

# Modellierungsseminar



**SPP1253**

Optimal Control-Based Feedback  
Stabilization of Multi-Field Flow  
Problems

**Navier-Stokes-Gleichung  
gekoppelt mit dem Transport von  
(reaktiven) Substanzen**

Heiko Weichelt

11. August 2010

**Betreuer:**

Prof. Dr. Peter Benner

Dr. René Schneider

**Institut:**

Fakultät für Mathematik der TU Chemnitz  
Mathematik in Industrie und Technik



## Inhaltsverzeichnis

<b>0</b>	<b>Notation</b>	<b>3</b>
<b>1</b>	<b>Aufgabenstellung</b>	<b>4</b>
1.1	Projekt . . . . .	4
1.2	Navier-Stokes-Strömung gekoppelt mit (passivem) Transport ei- niger (reaktiver) Substanzen . . . . .	4
<b>2</b>	<b>Einleitung</b>	<b>6</b>
2.1	Strömungslehre . . . . .	6
2.2	Regelungstheorie . . . . .	7
<b>3</b>	<b>Mathematische Modellierung</b>	<b>8</b>
3.1	Strömungsmodellierung . . . . .	8
3.2	Modell zur Ausbreitung des gelösten Stoffes . . . . .	8
3.3	Gekoppeltes Modell und Randbedingungen . . . . .	12
3.4	Grenzschichten . . . . .	13
3.5	Feedback-Steuerung . . . . .	15
<b>4</b>	<b>Numerische Simulation</b>	<b>19</b>
4.1	Umsetzung des Modells in NAVIER . . . . .	19
4.1.1	Mathematische Grundlagen für NAVIER . . . . .	20
4.1.2	Problemimplementation . . . . .	21
4.2	Beispielsimulation . . . . .	24
4.2.1	Gebietstriangulation . . . . .	24
4.2.2	Konfigurationsdaten sowie Anfangs- und Randbedingungen	25
4.2.3	Ergebnisauswertung . . . . .	28
4.3	Zusatzfunktionen für die Regelung . . . . .	34
4.3.1	<i>outflow</i> -Funktion . . . . .	34
4.3.2	Systemmatrizen . . . . .	34
4.3.3	Feedback-Funktion . . . . .	34
<b>5</b>	<b>Zusammenfassung/Ausblick</b>	<b>35</b>
5.1	Zusammenfassung . . . . .	35
5.2	Ausblick . . . . .	36
<b>6</b>	<b>Schlusswort</b>	<b>37</b>
<b>7</b>	<b>Quellenverzeichnis</b>	<b>38</b>
<b>8</b>	<b>Verzeichnisse</b>	<b>39</b>

## 0 Notation

Als Notation dient im Folgenden:

- Allgemeine Notation im  $\mathbb{R}^d$ :
  - skalare Variable/Parameter:  $x$
  - vektorwertige Variable/Parameter:  $\mathbf{x}$
  - skalare Funktion:  $f(\dots)$
  - vektorwertige Funktion:  $\mathbf{f}(\dots)$
  - Nablaoperator:  $\nabla = [\frac{\partial}{\partial x_1}, \dots, \frac{\partial}{\partial x_d}]^T$
  - Laplaceoperator:  $\Delta = \sum_{i=1}^d \frac{\partial^2}{\partial x_i^2}$
  - Gradient einer skalaren Funktion  $g$ :  $\text{grad } g := \nabla g = [\frac{\partial g}{\partial x_1}, \dots, \frac{\partial g}{\partial x_d}]^T$
  - Divergenz eines Vektorfeldes  $\mathbf{w}$ :  $\text{div } \mathbf{w} = \sum_{i=1}^d \frac{\partial w_i}{\partial x_i}$
  - Euklidisches Skalarprodukt zweier Vektoren:  $\mathbf{x} \cdot \mathbf{y} = \sum_{i=1}^d x_i \cdot y_i$
  - partielle Integration in mehreren Dimensionen über Gebiet  $V_0$ :

$$\int_{V_0} \mathbf{w} \cdot \text{grad } g \, d\mathbf{V} = \int_{\partial V_0} g \cdot \mathbf{w} \cdot \mathbf{n} \, ds - \int_{V_0} g \cdot \text{div } \mathbf{w} \, d\mathbf{V}$$

- Gaußscher Integralsatz:

$$\int_{V_0} \text{div } \mathbf{v} \, d\mathbf{V} = \int_{\partial V_0} \mathbf{v} \cdot \mathbf{n} \, ds$$

- Problemspezifische Variablen:
  - Raumdimension:  $d \in \mathbb{N}$
  - Zeit:  $t \in \mathbb{R}$
  - Ort:  $\mathbf{x} = [x_1, \dots, x_d]^T \in \mathbb{R}^d$
  - Gebiet:  $\Omega \subset \mathbb{R}^d$
  - Rand:  $\Gamma = \partial\Omega$
  - Normalenrichtung:  $\mathbf{n} \in \mathbb{R}^d$
  - Geschwindigkeit:  $\mathbf{v}(t, \mathbf{x}) = [v_1(t, \mathbf{x}), \dots, v_d(t, \mathbf{x})]^T \in \mathbb{R}^d$
  - Druck:  $p(t, \mathbf{x}) \in \mathbb{R}$
  - Konzentration:  $c(t, \mathbf{x}) \in \mathbb{R}$
  - von außen einwirkende Kraft:  $\mathbf{f}$
- Problemspezifische konstante Parameter:
  - Dichte:  $\rho \in \mathbb{R}$
  - dynamische Viskosität:  $\eta \in \mathbb{R}$
  - kinematische Viskosität:  $\nu := \frac{\eta}{\rho} \in \mathbb{R}$
  - Diffusionskonstante:  $D \in \mathbb{R}$
  - charakteristische Länge:  $d_{ch} \in \mathbb{R}$
  - charakteristische Geschwindigkeit:  $v_{ch} \in \mathbb{R}$

# 1 Aufgabenstellung

## 1.1 Projekt

Die Arbeit ist Teil des **DFG Schwerpunktprogrammes 1253: Optimierung mit partiellen Differentialgleichungen** mit dem Projekttitel:

*Optimal Control-Based Feedback Stabilization in Multi-Field Flow Problems.*

Das Projekt wird in Zusammenarbeit von Prof. Dr. Peter Benner (TU Chemnitz) und Prof. Dr. Eberhard Bänsch (FAU Erlangen) durchgeführt.

Weitere Informationen sind unter [DFG] zu finden. Ziel dieses Projektes ist die Herleitung und Untersuchung numerischer Algorithmen für die Stabilisierung von Mehrfeld-Strömungsproblemen über Randeingriff, basierend auf Optimalsteuerungsmethoden.

Dabei werden drei verschiedene Szenarien betrachtet:

1. Navier-Stokes-Strömung gekoppelt mit (passivem) Transport einiger (reaktiver) Substanzen
2. Phasenübergang flüssig-fest mit Konvektion
3. Stabilisierung einer Strömung mit freier kapillarer Oberfläche

Für das Modellierungsseminar wird das erste Szenario näher betrachtet. Eine genauere Beschreibung der Aufgabenstellung erfolgt im nächsten Abschnitt. Die Szenarien 2 und 3 werden hier nicht betrachtet.

