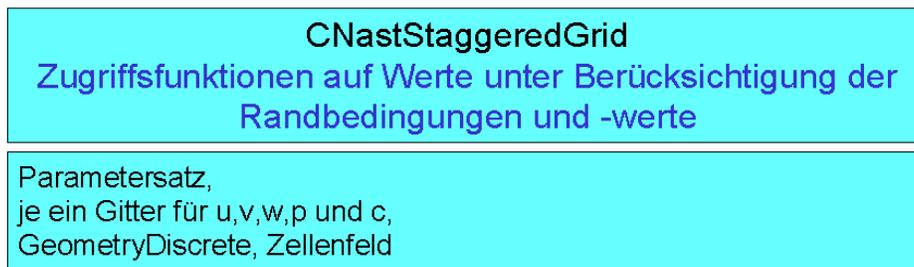


Abbildung 3.38: Quaderzelle des *Nast++*-Rechengitters (rot) mit ihren sechs unmittelbaren Nachbarn, für die Zugriffsfunktionen auf die gespeicherten Werte definiert sind



**Abbildung 3.39:** Die Klasse *CNastStaggeredGrid* für das versetzte Gitter mit den berechneten Werten für Geschwindigkeit, Druck und Temperatur beziehungsweise Konzentration mit den wichtigsten Klassenmitgliedern

### 3.5 Äußere Schnittstellen

*Nast++* wird bei der Simulation von Biofilmen ebenso wie bei vielen anderen Anwendungen nicht als alleinstehendes Programm benutzt, sondern in eine weitere Umgebung eingebunden. Dazu gehören hier die experimentellen Untersuchungen (Kapitel 1), die Zelluläre-Automaten-Software *CAsim* (Kapitel 4) und die Visualisierung der Ergebnisse (Kapitel 6). Klar definierte Schnittstellen erleichtern die Integration von *Nast++* in diesen Kontext und erhöhen gleichzeitig die Flexibilität im Hinblick auf die Ergänzung um weiterer Module im Zuge einer Modellerweiterung.

Ein- und Ausgabe erfolgen für *Nast++* im Wesentlichen über drei Dateien, die in den folgenden Abschnitten genauer beschrieben werden:

- *npa*: Parameterdatei (Eingabeschnittstelle),
- *nfv*: Geometrie und Werte für Geschwindigkeiten, Druck und Temperatur beziehungsweise Stoffkonzentration (Ein- und Ausgabeschnittstelle),
- *npc*: Randwerte (Eingabeschnittstelle).

Die Ausgabeschnittstelle wird durch verschiedene Ausgaben über die Standardausgabe (*stdout*) zur Laufzeit des Programmes ergänzt (Debugging-Informationen, aktuelle Residuen, Stand der Berechnung etc.).

#### 3.5.1 Parametereingabe

Die Eingabeschnittstelle verlangt die Beschreibung der Geometrie als 0-1-Feld, sowie verschiedene Parameter für die Gebietsgeometrie, die Löser, die Diskretisierung, die physikalischen Eigenschaften der Fluidströmung und des Wärme- bzw. Stofftransports sowie die gewünschte Ergebnisausgabe. Diese Parameter werden beim Start von *Nast++* über eine speziell formatierte Parameterdatei (*npa*-Datei, vgl. [63]) eingelesen. Für eine genaue Beschreibung der *npa*-Datei und der erforderlichen Parameter siehe Anhang A.1.

### 3.5.2 Geometrie- und Werteschnittstelle

Die – wie in 2.2.2 und 5.2 beschrieben – aus den CLSM-Schnittbildern gewonnene Geometrieinformation wird aus einer weiteren Datei (*nf v*-Datei, vgl. [63]) gelesen. Diese enthält neben dem 0-1-Feld für die Geometrie die Anzahl der Quader des Geometriegitters (einschließlich der hinzugefügten Hinderniszellen am Rand). Diese Information wird benötigt, um aus dem 0-1-Feld die Geometrie rekonstruieren zu können.

Die *nf v*-Datei dient jedoch nicht nur als Eingabeschnittstelle für die Geometrie des betrachteten Fluid-Hindernis-Gebiets, sondern ist gleichzeitig auch die Ausgabeschnittstelle für die berechneten Geschwindigkeits-, Druck- und Temperatur- beziehungsweise Konzentrationswerte. Gleichzeitig werden der Zeitpunkt, zu dem diese Werte gelten, und die Anzahl der enthaltenen Größen (1, falls nur die Geometrieinformation vorhanden ist, 6, falls Geometrie, Geschwindigkeiten, Druck und Konzentration angegeben werden) gespeichert. Die Geometrie steht in der ersten Spalte, die Geschwindigkeitskomponenten in der zweiten bis vierten, der Druck in der fünften und die Temperatur beziehungsweise Konzentration in der sechsten Spalte. Sind einige oder alle dieser Werte in der *nf v*-Datei angegeben, so werden sie beim Neustart von *Nast++* als Anfangswerte verwendet. Durch diese Identität von Ein- und Ausgabeschnittstelle kann das Programm jederzeit mit bereits berechneten oder von außen gegebenen Werten gestartet werden. Abbildung 3.40 zeigt ein Beispiel für den Anfang einer *nf v*-Datei. Zur Visualisierung der Ergebnisse mit AVS Express ([97]) wird die *nf v*-Datei mit Hilfe eines Postprozessors geeignet verarbeitet (siehe Abbildung 3.42 und 5.3) und in einer weiteren Datei (*nvs*-Datei) gespeichert. Diese Nachbearbeitung ist nötig, da die in *Nast++* berechneten Werte einem versetzten Gitter zugeordnet (siehe 3.2) und daher nicht unmittelbar zur Visualisierung geeignet sind, sondern durch Interpolation auf gemeinsame Gitterpunkte übertragen werden müssen.

### 3.5.3 Randbedingungen

Der dritte Teil der *Nast++*-Eingabeschnittstelle betrifft die Definition der Randwerte. Diese können prinzipiell für jede Grenzfläche zwischen einem Hindernis- und einem Fluidquader sowie für die Quaderflächen am äußeren Rand des Berechnungsgebiets beliebig aus den verfügbaren Randwerttypen (siehe 3.1) gewählt werden. Die Eingabe erfolgt über die sogenannte *nbc*-Datei. Diese besteht aus drei Teilen:

- allgemeine Parameter: Zellenanzahlen ohne künstliche Randzellen, Anzahl der Quader mit Randbedingungen, Anzahl der verwendeten Randbedingungstypen. Als verschiedene Randbedingungstypen zählen beispielsweise auch zwei INFLOW-Bedingungen mit unterschiedlichen zugehörigen Geschwindigkeiten.
- Spezifikation der verwendeten Randbedingungstypen: Nummer der Randbedingungsart nach Tabelle 3.2, benötigte Werte für diese Randbedingungsart (beispielsweise bis zu drei Geschwindigkeitskomponenten für INFLOW oder ein Druckwert für einen Dirichlet-Rand bezüglich des Drucks (CONSTPRESS)).
- Zuordnung der Randbedingungstypen zu den Grenzflächen der Fluidquader, die an einen Hindernisquader grenzen.

Zellenanzahlen		aktueller Zeitpunkt		
65	65	17		
0.030295e+00				
1	0.000000e+00	0.000000e+00	0.000000e+00	1.000000e+00
1	0.000000e+00	0.000000e+00	0.000000e+00	1.000000e+00
1	0.000000e+00	1.345849e+00	0.000000e+00	9.999298e-01
0	0.000000e+00	0.000000e+00	0.000000e+00	1.000000e+00
1	0.000000e+00	0.000000e+00	0.000000e+00	1.000000e+00
1	0.000000e+00	0.000000e+00	0.000000e+00	1.000000e+00
1	0.000000e+00	0.000000e+00	0.000000e+00	1.000000e+00
0	0.000000e+00	3.928459e-10	0.000000e+00	9.997724e-01
1	0.000000e+00	0.000000e+00	0.000000e+00	1.000000e+00
1	0.000000e+00	0.000000e+00	0.000000e+00	1.000000e+00
...	...	...	...	...
Geometrie	x-Geschwindigkeit	y-Geschwindigkeit	z-Geschwindigkeit	Druck

**Abbildung 3.40:** Beispiel für den Anfang einer *nfv*-Datei (dreidimensional, Wertebelegung für Geschwindigkeiten und Druck)

In Abhängigkeit davon, welche Teile des gesamten Strömungs- und Transportmodells tatsächlich simuliert werden sollen, müssen reine Strömungsrandwerte (0, 1, 2, 4, 5 und 19), reine Transportrandwerte (4, 5, 6 und 7) oder gemischte Strömungs- und Transportrandwerte (8, 9, 10, 11, 12, 13, 14, 15, 16, 17, 20 und 21) gewählt werden. Abbildung 3.41 zeigt beispielhaft den Anfang einer *nbc*-Datei.

## 3.6 Parallelisierung

### 3.6.1 Implementierung

Die Simulation von Strömungs- und Transportvorgängen in komplizierten Geometrien, wie sie in Biofilmsystemen auf der Mikroskala vorliegen, erfordert eine sehr hohe Auflösung des Berechnungsgitters. Der Speicher- und Rechenzeitaufwand ist daher sehr hoch. Eine typische Auflösung der CLSM-Schnittbilder eines Biofilms ist  $512 \times 512$  Bildpunkte. Will man beispielsweise bei hundert übereinanderliegenden Schichtbildern Strömung und Transport auf einem Gitter simulieren, dessen Auflösung gerade der Auflösung der CLSM-Daten entspricht, so benötigt man dafür 26.214.400 Gitterzellen. Dies entspricht bei einem Speicheraufwand von acht Byte pro Variable des Typs `double` einem Speicherbedarf von 1,05 GByte im Hauptspeicher für die fünf Funktionen  $u$ ,  $v$ ,  $w$ ,  $p$  und  $c$ . Um die Simulation auf Gebieten dieser oder

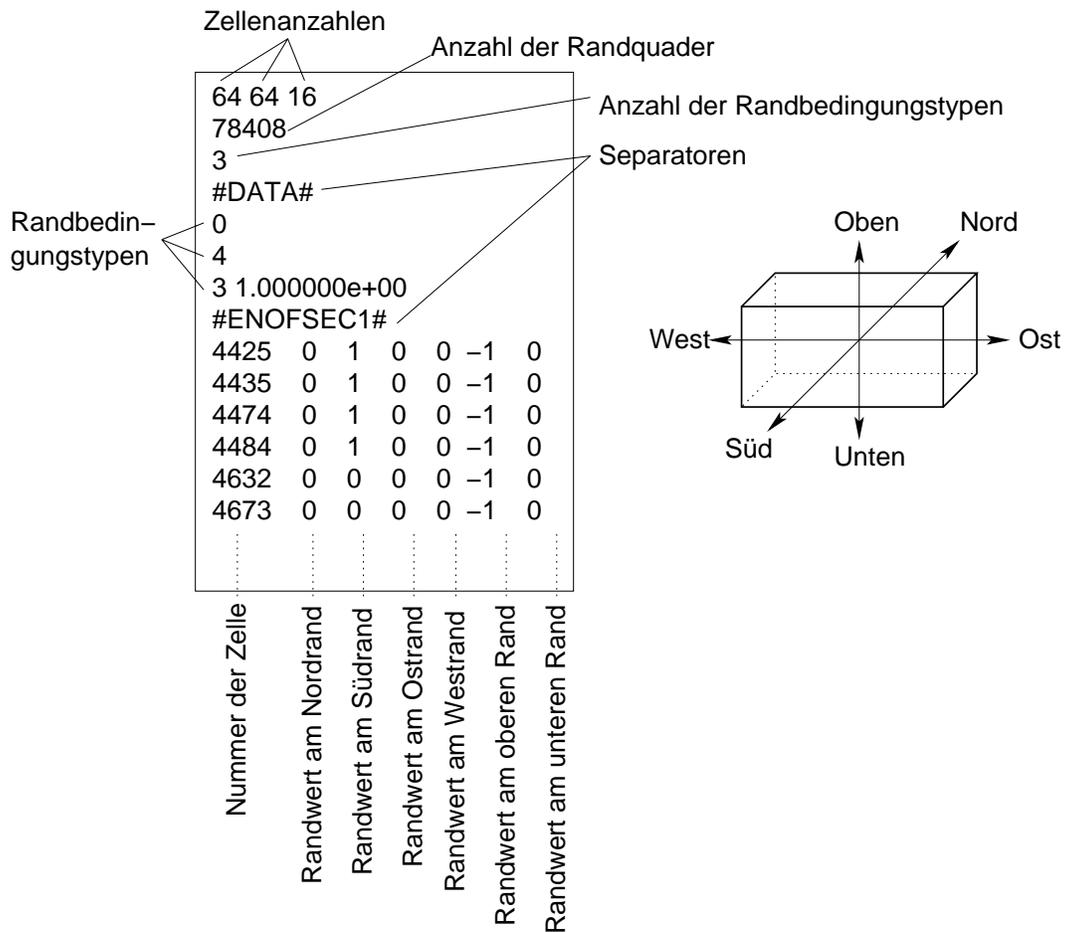
Randbedingungsart	Nummer
UNKOWN	-1
NOSLIP	0
SLIP	1
OUTFLOW	2
INFLOW	3
PERIODIC	4
ISOLATE	5
CONSTTRANS	6
DIFFUSION	7
NOSLIP_ISOLATE	8
NOSLIP_CONSTTRANS	9
NOSLIP_DIFFUSION	10
SLIP_ISOLATE	11
SLIP_CONSTTRANS	12
SLIP_DIFFUSION	13
INFLOW_ISOLATE	14
INFLOW_CONSTTRANS	15
OUTFLOW_ISOLATE	16
OUTFLOW_CONSTTRANS	17
CONSTPRESS	19
CONSTPRESS_ISOLATE	20
CONSTPRESS_CONSTTRANS	21
PERIODIC_CONSTPRESS	22

**Tabelle 3.2:** Tabelle der möglichen Randbedingungsarten in Nast++ (genaue Beschreibung in 3.1)

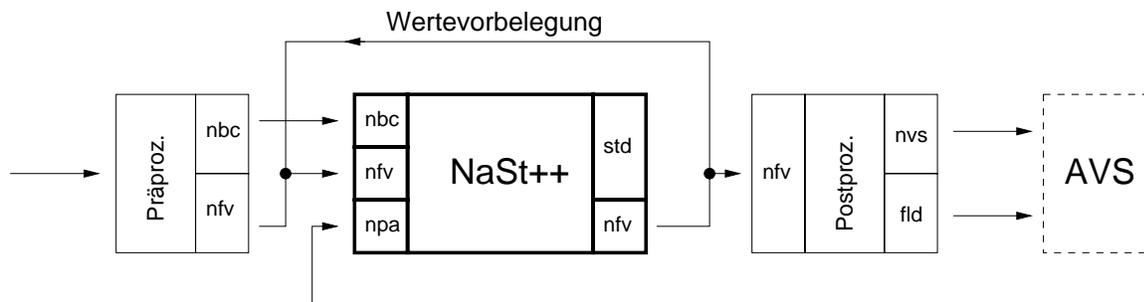
noch höherer Auflösung erst zu ermöglichen und gleichzeitig die Rechenzeiten erheblich zu verkürzen, wurde Nast++ unter MPI-II ([100]) sowohl für Workstation- und PC-Cluster als auch für Hochleistungsrechner wie zum Beispiel die VPP700 des Leibniz-Rechenzentrums München parallelisiert ([63]).

Das gesamte Simulationsgebiet wird dazu in Tensorproduktform zerlegt, das heißt, es werden durch Definition der Position trennender Ebenen in allen drei Raumrichtungen Teilquader festgelegt, die jeweils einem Prozess zugeordnet werden (siehe Abbildung 3.43). Die Gebietsunterteilung kann in diesem Sinne beliebig durchgeführt werden, die Teilquader müssen also nicht notwendigerweise gleich groß sein.

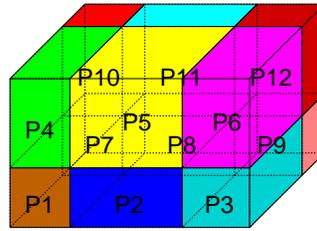
Es existiert nur eine einheitliche Programmversion von Nast++, die sowohl für sequentielle als auch für parallele Simulationsrechnungen verwendet wird. Die Unterscheidung zwischen sequentiell und parallel wird erst bei der Compilierung des Programmes über eine entsprechende Option vorgenommen. Auch die Ein- und Ausgabeschnittstellen (vgl. 3.5) bleiben dieselben. Allerdings wird im Fall der parallelen Rechnung vor dem eigentlichen Programm-  
lauf ein spezieller Präprozessor gestartet, der die ursprünglichen Eingabedateien (npa-, nfv- und nbc-Datei) in einzelne Dateien für jedes Teilgebiet beziehungsweise jeden Teilprozess



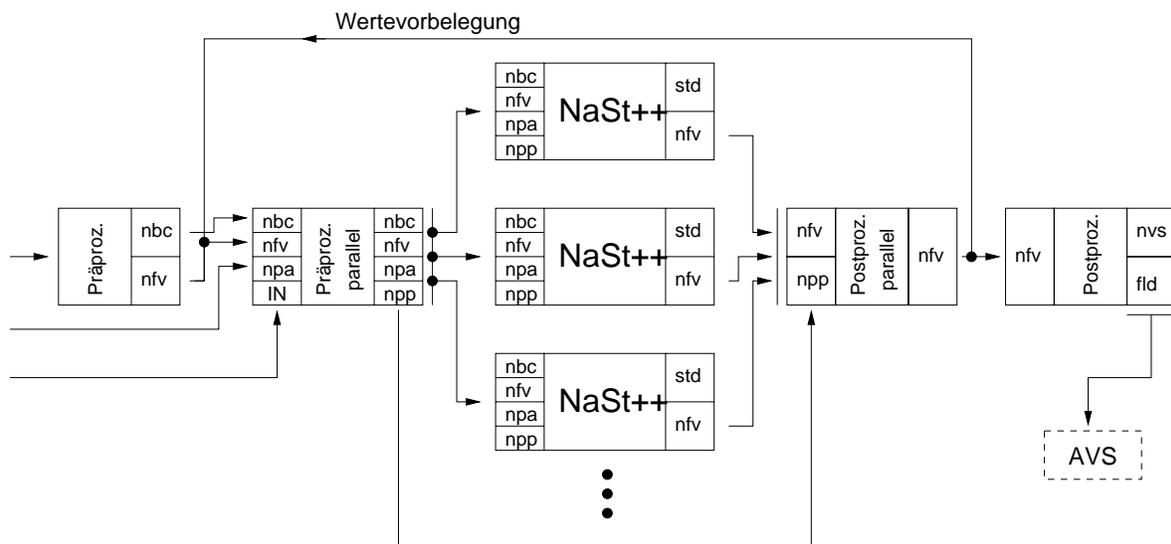
**Abbildung 3.41:** Beispiel für den Anfang einer nbc-Datei, der Eingabeschnittstelle von Nast++ für die Randbedingungen



**Abbildung 3.42:** Schematische Darstellung der Ein- und Ausgabeschnittstellen von Nast++, bestehend aus der nbc-Datei für die Randbedingungen, der nfv-Datei für die Fluid-Hindernis-Geometrie und die berechneten Funktionswerte für Geschwindigkeiten, Druck und Temperatur/Konzentration und der npa-Datei für die Modell- und Programmparameter (aus [63])



**Abbildung 3.43:** Aufteilung des Simulationsgebietes auf die einzelnen Prozesse in Tensorproduktform (vgl. [63])

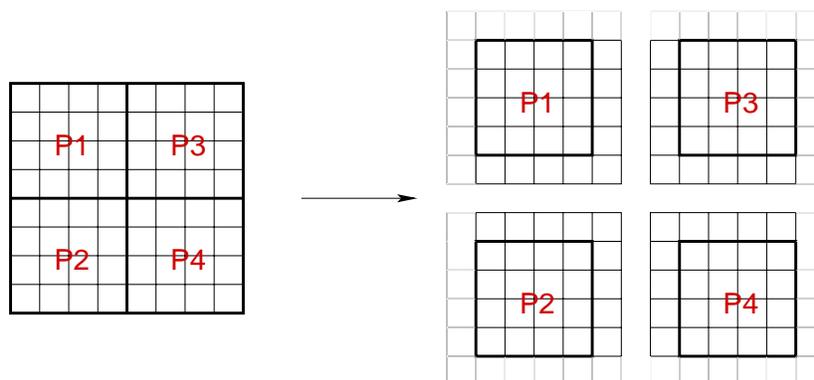


**Abbildung 3.44:** Komponenten der parallelen Simulation mit Nast++: Ergänzung der sequentiellen Prä- und Postprozessoren durch Weitere zur Aufteilung des Gebietes auf die einzelnen Prozesse (Präprozessor parallel) vor dem eigentlichen Programmstart und zur Rekonstruktion des Gesamtgebietes mit der zugehörigen Lösung der Gleichungen (Postprozessor parallel) nach Beendigung der parallelen Simulation (aus [63])

umwandelt. Über zusätzliche Eingabedaten erhält der Präprozessor die Informationen zur gewünschten Gebietsunterteilung. Die zusätzlich zu den npa-, nfv- und nbc-Dateien erzeugten npp-Dateien enthalten die nötigen Daten über die Position des Teilgebietes im Gesamtgebiet. Mit Hilfe dieser Dateien werden die Ergebnisse der Simulation nach Beenden von Nast++ von einem ebenfalls externen Postprozessor wieder zu einer einzigen nfv-Datei für das Gesamtgebiet zusammengesetzt (Abbildung 3.44).

Die Unterscheidung zwischen paralleler und sequentieller Version des ausführbaren Programmes erfolgt bei der Übersetzung des Quelltextes mit Hilfe der entsprechenden Option des Übersetzungsskriptes. Um die Compilierung unnötiger Programmteile wie zum Beispiel des Aufrufs der Kommunikationsfunktionen oder spezieller Klassenkonstruktoren und -destruktoren für die parallele Version bei der Erzeugung des sequentiellen Programmes zu verhindern, wurden entsprechende Compiler-Direktiven in den Programmcode von *Nast++* eingefügt. Dies betrifft vor allem die Klassen für das versetzte Gitter (*CNastStaggeredGrid*) und die diskrete Geometrie (*CNastGeometryDiscrete*).

*Nast++* arbeitet auf den einzelnen Teilgebieten völlig analog zur sequentiellen Version. Allerdings werden nach jedem Zeitschritt des Simulators und nach jeder Iteration des Poissonlösers Daten am Rand der Teilgebiete zwischen den Prozessen ausgetauscht. Weitere Kommunikation zwischen den Prozessen ist zur Berechnung der Residuen für das Gesamtgebiet erforderlich. Um den Kommunikationsaufwand gering zu halten, sollte die Oberfläche der Teilquader bei gegebener Anzahl der Prozesse möglichst klein sein. Daher sind insbesondere flache und langgezogene Teilgebiete ungünstig, während annähernd würfelförmige Teilquader optimal wären. Da zur Auswertung der in *Nast++* verwendeten Differenzensterne (siehe 3.2) jeweils Zugriffe auf die unmittelbaren Nachbarzellen nötig sind, muss jeder Prozess zusätzlich zum eigentlichen Teilgebiet in alle Raumrichtungen eine weitere Zellschicht kennen (Abbildung 3.45). Schreibender Zugriff ist jedoch nur auf die Zellen des Teilgebiets selbst möglich. In jedem Kommunikationsschritt werden die Werte dieser Randschicht anhand der im entsprechenden benachbarten Teilgebiet neu berechneten Werte aktualisiert (siehe [29]).