## 1.2 Navier-Stokes-Strömung gekoppelt mit (passivem) Transport einiger (reaktiver) Substanzen

Dieses Beispiel ist ein eher grobes Modell eines Reaktors, in dem chemische Substanzen durch ein Strömungsfeld transportiert werden und an der Oberfläche eines Hindernisses reagieren. Die Reaktion wird als schnell im Vergleich zu Diffusion und Transport angenommen, sodass sie durch eine homogene Dirichlet-Randbedingung vereinfacht modelliert werden kann.

Dieses Modell wird in der vorliegenden Arbeit näher untersucht und begründet.

Die Regelung erfolgt durch Variation der Einströmungs-Randbedingung. Dies ist dadurch motiviert, dass der Reaktionsprozess hauptsächlich durch den Transport der chemischen Substanz vom Einströmrand zur Reaktionsoberfläche beeinflusst wird.

Als einfaches Modellproblem wird die Anordnung wie in Abbildung 1 betrachtet. Das Gebiet  $\Omega$  ist durch den Rand  $\partial\Omega = \Gamma_{\text{in}} \cup \Gamma_{\text{out}} \cup \Gamma_{\text{wall}} \cup \Gamma_r$  begrenzt. Dabei beschreibt der Teilrand  $\Gamma_r$  ein Strömungshindernis, das umflossen wird und auf dessen Oberfläche die Reaktion stattfindet. Sämtliche Betrachtungen beziehen sich auf die Raumdimension  $d = 2$ , sodass  $\Omega \subset \mathbb{R}^2$ .

Die Berechnung soll im Zeitintervall  $t \in [t_0, t_{\text{end}}]$  betrachtet werden, sodass man als Lösungsraum den Raum-Zeit-Zylinder  $Q := \Omega \times [t_0, t_{\text{end}}]$  erhält. Die Randbedingungen werden in Abschnitt 3.4 definiert.

Vor dem Zeitpunkt  $t_0 = 0$  befindet sich das Fluid im Inneren von  $\Omega$  in Ruhe und die Konzentration des gelösten Stoffes ist Null. Nun lässt man am Rand  $\Gamma_{\text{in}}$  das Fluid mit dem gelösten Stoff einströmen, wodurch sich ein Strömungsfeld einstellt und der gelöste Stoff sich im Gebiet  $\Omega$  ausbreiten kann.

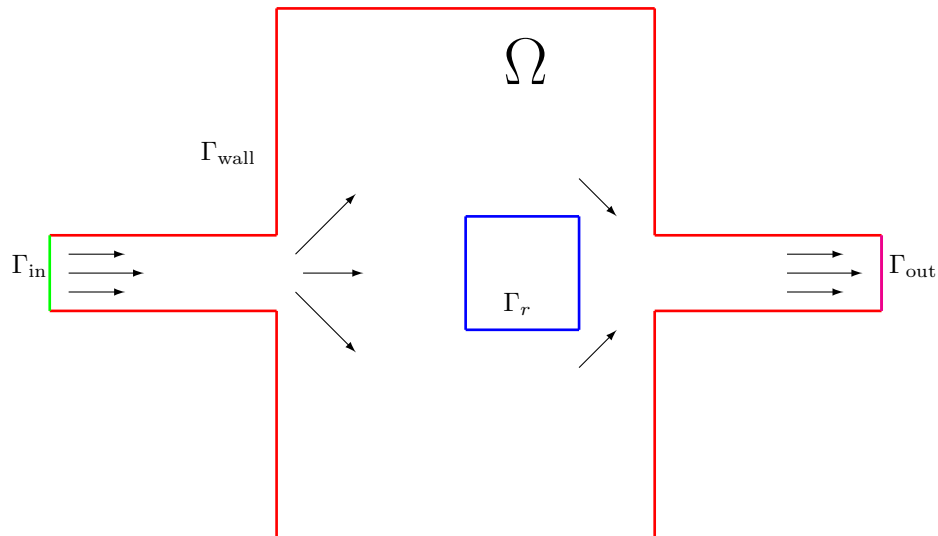


Abbildung 1: Gebietsdefinition

Für kleine Reynoldszahlen stellt sich relativ zügig eine stationäre Strömung ein. Gelangt der gelöste Stoff an den Reaktionsrand  $\Gamma_r$ , so wird dieser im Moment des Auftreffens durch Reaktion vollständig aufgebraucht, sodass kein gesonderter Reaktionsterm betrachtet werden muss. Ziel ist es nun, die Menge, die an Stoff aufgebraucht wird, durch Beeinflussung der Einströmbedingungen zu steuern bzw. zu stabilisieren. Der gelöste Stoff, der das Hindernis umfließt und zum Ausströmrand  $\Gamma_{\text{out}}$  gelangt, verlässt ohne Widerstand das Gebiet.

Die Zielstellung des Modellierungsseminars bestand darin, die mathematischen Modelle für Strömung sowie Stofftransport herzuleiten und zu begründen, den Ablauf zu simulieren sowie zu verifizieren und den Programmcode für die Steuerung vorzubereiten. Die mathematischen Modelle werden in Abschnitt (3) betrachtet.

## 2 Einleitung

### 2.1 Strömungslehre

Die Strömungslehre, als Teilgebiet der Mechanik, beschäftigt sich mit der Beschreibung sich bewegender *Fluide*. Unter einem Fluid versteht man „einen Stoff, der einer scherenden Beanspruchung unbegrenzt nachgibt“ [STM]. Beispiele für diese Stoffklasse sind Flüssigkeiten und Gase. Die Unterscheidung zwischen *Newtonschen* und *Nicht-Newtonschen Fluiden* fällt in das Gebiet der Rheologie, welches die Strömungsmechanik als Teilgebiet besitzt. *Nicht-Newtonsche Fluide* haben keine gleich bleibende Viskosität, was die mathematische Beschreibung sehr erschwert. In dieser Arbeit werden aber nur *Newtonsche Fluide* betrachtet.

Die Betrachtung von Fluiden im Sinne der Strömungslehre erfolgt in vielen Anwendungsgebieten, sei es im Flugzeug- oder Automobilbau, wo man *umströmende Gase* betrachtet, oder im Bereich von Kraftwerken zur Energiegewinnung, wo Gase durch Rohrleitungen fließen. Gerade im zuerst erwähnten Bereich wird derzeit noch viel mit Windkanälen gearbeitet, in denen Modelle unter angepassten Verhältnissen mit realen Versuchen getestet werden. Natürlich ist dies sehr aufwändig und kostenintensiv, zumal nicht alle möglichen Szenarien im Experiment simuliert werden können. Daraus ergibt sich der Wunsch, solche Versuche am Rechner zu simulieren. Dies erspart nicht nur Kosten und Zeit, sondern es könnten auch problemlos Grenzfälle für die verschiedenen Versuchsanordnungen betrachtet werden. Jedoch sind die mathematischen Modelle für solch komplexe Anordnungen sehr kompliziert und zeitintensiv in der Berechnung.

Für einige Versuchsanordnungen sind schon gute Modelle bekannt und verifiziert. Hier entsteht der Wunsch, durch Eingriffe von außen die Strömungen zu einem gewünschten Zustand oder einem angestrebten Verhalten zu steuern. Dazu wird die Simulation solcher Strömungen mit einer mathematischen Regelung verknüpft, welches der Grundgedanke des auf diese Arbeit aufbauenden Projektes ist.