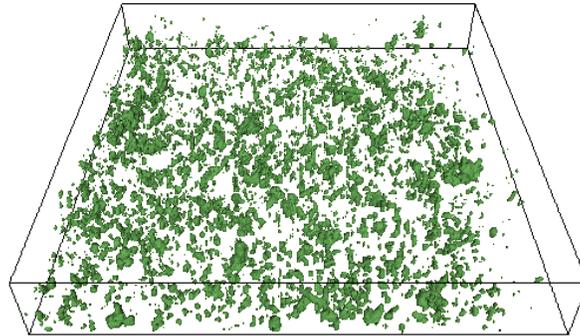


**Abbildung 3.45:** Zerlegung eines zweidimensionalen Gebietes in Teilgebiete mit zusätzlichen Randschichten

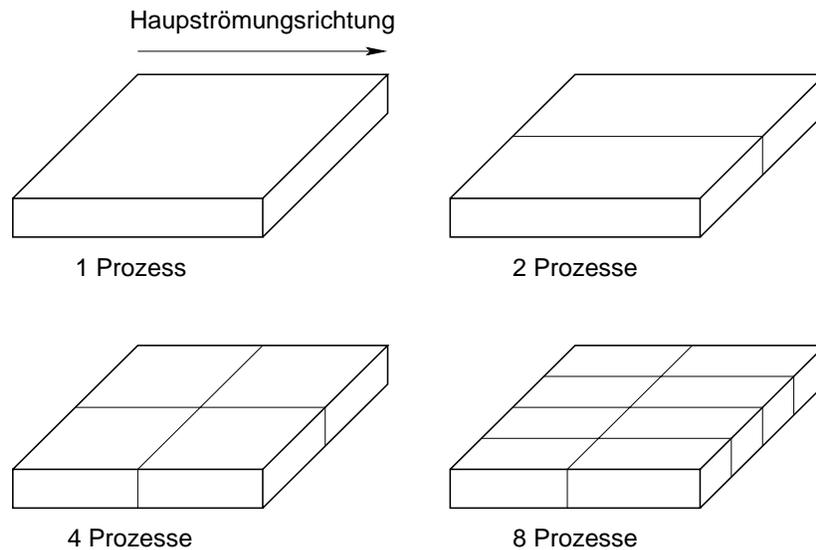
Da der Austausch der Werte der Teilgebiet-Randschichten jeweils erst nach einer kompletten Iteration des Poissonlösers über alle Zellen stattfindet, erhöht sich die Anzahl der benötigten Löser-Iterationen mit der Anzahl der Teilgebiete. Betrachte dazu den Extremfall der Zerlegung in Teilgebiete mit jeweils nur einer einzigen Zelle. In diesem Fall entartet beispielsweise das Gauß-Seidel-Verfahren bezogen auf das Gesamtgebiet zum Jacobi-Verfahren, da bei der Neuberechnung der Druckwerte in jedem Schritt jeweils auf „alte“ Werte in der Randschicht zugegriffen wird.

### 3.6.2 Beispiel

Um die parallele Effizienz und das Verhalten des Poissonl6sers bei steigender Anzahl an Prozessen zu 6berpr6fen, wurde das Str6mungsfeld in einem experimentell definierten Zwei-Kultur-Biofilm (siehe Abbildung 3.46) bei einer Gitteraufl6sung von  $200 \times 200 \times 30$  Zellen auf einem, zwei, vier und acht Prozessoren (Intel Pentium-III, 800 MHz) eines Dualboard-PC-Clusters mit 100Mbit Ethernet-Vernetzung berechnet. Abbildung 3.47 zeigt die zugeh6rigen Aufteilungen des Berechnungsgebiets auf die einzelnen Prozesse.



**Abbildung 3.46:** Geometrie des Beispielbiofilms (*Sphingomonas* sp. LB126 und *E. coli* EC62, Gebietsgr66e ca.  $200 \times 200 \times 30 \mu\text{m}^3$ ) in einer Gitteraufl6sung von  $200 \times 200 \times 30$  Zellen



**Abbildung 3.47:** Aufteilung des Berechnungsgebiets auf einen, zwei, vier beziehungsweise acht Prozesse

Berechnet wurde eine Str6mung mit Reynoldszahl zehn und folgenden Rand- und Anfangswerten:

- Rutschrandbedingungen (SLIP) am oberen, hinteren und vorderen Rand des Gebiets,
- Haftrandbedingungen (NOSLIP) am unteren Rand des Gebiets,

- Dirichlet-Randbedingungen für den Druck (CONSTPRESS) am rechten und linken Rand des Gebiets (links:  $p \equiv 1$ , rechts:  $p \equiv 0$ ),
- $\vec{u} \equiv 0$  und  $p \equiv 0$  im Inneren des Gebiets zum Zeitpunkt  $t = 0$ .

Als Poissonlöser wurde das SOR-Verfahren mit  $\omega = 1,8$  verwendet. Um die parallele Effizienz zu bestimmen, wurden jeweils 1.000 SOR-Iterationen für die Poissongleichung des ersten Zeitschritts durchgeführt:

parallele Effizienz			
Rechner	Prozesse	Rechenzeit	parallele Effizienz
1	1	1.556 sec	100%
1	2	812 sec	96%
2	2	788 sec	99%
2	4	426 sec	91%
4	4	413 sec	94%
4	8	222 sec	88%

Im Anschluss wurde nochmals der erste Zeitschritt der Navier-Stokes-Gleichungen berechnet, wobei jedoch die Iterationen des Poissonlösers erst bei der vorgegebenen Genauigkeitsschranke  $\epsilon_{poisson} = 10^{-3}$  abgebrochen wurden. Die folgende Tabelle zeigt den resultierenden Anstieg der Anzahl der benötigten Iterationen sowie die Verschlechterung der Konvergenzraten durch die verzögerte Aktualisierung der berechneten Werte an den trennenden Ebenen der den Prozessen zugeordneten Teilgebieten (siehe 3.6.1). Ferner wurde anhand der Rechenzeiten die Gesamteffizienz berechnet, die sich aus der parallelen Effizienz und der Effizienz des Löser ergibt.

Konvergenz des Poissonlösers			
Prozesse	Iterationen	Konvergenzrate	Gesamteffizienz
1	12.422	0,9988	100%
2	13.112	0,9989	95%
4	13.478	0,9989	89%
8	14.186	0,9990	80%

## Kapitel 4

# Numerische Simulationswerkzeuge 2 – *CAsim*

Das Softwarepaket *CAsim* ([57]) ist ein universeller Simulator für frei definierbare zelluläre Automaten. Die den Automaten definierenden Zustände und Regeln können mit Hilfe einer eigens entwickelten Skriptsprache flexibel gestaltet werden. Die Elemente dieser Skriptsprache werden in 4.3 und Anhang A.2 ausführlich beschrieben. Für die Simulation des Biofilmwachstums wurde speziell das in 2.2.2 beschriebene Modell realisiert. Die genaue Definition der dazu verwendeten Zustände und Regeln wird in [31] und 4.4 angegeben.

### 4.1 Leistungsumfang

*CAsim* ermöglicht die Simulation zellulärer Automaten auf Gittern mit endlicher Zellenzahl. Für die Zellen kann eine ebenfalls endliche Anzahl beliebiger zulässiger Zustände festgelegt werden, die im Laufe der Simulation anhand von beliebig definierbaren Automatenregeln geändert werden. Grundlage der Regeln sind dabei Bedingungen, die von der Zelle selbst und von frei wählbaren Nachbarzellen erfüllt werden müssen, damit ein Zustandsübergang stattfinden kann. Die Regeln hängen somit von der lokalen Umgebung einer Zelle ab, gelten aber global für alle Zellen des Automaten gleichermaßen und werden simultan auf alle Zellen angewandt. Das Ergebnis einer Automateniteration ist somit unabhängig von der Durchlaufreihenfolge über das Zellenfeld. Stochastische Zustandsübergänge werden durch das Random-Neighbour-Konstrukt ([31]) ermöglicht. Die definierenden Elemente des Automaten werden über eine Konfigurationsdatei (siehe 4.3 und Anhang A.2) eingelesen.

### 4.2 Implementierung

*CAsim* wurde objektorientiert in der Programmiersprache C++ implementiert. Der modulare Aufbau garantiert gleichzeitig eine hohe Flexibilität bei der Definition der einzelnen Bausteine eines Automaten (siehe 4.3) und bei der Ergänzung des Programmcodes um weitere Ele-

mente wie zum Beispiel das Random-Neighbour-Konstrukt (siehe [31]) für stochastische Zustandsübergänge. Die Klasse `CAUniverse` bildet die höchste hierarchische Ebene der *CAsim*-Module und enthält als Member alle Objekte, die das gesamte Universum des zellulären Automaten beschreiben, also die Zellen (`CACell`), Zustandsübergänge (`CATransition`) sowie die dazu benötigten Bedingungen (`CACondition`), Booleschen Ausdrücke (`CABooleanExpression`), Nachbarn (`CANeighbour`) und möglichen Zustände der Zellen (`CAState`). Abbildung 4.1 zeigt die zugehörige Klassenhierarchie. `CAUniverse` stellt darüber hinaus sämtliche Funktionen zur Initialisierung, Simulation und Speicherung des Zustands des zellulären Automaten zur Verfügung (siehe Abbildung 4.2). Da die Regeln des Automaten auf alle Zellen simultan angewandt werden müssen, werden für jede Automateniteration zwei Durchläufe über alle Zellen benötigt. Im ersten Durchlauf werden die Bedingungen für die Zustandsübergänge überprüft und der resultierende neue Zustand jeweils im Member `NextState` der Zelle gespeichert. Erst im zweiten Durchlauf darf der aktuelle Zustand (`CurrentState`) der Zellen durch die in `NextState` gespeicherten Daten ersetzt werden.

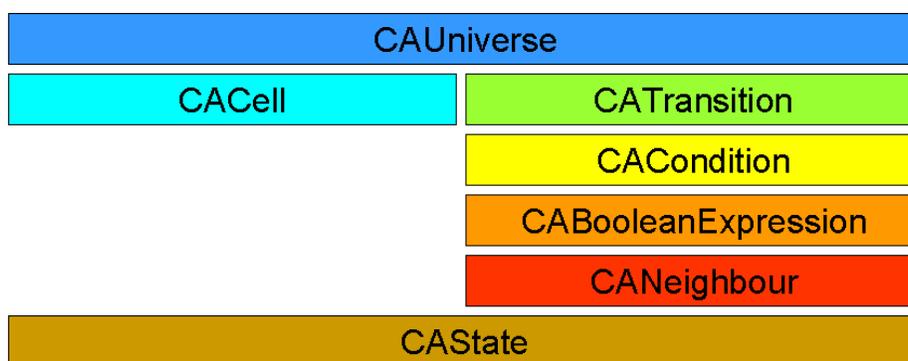


Abbildung 4.1: Die wichtigsten Klassen des Zelluläre-Automaten-Simulators *CAsim*

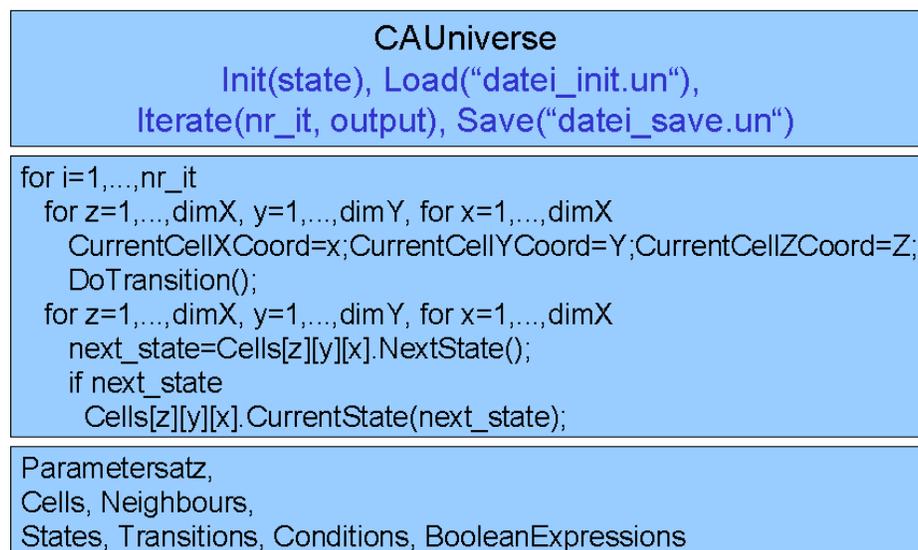
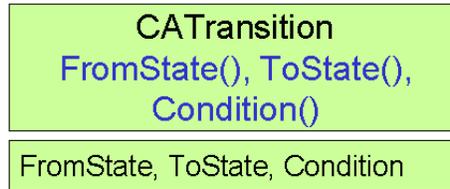
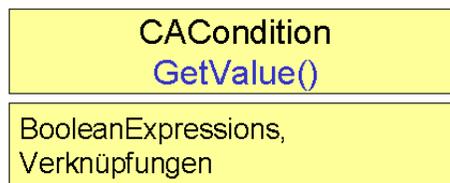


Abbildung 4.2: Die Klasse `CAUniverse` mit den Schnittstellen `Init`, `Load`, `Iterate` und `Save`, einer Ablaufskizze der Funktion `Iterate` und den wichtigsten Klassenmitgliedern

Zustandsübergänge werden durch Objekte der Klasse `CATransition` definiert und sind gegeben durch einen Ausgangszustand (`FromState`), einen Endzustand (`ToState`) und eine Bedingung (Abbildung 4.3). Die für die Zustandsübergänge benötigten Bedingungen implementiert die Klasse `CACondition` (Abbildung 4.4). Jede Bedingung setzt sich aus mehreren mit „und“ oder „oder“ verknüpften Booleschen Ausdrücken zusammen.



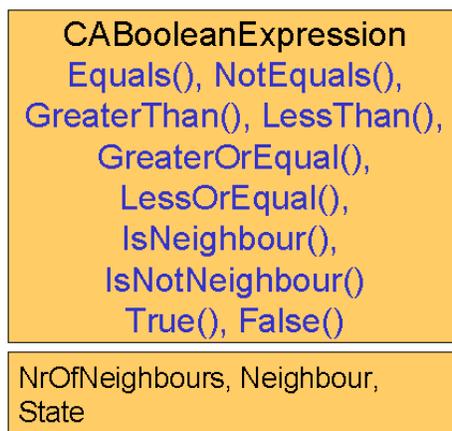
**Abbildung 4.3:** Die Klasse `CATransition` mit den Schnittstellen `FromState`, `ToState` und `Condition` und den wichtigsten Klassenmitgliedern



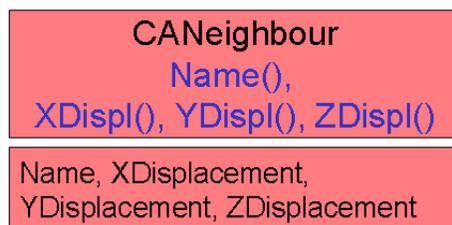
**Abbildung 4.4:** Die Klasse `CACondition` mit der Schnittstelle `GetValue` und den wichtigsten Klassenmitgliedern

Jeder einzelne Boolesche Ausdruck ist ein Objekt der Klasse `CABooleanExpression` und ist definiert durch einen festen Nachbarn `Neighbour`, eine Anzahl von Nachbarn `NrOfNeighbours` und einen Zustand `State` (Abbildung 4.5). Je nachdem, welche der Funktionen des Booleschen Ausdrucks aufgerufen wird, wird zur Auswertung des Ausdrucks der feste Nachbar (`IsNeighbour` und `IsNotNeighbour`) oder die definierte Anzahl an Nachbarn (`Equals`, `NotEquals`, `GreaterThan`, `LessThan`, `GreaterOrEqual` und `LessOrEqual`) verwendet. So wird zum Beispiel bei der Auswertung der Funktion `IsNeighbour` überprüft, ob sich der feste Nachbar `Neighbour` im definierten Zustand `State` befindet. In diesem Fall ist der Ausdruck wahr, ansonsten falsch. Die Funktion `Equals` testet dagegen, ob genau `NrOfNeighbours` Nachbarn im Zustand `State` sind. Die Funktionen `True` und `False` liefern immer „wahr“ beziehungsweise „falsch“ als Rückgabewert. Bei der Auswertung der Funktionen wird auf die im Objekt der Klasse `CAUniverse` gespeicherte aktuelle Zelle und ihre Nachbarzellen zugegriffen, um die aktuellen Zustände zu überprüfen.

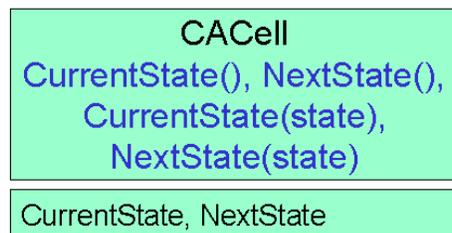
Ein Nachbar einer Zelle ist gegeben durch die relative Position ( $dx, dy, dz$ ) der Nachbarzelle zur aktuellen Zelle.  $(0, 0, 1)$  bezeichnet beispielsweise eine Zelle, die direkt rechts neben der aktuellen Zelle liegt. Die Klasse `CANeighbour` enthält außerdem noch einen Namen, der als Bezeichnung der einzelnen Nachbarn dient, zum Beispiel `NeighbourEast` für den Nachbarn unmittelbar rechts neben einer Zelle (Abbildung 4.6). Die Zellen des Automaten sind Objekte der Klasse `CACell` und enthalten jeweils ihren aktuellen Zustand und ihren nächsten Zustand (Abbildung 4.7). Die in der Klasse `CAState` implementierten Zustände sind dabei durch ihren Namen, eine Integer-Kodierung und gegebenenfalls Rot-Grün-Blau-Werte für die Onlinevisualisierung über OpenGL definiert (Abbildung 4.8).



**Abbildung 4.5:** Die Klasse `CBooleanExpression` mit den Schnittstellen `Equals`, `NotEquals`, `GreaterThan`, `LessThan`, `GreaterOrEqual`, `LessOrEqual`, `IsNeighbour`, `IsNotNeighbour`, `True` und `False` und den wichtigsten Klassenmitgliedern



**Abbildung 4.6:** Die Klasse `CANeighbour` mit den Schnittstellen `Name`, `XDispl`, `YDispl` und `ZDispl` und den wichtigsten Klassenmitgliedern



**Abbildung 4.7:** Die Klasse `CACell` mit den Schnittstellen `CurrentState` und `NextState` (jeweils eine lesende und eine schreibende Funktion) und den wichtigsten Klassenmitgliedern

Zum Programmpaket `CAsim` gehören ferner die Klassen `CAConfigParser` zum Einlesen der Konfigurationsdatei und zur Erzeugung des Universums sowie `GLout` zur graphischen Ausgabe der Ergebnisse für zweidimensionale zelluläre Automaten unter OpenGL.



**Abbildung 4.8:** Die Klasse *CAsim* mit den Schnittstellen *StateName*, *Code*, *RedComponent*, *GreenComponent* und *BlueComponent* und den wichtigsten Klassenmitgliedern

## 4.3 Äußere Schnittstellen

Die äußeren Schnittstellen des Simulators *CAsim* sind sowohl in Bezug auf die Definition des Automaten selbst als auch auf die Ein- und Ausgabe des aktuellen Zustands des Automaten eindeutig definiert. Dies ermöglicht die Kopplung des Programms mit weiteren Bestandteilen der gesamten Simulationsumgebung. Im Fall der Simulation von Biofilmsystemen betrifft dies konkret die Anbindung an *Nast++* (siehe 5.1).

### 4.3.1 Konfigurationsdatei

Das gesamte Universum des zellulären Automaten bestehend aus einem Zellgitter, endlich vielen Zuständen und den Regeln für die Zustandsübergänge wird in der sogenannten Konfigurationsdatei (*ca*-Datei) festgelegt, die beim Start von *CAsim* einmalig eingelesen wird. Eine genaue Beschreibung der Konfigurationsdatei findet sich im Anhang A.2. Die zu verwendende Konfigurationsdatei wird beim Programmaufruf mit der Option `-c <datei_konf.ca>` angegeben. Zur Vereinfachung der Erstellung der Konfigurationsdatei wurde in [31] ein Skript zur automatischen Erzeugung der Booleschen Ausdrücke, Bedingungen und Zustandsübergänge erstellt (siehe Anhang A.2).

### 4.3.2 Initialisierungs- und Ausgabedatei

Die Initialisierung und Ausgabe des Zustands der einzelnen Automatenzellen erfolgt ähnlich wie die Geometriein- und -ausgabe von *Nast++* über eine Datei (*un*-Datei), die für jede Zelle die Kodierung ihres Zustands enthält. Die Reihenfolge der Zellenaufzählung ist hier jedoch in erster Priorität von links nach rechts, dann von oben nach unten und schließlich von vorne nach hinten (siehe Abbildung 4.9).

Beim Aufruf von *CAsim* wird die Initialisierungsdatei mit der Option `-l <datei_init.un>` angegeben. Die Ausgabedatei nach Beenden der letzten Automateniteration hat dasselbe Format wie die Initialisierungsdatei. Die Fortsetzung der Iterationen auf der Basis bereits berechneter Zustände ist somit jederzeit möglich.



einteilen, wobei nur zwei dieser Gruppen zur eigentlichen Wachstumssimulation benötigt werden. Diese Gruppen mit den zugehörigen Übergängen sind im Folgenden aufgelistet. Mit '(a)' markierte Übergänge gehören dabei zum ersten, mit '(b)' markierte zum zweiten und mit '(c)' markierte Übergänge zum dritten Teilschritt einer Automateniteration.

### 1) Regeln für T-Zellen an der Oberfläche des Biofilms

- (a) Auswahl einer F-Nachbarzelle,
- (b) Auswahl einer markierten teilungsbereiten Nachbarzelle von der betreffenden F-Zelle aus,
- (c) eigentliche Zellteilung,
- (c) Zurücksetzen nicht geteilter Zellen auf den Zustand T

### 2) Regeln für T-Zellen im Inneren des Biofilms

- (a) Auswahl einer B-Nachbarzelle,
- (b) Auswahl einer markierten teilungsbereiten Nachbarzelle von der betreffenden B-Zelle aus,
- (c) eigentliches Vertauschen der teilungsbereiten mit nicht teilungsbereiten Zellen,
- (c) Zurücksetzen nicht geteilter Zellen auf den Zustand T

Die beiden übrigen Gruppen dienen der Beschleunigung des Programmablaufs und der Synchronisation der drei oben beschriebenen Teilschritte einer Automateniteration:

### 3) Beschleunigung

Da die Zustandsübergänge in der Reihenfolge ihrer Definition in der Konfigurationsdatei abgearbeitet werden, das heißt für jede Zelle die Liste der möglichen Übergänge so lange geprüft wird, bis ein Übergang tatsächlich stattfindet, wird die Simulation erheblich beschleunigt, wenn einfache, sehr häufig vorkommende Übergänge am Anfang der Liste stehen. Insbesondere betrifft dies den Übergang von Fluid- oder nicht teilungsbereiten Biofilmzellen in ihren Ausgangszustand, falls sie keine teilungsbereiten Nachbarn haben. In diesem Fall findet keine echte Zustandsänderung statt, die Einführung dieser künstlichen Übergänge verhindert aber die Prüfung der Bedingungen aller weiteren definierten Übergänge. Eine weitere Beschleunigung kann erreicht werden, wenn Regelgruppen mit hoher Anzahl von verschiedenen möglichen Übergängen – wie zum Beispiel die Auswahl einer B-Zelle für die Teilung einer T-Zelle im Inneren des Biofilms – möglichst am Ende der Liste definiert werden.

### 4) Synchronisation

Zur Synchronisation der einzelnen Teilschritte werden auch alle Automatenzellen, die nach den bisher genannten Regeln ihren Zustand nicht ändern, in den einzelnen Durchläufen des Automaten in Hilfszustände versetzt, die den aktuellen Teilschritt kennzeichnen.

Für eine genaue Beschreibung der Konfiguration siehe [31].



## Kapitel 5

# Einbettung der numerischen Simulation

Die numerische Simulationsumgebung stellt in dieser Arbeit kein in sich angeschlossenes System dar, sondern steht in enger Verbindung mit den in Kapitel 1 beschriebenen experimentellen Arbeiten auf der einen und der Visualisierung der berechneten Ergebnisse (Kapitel 6) auf der anderen Seite. Ferner werden die Simulationsrechnungen selbst mit Hilfe zweier getrennter Programme durchgeführt: Die kontinuumsmechanischen Strömungs- und Transportgleichungen (2.18) bis (2.22) werden mit Hilfe des objektorientierten Finite-Differenzen-Codes *Nast++* (Kapitel 3) gelöst. Der zelluläre Automat zur Wachstumssimulation wird mit dem Programmpaket *CAsim* (Kapitel 4) realisiert. Zur Durchführung der numerischen Simulation werden daher Anbindungen der Simulationswerkzeuge an die experimentellen Eingabedaten (siehe 5.2) und die Visualisierungswerkzeuge (siehe 5.3) sowie eine geeignete Kopplung der beiden Programme untereinander (siehe 5.1) benötigt.

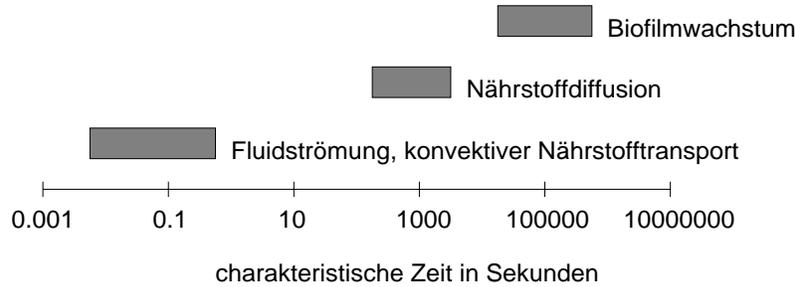
### 5.1 Kopplung von *Nast++* und *CAsim*

In den Kapiteln 3 und 4 wurden die Simulationswerkzeuge für die Lösung der Strömungs- und Transportgleichungen und für das Biofilmwachstum getrennt beschrieben. Um das Wachstum des Biofilms in Abhängigkeit von der Nährstoffverfügbarkeit simulieren zu können, sollen beide nun in eine gemeinsame Umgebung eingebunden werden. Dazu müssen zum einen eine gemeinsame Zeitdiskretisierung festgelegt, zum anderen durch Konvertierungsskripten Verbindungen zwischen den Ein- und Ausgabeschnittstellen hergestellt werden.

#### 5.1.1 Zeitdiskretisierung

Bei der Definition einer gemeinsamen Zeitdiskretisierung entstehen Schwierigkeiten weniger aufgrund der unterschiedlichen Qualität der den Simulationsprogrammen *Nast++* und *CAsim* zugrunde liegenden Modelle (kontinuierlich beziehungsweise diskret), sondern vor allem durch die sehr unterschiedlichen Zeitskalen der einzelnen Prozesse. Die typische Zeiteinheit

des Biofilmwachstums ist um mehrere Größenordnungen größer als die der Strömung und des Stofftransports – vor allem des konvektiven Transports (vgl. Abbildung 5.1). Wollte man alle Prozesse simultan zeitabhängig simulieren, wären daher mehrere Millionen Zeitschritte durchzuführen, bis sich die Biofilmgeometrie durch das Wachstum wahrnehmbar verändern würde.



**Abbildung 5.1:** Unterschiedliche Zeitskalen von Strömung, Nährstofftransport und Wachstum in Biofilmsystemen (aus [24])

Legt man zunächst eine geeignete Zeitschrittweite  $dt$  für das Biofilmwachstum fest, so muss gemäß 2.2.2 zur Simulation eines Wachstumsschritts eine Näherung der Größe

$$C_a(dt) = \int_{t^{(n)}}^{t^{(n+1)}} \left( \int_{\text{Zellvolumen}} \frac{qc}{K+c} \rho_b d\vec{x} \right) dt$$

gefunden werden (siehe Gleichung (2.24)). Da die Berechnungszellen von *Nast++* und *CA-sim* übereinstimmen, *Nast++* also für jede Zelle gerade eine Konzentration  $c_{\text{Zelle}}$  am Mittelpunkt der Zelle berechnet, ist

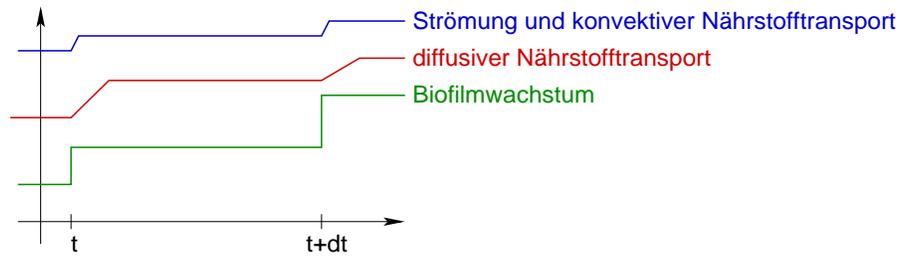
$$\int_{t^{(n)}}^{t^{(n+1)}} \left( dx \cdot dy \cdot dz \cdot \frac{qc_{\text{Zelle}}}{K+c_{\text{Zelle}}} \rho_b \right) dt$$

eine Approximation zweiter Ordnung für  $C_a(dt)$ .

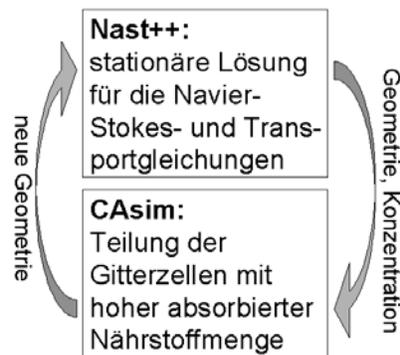
Die Konzentration  $c$  stellt sich nach der Durchführung eines Wachstumsschritts im Vergleich zur Schrittweite  $dt = t^{(n+1)} - t^{(n)}$  sehr schnell auf die neue Biofilmgeometrie ein, das heißt, sie wird nach kurzer Zeit bereits wieder stationär (Abbildung 5.2). Daher ist

$$\hat{C}_a(dt) = dt \cdot dx \cdot dy \cdot dz \cdot \frac{qc_{\text{Zelle}}^{(n)}}{K+c_{\text{Zelle}}^{(n)}} \rho_b$$

eine gute Näherung für  $C_a(dt)$ . Die Problematik der unterschiedlichen Zeitskalen lässt sich daher dadurch umgehen, dass nur das Wachstum des Biofilms echt zeitabhängig simuliert wird, während für die Strömungs- und Transportgleichungen jeweils nach jedem Wachstumsschritt nur die stationäre Lösung berechnet wird. Abbildung 5.3 veranschaulicht das Zusammenwirken der beiden Programme bei der Simulation des Biofilmwachstums.



**Abbildung 5.2:** Schnelles Erreichen des stationären Zustandes bezüglich der Strömung und der Nährstoffkonzentration nach einer wachstumsbedingten Geometrieänderung ( $dt$ : Zeitschrittweite der Wachstumssimulation)



**Abbildung 5.3:** Kopplung von *Nast++* und *CAsim* zur nährstoffabhängigen Simulation des Biofilmwachstums

### 5.1.2 Verbindung der Schnittstellen

Nachdem die Zeitdiskretisierung wie in 5.1.1 beschrieben festgelegt ist, benötigt man zur Kopplung der beiden Programme *Nast++* und *CAsim* geeignete Skripten, die die Ergebnisausgabe von *Nast++* in *CAsim*-Eingabedateien und umgekehrt die Ausgabe von *CAsim* wieder zurück in eine neue Eingabedatei für *Nast++* mit veränderter Geometrie konvertieren. Diese Vermittlungsfunktion übernehmen die Skripten `convert_2d` beziehungsweise `convert_3d` sowie `2d_revert` und `3d_revert` ([31]).

`convert_2d` und `convert_3d` konvertieren drei- beziehungsweise zweidimensionale Ausgabedateien von *Nast++* (`nfv`-Dateien) in Eingabedateien für die Definition des Anfangszustands des Automaten-simulators *CAsim* (`un`-Dateien). Dabei wird unter Berücksichtigung der unterschiedlichen Aufzählungsreihenfolge der Zellen (siehe 3.5 und 4.3) die Information zu Geometrie und Nährstoffkonzentration aus der `nfv`-Datei wie in 5.1.1 beschrieben in das von *CAsim* benötigte Feld der Startzustände aller Automatenzellen umgesetzt. Als Aufrufparameter muss dabei für die Unterscheidung in teilungsbereite und nicht teilungsbereite Biofilmzellen ein Grenzwert für die Konzentration  $c$  der einzelnen Zellen und somit indirekt für  $\hat{C}_a(dt)$  angegeben werden.

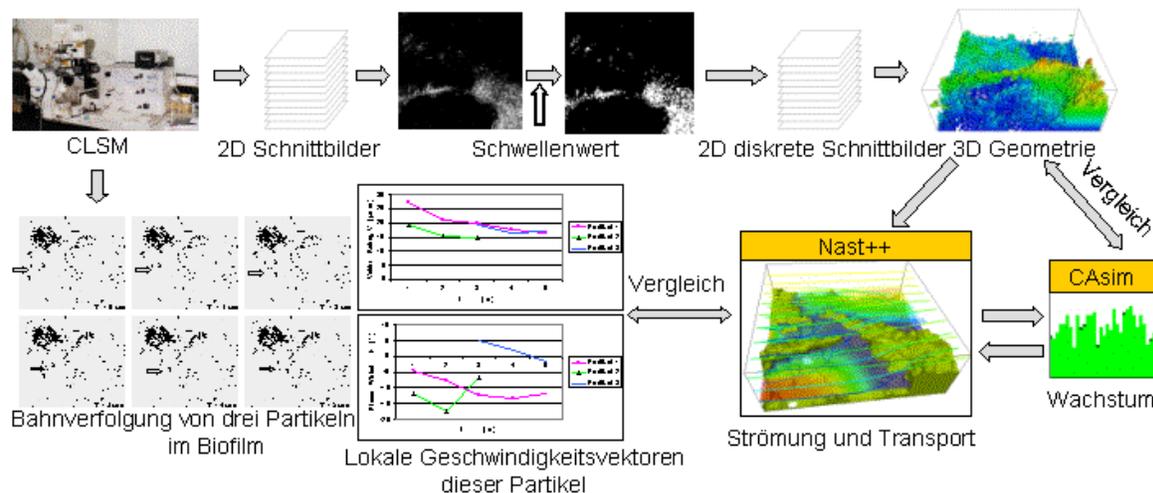
Umgekehrt wird aus den Endzuständen in der entsprechenden `un`-Datei nach dem *CAsim*-Programmablauf mit Hilfe der Skripten `2d_revert` und `3d_revert` eine neue Geometrie für

*Nast++* extrahiert. Als Vorbelegung für die Geschwindigkeits-, Druck- und Konzentrationsfelder werden die Ergebnisse der letzten Berechnung mit *Nast++* in die erzeugte *nf v*-Datei geschrieben. Dadurch wird bei der anschließenden Berechnung von Strömung und Transport die neue stationäre Lösung relativ schnell erreicht, da kleine Veränderungen der Geometrie durch den Wachstumsschritt auch nur kleine Veränderungen der Geschwindigkeiten, des Drucks und der Konzentrationen bewirken.

Die gesamte Simulation des Biofilmsystems beginnt und endet mit einem Programmablauf von *Nast++*.

## 5.2 Schnittstelle Experiment – Simulation

Die numerischen Simulationsrechnungen dieser Arbeit basieren stark auf experimentellen Daten (Kapitel 1). Insbesondere wird die Startgeometrie – wie in Kapitel 2 beschrieben – direkt aus CLSM-Schnittbildern im Fließkanal kultivierter Modellbiofilme bestimmt. Neben der reinen Strukturinformation fließen zudem auch die Ergebnisse der Partikelverfolgung (siehe 1.7) als Grundlage zur Validierung der Strömungsberechnungen in die numerische Simulation ein (siehe Abbildung 5.4).

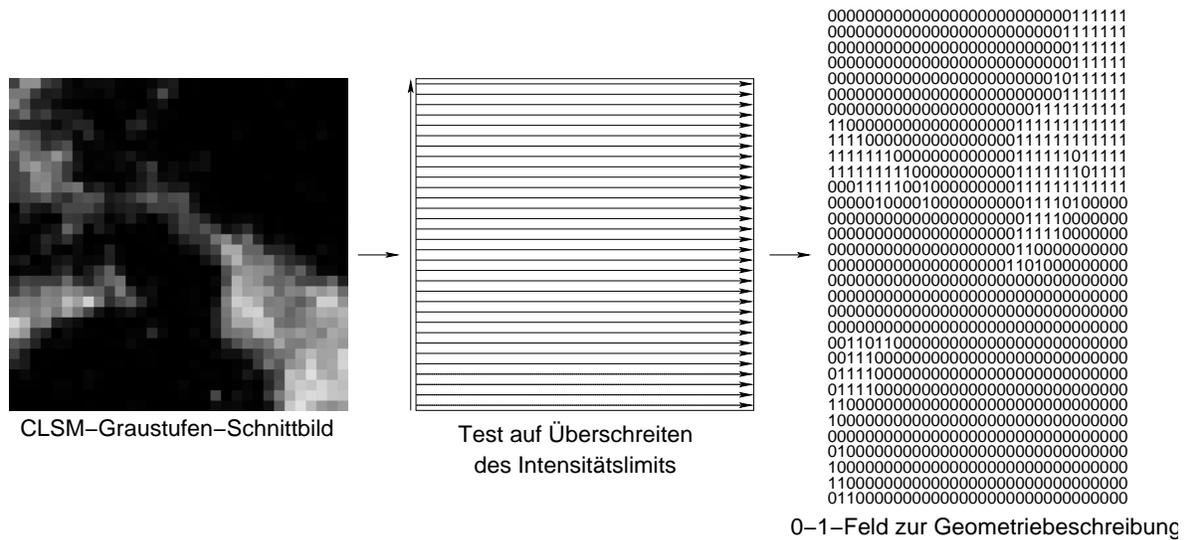


**Abbildung 5.4:** Zusammenhang zwischen Experiment und Simulation: experimentelle Bestimmung der Biofilmgeometrie und Validierung der berechneten Strömungsgeschwindigkeiten durch stichprobenartige experimentelle Messungen mittels Partikelverfolgung

Da die numerische Simulation von Strömung, Transport und Wachstum im betrachteten Biofilmsystem immer mit einem Programmablauf von *Nast++* beginnt, muss lediglich eine Anbindung der experimentellen Daten an den Strömungs- und Transportlöser *Nast++* und nicht an das Programmpaket *CAsim* definiert werden. *Nast++* benötigt als Eingaben eine Parameterdatei (*npa*-Datei), eine Datei zur Beschreibung der betrachteten Fluid-Hindernisse-Geometrie (*nf v*-Datei) sowie eine Datei zur Definition der Randbedingungen (*nbc*-Datei, siehe 3.5). Zur Erstellung der *nf v*- und der *nbc*-Datei wird ein spezieller Präprozessor verwendet. Dieser wendet wie in 2.2.2 geschildert eine Intensitätsschranke auf die Graustufen-Schnittbilder der

CLSM an. Jedem Bildpunkt der CLSM-Schnittbilder entspricht ein Quader des Geometriegitters. Das entstehende Fluid-Hinderniss-Quadrigitter wird neben den Zellenanzahlen des Gitters in den drei Raumdimensionen direkt als 0-1-Feld in die `nfv`-Datei geschrieben. Die Typen und Parameter der Randbedingungen werden aus der Parameterdatei (`npa`-Datei) gelesen, automatisch mit der so definierten Geometrie kombiniert und im erforderlichen Format als `nbc`-Datei ausgegeben. Größere Gitterauflösungen erhält man durch nachträgliche Verkleinerung der ursprünglichen CLSM-Bilder vor der Anwendung des Präprozessors.

Abbildung 5.5 beschreibt die Funktion des Präprozessors schematisch für ein zweidimensionales Schnittbild. Das Graustufenbild wird dabei von unten links nach oben rechts pixelweise abgearbeitet. Im entstehenden 0-1-Feld erscheinen die Punkte daher in (an der  $x$ -Achse) gespiegelter Reihenfolge. Die dreidimensionale Geometriebeschreibung erhält man durch Aneinanderreihung der 0-1-Felder zu den einzelnen Schnittbildern in aufsteigender  $z$ -Richtung.



**Abbildung 5.5:** Schematische Darstellung der Funktionsweise des Präprozessors bei der Verarbeitung eines CLSM-Graustufen-Schnittbilds zur digitalen Beschreibung der Hindernisstruktur eines Biofilms durch ein 0-1-Feld auf der Basis eines regulären strukturierten Rechteckgitters

Aus programminternen Gründen wird zusätzlich in jeder Raumrichtung nach Erreichen der maximalen Zellenzahl des eigentlichen Berechnungsgebiets eine weitere Hinderniszelle angefügt (siehe Abbildung 5.6).

### 5.3 Schnittstelle Simulation – Visualisierung

Um die Ergebnisse der numerischen Simulation so aufzubereiten, dass sie auch für den nicht mit den numerischen Details vertrauten Betrachter interpretierbar sind, werden sie mittels AVS Express ([97]) geeignet visualisiert.

Jede Simulation eines Biofilmsystems endet mit einem Programmablauf von `Nast++`. Ausgangspunkt für die Visualisierung von Strömung, Stofftransport und Biofilmwachstum ist



# Kapitel 6

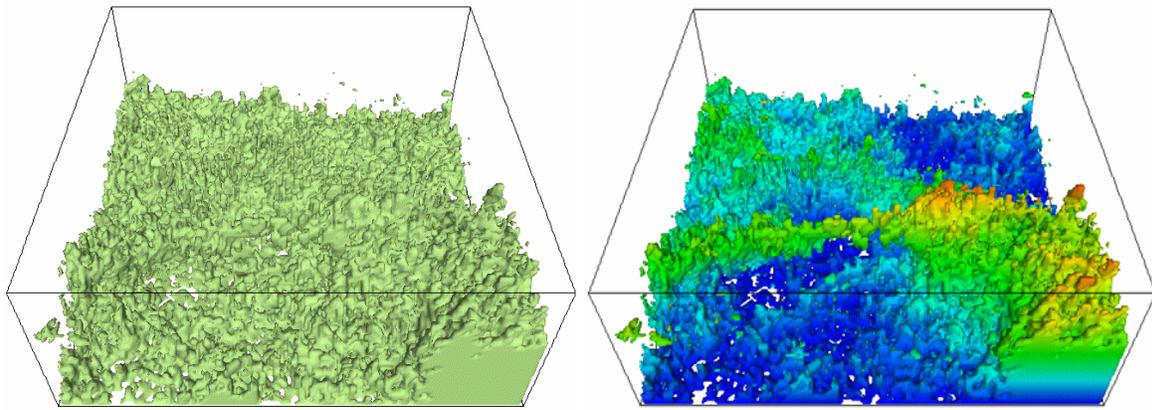
## Visualisierung

Sowohl experimentell (Kapitel 1) als auch numerisch (Kapitel 3 und 4) werden große Datenmengen zu Geometrie, Strömung und Nähr- beziehungsweise Schadstoffkonzentrationen in mikroskopischen Biofilmsystemen erzeugt. Da zweidimensionale CLSM-Schnittbilder und unverarbeitete Zahlenkolonnen aus den Ausgabedateien von *Nast++* kaum unmittelbar zum Verständnis der betrachteten Prozesse und ihrer Interaktion beitragen, müssen diese Daten zunächst geeignet aufbereitet werden. Die hierzu häufig eingesetzte Berechnung von Kennzahlen oder -linien ist zwar schnell implementier- und durchführbar und liefert einen groben Eindruck von einigen Eigenschaften – wie zum Beispiel der Porosität oder der Mikroben-dichte – des Systems, kann aber der Dreidimensionalität der Daten nicht gerecht werden. Da hier der detaillierte Einblick in die Prozesse unter Auflösung aller drei Raumdimensionen ein wichtiges Ziel ist, müssen daher stattdessen geeignete Visualisierungstechniken zur anschaulichen Darstellung der Datensätze angewandt werden. In dieser Arbeit wird die Visualisierung vor allem mit dem Programmpaket AVS Express ([97]) durchgeführt. Ergänzend wird ein spezielles Verfahren zur mobilen Farbstereoprojektion (siehe 6.3.2) angewandt, um echt dreidimensionale Ansichten zu erzeugen.

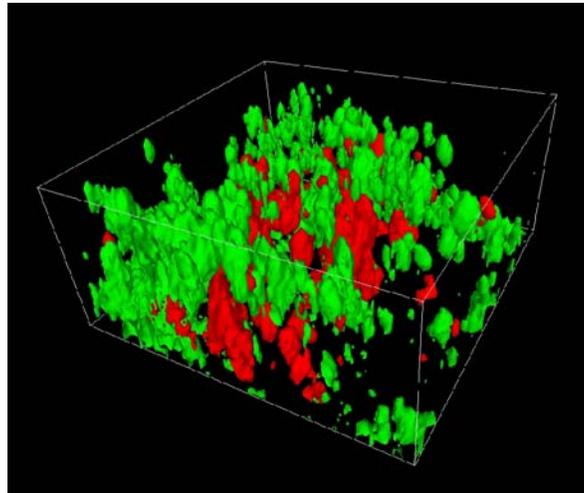
### 6.1 Geometrievisualisierung

Zur Visualisierung der Geometrie verwendet man die als Eingabe für das Simulationsprogramm *Nast++* erzeugten 0-1-Felder. Die durch diese beschriebene Quadergeometrie kann mit Hilfe von Isoflächen oder -volumen dargestellt werden. Bei komplizierten Geometrien ist diese einfache Darstellung jedoch oft sehr unübersichtlich, die dreidimensionale Struktur ist kaum noch erkennbar. Hier ist es hilfreich, die Isofläche oder die Oberfläche des Isovolumens in Abhängigkeit von der Höhe über der Aufwuchsfläche einzufärben (Abbildung 6.1).

Insbesondere im Fall mehrerer Biofilmkomponenten – wie beispielsweise bei Mehrkultur-Biofilmen oder der gleichzeitigen Erfassung von Mikroben und zugehöriger extrazellulärer polymerer Substanzen (EPS) – ist die gegenseitige Lage der Komponenten anhand der getrennten CLSM-Schnittbilder kaum, in der Visualisierung mit Hilfe unterschiedlich eingefärbter Isoflächen jedoch sofort detailliert erkennbar (Abbildungen 6.2 und 6.3).