Zur mathematischen Beschreibung von sich bewegenden Fluiden gibt es mehrere Ansätze. Eine wichtige Unterscheidung hierfür bildet die Eigenschaft der *Kompressibilität*. Eine recht allgemeine Form zur Beschreibung von Fluiden, die keinen großen Temperaturschwankungen ausgesetzt sind, bilden die *kompressiblen Navier-Stokes-Gleichungen*

$$\rho \left( \frac{\partial \mathbf{v}}{\partial t} + \sum_{i=1}^d v_i \frac{\partial \mathbf{v}}{\partial x_i} \right) - \eta \Delta \mathbf{v} - (3\lambda + \eta) \nabla \nabla \cdot \mathbf{v} + \nabla p = \mathbf{f}, \quad (1)$$

$$\frac{\partial \rho}{\partial t} + \nabla \cdot (\rho \mathbf{v}) = 0, \quad (2)$$

$$k\rho^\gamma = p. \quad (3)$$

Durch Vereinfachung verschiedener Annahmen können weitere Modelle abgeleitet werden. Die hier geführte Annahme der *Inkompressibilität*, also einer räumlich und zeitlich konstanten Dichte, führt zu den inkompressiblen Navier-Stokes-Gleichungen, die in Abschnitt 3.1 vorgestellt werden. Diese Annahme ist insofern gerechtfertigt, da sich mit Flüssigkeiten und langsam strömenden Gasen beschäftigt wird, bei denen Dichteänderungen vernachlässigbar sind.

Um die Strömungen nun in einen erwünschten Zustand zu führen, wird sich der mathematischen Regelungstheorie bedient.

## 2.2 Regelungstheorie

Die Grundidee der Regelungstheorie besteht darin, Systeme zu steuern, um ein gewünschtes Verhalten zu erzielen. Hierbei können die Zustandsvariablen  $\mathcal{X}(t)$  im Inneren des Systems oft nicht direkt beobachtet und beeinflusst werden. Meist kann man nur die Ausgangsvariablen  $\mathcal{Y}(t)$  beobachten bzw. messen, die zur Auswertung eines Kostenfunktional  $J(\mathcal{X}, \mathcal{U})$  genutzt werden. Die Eingangsvariablen  $\mathcal{U}(t)$  werden durch vorgegebene Funktionen beeinflusst.

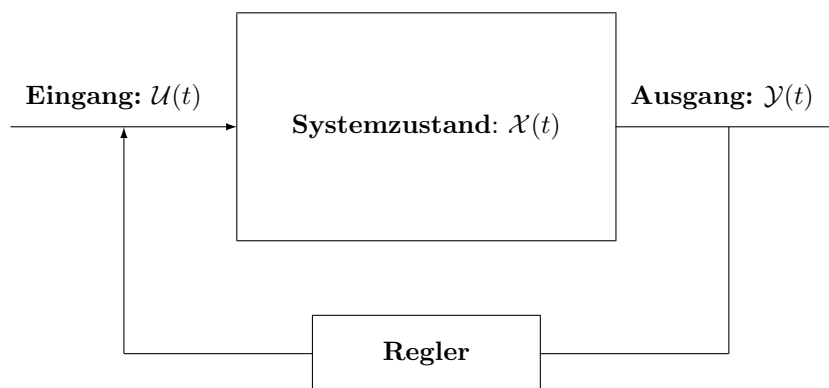


Abbildung 2: Geschlossener Regelkreis

Werden die Ausgangs- oder Zustandsvariablen, nachdem sie durch einen *Regler* modifiziert worden sind, auf den Eingang zurückgeführt, so spricht man von einem *geschlossenen Regelkreis* bzw. einer *Feedback-Steuerung*. Schematisch ist dies in Abbildung 2 für eine Ausgangsrückführung dargestellt. Der Regler hat dabei die Aufgabe, die Eingangsdaten so zu beeinflussen, dass die Ausgangsdaten in einen erwünschten Zustand gebracht werden, z.B. das Kostenfunktional minimiert wird. Die Kosten, die die Regelung hervorruft, können unter verschiedenen Kriterien betrachtet werden. Beispiel dafür sind wegoptimale beziehungsweise zeitoptimale Steuerungen, die betrachtet werden können. Dementsprechend muss das Kostenfunktional an diese Kriterien angepasst werden. Einige mathematische Grundlagen, bezogen auf das Thema, werden in Abschnitt 3.5 erläutert. Grundlagen zur Regelungstheorie sind ausführlich in [SRT] nachzulesen. Im Rahmen des Modellierungsseminars ist noch keine Feedbacksteuerung zu entwickeln. Dies soll in weiteren Arbeiten geschehen. Lediglich der Programmcode soll für eine derartige Steuerung vorbereitet werden. Dazu zählen die Auswertung von Kostenfunktionalen und der Einfluss auf die Eingangsgrößen, wie auch das Herausschreiben der Systemmatrizen, die zur Berechnung einer Regelung benötigt werden.

### 3 Mathematische Modellierung

Zur Beschreibung von physikalischen und chemischen Prozessen werden häufig *partielle Differentialgleichungen (PDE)* benutzt. Dabei handelt es sich um Gleichungen bzw. Gleichungssysteme, die den Zusammenhang verschiedener unbekannter Funktionen mehrerer Veränderlichen und deren partiellen Ableitungen darstellen. Als *Ordnung* der PDE wird die Zahl  $n \in \mathbb{N}$  bezeichnet, welche den Grad der höchsten partiellen Ableitung beschreibt. Eine PDE zweiter Ordnung in impliziter Form für eine unbekannte Funktion  $u(x_1, x_2)$  die von zwei Variablen abhängt, hat die Form

$$F(x_1, x_2, u, \frac{\partial u}{\partial x_1}, \frac{\partial u}{\partial x_2}, \frac{\partial^2 u}{\partial x_1 \partial x_2}, \frac{\partial^2 u}{\partial^2 x_1}, \frac{\partial^2 u}{\partial^2 x_2}) = 0.$$

Die explizite Lösung solcher PDEs ist nur für einfache Spezialfälle möglich. Numerische Lösungsansätze und -techniken werden im 4. Kapitel vorgestellt. Einen guten Einblick in dieses Themengebiet findet man bei [NPDE].

Die PDEs für Szenario 1 werden wie folgt motiviert.

#### 3.1 Strömungsmodellierung

Das Modell für die Strömung des Fluids wird durch die *inkompressiblen Navier-Stokes-Gleichungen* beschrieben (NSE):

$$\frac{\partial \mathbf{v}}{\partial t} - \frac{1}{\text{Re}} \Delta \mathbf{v} + \mathbf{v} \cdot \nabla \mathbf{v} + \nabla p = \mathbf{f} \quad \text{auf } \Omega, \quad (4)$$

$$\nabla \cdot \mathbf{v} = 0 \quad \text{auf } \Omega. \quad (5)$$

Dies ist eine PDE der Ordnung 2 für die gesuchten Funktionen  $\mathbf{v}, p$ , die beide abhängig von Zeit und Ort sind. Zur Entdimensionierung der PDE dient die *Reynoldszahl*, die durch

$$\text{Re} := \frac{\rho \cdot v_{\text{ch}} \cdot d_{\text{ch}}}{\eta}$$

beschrieben wird.