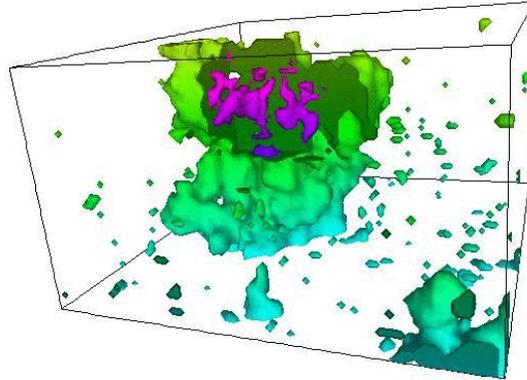


**Abbildung 6.1:** Visualisierung eines sechs Tage alten *E. coli* strain EC12 Monokulturbiofilms mit Hilfe einer Isofläche (links: einfarbige Isofläche, rechts: höhenabhängig eingefärbte Isofläche, Gebietsgröße ca.  $130 \times 130 \times 50 \mu\text{m}^3$ )

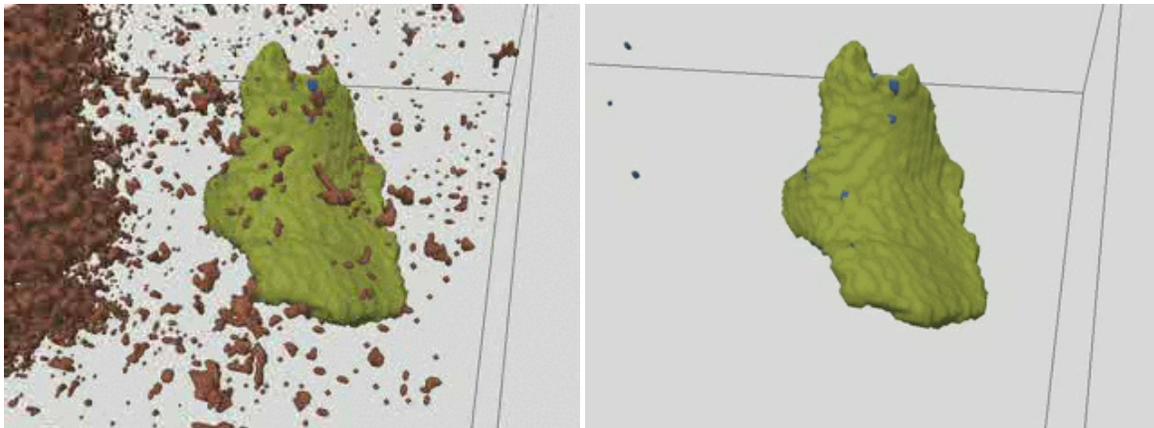


**Abbildung 6.2:** Visualisierung eines Zwei-Kultur-Biofilms auf der Mikroskala (Gebietsgröße ca.  $200 \times 200 \times 40 \mu\text{m}^3$ )

In einigen Anwendungen überschneiden sich die unterschiedlichen mittels CLSM erfassten Biofilmkomponenten. Dies trifft beispielsweise bei der Untersuchung des sogenannten Gentransfers zu. Dazu werden Donoren (Genlieferanten) und Rezipienten (Genempfänger) in einem fest gewählten Bereich des Biofilms getrennt fluoreszenzmarkiert und mittels CLSM aufgenommen. Besonderes Interesse gilt jedoch weder den Donoren noch den Rezipienten sondern den Transkonjuganten, das sind diejenigen Rezipienten, die Gene der Donoren übernommen haben. Diese sind dadurch identifizierbar, dass sie sowohl bei der Markierung der Rezipienten als auch der Donoren fluoreszieren. Ihre räumliche Position ergibt sich also aus dem Schnitt der CLSM-Bilder für Donoren und Rezipienten. Um ein geeignetes 0-1-Feld für die Darstellung solcher Schnittmengen zu erzeugen, wurde der Präprozessor von *Nast++* (siehe 5.2) entsprechend erweitert. Abbildung 6.4 zeigt eine Visualisierung zur Analyse der Lage der Transkonjuganten.



**Abbildung 6.3:** Visualisierung eines Zellclusters einer *Sphingomonas* sp. LB126 Monokultur (grün) mit eingeschlossener EPS (pink) (Cluster-Durchmesser ca. 20  $\mu\text{m}$ )



**Abbildung 6.4:** Visualisierung zur räumlichen Analyse der Transkonjugantenverteilung in einem Zwei-Kultur-Biofilm mit (links) und ohne (rechts) Darstellung der Rezipienten (grün: Donoren *E. coli* strain CM435 (*pRK415::eyfp*), rot: Rezipienten *R. eutropha* strain AE104, blau: Transkonjuganten)

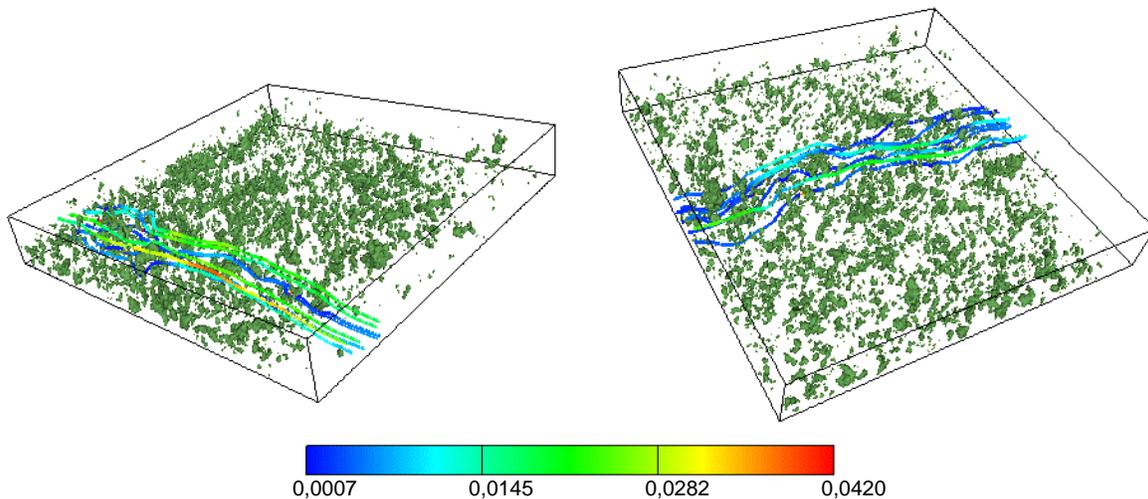
## 6.2 Visualisierung der Simulationsergebnisse

Die Ergebnisse der Simulationsrechnungen für Fluidströmung, Stofftransport und Biofilmwachstum werden aufbauend auf der Geometrievisualisierung mit Hilfe zusätzlicher Module dargestellt.

### 6.2.1 Strömungsvisualisierung

Da die Fluidströmung in den betrachteten Biofilmsystemen immer laminar, stabil und quasi-stationär in Bezug auf das Biofilmwachstum ist, benötigt man zur Visualisierung ausschließlich Methoden für stationäre Geschwindigkeitsverteilungen. Verlauf und Stärke der Strömung,

die im Hinblick auf mechanische Belastungen des Biofilms und ein mögliches Vordringen kleiner Partikel in Poren und Kanäle der Geometrie vor allem nahe der Biofilmoberfläche von Interesse sind, lassen sich sehr anschaulich durch Stromlinien darstellen (Abbildungen 6.5 und 6.8).



**Abbildung 6.5:** In Abhängigkeit von der Absolutgeschwindigkeit eingefärbte Stromlinien in einem Zwei-Kultur-Biofilm (*Sphingomonas* sp. LB126 und *E. coli* EC62, Gebietsgröße ca.  $200 \times 200 \times 32 \mu\text{m}^3$ ) bei einer Durchströmung mit Reynoldszahl zehn

Zusätzlich kann die farbkodierte Visualisierung der Absolut-,  $x$ -,  $y$ - und  $z$ -Geschwindigkeiten sowie des Drucks auf verschiedenen Schnittebenen durch das betrachtete Gebiet detaillierte Informationen über lokal auf den Biofilm wirkende Schub- und Druckkräfte sowie Dehnspannungen liefern (Abbildungen 6.6 und 6.7).

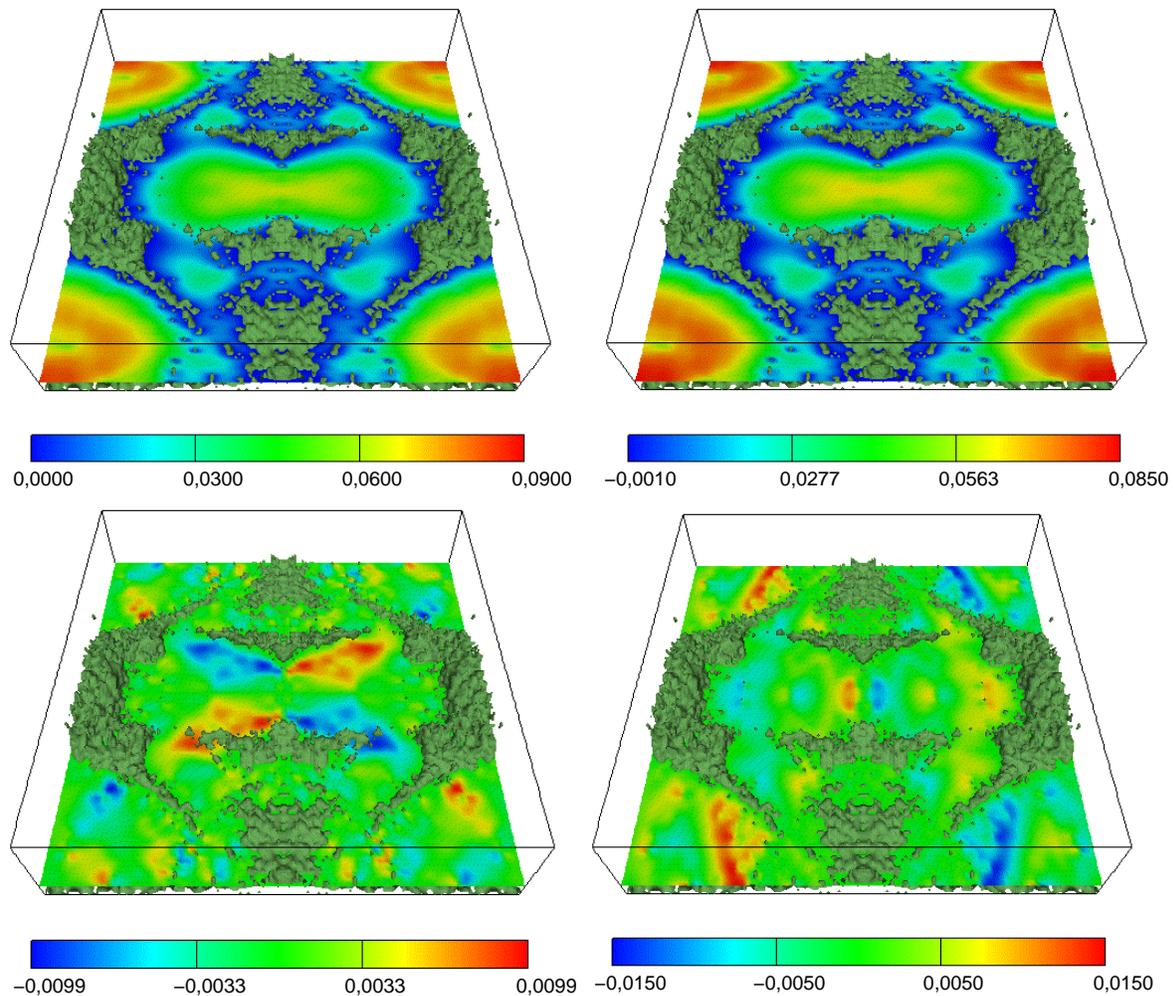
## 6.2.2 Visualisierung der Konzentration gelöster Stoffe

Die Nähr- beziehungsweise Schadstoffkonzentration im Biofilmsystem kann analog zur Strömungsvisualisierung über die dimensionsreduzierte Abbildung auf Schnittebenen veranschaulicht werden (Abbildungen 6.8 und 6.9). Zur Darstellung des Verlaufs einer bestimmten Konzentrationsgrenze eignen sich (halbtransparente) Isoflächen (Abbildung 6.10).

## 6.2.3 Visualisierung des Wachstums

Durch Zeitreihenbilder der Geometrie mit oder ohne Visualisierungselemente für Strömung und Stofftransport lässt sich die Entwicklung des Biofilms über mehrere Zeitschritte des Wachstumssimulators *CAsim* verfolgen. Grundlage hierfür sind die Daten der Ausgaben von *Nast++* unmittelbar vor der Durchführung der Wachstumsschritte mittels *CAsim* (Abbildungen 6.11 bis 6.13).

Um die interne Funktionsweise des zellulären Automaten überprüfen und verschiedene Variationen der Automatenkonfiguration testen zu können, wurden zusätzlich verschiedene Visua-



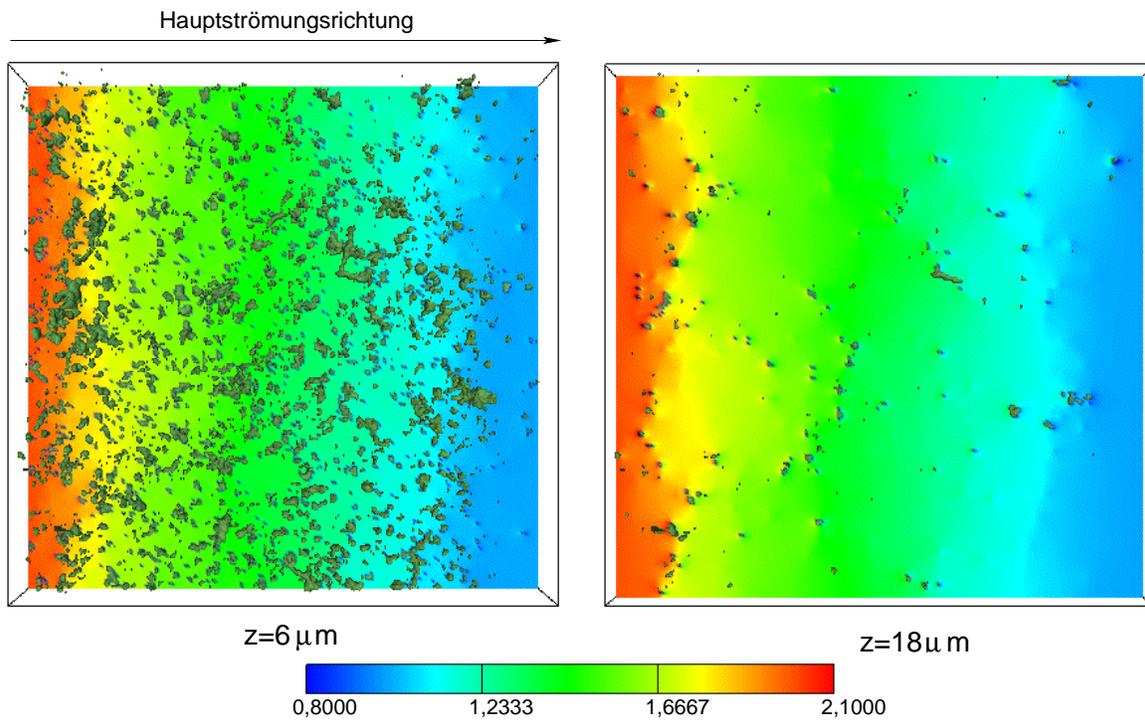
**Abbildung 6.6:** Orthoslices für Geschwindigkeiten in einem sechs Tage alten *E. coli* strain EC12 Monokultur-Biofilm bei einer Durchströmung mit Reynoldszahl zehn (links oben: Absolutgeschwindigkeit, rechts oben: Geschwindigkeit in  $x$ -Richtung, links unten: Geschwindigkeit in  $y$ -Richtung, rechts unten: Geschwindigkeit in  $z$ -Richtung, Gebietsgröße ca.  $250 \times 250 \times 60 \mu\text{m}^3$ )

lisierungen der einzelnen Automateniterationen (Abbildungen 6.14 und 6.15) und der drei Zwischenschritte einer einzelnen Iteration (Abbildung 6.16) durchgeführt ([31]).

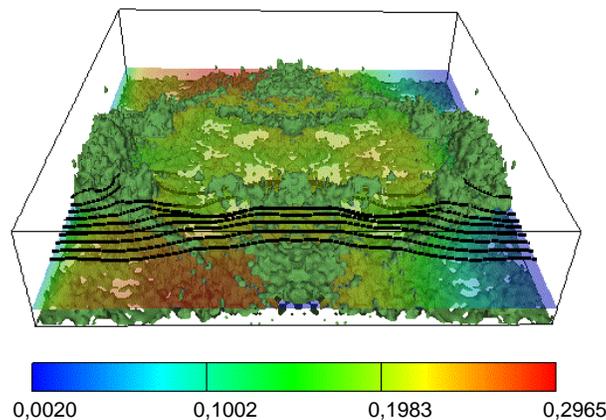
## 6.3 Animierte und 3D-Visualisierung

### 6.3.1 Animationen

Die oben beschriebenen statischen Visualisierungstechniken können der komplexen dreidimensionalen Geometrie und der Geschwindigkeits- und Nährstoffverteilung nicht in vollem Umfang gerecht werden. Um beispielsweise alle Poren und Kanäle eines Biofilms darstellen zu können, sind zumindest mehrere Bilder aus unterschiedlichen Perspektiven erforderlich.

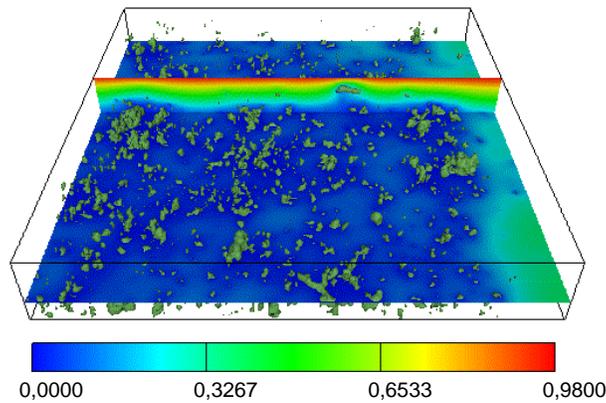


**Abbildung 6.7:** Orthoslices für den Druck in einem Zwei-Kultur-Biofilm (*Sphingomonas* sp. LB126 und *E. coli* EC62, Gebietsgröße ca.  $200 \times 200 \times 32 \mu\text{m}^3$ ) bei einer Durchströmung mit Reynoldszahl zehn



**Abbildung 6.8:** Stromlinien und Orthoslice für die Nährstoffkonzentration in der fluiden Phase eines zweifach gespiegelten, sechs Tage alten *E. coli* strain EC12 Monokultur-Biofilms (Gebietsgröße ca.  $250 \times 250 \times 60 \mu\text{m}^3$ ) bei einer Durchströmung mit Reynoldszahl zehn und  $D^{(f)} = D^{(b)} = 0,01$ ,  $k_1 = 1,0$ ,  $k_2 = 100$

Einen optimalen Eindruck vom gesamten dreidimensionalen Datensatz können jedoch nur animierte Visualisierungen vermitteln. Daher wurden virtuelle Flüge durch experimentell definierte Biofilmgeometrien und entlang der Stromlinien der Fluidströmung (siehe [94]) sowie Filmsequenzen zur zeitabhängigen Ausbreitung eines Nähr- oder Schadstoffs und zum



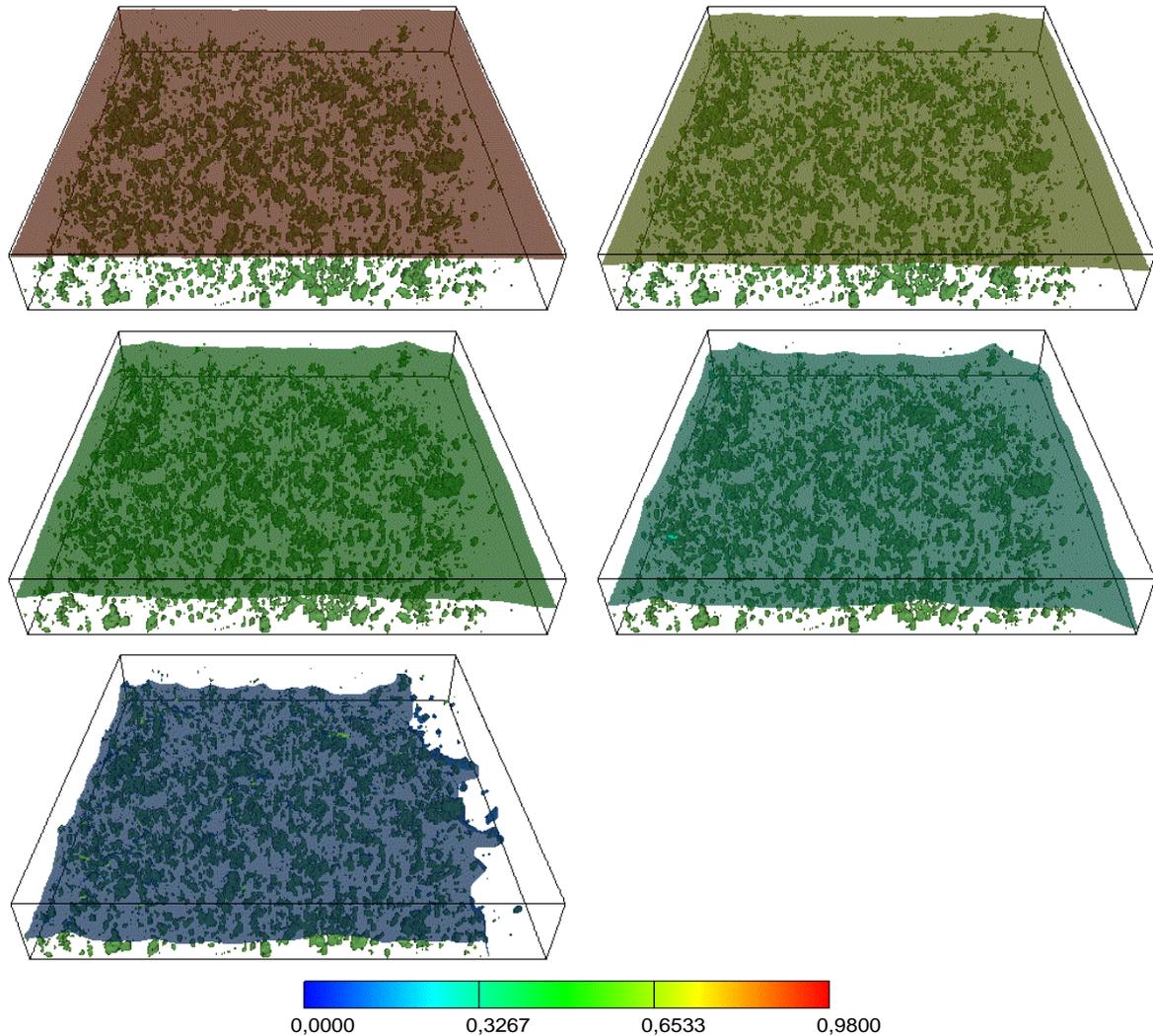
**Abbildung 6.9:** Orthoslices für die Stoffkonzentration in einem Zwei-Kultur-Biofilm (*Sphingomonas sp. LB126* und *E. coli EC62*, Gebietsgröße ca.  $200 \times 200 \times 32 \mu\text{m}^3$ ) mit den Parametern  $D^{(f)} = D^{(b)} = 0,01$ ,  $k_1 = 0,5$ ,  $k_2 = 100$

Wachstum des Biofilms (siehe [95] und [96]) erzeugt.