Als rechte Seite  $\mathbf{f}$  betrachtet man Volumenkräfte, wie die Schwerkraft. Solche Kräfte werden in unserem Modell vernachlässigt, sodass die rechte Seite stets Null ist. Die hier gezeigte NSE ist ein Sonderfall der *kompressiblen Navier-Stokes-Gleichung*. Da Flüssigkeiten und langsam strömende Gase betrachtet werden, kann man eine konstante Dichte voraussetzen (Inkompressibilität), wodurch die Gleichungen (1) und (2) sich zu (4) und (5) vereinfachen.

Auf eine Herleitung wird im Folgenden verzichtet, diese kann aber unter [STM] nachgelesen werden.

#### 3.2 Modell zur Ausbreitung des gelösten Stoffes

Da nicht nur die Strömung eines Fluids, sondern auch die Ausbreitung eines gelösten Stoffes betrachtet wird, muss zusätzlich zur NSE eine *Diffusions-Konvektions-Gleichung (DCE)*

$$\frac{\partial c}{\partial t} - \frac{1}{\text{Re Sc}} \Delta c + \mathbf{v} \cdot \nabla c = 0 \quad \text{auf } \Omega, \quad (6)$$



gelöst werden. Hierbei beschreibt  $c := c(t, \mathbf{x})$  die Konzentration des gelösten Stoffes als Verhältnis von Masse und Volumen mit der Einheit  $[c] = \frac{\text{kg}}{\text{m}^3}$ . Entdimensioniert wird die Gleichung durch die Reynoldszahl  $\text{Re}$  aus (4) und der Schmidtzahl, die in [STM] durch

$$\text{Sc} := \frac{\nu}{D} = \frac{\eta}{\rho D}$$

definiert ist. Die DCE beinhaltet einen *konvektiven* Anteil

$$\mathbf{v} \cdot \nabla c,$$

welcher den Zusammenhang zwischen Konzentration und Geschwindigkeit herstellt, sowie den *Diffusionsanteil*

$$\frac{1}{\text{Re Sc}} \Delta c.$$

In Summe beschreiben sie die zeitliche Veränderung der Konzentration in dem Gebiet  $\Omega$ .

Das Vektorfeld  $\mathbf{v} := \mathbf{v}(t, \mathbf{x})$  wird dabei als bekannte Funktion vorausgesetzt, da es gesondert aus der NSE hervorgeht und durch die DCE nicht beeinflusst wird. Hierdurch erhält man eine einseitige Kopplung der NSE zur DCE in dieser Richtung.

Diese Gleichung wird im Folgenden hergeleitet.

Die Ausbreitung des gelösten Stoffes kann durch die Teilchenstromdichte  $\mathbf{J}$  beschrieben werden. Nach dem *1. Fickschen Gesetz* [WST], ist die Teilchenstromdichte bei Diffusion proportional zum Konzentrationsgradienten in Richtung des negativen Gradienten, da Stoff vom Ort hoher Konzentration zum Ort niedriger Konzentration fließt.

$$\mathbf{J} = -D \nabla c,$$

mit der Diffusionskonstanten  $D > 0$  ( $[D] = \frac{\text{m}^2}{\text{s}}$ ). Zusätzlich zur Diffusion kommt ein konvektiver Anteil  $\mathbf{v} \cdot c$  hinzu. Dadurch ergibt sich für die Teilchenstromdichte

$$\mathbf{J} := -D \nabla c + \mathbf{v} \cdot c = \begin{pmatrix} J_x \\ J_y \end{pmatrix}. \quad (7)$$

Nun wird die Teilchenstromdichte  $\mathbf{J}$  durch ein Kontrollvolumen  $\Delta V := \Delta x \times \Delta y$  betrachtet, wie in Abbildung 3 dargestellt.

Der Teilchenstrom durch die verschiedenen Grenzflächen ergibt sich durch Integration mittels Mittelpunktregel und einer Approximation dieser durch eine Taylorreihe des Seitenmittelpunktes in Bezug auf den Referenzpunkt  $(x, y)$ .

Daraus folgt

**für die untere Fläche**

$$-J_y(x + \frac{\Delta x}{2}, y) \Delta x \approx \left[ -J_y(x, y) - \frac{1}{2} \frac{\partial J_y}{\partial x} \Delta x + \mathcal{O}(\Delta x)^2 \right] \Delta x,$$

**für die obere Fläche**

$$J_y(x + \frac{\Delta x}{2}, y + \Delta y) \Delta x \approx \left[ J_y(x, y) + \frac{1}{2} \frac{\partial J_y}{\partial x} \Delta x + \mathcal{O}(\Delta x)^2 + \frac{\partial J_y}{\partial y} \Delta y + \mathcal{O}(\Delta y)^2 \right] \Delta x,$$

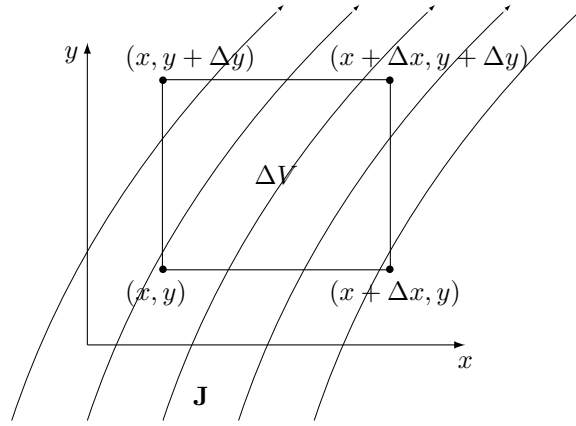


Abbildung 3: Konzentrationsveränderung in Kontrollvolumen

für die linke Fläche

$$-J_x(x, y + \frac{\Delta y}{2})\Delta y \approx \left[ -J_x(x, y) - \frac{1}{2} \frac{\partial J_x}{\partial y} \Delta y + \mathcal{O}(\Delta y)^2 \right] \Delta y,$$

und für die rechte Fläche

$$J_x(x + \Delta x, y + \frac{\Delta y}{2})\Delta y \approx \left[ J_x(x, y) + \frac{\partial J_x}{\partial x} \Delta x + \mathcal{O}(\Delta x)^2 + \frac{1}{2} \frac{\partial J_x}{\partial y} \Delta y + \mathcal{O}(\Delta y)^2 \right] \Delta y.$$

Die gesamte Teilchenstrombilanz  $\mathbf{T}_{\Delta \mathbf{v}}$  für das Volumen  $\Delta V$  ergibt sich als Summe dieser Ergebnisse. Wird vorausgesetzt, dass  $\Delta x \sim \Delta y$  so kann man  $h := \max(\Delta x, \Delta y)$  definieren und erhält

$$\mathbf{T}_{\Delta \mathbf{v}} := \left[ \frac{\partial J_x}{\partial x} + \frac{\partial J_y}{\partial y} \right] \Delta x \Delta y + \mathcal{O}(h^3). \quad (8)$$

Wird dies nun durch das Volumen  $\Delta V := \Delta x \times \Delta y \approx h^2$  geteilt, so erhält man  $\mathbf{T}_h$  mit

$$\mathbf{T}_h := \frac{\partial J_x}{\partial x} + \frac{\partial J_y}{\partial y} + \mathcal{O}(h), \quad (9)$$

$$= \operatorname{div} \mathbf{J} + \mathcal{O}(h). \quad (10)$$

Aufgrund des Masseerhaltungssatzes zieht  $\mathbf{T}_h \neq 0$  eine zeitliche Veränderung der Konzentration nach sich, das heißt

$$\operatorname{div} \mathbf{J} + \mathcal{O}(h) + \frac{\partial c}{\partial t} = 0, \quad (11)$$

wodurch  $\mathbf{T}_h$  als Quelldichte bezeichnet werden kann.