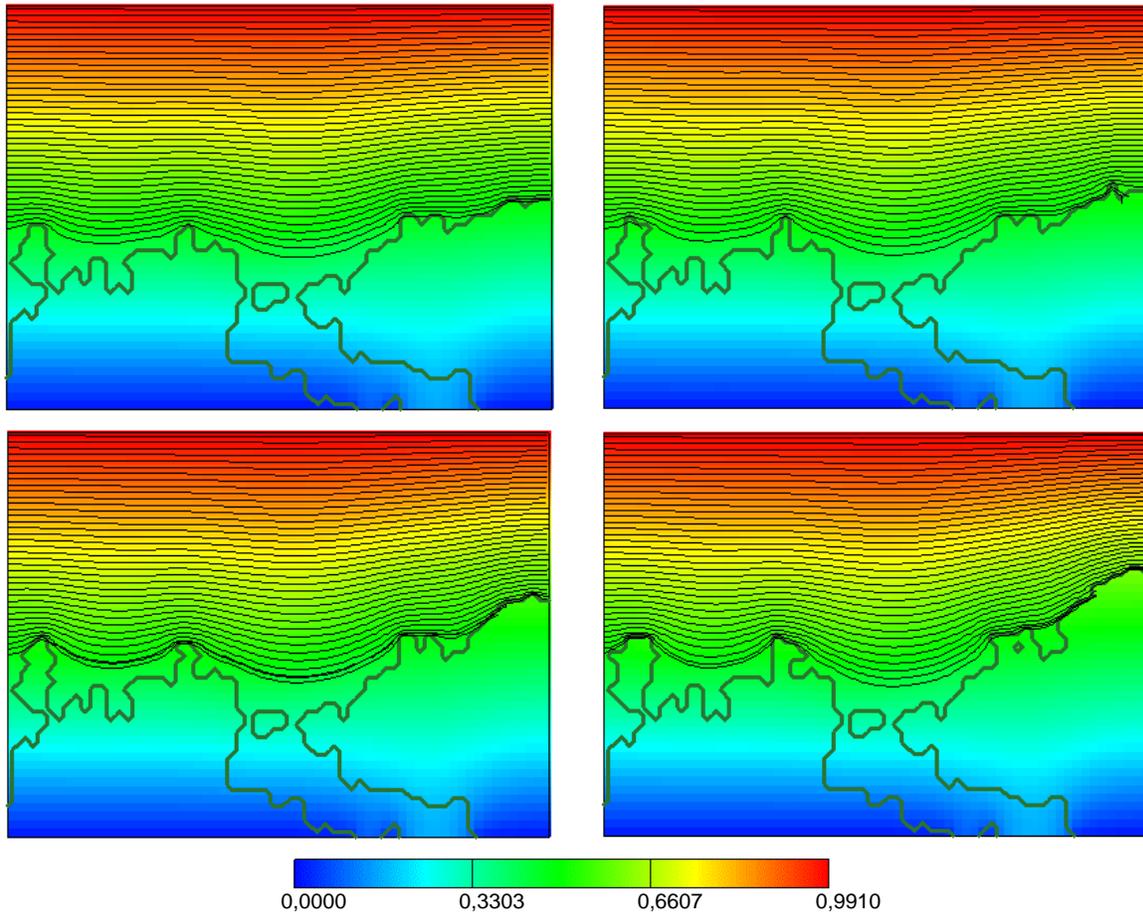
### 6.3.2 3D-Visualisierung

Bei der Projektion des Bilds eines dreidimensionalen Objekts auf die zweidimensionale Darstellungsebene geht ein Großteil des räumlichen Tiefeneindrucks verloren. Um die Ergebnisse dreidimensionaler Berechnungen in idealer Weise veranschaulichen zu können, benötigt man daher spezielle Visualisierungstechniken, die durch getrennte Bilder für das linke und das rechte Auge eine echt dreidimensionale Ansicht erzeugen. Die bekanntesten Techniken sind die sogenannten Rot-Grün-Bilder und die Stereoprojektion, bei der die beiden Bilder für das linke und das rechte Auge unterschiedlich polarisiert werden. Während die Rot-Grün-Bilder vor allem im Hinblick auf Farbdarstellungen meist nur unbefriedigende Bildqualität erreichen, ist die Stereoprojektion, die sich durch hervorragende Bildqualität und hohe Flexibilität der Betrachterposition auszeichnet, bisher mit hohem technischen Aufwand verbunden, der den Einsatz dieser Technik auf ortsfeste Einrichtungen beschränkt.

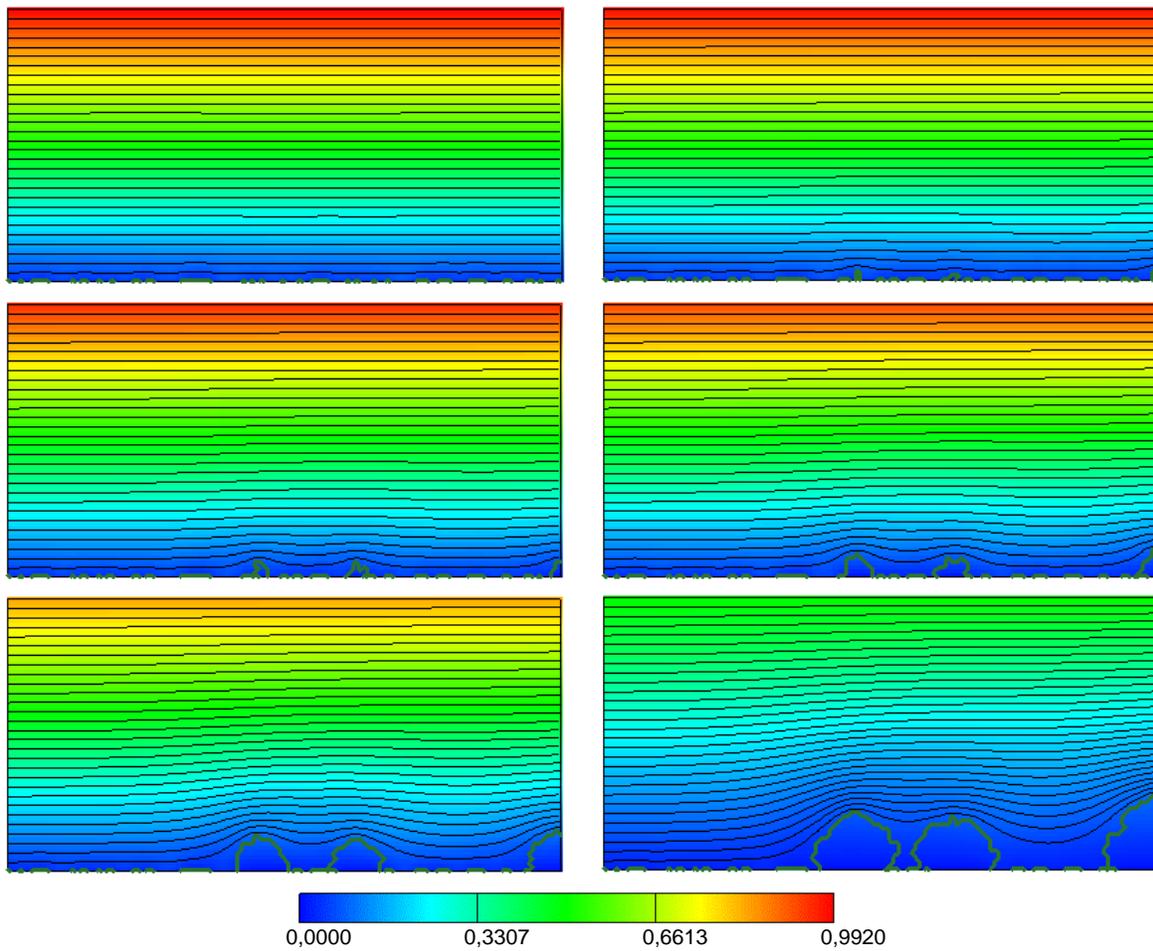
Wie in [93] beschrieben ist es jedoch möglich, Stereoprojektionen auch mit relativ einfachen Hilfsmitteln durchzuführen. Benötigt werden neben einem Rechner zur Erzeugung der Bilder lediglich zwei gleichartige LCD-Videoprojektoren, zwei Polarisations-Filterfolien und Polarisationsbrillen für den Betrachter. Das Stereobild entsteht durch die versetzte Übereinanderprojektion eines Doppelbilds der darzustellenden Szene. Die linke Hälfte des Doppelbilds enthält dabei die Ansicht für das linke Auge, die rechte Hälfte die Ansicht für das rechte Auge. Die jeweils nicht benötigten Bildhälften werden bei der Projektion ausgeblendet (siehe Abbildung 6.17). Auf diese Weise können zu allen oben beschriebenen Visualisierungsarten einschließlich der Animationen 3D-Darstellungen erzeugt werden. Die Generierung von zwei Bildern mit zueinander leicht versetzter Perspektive lässt sich unter AVS Express über Multiple-Window-Applikationen realisieren.



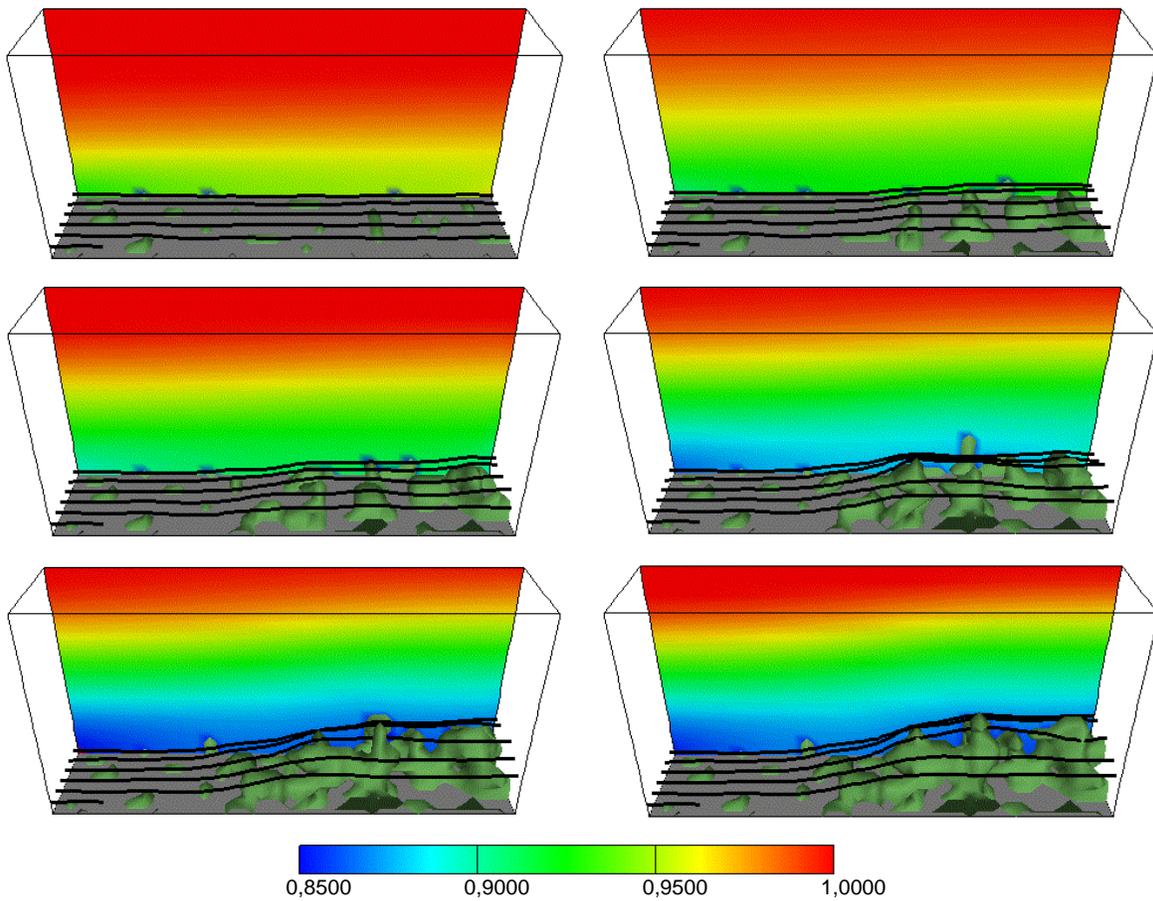
**Abbildung 6.10:** Konzentrationsvisualisierung mit Hilfe von halbtransparenten Isoflächen in einem Zwei-Kultur-Biofilm (*Sphingomonas* sp. LB126 und *E. coli* EC62, Gebietsgröße ca.  $200 \times 200 \times 32 \mu\text{m}^3$ ) mit den Parametern  $D^{(f)} = D^{(b)} = 0,01$ ,  $k_1 = 0,5$ ,  $k_2 = 100$



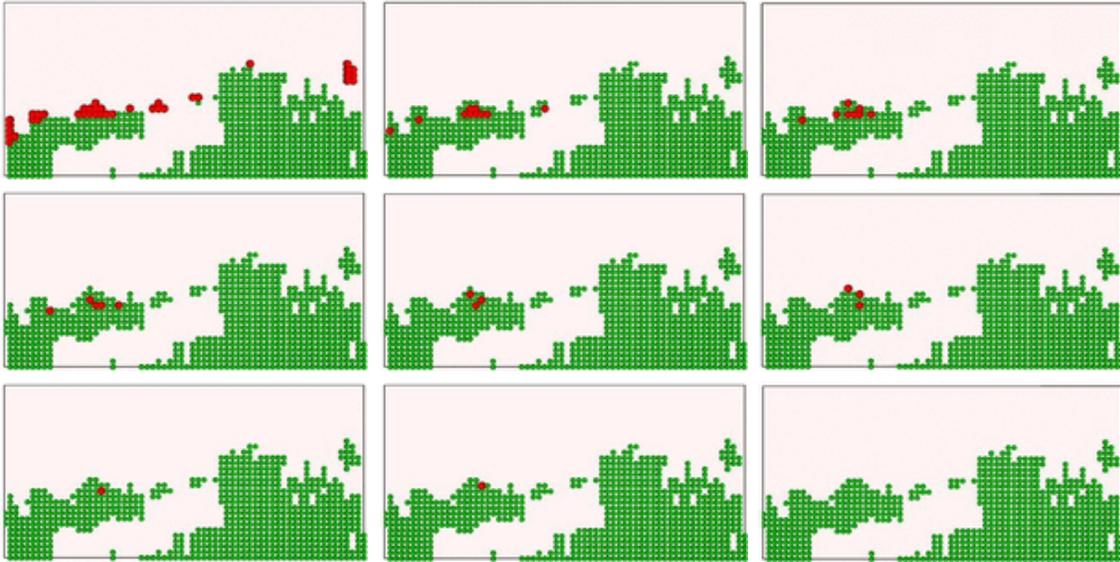
**Abbildung 6.11:** Visualisierung des weiteren Wachstums eines zweidimensionalen Schnitts durch einen experimentell definierten *E. coli* strain EC12 Monokulturbiofilm mit Stromlinien und Nährstoffkonzentration (Gebietsgröße ca.  $500 \times 75 \mu\text{m}^2$ ) bei Reynoldszahl zehn und schwacher metabolischer Aktivität (Transportparameter:  $D^{(f)} = D^{(b)} = 0,01$ ,  $k_1 = 0,01$ ,  $k_2 = 100$ )



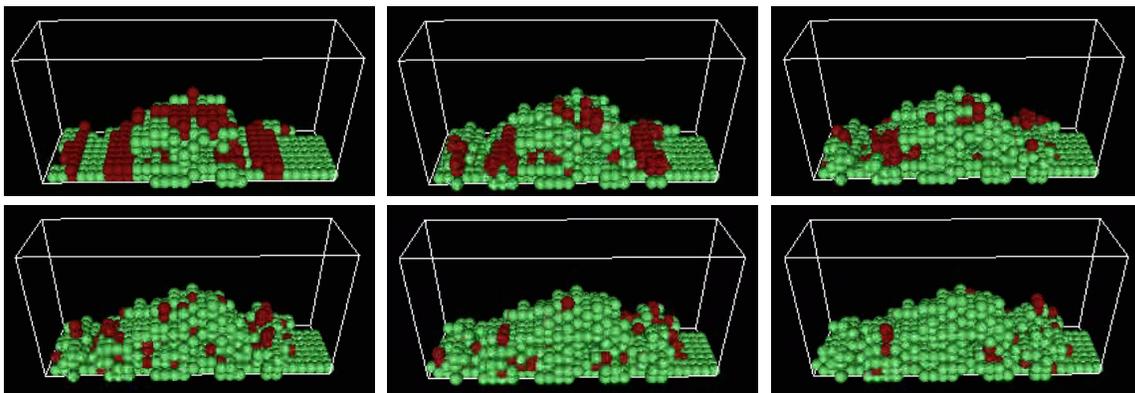
**Abbildung 6.12:** Zweidimensionale Visualisierung der Entwicklung eines Biofilms aus einer stochastisch erzeugten Startpopulation mit Stromlinien und Nährstoffkonzentration bei Reynoldszahl zehn und schwacher metabolischer Aktivität (Transportparameter:  $D^{(f)} = D^{(b)} = 0,01$ ,  $k_1 = 0,01$ ,  $k_2 = 100$ )



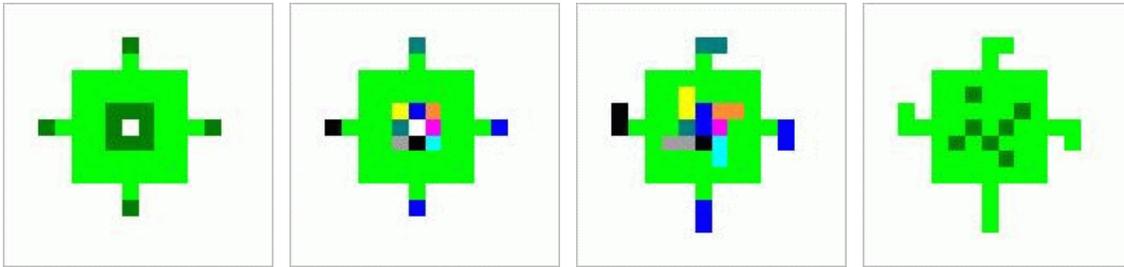
**Abbildung 6.13:** Visualisierung der Entwicklung eines Biofilms aus einer stochastisch erzeugten Startpopulation mit Stromlinien und Nährstoffkonzentration bei Reynoldszahl zehn und stärkerer metabolischer Aktivität (Transportparameter:  $D^{(f)} = D^{(b)} = 0,01$ ,  $k_1 = 1,0$ ,  $k_2 = 100$ )



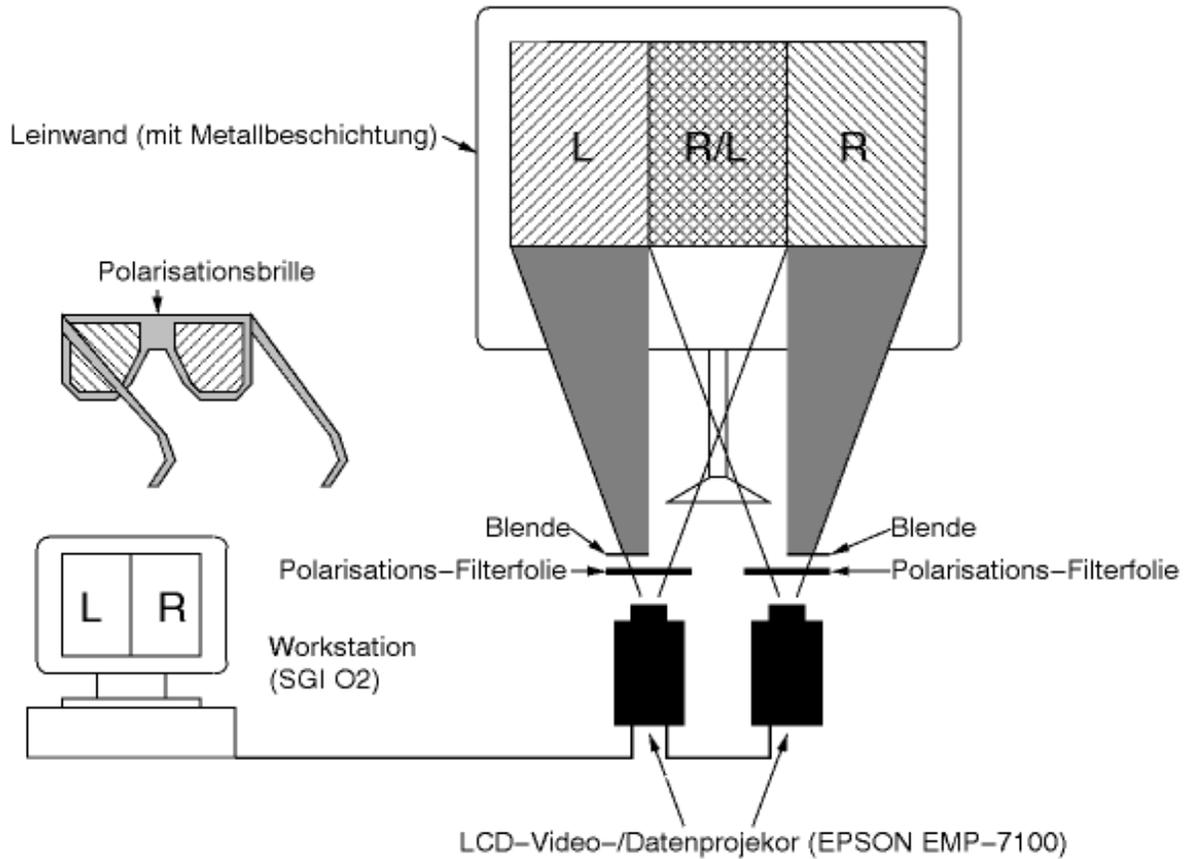
**Abbildung 6.14:** Visualisierung der einzelnen Iterationen eines Wachstumsschritts eines zweidimensionalen Schnitts durch einen experimentell definierten *E. coli* strain EC12 Monokulturbiofilm (rot: teilungsbereite Biofilmzellen, grün: nicht teilungsbereite Biofilmzellen)



**Abbildung 6.15:** Visualisierung der ersten Iterationen eines Wachstumsschritts ausgehend von einem künstlich erzeugten Startzustand (rot: teilungsbereite Biofilmzellen, grün: nicht teilungsbereite Biofilmzellen)



**Abbildung 6.16:** Visualisierung der einzelnen Schritte einer CAsim-Iteration mit Übergangszuständen in einer künstlichen zweidimensionalen Testgeometrie (dunkelgrün: teilungsbereite Biofilmzellen, hellgrün: nicht teilungsbereite Biofilmzellen, weiß: Fluidzellen, bunt: Übergangszustände)



**Abbildung 6.17:** Mobile Farbsteroprojektion zur animierten Darstellung dreidimensionaler Daten (aus [93])



# Zusammenfassung

In dieser Arbeit wurde die numerische Simulation dreidimensionaler Biofilmsysteme auf der Mikroskala anhand eines noch stark vereinfachenden Modells beschrieben. Biologische Prozesse wie Wettbewerb und Auslese der Mikrobenarten oder das Absterben von Zellen wurden dabei ebenso vernachlässigt wie die Ablösung von Teilen des Biofilms aufgrund mechanischer Beanspruchung durch die Strömung, die jedoch nach [21] im Allgemeinen auch zu gering für eine Beschädigung des Biofilms ist. Explizit simuliert wurden dagegen die für die Entwicklung eines Biofilms wesentlichen physikalischen und chemischen Prozesse Fluidströmung, Transport und Abbau gelöster Stoffe – insbesondere sind dies Nähr- beziehungsweise Schadstoffe und Sauerstoff – und darauf aufbauend das Wachstum des Biofilms. Als weitere Vereinfachung wurden sowohl die experimentellen als auch die numerischen Arbeiten im Wesentlichen auf Monokulturbiofilme beschränkt, die unter definierten Bedingungen im Laborfließkanal kultiviert wurden (Kapitel 1). Insbesondere bei der Bestimmung einer repräsentativen Biofilmgeometrie stützt sich die numerische Simulation unmittelbar auf experimentelle Daten.

Im Bewusstsein der Unvollständigkeit und eventuellen Korrekturbedürftigkeit des zugrunde liegenden Modells wurden für die numerische Simulation modulare und flexible Programme weiter beziehungsweise neu entwickelt. Das Ergebnis dieser Arbeit ist also nicht eine abgeschlossene Simulationsumgebung für ein optimales Modell, sondern vielmehr ein Baukasten bestehend aus einzelnen, leicht erweiter- und veränderbaren Modulen, der als Ausgangsbasis für die Simulation weiter verbesserter und ergänzter Modelle (siehe Anhang B.1) dienen soll. Die Hauptmodule stellen dabei das Strömungs- und Transportprogramm *Nast++* (Kapitel 3), das für die Simulation von Biofilmsystemen um Module zur Simulation des konvektiv-diffusiven Stofftransports in der fluiden Phase sowie des diffusiv-reaktiven Stofftransports im Biofilm selbst erweitert wurde, und das neu entwickelte Programmpaket *CAsim* (Kapitel 4) zur Simulation beliebiger zellulärer Automaten, das zur Simulation des Biofilmwachstums verwendet wurde. Diese Aufteilung entspricht gerade der Zusammensetzung des eingesetzten Biofilmmodells aus einem kontinuierlichen Modell für Strömung und Stofftransport und einem diskreten Wachstumsmodell (siehe 2.2.2).

Den größten Teil der gesamten Rechenzeit erfordert bei der Simulation des Gesamtmodells die Lösung der Strömungs- und Transportgleichungen mit *Nast++*. Um den durch die oft hochkomplexe Biofilmgeometrie entstehenden hohen Anforderungen hinsichtlich Speicherbedarf und Rechenzeit gerecht zu werden, wurde *Nast++* unter MPI-II parallelisiert (siehe 3.6 und [63]) und ein geeignetes Mehrgitterverfahren zur Lösung der als Teil der Navier-Stokes-Gleichungen auftretenden Poissongleichung (siehe 3.2) entwickelt und implementiert. Für

einen typischen Beispielbiofilm konnte die Poissongleichung bei einer Gitterauflösung von  $256 \times 256 \times 64$  Zellen mit diesem Verfahren 14-mal schneller gelöst werden als mit dem bisher verwendeten SOR-Verfahren (siehe 3.3.3).

Aus Sicht der Mikrobiologie können bereits einige qualitative Ergebnisse aus den Simulationsergebnissen und den Visualisierungen (Kapitel 6) abgeleitet werden. So können die Geometrievisualisierungen zur räumlichen Analyse der gegenseitigen Lage der Mikrobencluster unterschiedlicher Bakterienspezies oder der Verteilung der extrazellulären polymeren Substanzen (EPS) verwendet werden. Die berechneten Strömungsgeschwindigkeiten liefern Hinweise auf mögliche mechanische Beanspruchungen des Biofilms (Schub- und Dehnspannungen, vgl. [21]) und die Bedingungen für das Vordringen kleiner Partikel – wie zum Beispiel Donorenzellen bei der Untersuchung des Gentransfers – in Poren und Kanäle des Films. Ob und in welchem Umfang aufgrund von Nähr- oder Sauerstoffmangel im Inneren des Biofilms Zellen absterben oder EPS abgebaut werden, lässt sich anhand der berechneten Nährstoffkonzentrationen abschätzen. Die genauen Diffusions- und Reaktionskoeffizienten sind zwar meist nicht bekannt, für erste qualitative Aussagen genügen jedoch Simulationsrechnungen mit unterschiedlichen geschätzten Koeffizienten. Die angekoppelte Wachstumssimulation ermöglicht eine grobe Analyse des Zusammenhangs zwischen Strömung, Nährstoffangebot und Wachstumsgeschwindigkeit sowie Morphologie des entstehenden Biofilms.

# Anhang A – Ergänzungen zu den Schnittstellen von *Nast++* und *CAsim*

## A.1 Beschreibung der Parameterdatei von *Nast++*

Im Folgenden sind alle notwendigen und optionalen Elemente der Eingabedatei von *Nast++* für die Problem- und Löserparameter (*npa*-Datei) aufgelistet. Die Reihenfolge der einzelnen Definitionen ist dabei beliebig.

- Sollen die Geometrie und die Randwerte des Problems automatisch mit Hilfe des Präprozessors aus Graustufenschnittbildern erzeugt werden (siehe 5.2), so muss die *npa*-Datei dazu folgende Angaben enthalten:
  - Randbedingungen: Für jede äußere Gebietsgrenze können ein Randbedingungstyp (entsprechend Tabelle 3.2) und die dazu benötigten Werte für Geschwindigkeiten, Druck und Temperatur beziehungsweise Konzentration vorgegeben werden. So erfordert beispielsweise der Randbedingungstyp 12 (*SLIP\_CONSTTRANS*) die Angabe eines entsprechenden Temperatur-/Konzentrationswertes, während für den Randbedingungstyp 22 (*PERIODIC\_CONSTPRESS*) ein Druckwert festgelegt werden muss:

```
wall.up           = n:12,t:1.0
wall.down        = n:10
wall.north       = n:4
wall.south       = n:4
wall.east        = n:22,p:0.0
wall.west        = n:22,p:1.0
```

Für alle in dieser Aufzählung nicht vorkommenden äußeren Ränder sowie für alle Grenzflächen zu inneren Hindernissen wird eine Default-Randbedingung gesetzt:

```
boundary.default = n:10
```

Fehlt diese Angabe, wird als Default-Bedingung der Randbedingungstyp 0 (*NOSLIP*) verwendet.

- Intensitätslimit: Für die Zuordnung der einzelnen Bildpunkte zum Hindernis- oder Fluidgebiet kann ein Intensitätslimit vorgegeben werden:

```
intensity.lim = 60
```

Fehlt dieses, so wird standardmäßig 125 angenommen. Alle Bildpunkte mit Intensitäten *oberhalb* des Limits werden als Hindernis interpretiert (siehe 5.2).

- Für die Geometrie des Berechnungsgebiets müssen folgende Parameter gesetzt werden:

- Herkunft der Geometrie:

```
geometrie.problem = "file"
```

Die Geometrie- und Randwertdefinition erfolgt entweder über die *nfv*- und *nbc*-Dateien (*file*) oder für einige einfache Beispiele direkt innerhalb von *Nast++* (*dcavity*, *stufe*, *karman*, *convection*, *rayleigh* oder *fluidfalle*). Im ersten Fall müssen zusätzlich die Namen der beiden entsprechenden Dateien gegeben sein:

```
geometrie.boundary = "biofilm32.nbc"
geometrie.obstacle = "biofilm32.nfv"
```

- Abmessungen des Gebietes und Anzahl der Gitterzellen in jede Raumrichtung:

```
geometrie.laenge.x = 2.56
geometrie.laenge.y = 2.56
geometrie.laenge.z = 0.5
geometrie.imax = 32
geometrie.jmax = 32
geometrie.kmax = 8
```

- Die Fluidströmung ist durch folgende physikalische Parameter bestimmt:

- Reynoldszahl:

```
fluid.reynolds = 10
```

- Äußere Kraft  $\vec{g}$  (konstant im gesamten Gebiet):

```
fluid.gx = 0
fluid.gy = 0
fluid.gz = 0
```

- Für die Transportgleichungen werden die folgenden physikalischen Parameter benötigt:

- Schmidt- oder Prandtlzahl  $Sc = \frac{1}{ReD(\Gamma)}$  (siehe Gleichung (2.20)):

```
transpar.prandtl = 10
```

- Volumenexpansionskoeffizient  $\beta$  (null für den Transport gelöster Stoffe, siehe 3.1):

```
transpar.beta = 0.0
```

- Diffusionskoeffizient  $D^{(b)}$  für die Diffusion im Biofilm:

```
transpar.diff = 0.01
```

- Reaktionskoeffizienten  $k_1 := -q\rho_b$  und  $k_2 := K$  (siehe Gleichung (2.23)):

```
transpar.k1 = -0.01
```

```
transpar.k2 = 100
```

- Für die Ergebnisausgabe werden der Name des Ausgabefiles, der Zeitpunkt der ersten Ausgabe und der zeitliche Abstand der Ausgaben (jeweils bezüglich der Zeit  $t$  des simulierten Problems) gefordert:

```
ausgabe.file = "biofilm32"
```

```
ausgabe.start = 0.0
```

```
ausgabe.delta = 5e-2
```

Zu jedem so definierten Zeitpunkt schreibt *Nast++* die aktuellen Geometrie-, Geschwindigkeits-, Druck und eventuell Temperatur- beziehungsweise Konzentrationswerte in eine Datei `filename.zähler.nfv`.

- Die Zeitdiskretisierung wird durch die Parameter Zeitschrittweite, Endzeitpunkt und einen Sicherheitsfaktor  $\tau$  für die Zeitschrittweite festgelegt, mit dem die in *Nast++* berechnete maximal zulässige Schrittweite (siehe 3.2.6) multipliziert wird:

```
simulator.tdelta = 1.0
```

```
simulator.tend = 1e-10
```

```
simulator.tau = 0.5
```

- Die Diskretisierung der kontinuierlichen Modellgleichungen wird durch den Typ der verwendeten Differenzen-Quotienten (bisher nur aus Donor-Cell-Schema und zentralen Differenzen gemischte Quotienten möglich, siehe 3.2.4) sowie spezielle Parameter für den gewählten Differenzen-Quotienten definiert:

```
diff.typ = "dcell"
```

```
diff.dcell.gamma = 0.9
```

- Für den Löser der Navier-Stokes-Gleichungen werden der Typ des Lösers (`explizit = mac` oder `semiimplizit = simple`), eine Genauigkeitsschranke  $eps_{ns}$  für die diskrete  $L_2$ -Norm (siehe 3.3.3) der stationären Impulsgleichung

$$(\vec{u} \cdot \nabla) \vec{u} = \frac{1}{Re} \Delta \vec{u} - \nabla p + \vec{g}$$

und gegebenenfalls Parameter für den semiimpliziten Löser benötigt:

```
nastsolver.typ = "simple"
nastsolver.eps = 1e-3

nastsolver.imp_eps = 1e-5
nastsolver.imp_omg = 1.0
nastsolver.imp_itmax = 100
```

- Dem Löser der aus der Kontinuitätsgleichung (2.19) entstehenden Poissongleichung für den Druck (Gleichung (3.10)) sind die Parameter Lösertyp (SOR oder Mehrgitter, siehe 3.3), Genauigkeitsschranke  $\epsilon_{poisson}$ , Maximalzahl der Iterationen und spezielle Parameter für den gewählten Löser (siehe 3.3) zugeordnet:

```
loeser.typ = "MG"
loeser.eps = 1e-3
```

Parameter für das SOR-Verfahren:

```
loeser.SOR.itermax = 2000
loeser.SOR.omega = 1.8
loeser.SOR.schema = 0
loeser.SOR.debug = 0
```

Parameter für das Mehrgitterverfahren:

```
loeser.MG.smoother = "SOR"
loeser.MG.resinter = "CONST"
loeser.MG.itermax = 1000
loeser.MG.levels = 8
loeser.MG.debug = 0
loeser.SOR.itermax = 2
loeser.SOR.omega = 1.0
loeser.SOR.schema = 0
loeser.SOR.debug = 0
```

Der Parameter `loeser.SOR.schema` gibt dabei an, ob für das SOR-Verfahren eine lexikographische Durchlaufreihenfolge (0) über die Zellen angewandt oder das Red-Black-Schema (1, siehe [29]) verwendet werden soll. Die Parameter `loeser.SOR.debug` und `loeser.MG.debug` regeln die Ausgabe von Zwischenergebnissen zur Unterstützung der Fehlersuche (0: keine Ausgabe, 1: Ausgabe des Drucks und des Residuums nach der Lösung der Gleichung, 2: Ausgabe des Drucks und des Residuums nach jeder Iteration beziehungsweise jedem V-Zyklus des Lösertyps).

- Schließlich muss noch der Löser der Transportgleichungen über den Lösertyp, die Genauigkeitsschranke  $\epsilon_t$  bezüglich der  $L_2$ -Norm (siehe 3.3.3) des Residuums der stationären Transportgleichungen

$$\begin{aligned} (\vec{u} \cdot \nabla) c &= D^{(f)} \Delta c && \text{in } \Omega^{(f)} \text{ und} \\ 0 &= D^{(b)} \Delta c + \frac{k_1 c}{k_2 + c} && \text{in } \Omega^{(b)} \end{aligned}$$

und gegebenenfalls Parameter für den semiimpliziten Löser festgelegt werden:

```
transport.typ      = "simple"
transport.eps      = 1e-3

transport.imp_eps  = 1e-5
transport.imp_omg  = 1.0
transport.imp_itmax = 100
```

Enthält die `npa`-Datei keine physikalischen und Löser-Parameter für die Transportgleichungen, so wird nur die Fluidströmung berechnet. Dasselbe gilt für die Berechnung des diffusiv-reaktiven Transportes im Hindernisgebiet. Diese wird nicht durchgeführt, falls  $D^{(b)}$ ,  $k_1$  und  $k_2$  nicht angegeben werden.

## A.2 Beschreibung der *CAsim*-Konfigurationsdatei

Die *CAsim*-Konfigurationsdatei dient der Definition der Zustände und Regeln des zu simulierenden zellulären Automaten. Jede Zeile der Konfigurationsdatei beginnt mit '#' und besteht aus der Nummer der Zeile, einem Schlüsselwort und bestimmten Parametern, deren Art und Anzahl vom Schlüsselwort bestimmt werden. Grundlegende Parameter wie die Dimension des Automaten und die Anzahl der einzelnen Elemente des Automaten werden im ersten Abschnitt definiert ([57]):

#0	=	Dimension(3);	Dimension (1, 2 oder 3)
#10	=	Width(64);	Zellen in $x$ -Richtung
#20	=	Height(32);	Zellen in $z$ -Richtung
#30	=	Depth(16);	Zellen in $y$ -Richtung
#40	=	Neighbours(4);	Nachbarn
#60	=	RGBColors(2);	Visualisierungsfarben
#70	=	States(2);	mögliche Zustände
#80	=	BooleanExpressions(2);	Boolsche Ausdrücke
#90	=	Conditions(2);	Bedingungen
#100	=	Transitions(2);	Zustandsübergänge

Sind diese Parameter festgelegt, müssen nacheinander die Automaten-elemente angegeben werden. Dabei müssen zunächst die Elemente definiert werden, die im Folgenden zur Definition weiterer Elemente benötigt werden (siehe [57]):

- Vor der Definition der Regeln zum Zustandsübergang müssen die Nachbarn angegeben werden, deren Zustand in jeder Automatenzelle bei der Entscheidung über die Zustandsänderung berücksichtigt werden soll. Ein Nachbar ist durch die vier Parameter Name, relative Position in  $x$ -Richtung, relative Position in  $y$ -Richtung und relative Position in  $z$ -Richtung gegeben:

```
#110 = Neighbour('NorthNeighbour', 0, 1, 0);
#115 = Neighbour('SouthNeighbour', 0, -1, 0);
```

- Falls der Automat auf einem zweidimensionalen Gitter arbeitet, können Farben definiert werden, die bei der Onlinevisualisierung der Automateniterationen zur Darstellung der Zustände der Zellen verwendet werden. Eine Farbe wird über ihre Rot-, Grün- und Blauanteile (zwischen 0 und 255) festgelegt:

```
#135 = RGBColor('Green', 0, 255, 0);
#140 = RGBColor('Red', 255, 0, 0);
```

- Um die möglichen Zustände zu definieren, müssen jeweils die drei Parameter Name des Zustands, Kodierungszeichen und die zugeordnete Farbe übergeben werden:

```
#145 = State('dividing cell', 'T', #140);
#150 = State('non-dividing cell', 'B', #135);
```

- Boolesche Ausdrücke sind die kleinsten Bausteine der Regeln zum Zustandsübergang. Als Parameter werden ein fester Nachbar, ein Zustand und eine Anzahl von Nachbarn benötigt. Je nach Verwendung der Ausdrücke bei der Generierung der Bedingungen für die Zustandsübergänge (siehe unten) wird der Ausdruck über einen Vergleich dieser Anzahl mit der Anzahl der Nachbarn, die sich im entsprechenden Zustand befinden oder über den Test, ob sich der definierte Nachbar in diesem Zustand befindet, ausgewertet (siehe 4.2).

```
#155 = BooleanExpression(#110, #150, 3);
#160 = BooleanExpression(#115, #150, 0);
```

- Die Bedingungen setzen sich aus Booleschen Ausdrücken, jeweils einer Funktion zur Auswertung des Booleschen Ausdrucks und logischen Verknüpfungen zusammen:

```
#170 = Condition(#155(IN) & #160(IN));
#175 = Condition(#155(GT));
```

Die erste dieser Bedingungen bedeutet beispielsweise, dass die im Booleschen Ausdruck in Zeile #155 benutzte Nachbarzelle den ebenfalls in diesem Ausdruck benutzten Zustand besitzen muss und gleichzeitig die zum Booleschen Ausdruck aus Zeile #160 gehörige Nachbarzelle im diesem Ausdruck zugeordneten Zustand sein muss. Mögliche Funktionen zur Auswertung eines Booleschen Ausdrucks sind im Folgenden aufgelistet.

- EQ: Genau die gegebene Anzahl an Nachbarn befindet sich im entsprechenden Zustand.
- NEQ: Die Anzahl der Nachbarn im entsprechenden Zustand entspricht nicht der gegebenen Anzahl.
- GT: Die Anzahl der Nachbarn im entsprechenden Zustand ist größer als die gegebene Anzahl.

- LT: Die Anzahl der Nachbarn im entsprechenden Zustand ist kleiner als die gegebene Anzahl.
- GE: Die Anzahl der Nachbarn im entsprechenden Zustand ist größer oder gleich der gegebenen Anzahl.
- LE: Die Anzahl der Nachbarn im entsprechenden Zustand ist kleiner oder gleich der gegebenen Anzahl.
- IN: Der gegebene Nachbar ist im entsprechenden Zustand.
- INN: Der gegebene Nachbar ist nicht im entsprechenden Zustand.
- TRU: Die Bedingung ist immer erfüllt.
- FALSE: Die Bedingung ist nie erfüllt.

Als Verknüpfungs-Operatoren dürfen

- (: öffnende Klammer,
- ): schließende Klammer,
- &: logisches „und“,
- |: logisches „oder“,
- !: Negation

verwendet werden.

- Ein Ausgangs-, ein Zielzustand und eine Übergangsbedingung definieren schließlich einen Zustandsübergang:

```
#185 = Transition(#145, #150, #170);
#190 = Transition(#150, #145, #175);
```

Stochastische Zustandsübergänge werden über sogenannte `RandomNeighbours` realisiert. `RandomNeighbour` wählt aus den aufgelisteten Nachbarn einen aus, wobei zusätzlich zu den betreffenden Nachbarn die zugehörigen Wahrscheinlichkeiten (in Prozent) angegeben werden müssen.

```
#50 = RandomNeighbours(1);

#120 = Neighbour('myself', 0, 0, 0);
#125 = Neighbour('out', 0, 1000, 0);

#130 = RandomNeighbour(#120,25,#125,75);
```

`RandomNeighbour` kann genau wie normale Nachbarn in der Definition Boolescher Ausdrücke verwendet werden. Um Boolesche Ausdrücke definieren zu können, deren Funktionswerte mit einer Wahrscheinlichkeit, die nicht vom Ort der aktuellen Zelle abhängt, wahr oder falsch sind, werden bei der Definition des `RandomNeighbour` die beiden Nachbarn `myself` und `out` verwendet, deren Zustände jeweils bekannt sind, da die Zelle selbst sich immer in ihrem eigenen Zustand befindet und der Nachbar `out` außerhalb des simulierten Gebietes liegt, also nie in irgendeinem Zustand ist.

```
#165 = BooleanExpression(#130,#150,1);
```

Wird der entsprechende Boolesche Ausdruck mit der Funktion `IN` ausgewertet, so erhält man einen Baustein einer Bedingung, der gerade mit der Wahrscheinlichkeit erfüllt ist, mit der `RandomNeighbour` die Zelle selber ist, falls sich die Zelle im entsprechenden Zustand befindet. Falls die Zelle nicht in diesem Zustand ist, ist der Ausdruck nie wahr.

```
#180 = Condition(#155(IN) & #165(IN));
```

```
#195 = Transition(#145,#150,#180);
```

Nach obiger Definition geht die aktuelle Zelle somit mit einer Wahrscheinlichkeit von 25 Prozent in den Zustand `non-dividing` über, falls sie aktuell im Zustand `dividing` und der nördliche Nachbar im Zustand `non-dividing` ist.

Zur Vereinfachung der Erstellung der Konfigurationsdatei wurde in [31] ein Skript zur automatischen Erzeugung der Booleschen Ausdrücke, Bedingungen und Zustandsübergänge erstellt. Als Eingabe werden der erste Teil einer Konfigurationsdatei mit den Daten zu Nachbarn, stochastischen Nachbarn, RGB-Farben und Zuständen sowie eine zusätzliche Datei, die die Booleschen Ausdrücke, Bedingungen und Zustandsübergänge in vereinfachter Form definiert, benötigt (siehe [31]).

# Anhang B – Ausblick

## B.1 Modellerweiterungen

Das in dieser Arbeit verwendete Modell für mikroskopische Biofilmsysteme ist nicht als fertiges endgültiges Abbild realer Biofilme, sondern als Ausgangsbasis für die Entwicklung realistischerer, das heißt detaillierterer und hinsichtlich der Parameterbestimmung genauer definierter Modelle, zu verstehen. Im Folgenden sind einige wichtige Modellerweiterungen aufgelistet. Die nötigen Grundlagen für die konkrete Ausgestaltung sind für einige dieser Erweiterungen nach dem aktuellen Kenntnisstand so gut wie vollständig bekannt. Dies trifft insbesondere auf den Transport mehrerer gelöster Stoffe, Mehrkulturbiofilme und den advektiven Transport kleiner Partikel zu. Für andere Erweiterungen gibt es lediglich mehr oder weniger heuristische beziehungsweise empirische Anhaltspunkte über mögliche Modelle. Hierzu ist die Modellierung der extrazellulären polymeren Substanzen (EPS), die Ablösung von Teilen des Biofilms und die Viskoelastizität der gesamten Struktur zu zählen. Bei den meisten der *biologischen* Prozesse sind die grundlegenden Gesetzmäßigkeiten sogar weitgehend unbekannt, da empirische Messungen zur Untersuchung dieser Phänomene kaum durchführbar sind.

- **Mehrere gelöste Stoffe**

Der Transport mehrerer gelöster Stoffe wird analog zu Gleichung (2.20) durch Konvektions-Diffusions-Gleichungen beschrieben. Können die einzelnen Stoffe miteinander reagieren, so müssen die Gleichungen um entsprechende Reaktionsterme ergänzt werden. Im Biofilm selbst gelten ebenfalls zu Gleichung (2.21) analoge Diffusions-Reaktions-Gleichungen. Der Vektor  $\vec{r}(\vec{c})$  der vorkommenden Reaktionsterme kann dabei beispielsweise durch Monod-Kinetik für mehrere Stoffe (siehe [60]) definiert werden.

- **Mehrkulturbiofilme**

Mehrkulturbiofilme lassen sich auf einfache Weise durch jeweils eine Gleichung (2.21) entsprechende Diffusions-Reaktions-Gleichung für jede Mikrobenspezies und einen erweiterten zellulären Automaten mit leicht abgewandelten Regeln und zusätzlichen Zuständen zur Unterscheidung der Bakterienarten beschreiben. Auch der Grenzwert für die Unterscheidung in teilungsbereite und nicht teilungsbereite Biofilmzellen muss für jede Spezies separat festgelegt werden.