Für  $h \rightarrow 0$  ergibt sich

$$\operatorname{div} \mathbf{J} + \frac{\partial c}{\partial t} = 0. \quad (12)$$

Mit (7) eingesetzt, folgt

$$\begin{aligned}\frac{\partial c}{\partial t} &= -\operatorname{div} \mathbf{J} \\ &= -\operatorname{div} (-D\nabla c + \mathbf{v} \cdot c) \\ &= -\left[ \frac{\partial}{\partial x} \left( -D \frac{\partial c}{\partial x} + v_x \cdot c \right) + \frac{\partial}{\partial y} \left( -D \frac{\partial c}{\partial y} + v_y \cdot c \right) \right].\end{aligned}$$

Mit  $D$  konstant ergibt sich

$$\begin{aligned}\frac{\partial c}{\partial t} &= D \left[ \frac{\partial^2 c}{\partial x^2} + \frac{\partial^2 c}{\partial y^2} \right] - c \cdot \left( \frac{\partial v_x}{\partial x} + \frac{\partial v_y}{\partial y} \right) - \left( v_x \frac{\partial c}{\partial x} + v_y \frac{\partial c}{\partial y} \right) \\ &= D\Delta c - c \cdot \operatorname{div} \mathbf{v} - \mathbf{v} \cdot \nabla c.\end{aligned}\quad (13)$$

Da  $\mathbf{v}$  als gegeben aus der NSE (4) in die Gleichung eingeht, folgt mit Bedingung (5), dass der Term  $c \cdot \operatorname{div} \mathbf{v}$  verschwindet, und man erhält

$$\frac{\partial c}{\partial t} - D\Delta c + \mathbf{v} \cdot \nabla c = 0. \quad (14)$$

Diese Gleichung muss für weitere Berechnungen nun entdimensioniert werden. Dazu betrachtet man folgende dimensionslosen Variablen:

$$c^* = \frac{c}{\rho}, \quad \mathbf{v}^* = \frac{\mathbf{v}}{v_{ch}}, \quad t^* = \frac{v_{ch}}{d_{ch}} t, \quad \mathbf{x}^* = \frac{\mathbf{x}}{d_{ch}}. \quad (15)$$

Dies führt zu den Gleichungen:

$$\frac{\partial c}{\partial t} = \rho \frac{\partial c^*}{\partial t^*} \frac{\partial t^*}{\partial t} = \frac{\rho v_{ch}}{d_{ch}} \frac{\partial c^*}{\partial t^*}, \quad (16)$$

$$\frac{\partial c}{\partial x_i} = \rho \frac{\partial c^*}{\partial x_i^*} \frac{\partial x_i^*}{\partial x_i} = \frac{\rho}{d_{ch}} \frac{\partial c^*}{\partial x_i^*}, \quad (17)$$

$$\begin{aligned}D\Delta c &= D \sum_{i=1}^d \frac{\partial}{\partial x_i} \left( \frac{\partial c}{\partial x_i} \right) \stackrel{(17)}{=} D \sum_{i=1}^d \frac{\partial}{\partial x_i^*} \left( \frac{\rho}{d_{ch}} \frac{\partial c^*}{\partial x_i^*} \right) \frac{\partial x_i^*}{\partial x_i} \\ &= \frac{\rho D}{d_{ch}^2} \sum_{i=1}^d \frac{\partial^2 c^*}{\partial x_i^{*2}}.\end{aligned}\quad (18)$$

Die Gleichungen (15)-(18) eingesetzt in (14) ergeben

$$\frac{\rho v_{ch}}{d_{ch}} \frac{\partial c^*}{\partial t^*} - \frac{\rho D}{d_{ch}^2} \sum_{i=1}^d \frac{\partial^2 c^*}{\partial x_i^{*2}} + \frac{\rho v_{ch}}{d_{ch}} \mathbf{v}^* \cdot \nabla c^* = 0, \quad (19)$$

dividiert durch  $\frac{\rho v_{ch}}{d_{ch}}$ , ergibt sich

$$\frac{\partial c^*}{\partial t^*} - \frac{\rho D}{d_{ch}^2} \frac{d_{ch}}{\rho v_{ch}} \sum_{i=1}^d \frac{\partial^2 c^*}{\partial x_i^{*2}} + \mathbf{v}^* \cdot \nabla c^* = 0. \quad (20)$$

Betrachtet man nun

$$\frac{\rho D}{d_{ch}^2} \cdot \frac{d_{ch}}{\rho v_{ch}} = \frac{1}{\frac{d_{ch} \cdot v_{ch}}{D}} = \frac{1}{\frac{d_{ch} \cdot v_{ch} \cdot \rho}{\eta}} \cdot \frac{1}{\frac{\eta}{\rho D}} = \frac{1}{\operatorname{Re} \operatorname{Sc}}, \quad (21)$$

mit der Reynoldszahl  $\text{Re} = \frac{d_{ch} \cdot v_{ch} \cdot \rho}{\eta}$  aus der NSE (4) und der Schmidtzahl  $\text{Sc} = \frac{\eta}{\rho D}$ , folgt aus (20)

$$\frac{\partial c^*}{\partial t^*} - \frac{1}{\text{Re Sc}} \Delta c^* + \mathbf{v}^* \cdot \nabla c^* = 0. \quad (22)$$

Die dimensionslosen Variablen werden ab nun wieder ohne '\*' geschrieben. Somit erhält man die DCE (6).

### 3.3 Gekoppeltes Modell und Randbedingungen

Die vollständige Beschreibung unseres Modells erfolgt durch einseitige Kopplung der NSE (4),(5) und DCE (6). Als PDE ergibt sich

$$\left. \begin{aligned} \frac{\partial \mathbf{v}}{\partial t} - \frac{1}{\text{Re}} \Delta \mathbf{v} + \mathbf{v} \cdot \nabla \mathbf{v} + \nabla p &= \mathbf{f} \\ \nabla \cdot \mathbf{v} &= 0 \\ \frac{\partial c}{\partial t} - \frac{1}{\text{Re Sc}} \Delta c + \mathbf{v} \cdot \nabla c &= 0 \end{aligned} \right\} \text{ auf } \Omega, \quad (23)$$

mit den im Folgenden beschriebenen Randbedingungen.

Der Rand  $\Gamma$  unseres Gebietes  $\Omega$  ist, wie in Abbildung 1 zu sehen, in vier Abschnitte eingeteilt. Dort gelten folgende Randbedingungen.