- **Partikelsimulation**

Bei praktischen Fragestellungen – zum Beispiel im Zusammenhang mit Untersuchungen zum Gentransfer – ist häufig nicht nur die Konzentrationsverteilung gelöster Stoffe, sondern auch der advektive Transport kleiner Partikel von Interesse. Geht man davon aus, dass die Partikel dieselbe Dichte wie das freie Fluid besitzen und somit keine Trägheits- und Schwerkrafteffekte auftreten, bewegen sie sich in einer stationären Strömung entlang der Stromlinien. Dies gilt jedoch nur, solange keine zu engen Poren oder Kanäle den Weg des Partikels versperren. Der Partikeltransport lässt sich also durch die Stromlinien und zusätzliche Restriktionen bezüglich des freien Raumes um jede Partikelposition modellieren.

- **Extrazelluläre polymere Substanzen (EPS)**

Der Anteil extrazellulärer polymerer Substanzen und damit ihre Relevanz schwanken stark von Biofilm zu Biofilm und im zeitlichen Verlauf der Biofilmentwicklung ([47]). Die EPS können jedoch bis zu neunzig Prozent der Trockenmasse des Biofilms ausmachen, und sollten daher in die Biofilmmodelle integriert werden. Dazu sind noch verschiedene Fragen offen:

- Müssen die EPS als poröses Medium modelliert werden oder ist im Rahmen der geforderten Genauigkeit auf der betrachteten räumlichen Skala eine Modellierung als dichte, homogene Phase zulässig? Die Modellierung der EPS als poröses Medium würde neue Gleichungen für Strömung und Stofftransport erfordern, die durch Homogenisierung der Navier-Stokes-, Konvektions-Diffusions- und Diffusions-Reaktions-Gleichungen bestimmt werden müssten. Hierbei spielt die Größe und Gleich- oder Ungleichmäßigkeit der Poren eine wichtige Rolle.
- Darf die EPS als feste Phase betrachtet werden oder spielen viskoelastische Eigenschaften eine wesentliche Rolle? Im zweiten Fall muss die Gleichung für die Fluidströmung mit einer strukturmechanischen Gleichung für die Reaktion der EPS-Matrix auf die auftretenden hydrodynamischen Kräfte gekoppelt werden.
- Welche biologischen, chemischen und physikalischen Prozesse finden innerhalb der EPS statt (zum Beispiel physikalische oder chemische Bindung von Schadstoffen) oder werden durch sie verursacht (beispielsweise Änderungen der metabolischen Eigenschaften der Mikroben durch die Bindung wichtiger Enzyme in den EPS)?

Aufgrund des extrem verschiedenartigen und komplexen Aufbaus der EPS liegen bisher kaum experimentelle Ergebnisse vor, die eine gesicherte Beantwortung obiger Fragen ermöglichen.

- **Detachment**

Die Ablösung von Teilen des Biofilms lässt sich grundsätzlich ähnlich wie das Biofilmwachstum im Rahmen eines zellulären Automaten modellieren. Dabei entstehen jedoch Schwierigkeiten, da die Kenntnisse über die Reaktionen der Mikroben auf mechanische Beanspruchung zu gering sind, um entsprechende Automatenregeln und Kriterien zur Übersetzung der berechneten Fluidströmung in geeignete Startzustände der

Automatenzellen definieren zu können. Das Biofilm-Detachment wird daher fast ausschließlich in eindimensionalen Modellen in Form eines gemittelten Biomassenverlusts berücksichtigt ([54]).

- **Biologische Prozesse**

Bisherige Modelle vernachlässigen in der Regel sämtliche biologischen Prozesse mit Ausnahme der metabolischen Abbauleistung der Mikroben, da zum einen zu wenig über die grundlegenden Mechanismen von Phänomenen wie Wettbewerb zwischen unterschiedlichen Mikrobenarten oder Fressverhalten und Eigenbewegungen von Protozoen im Biofilm bekannt ist, zum anderen auch die Relevanz dieser Prozesse stark umstritten ist.

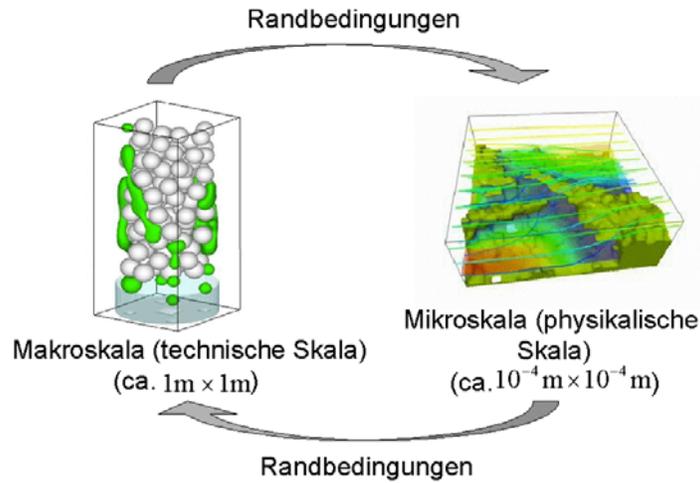
## B.2 Kopplung der Mikro- und Makroskala

Das langfristige Ziel aller Arbeiten zur Modellierung und Simulation von Biofilmsystemen ist die Optimierung großer Systeme wie zum Beispiel Abwasser-Aufbereitungsanlagen oder die Gewinnung von Kenntnissen zur Vermeidung der Bildung unerwünschter Biofilme (zum Beispiel auf Schiffsrümpfen). In beiden Fällen reicht die Betrachtung der mikroskopischen Skala nicht aus. Zwar gewähren experimentelle und numerische Untersuchungen auf dieser Skala Einblick in die genaue Funktionsweise des Biofilms, der Effekt der einzelnen Prozesse auf das sehr viel größere Gesamtsystem kann jedoch nicht unmittelbar abgeschätzt werden. Genausowenig können Simulationen des Gesamtsystems für sich genommen alle Prozesse berücksichtigen, die auf dieser Skala messbare Effekte auslösen können, da aufgrund der begrenzten Speicher- und Rechenzeitressourcen die einzelnen Mikroben des Biofilms nicht mehr aufgelöst werden können (vgl. Kapitel 2).

So kann zum Beispiel die Frage nach dem Einfluss der durch den Biofilmbewuchs verursachten Oberflächenrauigkeit der Aufwuchskörper auf die Strömungsverhältnisse und damit den Nähr- und Sauerstofftransport sowie die Abbauleistung im Festbettreaktor nicht durch Simulationen auf einer der beiden Skalen allein beantwortet werden. Eine Anpassung der Oberfläche der Aufwuchskörper wie in [58] vorgeschlagen, kann auf der makroskopischen Skala nur im Rahmen der Auflösung des verwendeten Rechengitters erfolgen und daher die tatsächliche Komplexität der Biofilmstruktur nur ungenügend wiedergeben. Umgekehrt kann der Einfluss der Strömungsverhältnisse im Gesamtsystem auf das Wachstum und die Struktur des Biofilms nicht allein auf der Mikroskala bestimmt werden, da typische Strömungssituationen und damit strömungsinduzierte mechanische Belastungen sowie die Nährstoffverfügbarkeit nur aus makroskopischen Untersuchungen abgeleitet werden können. Makroskopische Simulationen liefern also die Randbedingungen für die mikroskopischen Rechnungen und umgekehrt (Abbildung B.1).

### B.2.1 Strömung und Nährstofftransport

Die Kopplung von makro- und mikroskopischen Strömungs- und Transportsimulationen wird über eine Iteration realisiert, in deren Verlauf abwechselnd auf den beiden Skalen gerechnet wird und nach jeder Rechnung die Randbedingungen (für die Aufwuchskörperoberfläche



**Abbildung B.1:** Kopplung makro- und mikroskopischer Simulationen über Randbedingungen

beziehungsweise die obere Grenzfläche des Berechnungsgebietes) der jeweils anderen Skala aktualisiert werden. Die Iteration beginnt beispielsweise mit einer makroskopischen Simulation und einfachen Randbedingungen (zum Beispiel Haftbedingungen und homogene Dirichlet-Randbedingungen für die Konzentration eines gelösten Stoffes an der Oberfläche der Aufwuchskörper). Daraus lassen sich lokal in einem kleinen definierten Abstand von der Aufwuchskörperoberfläche die Konzentration  $\bar{c}$  des gelösten Stoffes und die zu dieser Oberfläche parallelen Geschwindigkeitskomponenten  $\bar{u}$  und  $\bar{v}$  ermitteln. Daraus ergeben sich für die mikroskopische Simulation am oberen Gebietsrand Dirichlet-Randbedingungen

$$u = \bar{u}, \quad v = \bar{v}, \quad w = 0, \quad (\text{B.1})$$

$$c = \bar{c}. \quad (\text{B.2})$$

Die Ergebnisse der mikroskopischen Simulationsrechnungen liefern wiederum die notwendigen Daten zur Bestimmung der Verbrauchsrate  $r_v$  des gelösten Stoffes nahe der Aufwuchsoberfläche sowie des gemittelten Geschwindigkeitsgradienten  $\bar{u}_z$  senkrecht zur Aufwuchsoberfläche.  $r_v$  und  $\bar{u}_z$  hängen dabei von den Randwerten  $\bar{c}$ ,  $\bar{u}$  und  $\bar{v}$  sowie von der Struktur des Biofilms ab. Für Strömung und Transport erhält man daher die nichtlinearen Randbedingungen

$$u_{\perp} = 0, \quad \frac{\partial \vec{u}_{\parallel}}{\partial \vec{n}} = \bar{u}_z(\vec{u}_{\parallel}), \quad (\text{B.3})$$

$$\frac{\partial c}{\partial \vec{n}} = r_v(c, \vec{u}), \quad (\text{B.4})$$

wobei  $u_{\perp}$  die Geschwindigkeitskomponente senkrecht zur Oberfläche der Aufwuchskörper und  $\vec{u}_{\parallel}$  die Komponente parallel zur Oberfläche der Aufwuchskörper bezeichnen.

Die Iteration wird solange fortgesetzt, bis ein stationärer Zustand erreicht ist, das heißt die Randbedingungen sich nicht mehr oder kaum noch ändern. Der Rechenaufwand wird dadurch auf ein akzeptables Maß reduziert, dass nicht für alle Ausschnitte der Aufwuchskörperoberflächen des Festbettreaktors, sondern nur für typische Strömungs- und Nährstoffsituationen (gekennzeichnet durch  $\bar{c}$ ,  $\bar{u}$  und  $\bar{v}$ ) und repräsentative Ausschnitte verschiedener Biofilmtypen mikroskopische Simulationen durchgeführt werden. Die Werte von  $\bar{u}_z$  und  $r_v$  für jede andere Situation lassen sich durch eine geeignete Interpolation berechnen.

## B.2.2 Biofilmwachstum

Das Biofilmwachstum wird auf der Makroskala im einfachsten Fall durch eine Erhöhung des Durchmessers der Aufwuchskörper modelliert. Die zugehörige Wachstumsrate wird als Mittelwert aller berechneten lokalen mikroskopischen Wachstumsraten in repräsentativen mikroskopischen Biofilmausschnitten bestimmt. Sie ist also ähnlich wie die Verbrauchsrate  $r_v$  beim Stofftransport eine Funktion der durch  $\bar{c}$ ,  $\bar{u}$  und  $\bar{v}$  gegebenen Strömungs- und Nährstoffsituation, des Biofilmtyps sowie zusätzlich der bereits erreichten Biofilmdicke. Diese Funktion wird ebenfalls wie oben beschrieben durch Interpolation der Ergebnisse für ausgewählte typische Parameterkonstellationen berechnet.

## B.3 Verbesserungen der numerischen Algorithmen

### B.3.1 Adaptivität

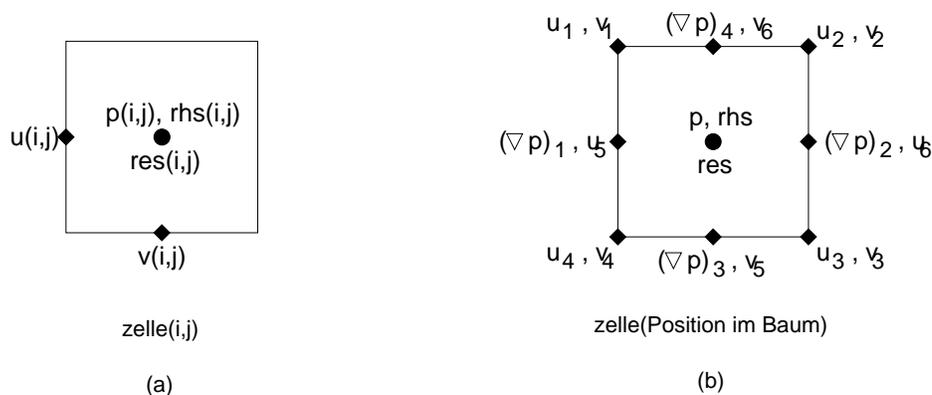
Die Komplexität der Biofilmgeometrie auf der in dieser Arbeit betrachteten räumlichen Skala erfordert sehr hohe Gitterauflösungen und stellt damit die verwendeten numerischen Algorithmen vor große Herausforderungen. Dies gilt insbesondere für die rechenaufwändige Simulation von Strömungs- und Transportvorgängen (Kapitel 3). Die speziellen Anforderungen der Biofilmsimulation – Gebiete mit starker geometrischer Heterogenität und daneben sehr homogene Gebiete im freien Fluid oder in tieferen Schichten des Biofilms – können besonders gut mit adaptiven Verfahren erfüllt werden. Im Falle adaptiver strukturierter Quadergitter – der natürlichen Erweiterung der bisher verwendeten regulären strukturierten Quadergitter – werden die auf dem Gitter gespeicherten Daten sinnvollerweise in Bäumen (im Dreidimensionalen beispielsweise Oktalbäumen) organisiert.

Bisher werden in *Nast++* die Werte der gesuchten physikalischen Größen in Arrays gespeichert. Bei der Auswertung der diskreten Differential-Operatoren einer Zelle  $(i, j, k)$  wird jeweils auf Werte der Nachbarzellen zugegriffen. Bei dieser punktbasierten Diskretisierung wird das Residuum  $res = \Delta p - rhs$  der Poissongleichung (Gleichung (3.10)) nach folgendem Algorithmus berechnet, falls  $dx = dy = dz = h$ :

<p>für alle <math>(i, j)</math></p> $res_{i,j,k} := (p_{i+1,j,k} + p_{i-1,j,k} + p_{i,j+1,k} + p_{i,j-1,k} + p_{i,j,k-1} + p_{i,j,k+1} - 4p_{i,j,k})/h^2 - rhs_{i,j,k}.$
--

Man benötigt also für die Berechnung jedes Residuumswertes  $res_{i,j,k}$  Zugriffe auf die Druckwerte der sechs Nachbarzellen  $(i \pm 1, j, k)$ ,  $(i, j \pm 1, k)$  und  $(i, j, k \pm 1)$ .

Im Gegensatz zur Organisation der Daten in Arrays ist der Zugriff auf Nachbarzellen in Baumstrukturen jedoch teurer und komplizierter. Daher sollte für eine adaptive Version von *Nast++* die Diskretisierung neu organisiert werden, um eine zellbasierte Diskretisierung (wie sie beispielsweise bei Finite-Element-Verfahren verwendet wird) statt der bisherigen punktbasierten Diskretisierung zu erhalten (siehe Abbildung B.2).



**Abbildung B.2:** Zweidimensionale Diskretisierung der Geschwindigkeiten des Drucks und des Residuums der Poissongleichung ((a): punktbasierte Diskretisierung, (b): zellbasierte Diskretisierung, aus [12])

Das Residuum  $res = [\Delta p] - rhs$  der Poissongleichung wird bei dieser zellbasierten Diskretisierung folgendermaßen berechnet:

für alle Zellen  
 $(\nabla p)_1 := 0, (\nabla p)_2 := 0, (\nabla p)_3 := 0, (\nabla p)_4 := 0, (\nabla p)_5 := 0, (\nabla p)_6 := 0$   
 für alle Zellen  
 $(\nabla p)_{1+} = p, (\nabla p)_{2-} = p, (\nabla p)_{3+} = p, (\nabla p)_{4-} = p, (\nabla p)_{5+} = p, (\nabla p)_{6-} = p,$   
 für alle Zellen  
 $res := ((\nabla p)_2 - (\nabla p)_1 + (\nabla p)_4 - (\nabla p)_3 + (\nabla p)_6 - (\nabla p)_5)/h^2 - rhs$

Es werden also keine Zugriffe auf Nachbarzellen durchgeführt, alle Operationen finden lokal in der aktuellen Zelle statt. Eine adaptive Version von *Nast++* ist so weitgehend ohne Verluste durch die Baumstruktur der Daten machbar.

### B.3.2 Weiterentwicklung des Mehrgitterverfahrens

#### Verbesserung der Robustheit des Poissonlösers

Ein wesentliches Problem für die Robustheit des bisher implementierten Mehrgitterlösers für die Poissongleichung (3.10) ist der Verlust von Informationen über trennende Hindernisse auf den groben Gittern (siehe Abbildung 3.27 in 3.3.3). Diese Schwierigkeit tritt nicht nur bei der Verwendung versetzter Gitter auf, sondern besteht auch auf nicht-versetzten Gittern und operatorabhängiger Interpolation und Restriktion. Man betrachte dazu beispielsweise die 3-Punkt-Stern-Diskretisierung

$$L = \frac{1}{dx^2} \begin{bmatrix} 1 & -2 & 1 \end{bmatrix}$$

des eindimensionalen Laplace-Operators auf einem nicht-versetzten Gitter und den häufig verwendeten Restriktionsoperator (Full-Weighting, siehe [82])

$$R = \left[ \begin{array}{ccc} \frac{1}{4} & \frac{1}{2} & \frac{1}{4} \end{array} \right]$$

sowie den linearen Interpolationsoperator

$$I = \left[ \begin{array}{ccc} \frac{1}{2} & 1 & \frac{1}{2} \end{array} \right].$$

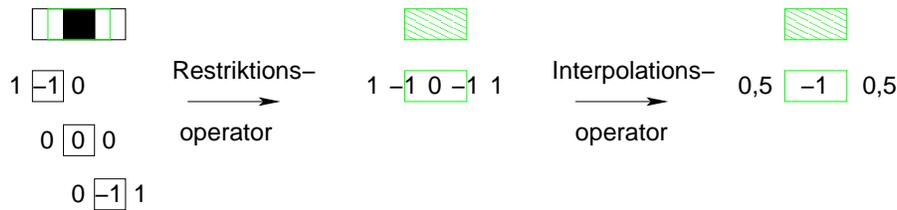
Verallgemeinert man die Restriktions- und Interpolationsoperatoren auf Gebiete mit nicht konstanten Diffusionskoeffizienten, also allgemeinen Sternen  $\left[ \begin{array}{ccc} l_w & l_m & l_e \end{array} \right]$ , so erhält man operatorabhängig (siehe [82])

$$R = \left[ \begin{array}{ccc} -\frac{l_w}{2l_m} & \frac{1}{2} & -\frac{l_e}{2l_m} \end{array} \right] \text{ und } I = \left[ \begin{array}{ccc} -\frac{l_w}{l_m} & 1 & -\frac{l_e}{l_m} \end{array} \right],$$

wobei  $\left[ \begin{array}{ccc} l_w & l_m & l_e \end{array} \right]$  der Differenzen-Stern der mittleren Feingitterzelle ist. Diese Formeln lassen sich unmittelbar nur für  $l_m > 0$  anwenden. Bei der diskreten Poissongleichung in *Nast++* ist jedoch in allen Hinderniszellen  $l_w = l_e = l_m = 0$ . Es findet also keine Diffusion über Hinderniszellen hinweg statt. Das Verhalten dieser Zellen kann daher als Grenzwert einer Zelle mit sehr schwacher Diffusion (Differenzen-Stern  $\left[ \begin{array}{ccc} \epsilon & -2\epsilon & \epsilon \end{array} \right]$ ,  $0 < \epsilon \ll 1$ ) interpretiert werden. Als Interpolations- und Restriktionsoperatoren ergeben sich so

$$R = \lim_{\epsilon \rightarrow 0} \left[ \begin{array}{ccc} \frac{\epsilon}{4\epsilon} & \frac{1}{2} & \frac{\epsilon}{4\epsilon} \end{array} \right] = \left[ \begin{array}{ccc} \frac{1}{4} & \frac{1}{2} & \frac{1}{4} \end{array} \right] \text{ und } I = \lim_{\epsilon \rightarrow 0} \left[ \begin{array}{ccc} \frac{\epsilon}{2\epsilon} & 1 & \frac{\epsilon}{2\epsilon} \end{array} \right] = \left[ \begin{array}{ccc} \frac{1}{2} & 1 & \frac{1}{2} \end{array} \right].$$

Abbildung B.3 zeigt an einem Beispiel die Aufhebung der Trennung von Fluidzellen durch Hindernisse bei der resultierenden Vergrößerung der Differenzen-Sterne.



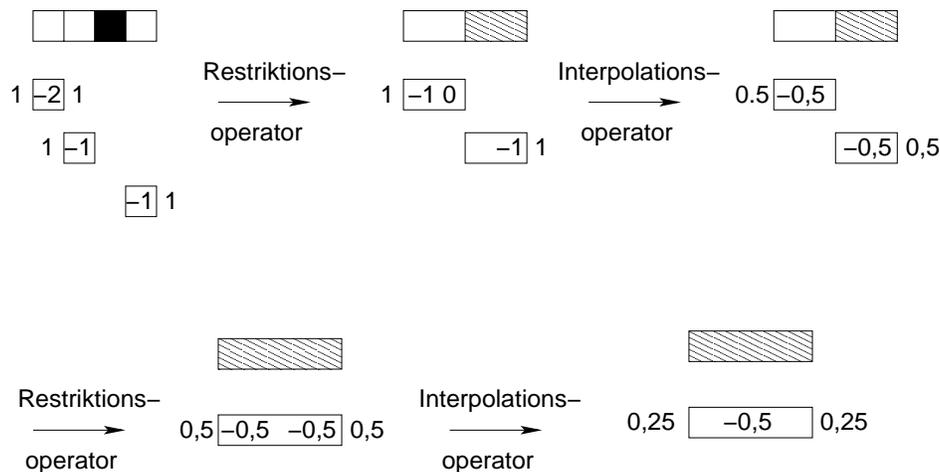
**Abbildung B.3:** Aufhebung der Trennung von Fluidzellen durch ein Hindernis bei der Gittervergrößerung auf nicht versetzten Gittern mit operatorabhängiger Restriktion und Interpolation

Auch der Übergang von der konstanten zur linearen Interpolation

$$I = \left[ \begin{array}{cccc} \frac{1}{4} & & & \end{array} \right] 1 \quad 3 \quad 3 \quad 1 \quad \left[ \begin{array}{c} \end{array} \right]$$

auf dem versetzten Gitter löst das Problem der zusammenwachsenden Bereiche nicht (siehe Abbildung B.4).

Die Speicherung mehrerer Funktionswerte und Differenzen-Sterne für Grobgitterzellen mit topographisch getrennten Fluid-Teilgebieten ist dagegen eine Möglichkeit, die Trennung der Gebiete beizubehalten, ohne an Genauigkeit zu verlieren. Die groben Gitter wären somit nicht mehr regulär sondern adaptiv. Die Anzahl der Gitterpunkte sinkt bei der Vergrößerung dennoch ausreichend stark ab, falls die Hindernisoberflächen nicht zu komplex sind. Ein robustes Mehrgitterverfahren ließe sich daher mit einer adaptiver Version von *Nast++* auf einfache Weise realisieren.



**Abbildung B.4:** Zusammenwachsen von Teilgebieten bei Verwendung versetzter Gitter und linearer Interpolation im Laufe der Vergrößerung der Differenzen-Sterne

### Mehrgitterverfahren für die zeitabhängigen Navier-Stokes- und Transportgleichungen

Die für adaptive Verfahren benötigte Datenspeicherung in Baumstrukturen beinhaltet eine implizite Hierarchie der Zellen. Diese Tatsache kann unmittelbar zur Hierarchisierung der Daten und in einem weiteren Schritt zur Entwicklung von Mehrgittermethoden genutzt werden, die nicht nur auf die Poissongleichung für den Druck, sondern auf die gesamten zeitabhängigen Navier-Stokes-Gleichungen angewandt werden können. Dafür kann die sogenannte „Waveform Relaxation“ ([83]) eingesetzt werden, die in gewissem Sinne mit Full-Multigrid-Methoden (FMG, [8]) verwandt ist, aber im Gegensatz zu diesen auch für zeitabhängige (parabolische oder hyperbolische) statt nur für stationäre (elliptische) Differentialgleichungen eingesetzt werden kann. Aufgrund der Nichtlinearität der zeitabhängigen Navier-Stokes-Gleichungen bezüglich der Fließgeschwindigkeit  $\vec{u}$ , muss dabei ein „Full Approximation“ Schema ([82]) für  $\vec{u}$  und  $p$  verwendet werden, das heißt die Grobgittergleichungen dürfen nicht wie bei den ursprünglichen Mehrgitterverfahren für Druck- und Geschwindigkeitskorrekturen aufgestellt und gelöst werden, sondern müssen auch auf den groben Gittern eine Approximation für die Geschwindigkeiten und den Druck selbst liefern. Besitzen die Daten wie im Fall der Speicherung in Oktalbäumen bereits eine inhärente Hierarchie, können die dafür notwendigen Transformationen von Feingitter- und Grobgitterlösungen und umgekehrt einfach und günstig in Bezug auf den Speicherbedarf und die Rechenzeit durchgeführt werden. Die Transformationen entsprechen im Wesentlichen der Hierarchisierung und Dehierarchisierung der gespeicherten Daten. Zusätzliche Datenstrukturen und damit zusätzlicher Speicherplatz für die Grobgitterdaten werden nicht benötigt ([28]).

# Literaturverzeichnis

- [1] R. E. Alcouffe, A. Brandt, J. E. Dendy und J. W. Painter. 1981. *The multigrid method for the diffusion equation with strongly discontinuous coefficients*. SIAM J. Sci. Comput. 2, S. 430-454.
- [2] H. Babovsky. 1998. *Die Boltzmann-Gleichung: Modellbildung – Numerik – Anwendungen*. B. G. Teubner, Stuttgart & Leipzig.
- [3] D. de Beer, P. Stoodley und Z. Lewandowski. 1994. *Liquid flow in heterogeneous biofilms*. Biotech. Bioeng. 44: S. 636-641.
- [4] D. de Beer, P. Stoodley und Z. Lewandowski. 1996. *Liquid flow and mass transport in heterogeneous biofilms*. Water Res. 30 (11), S. 2761-2765.
- [5] D. de Beer und P. Stoodley. 1995. *Relation between the structure of an aerobic biofilm and transport phenomena*. Water Sci. Tech. 32, S. 11-18.
- [6] E. R. Berlekamp, J. H. Conway, R. K. Guy. 1982. *Winning Ways for Your Mathematical Plays*. Academic Press, New York.
- [7] P. L. Bishop. 1997. *Biofilm Structure and Kinetics*. Water Sci. Tech. 36 (1), S. 287-194.
- [8] A. Brandt. 1977. *Multi-Level Adaptive Solutions to Boundary Value Problems*. Math. Comput. 31, S. 333-390.
- [9] P. Breitling, H.-J. Bungartz und A. Frank. 1999. *Hierarchical Concepts for Improved Interfaces between Modelling, Simulation and Visualization*. Proc. Vision, Modelling, and Visualization, St. Augustin, S. 269-276.
- [10] B. Brück. 1998. *Nast++: Ein objektorientiertes Framework zur modularen Strömungssimulation*. Diplomarbeit, TU München.
- [11] H.-J. Bungartz, M. Kuehn, M. Mehl, M. Hausner und S. Wuertz. 2000. *Fluid flow and transport in defined biofilms: experiments and numerical simulations on a microscale*. Water Sci. Tech. 41 (4-5), S. 331-338.
- [12] H.-J. Bungartz, M. Kuehn, M. Mehl und S. Wuertz. 2001. *Space- and Time-Resolved Simulations of Processes in Biofilm Systems on a Microscale*. In: W. Alt, M. Chaplain, M. Griebel, J. Lenz (Hrsg.), *Polymer and Cell Dynamics – Multiscale Modelling and Numerical Simulations*, Birkhäuser-Verlag, Basel. In Druck.

- [13] H.-J. Bungartz und M. Mehl. 2001. *Beyond models: requirements and chances of computational biofilms*. In: S. Wuertz, P. Wilderer, P. Bishop (Hrsg.), *Biofilms in wastewater treatment: An interdisciplinary approach*, Technomic Publishing. In Druck.
- [14] S. Chapman, T. G. Cowling. 1990. *The Mathematical Theory of Non-Uniform Gases*. Cambridge University Press.
- [15] O. Cipka und R. Helmig. 1994. *Numerical simulation of contaminat transport and biodegradation in porous and fractured-porous media*. Numerical Methods in Water Resources, A. Peters et al. (Hrsg.), Kluwer Academic Publishers, Heidelberg.
- [16] J. W. Costerton, Z. Lewandowski, D. de Beer, D. Caldwell, D. Korber und G. James. 1994. *Biofilms, the customized microniche*. J. of Bacteriol. 176: S. 2137-2142.
- [17] M. A. Champ, W. L. Pugh. 1987. *The dilemma of high-technology antifouling paints*. Oceanus 30, S. 68-77.
- [18] H. T. Chang, F. Zhu und M. Ahmed. 1999. *A Duality Model Unifying Suspended and Attached Growth in a Biological Reactor*. Water Environment Research.
- [19] W. G. Charaklis und K. C. Marshall. 1990. *Biofilms*. Wiley-Interscience, New York.
- [20] K. Debus. 1998. *Numerische Simulation der Durchströmung von Kugelschüttungen mit Hilfe einer dreidimensionalen Chimären-Gittertechnik*. Dissertation, TU München.
- [21] A. Delgado. 2000. *Numerische Simulation lokaler Strömungsvorgänge in einem Festbett-Biofilmreaktor*. In: Sonderforschungsbereich 411 – Grundlagen der aeroben biologischen Abwasserreinigung. Arbeits- und Ergebnisbericht 1998-1999-2000.
- [22] A. Dybbs, R. V. Edwards. 1995. *A new look at Porous Media Fluid Mechanics – Darcy to Turbulent*, NATO ASI Series 82 (1), S. 199.
- [23] H. Eberl, D. Parker, M. van Loosdrecht. 2001. *A new Deterministic Spatio-Temporal Continuum Model For Biofilm Development*. J. of Theoretical Medicine. In Druck.
- [24] H. Eberl, C. Picioreanu, M. C. M. van Loosdrecht. 2000. *Modeling geometrical heterogeneity in biofilms*. In: A. Pollard et al. (Hrsg.), *High Performance Computing Systems and Applications*, Kluwer Acad. Publishers.
- [25] H.-C. Flemming. 1994. *Biofilme, Biofouling und mikrobielle Schädigung von Werkstoffen*. Habilitationsschrift, Forschungs- und Entwicklungsinstitut für Industrie- und Siedlungswasserwirtschaft sowie Abfallwirtschaft e. V. Stuttgart, Kommissionsverlag R. Oldenburg, München, Stuttgarter Berichte zur Siedlungswasserwirtschaft, Band 129.
- [26] H.-C. Flemming, P. Wingender, R. Moritz, W. Borchard und C. Mayer. 1998. *Extracellular polymeric substances – the material which forms the biofilm*. Conf. Proc. Forum for Appl. Biotech., 24.-25. September, Brügge.
- [27] D. G. Green. 1990. *Cellular Automata Models in Biology*. Math. Comp. Mod. 13, S. 69-74.

- [28] M. Griebel. 1989. *Zur Lösung von Finite-Differenzen- und Finite-Element-Gleichungen mittels der Hierarchischen-Transformations-Mehrgitter-Methode*. Dissertation, TU München.
- [29] M. Griebel, T. Dornseifer und T. Neunhoeffler. 1995. *Numerische Simulation in der Strömungsmechanik – eine praxisorientierte Einführung*. Vieweg, Braunschweig.
- [30] D. H. Grunau, S. Chen, K. Eggert. 1993. *A Lattice Boltzmann Modell for multiphase flow*. Phys. Fluids A5 (10), S. 2557-2561.
- [31] H. Gundlach. 2000. *Simulation und Visualisierung von Biofilmwachstum*. Diplomarbeit, TU München.
- [32] M. Henze, W. Gujer, T. Mino, M. van Loosdrecht. 2000. *Activated Sludge Models ASM1, ASM2, ASM2d and ASM3*. Scientific and Technical Report no. 9, IWA Task Group on Mathematical Modelling for Design and Operation of Biological Wastewater Treatment.
- [33] S. W. Hermanowicz. 1999. *Two-dimensional Simulations of Biofilm Development: Effects of External Environmental Conditions*. Water Sci. Tech. 39 (7), S. 107-114.
- [34] S. W. Hermanowicz, U. Schindler und P. Wilderer. 1996. *Fractal Structure of Biofilms: New Tools for Investigation of Morphology*. Water Sci. Tech. 32 (8), S. 99-105.
- [35] V. Hösel und S. Walcher. 1999. *On the mathematical modeling of biofilms*. GSF-Bericht 10/99, GSF-Forschungszentrum für Umwelt und Gesundheit.
- [36] H. Horn und D. C. Hempel. 1997. *Substrate Utilization and Mass Transfer in an Autotrophic Biofilm System: Experimental Results and Simulation*. Biotech. Bioeng. 53 (4), S. 363-371.
- [37] H. Horn und M. Wulkow. 1999. *RIOVAL: A management tool for water quality modelling in small streams*. <http://www.wst.shuttle.de/cit/p1rioval.pdf>.
- [38] H. Horn, T. R. Neu und M. Wulkow. 1999. *Modelling the Structure and Function of Extracellular Polymeric Substances in Biofilms with new Numerical Techniques*. International Specialist Conference EPS – Extracellular Polymeric Substances. Duisburg.
- [39] A. R. Johnsen, M. Hausner, A. Schnell und S. Wuertz. 2000. *Evaluation of fluorescently labeled lectins for the localization of EPS in Sphingomonas biofilms*. Appl. Environ. Microbiol. 66, S. 3487-3491.
- [40] P. Knabner. 1991. *Mathematische Modelle für Transport und Sorption gelöster Stoffe in porösen Medien*. In: Methoden und Verfahren der mathematischen Physik, Band 36, Verlag Peter Lang, Frankfurt/Main.
- [41] C. Kopp und R. Nießner. 1998. *Photoacoustic Depth Profiling of Layered Samples*. In: Analyst 123, S. 547-550.
- [42] C. Kopp und R. Nießner. 1999. *Optoacoustic sensor head for depth profiling*. Appl. Phys. B68, S. 719-725.

- [43] D. R. Korber, J. R. Lawrence, M. J. Hendry und D. E. Caldwell. 1992. *Programs for determining statistically representative areas of microbial biofilms*. Binary 4: 204-210.
- [44] M. Krafczyk, P. Lehmann, O. Filippova, D. Hänel und U. Lantermann. 1999. *Lattice Boltzmann Simulations of Complex Multiphase Flows*. In: *Multifield Problems in Solid and Fluid Mechanics*, W. Wendland (Hrsg.), Springer Proceedings, Berlin.
- [45] M. Kuehn. 2001. *In-Situ-Analyse der Architektur von Biofilmen bei unterschiedlichen Strömungsbedingungen*. Dissertation, TU München. In Fertigstellung.
- [46] M. Kuehn, M. Hausner, H.-J. Bungartz, M. Wagner, P. A. Wilderer und S. Wuertz. 1998. *Automated confocal laser scanning microscopy and semiautomated image processing for the analysis of biofilms*. Appl. Environ. Microbiol. 64: S. 4115-4127.
- [47] M. Kuehn, M. Mehl, M. Hausner, H.-J. Bungartz und S. Wuertz. 2001. *Time-Resolved Study of Biofilm Architecture and Transport Processes Using Experimental and Simulation Techniques: The Role of EPS*. Water Sci. Tech. 43 (8), S. 143-151.
- [48] M. Kuehn, U. Schindler, P. A. Wilderer und S. Wuertz. 2001. *Quantitative microscopy in biofilm studies*. In: H.-C. Flemming, T. Neu und T. Giebe (Hrsg.). *Investigation of Biofilms*, Technomic Publishing. In Druck.
- [49] J. R. Lawrence, G. M. Wolfaardt und D. R. Korber. 1994. *Determination of Diffusion Coefficients in Biofilms by Confocal Laser Microscopy*. Appl. Environ. Microbiol. 60, S. 1166-1173.
- [50] Z. Lewandowski und P. Stoodley. 1995. *Flow induced vibrations, drag force, and pressure drop in conduits covered with biofilm*. Water Sci. Tech. 32 (8), S. 19-26.
- [51] Z. Lewandowski, D. Webb, M. Hamilton und G. Harkin. 1999. *Quantifying Biofilm Structure*. Water Sci. Tech. 39 (7), S. 71-76.
- [52] M. C. M. van Loosdrecht, C. Picioreanu, J. J. Heijnen. 1997. *A More Unifying Hypothesis for Biofilm Structures*. FEMS Microbiol. Ecol. 24, S. 181-183.
- [53] J. Monod. 1949. *Recherches sur la croissance des cultures bactériennes*. Hermann Verlag, Paris.
- [54] E. Morgenroth, H. Eberl und M. C. M. van Loosdrecht. 1999. *Evaluating 3-D and 1-D mathematical models for mass transport in heterogeneous biofilms*. Proc. 4th IAWQ Conference on Biofilm Systems, 17.-20. Oktober, New York.
- [55] G. Müller. 1999. *Radial Void Fraction Correlation for Annular Packed Beds*. AIChE Journal 45 (11), S. 2485-2460.
- [56] R. Mundani. 2000. *Strömungssimulation auf Oktalbäumen*. Diplomarbeit, Institut für Informatik, TU München.
- [57] I. Nagy. 1999. *Simulation und Visualisierung von Biofilmwachstum mittels zellulärer Automaten. Einführung in das Programmpaket CAsim*. Systementwicklungsprojekt, Institut für Informatik, TU München.

- [58] C. Nicolella, M. C. M. van Loosdrecht. 1999. *Terminal Settling Velocity and Bed-Expansion Characteristics of Biofilm-Coated Particles*. Biotech. Bioeng. 62, S. 62-70.
- [59] P. H. Nielsen, K. Andreasen, N. Lee und M. Wagner. 1999. *Use of microautoradiography and fluorescent in situ hybridization for characterization of microbial activity in activated sludge*. In Water Sci. Tech. 39 (1), S. 1-9.
- [60] D. R. Noguera, G. Pizarro, D. A. Stahl und B. E. Rittmann. 1999. *Simulation of Multispecies Biofilm Development in Three Dimensions*. Water Sci. Tech. 39 (7), S. 123-130.
- [61] C. Picioreanu, M. C. M. van Loosdrecht, J. J. Heijnen. 1998. *A new Combined Differential-Discrete Cellular Automaton Approach for Biofilm Modeling: Application for Growth in Gel Beads*. Biotech. Bioeng. 57 (6).
- [62] C. Picioreanu, M. C. M. van Loosdrecht und J. J. Heijnen. 1999. *Discrete-differential modeling of biofilm structure*. Water Sci. Tech. 39 (7), S. 115-122.
- [63] M. Pögl. 1998. *Erweiterungen von Nast++: Drei Raumdimensionen, Geometriebeschreibung und Parallelisierung*. Systementwicklungsprojekt, Institut für Informatik, TU München.
- [64] M. Pögl. 2000. *Verteilte transiente Simulation einer Fluid-Struktur-Wechselwirkung in einer Mikropumpe*. Diplomarbeit, Institut für Informatik, TU München.
- [65] Y. H. Quian, D. d'Humieres und P. Lallemand. 1992. *Lattice BGK models for Navier Stokes equations*. Europhys. Lett. 17 (6), S. 479-484.
- [66] E. Rank und M. Krafczyk. 2000. *Computersimulation von Mehrphasenströmungen in Festbett-Biofilm-Reaktoren*. In: Sonderforschungsbereich 411 – Grundlagen der aeroben biologischen Abwasserreinigung. Arbeits- und Ergebnisbericht 1998-1999-2000.
- [67] P. Reichert. 1994. *AQUASIM – A tool for simulation and data analysis of aquatic systems*. Water Sci. Tech. 30 (2), S. 21-30.
- [68] P. Reichert und O. Wanner. 1997. *Movement of solids in biofilms: Significance of liquid phase transport*. Water Sci. Tech. 36 (1), S. 321-328.
- [69] G. Ristow. 1994. *Granular Dynamics: a Review about Recent Molecular Dynamics Simulations of Granular Materials*. Annual Reviews of Computational Physics I, S. 275-308.
- [70] B. E. Rittmann, M. Pettis, H. W. Reeves und D. A. Stahl. 1997. *How biofilm clusters affect substrate flux and ecological selection*. Preprint.
- [71] A. Roe. 1994. *Biofilm heterogeneity*. Center of Biofilm Engineering News. Vol. 2, Bozeman, Montana.
- [72] K. Rottschäfer und D. Vortmeyer. 1996. *Geschwindigkeitsverteilungen in durchströmten Füllkörperschüttungen*. Dissertation, TU München.
- [73] H. Samet. 1984. *The Quadtree and Related Hierarchical Data Structures*. ACM Computing Surveys 16, S. 187-260.

- [74] R. Scheidegger. 1974. *The physics of flow through porous media*. University of Toronto Press.
- [75] M. Schelkle. 1996. *LB-Verfahren zur Simulation dreidimensionaler Zweiphasenströmungen mit freien Oberflächen*. Dissertation, Universität Stuttgart.
- [76] A. Schnell, P. Wattiau, D. Springael und S. Wuertz. *Dual in situ detection of PAH-degrading Sphingomonas sp. LB126 based on gfp labelling and strain-specific 16S rRNA-targeted oligonucleotide*. Eingereicht.
- [77] P. S. Steward, R. Murga, R. Srinivasan und D. de Beer. 1995. *Biofilm structural heterogeneity visualized by three microscopic methods*. Water Res. 29, S. 2006-2009.
- [78] J. Stoer und R. Bulirsch. 1990. *Numerische Mathematik 2*. Dritte Auflage, Springer-Lehrbuch, Heidelberg.
- [79] P. Stoodley, D. de Beer und Z. Lewandowski. 1994. *Liquid flow in biofilm systems*. Appl. Environ. Microbiol. 60: S. 2711-2716.
- [80] K. Stüben. 2001. *Algebraic Multigrid (AMG): An Introduction with Applications*. In: U. Trottenberg, C. W. Oosterlee, A. Schüller: Multigrid, Academic Press, London, San Diego.
- [81] J. Tölke, M. Krafczyk, M. Schulz und E. Rank. 2001. *Discretization of the Boltzmann-Equation in velocity space using a Galerkin approach*. Journal of Computer Physics Communications. In Druck.
- [82] U. Trottenberg, C. W. Oosterlee und A. Schüller. 2001. *Multigrid*. Academic Press, London, San Diego.
- [83] S. Vandewalle. 1993. *Parallel Multigrid Waveform Relaxation for Parabolic Problems*. B. G. Teubner Verlag, Stuttgart.
- [84] M. Wagner, S. Juretschko, G. Lind, H. Lemmer und K.-H. Schleifer. 1998. *Neueste Entwicklungen in der in-situ-Charakterisierung mikrobieller Biozönosen in Abwasser, Oberflächengewässern, Grund- und Trinkwasser. Die Anwendung von in-situ-Hybridisierung zur Identifizierung der an der Stickstoff- und Phosphorentfernung beteiligten Bakterien in Kläranlagen*. In: Münchener Beiträge zur Abwasser-, Fischerei- und Flussbiologie Bd. 52. R. Oldenbourg Verlag, München, Wien.
- [85] O. Wanner, P. Reichert. 1996. *Mathematical Modeling of Mixed-Culture Biofilm*. Biotech. Bioeng. 49, S. 172-184.
- [86] P. A. Wilderer, H.-J. Bungartz, H. Lemmer, M. Wagner und S. Wuertz. 2001. *From Microscale to Macroscale: How Modern Analytical Tools May Help the Engineer to Do a Better Job*. Water Research. In Druck.
- [87] J. W. T. Wimpenny und R. Colasanti. 1997. *A Unifying Hypothesis for the Structure of Microbial Biofilms Based on Cellular Automaton Models*. FEMS Microbiol. Ecol. 22 (1), S. 1-16.
- [88] J. Wingender, T. Neu und H.-C. Flemming. 1999. *Bacterial extracellular polymer substances*. In: Wingender, J., Neu, T., Flemming, H.-C. (Hrsg.): Bacterial extracellular polymer substances, Springer, Heidelberg, Berlin.

- [89] G. M. Wolfaardt, J. R. Lawrence, R. D. Robarts und D. E. Caldwell. 1995. *Bioaccumulation of the herbicide diclofop in extracellular polymers and its utilization by biofilm community during starvation*. Appl. Environ. Microbiol. 61, S. 152-158.
- [90] T. C. Zhang und P. L. Bishop. 1994. *Density, Porosity and Pore Structure of Biofilms*. Water Res. 28, S. 2267-2277.
- [91] NASA Langley Research Center, Aero- and Gas-Dynamics Division  
<http://ad-www.larc.nasa.gov/cab/People/SP/Web/IMAGES/eet4.gif>
- [92] Ruhr-Universität Bochum, Medizinische Fakultät, Abteilung für Allgemeine und Spezielle Pathologie  
<http://homepage.ruhr-uni-bochum.de/Olaf.Anhenn/CLSM.htm>
- [93] Technische Universität München, Lehrstuhl für Informatik V  
<http://www5.in.tum.de/forschung/visualisierung/3dbeam.html>
- [94] Technische Universität München, Lehrstuhl für Informatik V  
<http://www5.in.tum.de/persons/schultem/forschung/SFB411/A4.html>
- [95] Technische Universität München, Lehrstuhl für Informatik V  
<http://www5.in.tum.de/persons/schultem/forschung/SFB411/Nast++.html>
- [96] Technische Universität München, Lehrstuhl für Informatik V  
<http://www5.in.tum.de/persons/schultem/forschung/SFB411/Trans3d.html>
- [97] Advanced Visual Systems  
[http://www.AVS.com/software/soft\\_t/avsxps.html](http://www.AVS.com/software/soft_t/avsxps.html)
- [98] Fachhochschule Bingen, Fachbereich Verfahrenstechnik  
<http://www.fw.online.de/3kapitel.htm>
- [99] Fraunhofer Institut für Grenzflächen- und Bioverfahrenstechnik  
[http://www.igb.fhg.de/Presse/dt/1998/PI\\_Festbettreaktor.dt.html](http://www.igb.fhg.de/Presse/dt/1998/PI_Festbettreaktor.dt.html)
- [100] Message Passing Interface Forum  
<http://www.mpi-forum.org/>
- [101] Zeiss AG  
<http://www.zeiss.de>