**$\Gamma_{\text{in}}$ :** Einströmrand, durch den das Fluid mit dem gelösten Stoff einströmt. Beschrieben wird das Verhalten durch die vorgegebenen Dirichlet-Randbedingungen

$$\begin{aligned} \mathbf{v}(t, \mathbf{x}) &= u_v(t) \cdot \mathbf{v}_0(\mathbf{x}) \\ c(t, \mathbf{x}) &= u_c(t) \cdot c_0(\mathbf{x}) \end{aligned} \quad \text{auf } \Gamma_{\text{in}} \quad (24)$$

für Strömung und Konzentration. Die Einströmbedingungen  $\mathbf{v}_0, c_0$  müssen zu den angrenzenden Randbedingungen auf  $\Gamma_{\text{wall}}$  passen, sodass an den gemeinsamen Ecken keine inkonsistenten Bedingungen entstehen. Die genaue Wahl der Einströmbedingung wird in Abschnitt 4.2.2 diskutiert. Die Feedbacksteuerung greift dabei durch die Funktion  $\mathcal{U}(t) := [u_v(t), u_c(t)]^T$  ein. Da man die Steuerung in dieser Arbeit noch nicht betrachtet, wird  $\mathcal{U}(t) = [1 \ 1]^T$  gesetzt.

**$\Gamma_{\text{out}}$ :** Ausströmrand, durch den das Fluid samt gelöstem Stoff ohne Hinderung ausströmen kann. Diese Bedingung wird im Programm durch eine spezielle Ausströmbedingung realisiert. Genauere Beschreibungen sind im Abschnitt 4 zu finden.

**$\Gamma_{\text{wall}}$ :** Gebietsbegrenzung, an welcher Haftbedingung für die Strömung gilt. Dies ist sinnvoll, da aufgrund von Reibung die Geschwindigkeit unmittelbar am Rand Null ist. Das dadurch auftretende Phänomen der Grenzschichten wird in Abschnitt 3.4 näher beschrieben. Außerdem wird Undurchlässigkeit für den gelösten Stoff gefordert, was durch eine Neumann-Randbedingung realisiert wird. Dadurch ergibt sich

$$\begin{aligned} \mathbf{v}(t, \mathbf{x}) &= 0, \\ \partial_{\mathbf{n}} c(t, \mathbf{x}) &= 0, \end{aligned} \quad \text{auf } \Gamma_{\text{wall}}. \quad (25)$$

Konvektion sowie Diffusion durch den Rand  $\Gamma_{\text{wall}}$  kann daher ausgeschlossen werden.

$\Gamma_r$ : Strömungshindernis und Reaktionsrand. Für die Geschwindigkeit gelten äquivalent die Dirichletbedingungen (25) wie bei  $\Gamma_{\text{wall}}$ . Die Reaktion, die hier stattfindet, wird als sehr schnell im Vergleich zur Geschwindigkeit betrachtet. Sobald der gelöste Stoff den Rand erreicht, soll dieser durch die Reaktion vollständig aufgebraucht werden. Dies verwirklicht das Modell dadurch, dass ebenfalls homogene Dirichlet-Randbedingungen für die Konzentration gesetzt werden. Es ergibt sich

$$\begin{aligned} \mathbf{v}(t, \mathbf{x}) &= 0, \\ c(t, \mathbf{x}) &= 0, \end{aligned} \quad \text{auf } \Gamma_r. \quad (26)$$

Um das Ziel der Feedbacksteuerung zu erreichen, kann die Menge des verbrauchten Stoffes berechnet werden. Dies wird im Abschnitt 3.5 beschrieben.

An den Rändern mit Dirichlet-Randbedingungen sowie im Inneren der Strömung können, wie oben erwähnt, Grenzschichten auftreten. Mehr dazu im folgenden Abschnitt.

### 3.4 Grenzschichten

Bei den Modellen NSE und DCE handelt es sich um parabolische PDEs, da der Faktor  $\varepsilon \in \left\{ \frac{1}{\text{Re}}, \frac{1}{\text{Re Sc}} \right\}$  vor dem Term höchster Ordnung größer als Null ist. Für  $\varepsilon = 0$  wäre dies eine hyperbolische PDE. Da jedoch  $0 < \varepsilon \ll 1$  betrachtet wird, erhält man eine *im Wesentlichen hyperbolische PDE*, wodurch sich der Charakter eines singular gestörten Problems ausprägt. Dies führt zur Grenzschichtbildung. Das Gebiet wird dadurch, im laminaren Fall, in Außenströmung und Grenzschicht unterteilt. Im Falle einer turbulenten Strömung müsste sogar eine komplizierte Mehrschichtenstruktur modelliert werden. Anschaulich betrachtet, muss „in der Grenzschicht der Übergang von der physikalisch korrekten Randbedingung an der Wand (Haftbedingung) auf die Randbedingung erfolgen, die von der Außenströmung eingehalten wird, damit beide Teilgebiete anschließend zu einer einheitlichen Lösung zusammengesetzt werden können.“[STM] Am Beispiel der Geschwindigkeit kann man dies für Grenzschichten am Rand dadurch erklären, dass die Teilchen in unmittelbarer Umgebung zum Rand durch die Reibung in Ruhe sind. Der Einfluss nimmt mit dem Abstand zum Rand, für  $\varepsilon \ll 1$  annähernd exponentiell, ab. Da der Einfluss theoretisch betrachtet nie Null wird, spricht man solange von dem Bereich der Grenzschicht, bis sich die Geschwindigkeit zu 99% der Umgebungsgeschwindigkeit wieder angepasst hat. Im Bereich der Außenströmung wird der Einfluss der Reibung im Modell vernachlässigt. „Die Grenzschichttheorie ist nicht auf Wandgrenzschichten begrenzt, sondern gilt z.B. auch für Trennungsschichten zwischen zwei angrenzenden Strömungen.(...) Allen Grenzschichtströmungen gemeinsam ist die asymptotisch kleine Querabmessung der Grenzschicht sowie das Auftreten von Reibungseffekten anders als in der umgebenden Außenströmung.“[STM] Schematisch ist dies in Abbildung 4 dargestellt.

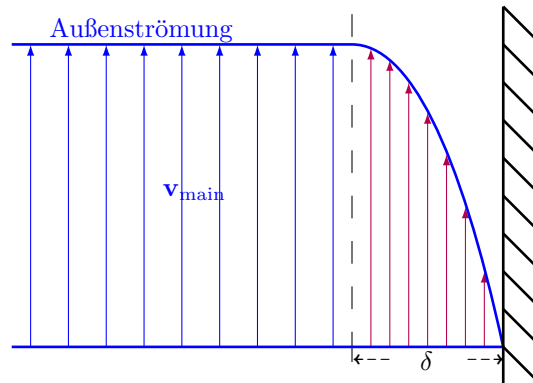


Abbildung 4: Schematische Darstellung der Grenzschicht

Die Grenzschichtdicke  $\delta$  hängt zum einen von der Richtung der Strömung zu den Bereichen, zum anderen auch vom Diffusionsparameter  $\varepsilon$  vor dem Term höchster Ordnung in der PDE, ab. Dabei gilt: für  $\varepsilon \rightarrow 0$  geht  $\delta \rightarrow 0$ . Die genaue Betrachtung solcher Grenzschichten ist sehr aufwändig. Für unser Modell sind sie in sofern wichtig, dass die Bereiche der Grenzschichten in der numerischen Berechnung fein genug durch das Rechengitter aufgelöst werden müssen. Einen Einblick in die dahinter stehende Theorie ist in [ROOS] zu finden. Die dort erwähnten, speziell für die Grenzschichten angepassten *Shishkin-Gitter*, werden in dieser Arbeit nicht verwendet. Zum Einen ist für die nichtlinearen NSE die Bestimmung der Gitter-Parameter nicht durch die gängige a-priori Theorie möglich. Aufgrund der am Rand verschwindenden Geschwindigkeit ist auch für die DCE die standard a-priori Theorie nicht anwendbar. Es wird sich daher auf moderate Werte für  $Re$  ( $\leq 100$ ) und  $Sc$  ( $\leq 100$ ) beschränkt, sodass mit heuristisch angepassten Gittern noch gute Ergebnisse erzielt werden. Dabei betrachtet man folgende Approximation für die Grenzschichtdicke  $\delta$

$$\delta_x^{(P)}$$

- beschreibt die Grenzschichtdicke in Bereichen parallel zur Strömungsrichtung
- für die Geschwindigkeit gilt:

$$\delta_v^{(P)} = \frac{1}{\sqrt{Re}}$$

- für die Konzentration gilt:

$$\delta_c^{(P)} = \frac{1}{\sqrt{Re Sc}}$$

$\delta_{\mathbf{x}}^{(S)}$ 

- beschreibt die Grenzschichtdicke in Bereichen senkrecht zur Strömungsrichtung
- für die Geschwindigkeit gilt:

$$\delta_v^{(S)} = \frac{1}{\text{Re}}$$

- für Konzentration gilt:

$$\delta_c^{(S)} = \frac{1}{\text{Re} \text{ Sc}}$$

Beispiele für konkrete Werte und die damit verbundenen Schwierigkeiten werden im Abschnitt 4.2.3 näher beschrieben.

### 3.5 Feedback-Steuerung

Wie in Abschnitt 1.1 beschrieben, ist das Ziel, die Reaktion auf dem Reaktionsrand  $\Gamma_r$  zu steuern. Die im Abschnitt 2.2 eingeführten Systemgrößen  $\mathcal{U}(t)$ ,  $\mathcal{X}(t)$ ,  $J(\mathcal{X}, \mathcal{U})$  auf unser Problem bezogen sind wie folgt:

$\mathcal{X}(t)$ : Zustandsgrößen, die das Systeminnere beschreiben. Dazu zählen die gesuchten Funktionen, die auf einer festen Diskretisierung betrachtet werden und dadurch nur noch von der Zeit  $t$  abhängen. Dadurch erhält man Vektoren verschiedener Länge, die abhängig von den Anzahlen der Gitterpunkte und der gewählten Ansatzfunktionen ist. Die Funktionen werden mit  $\mathbf{v}_D(t)$  für die Geschwindigkeit,  $\mathbf{p}_D(t)$  für den Druck und  $\mathbf{c}_D(t)$  für die Konzentration bezeichnet. So ergibt sich

$$\mathcal{X}(t) := [\mathbf{v}_D(t)^T, \mathbf{p}_D(t)^T, \mathbf{c}_D(t)^T]^T. \quad (27)$$

Als Grundlage unseres Steuerproblems wird eine DAE (*System von Differential Algebraische Gleichungen*) betrachtet.

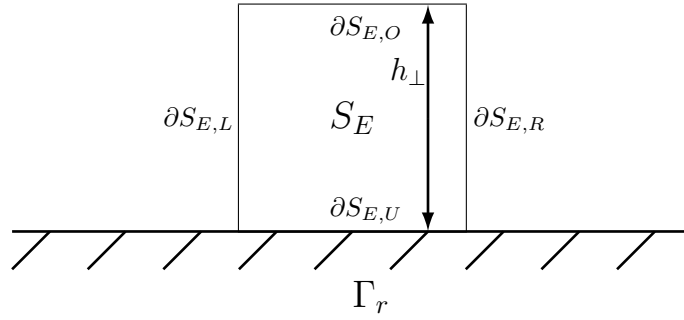
$\mathcal{U}(t)$ : Eingangsgrößen für das System sind die Funktionen  $u_v(t)$  und  $u_c(t)$ , die den zeitlichen Verlauf der Intensität der Dirichletrandbedingungen (24) auf dem Randteil  $\Gamma_{\text{in}}$  beschreiben. Dadurch ergibt sich

$$\mathcal{U}(t) := [u_v(t), u_c(t)]^T. \quad (28)$$

$J(\mathcal{X}, \mathcal{U})$ : Als Kostenfunktional, welches minimiert werden soll, wird das Integral

$$J(\mathcal{X}, \mathcal{U}) := \int_{t_0}^{t_{\text{end}}} \underbrace{\left[ \int_{\Gamma_r} \alpha \left| \frac{\partial c}{\partial \mathbf{n}} - q \right|^2 ds \right]}_{T_1} dt \quad (29)$$

genutzt. Der Term ( $T_1$ ) ist daher motiviert, dass die Menge an gelöstem Stoff, die am Rand  $\Gamma_r$  durch die Reaktion verbraucht wird, auf ein gewisses Niveau  $q$  gesteuert werden soll.

Abbildung 5: Kontrollvolumen  $S_E$  am Rand  $\Gamma_r$ 

Man betrachtet dazu in Analogie zu Abschnitt 3.2 ein Kontrollvolumen  $S_E$  am Reaktionsrand  $\Gamma_r$ . Der Rand des Kontrollvolumens ist in vier Bereiche unterteilt, wobei  $\partial S_{E,U} \subset \Gamma_r$ , wie in Abbildung 5 dargestellt. Ziel ist es, den Fluss durch  $\partial S_{E,U}$  zu bestimmen.

Aus (7) eingesetzt in (12) erhält man

$$\begin{aligned} \frac{\partial c}{\partial t} &= -\operatorname{div} \mathbf{J} \\ &= -\operatorname{div}(-D\nabla c + \mathbf{v} \cdot c) \\ &= \operatorname{div}(D\nabla c - \mathbf{v} \cdot c). \end{aligned}$$

Die Menge  $Q$  an gelöstem Stoff, die pro Zeiteinheit über das Volumen  $S_E$  aufgenommen wird, ergibt sich durch Integration.

$$\begin{aligned} Q &= \int_{S_E} \frac{\partial c}{\partial t} d\mathbf{V} = \int_{S_E} \operatorname{div}(D\nabla c - \mathbf{v} \cdot c) d\mathbf{V} \\ &= \underbrace{\int_{S_E} \operatorname{div}(D\nabla c) d\mathbf{V}}_{T_2} - \underbrace{\int_{S_E} \operatorname{div}(\mathbf{v} \cdot c) d\mathbf{V}}_{T_3}. \end{aligned} \quad (30)$$

Wird zunächst ( $T_3$ ) betrachtet mit

$$\int_{S_E} \operatorname{div}(\mathbf{v} \cdot c) d\mathbf{V} = \int_{S_E} c \cdot \operatorname{div} \mathbf{v} d\mathbf{V} + \int_{S_E} \mathbf{v} \cdot \nabla c d\mathbf{V},$$

so ergibt sich mit partieller Integration für den 2. Summanden

$$\begin{aligned} &= \int_{S_E} c \cdot \operatorname{div} \mathbf{v} d\mathbf{V} + \int_{\partial S_E} c \cdot \mathbf{v} \cdot \mathbf{n} ds - \int_{S_E} c \cdot \operatorname{div} \mathbf{v} d\mathbf{V} \\ &= \int_{\partial S_E} c \cdot \mathbf{v} \cdot \mathbf{n} ds. \end{aligned} \quad (31)$$

Für ( $T_2$ ) erhält man

$$\begin{aligned} \int_{S_E} \operatorname{div}(D\nabla c) d\mathbf{V} &\stackrel{(\text{D const.})}{=} D \int_{S_E} \operatorname{div} \nabla c d\mathbf{V} \\ &\stackrel{(\text{Gauß})}{=} D \int_{\partial S_E} \nabla c \cdot \mathbf{n} ds = D \int_{\partial S_E} \frac{\partial c}{\partial \mathbf{n}} ds. \end{aligned} \quad (32)$$



Zusammen ergibt sich für (30)

$$Q = \int_{S_E} \operatorname{div}(D\nabla c - \mathbf{v} \cdot c) \, d\mathbf{V} = D \int_{\partial S_E} \frac{\partial c}{\partial \mathbf{n}} \, ds + \int_{\partial S_E} c \cdot \mathbf{v} \cdot \mathbf{n} \, ds. \quad (33)$$

Die Menge an gelöstem Stoff  $Q$ , die in das Volumen  $S_E$  aufgenommen wird, ergibt sich durch den Fluss über den Rand  $\partial S_E$ , als Summe der Integrale über die Teilränder,

$$Q = \sum_{i \in \{O, U, R, L\}} \int_{\partial S_{E,i}} D \frac{\partial c}{\partial \mathbf{n}_i} + c \cdot \mathbf{v} \cdot \mathbf{n}_i \, ds, \quad (34)$$

wobei  $\mathbf{n}_i$  den Normalenvektor der jeweiligen Seite bezeichnet. Lässt man nun  $h_\perp \rightarrow 0$  gehen, geht das Volumen von  $S_E$  gegen Null, wodurch auch  $Q = 0$  wird. Zugleich verschwinden die Anteile von  $\partial S_{E,L}$  und  $\partial S_{E,R}$  und es ergibt sich

$$Q = 0 = \int_{\partial S_{E,O}} D \frac{\partial c}{\partial \mathbf{n}_O} + c \cdot \mathbf{v} \cdot \mathbf{n}_O \, ds + \int_{\partial S_{E,U}} D \frac{\partial c}{\partial \mathbf{n}_U} + c \cdot \mathbf{v} \cdot \mathbf{n}_U \, ds.$$

Außerdem fällt  $\partial S_{E,O}$  mit  $\partial S_{E,U}$  zusammen, mit  $\mathbf{n}_O = -\mathbf{n}_U$ , sodass gilt

$$\int_{\partial S_{E,O}} D \frac{\partial c}{\partial \mathbf{n}_O} + c \cdot \mathbf{v} \cdot \mathbf{n}_O \, ds = \int_{\partial S_{E,U}} D \frac{\partial c}{\partial \mathbf{n}_O} + \underbrace{c \cdot \mathbf{v} \cdot \mathbf{n}_O}_{T_4} \, ds =: Q_{\partial S}. \quad (35)$$

Das heißt, der Fluss durch  $\partial S_{E,O}$  ist gleich dem gesuchten Fluss  $Q_{\partial S}$  durch  $\partial S_{E,U} \subset \Gamma_r$ . Auf  $\Gamma_r$  gilt die RB (26), dadurch wird  $(T_4)$  zu Null.

Da man sich für den Fluss über den gesamten Rand  $\Gamma_r$  interessiert, summiert man  $Q_{\partial S}$  über alle Teilabschnitte von  $\Gamma_r$  und erhält

$$Q_{\Gamma_r} := \int_{\Gamma_r} D \cdot \frac{\partial c}{\partial \mathbf{n}} \, ds, \quad (36)$$

als Fluss  $Q_{\Gamma_r}$  des gelösten Stoffes durch den Rand  $\Gamma_r$ . Zieht man die Konstante  $D$  vor das Integral und skaliert die Gleichung damit, so ergibt sich

$$\frac{Q_{\Gamma_r}}{D} = \int_{\Gamma_r} \frac{\partial c}{\partial \mathbf{n}} \, ds =: \text{outflow}. \quad (37)$$

Die so definierte Funktion *outflow* wird in Abschnitt 4.2.3 verwendet, um die Menge an gelöstem Stoff, die über den Rand  $\Gamma_r$  aufgenommen wird, zu messen.

Wird nun wieder die Gleichung (36) genutzt und über dem Intervall  $[t_0, t_{end}]$  integriert, so folgt

$$\int_{t_0}^{t_{end}} \left[ \int_{\Gamma_r} D \cdot \frac{\partial c}{\partial \mathbf{n}} \, ds \right] dt, \quad (38)$$

als Menge des gelösten Stoffes, der in der in der Zeit  $t := t_{end} - t_0$  über den Rand  $\Gamma_r$  aufgenommen wird.

Zur Regelung wird diese Menge auf ein gewisses skaliertes Niveau  $\tilde{q} := D \cdot q$  gebracht. Dazu wird im Integrand die Differenz mit  $\tilde{q}$  gebildet und das Normquadrat dieser betrachtet. Daraus folgt

$$\begin{aligned}
 & \int_{t_0}^{t_{end}} \left[ \int_{\Gamma_r} \left| D \cdot \frac{\partial c}{\partial \mathbf{n}} - \tilde{q} \right|^2 ds \right] dt \\
 &= \int_{t_0}^{t_{end}} \left[ \int_{\Gamma_r} \left| D \cdot \frac{\partial c}{\partial \mathbf{n}} - D \cdot q \right|^2 ds \right] dt \\
 &= \int_{t_0}^{t_{end}} \left[ \int_{\Gamma_r} D^2 \cdot \left| \frac{\partial c}{\partial \mathbf{n}} - q \right|^2 ds \right] dt. \tag{39}
 \end{aligned}$$

Wird  $D^2 =: \alpha$  gesetzt, ergibt sich aus (39) die Gleichung  $(T_1)$  in (29)

$$\int_{t_0}^{t_{end}} \left[ \int_{\Gamma_r} \alpha \left| \frac{\partial c}{\partial \mathbf{n}} - q \right|^2 ds \right] dt.$$

Zur Implementierung im Programm sei auf Abschnitt (4.3) verwiesen.

Zur Regularisierung des Kostenfunctionals in Bezug auf technische Zusammenhänge kann man durch die Funktion  $S(t)$  zusätzlich noch die Kosten, die die Regelung verursacht, gewichten und man erhält

$$\int_{t_0}^{t_{end}} \left[ \int_{\Gamma_r} \alpha \left| \frac{\partial c}{\partial \mathbf{n}} - q \right|^2 ds \right] + S(t) dt.$$

Hierfür sind verschiedene Ansätze möglich, wobei in dieser Arbeit nicht näher darauf eingegangen wird.

## 4 Numerische Simulation

Wie in Kapitel 3 beschrieben, ist die analytische Lösung von PDEs nur für einfache Geometrien möglich. Um allgemeinere Probleme auf komplizierteren Geometrien zu behandeln, bedient man sich der Mittel der numerischen Mathematik. Die PDEs werden dabei näherungsweise durch numerische Verfahren gelöst. Eine Möglichkeit besteht darin, dass man die Gebiete durch Triangulationen diskretisiert, wodurch die Definitionsbereiche der Operatoren in einen linearen Unterraum mit endlicher Dimension transformiert werden. Diese Unterräume können durch verschiedene Ansatzfunktionen aufgespannt werden. Beispiele hierfür sind Finite-Differenzen, Finite-Elemente-, oder Finite-Volumen-Ansätze.

Die erhaltenen Lösungen können visualisiert oder durch *Postprocessing* weiterverarbeitet werden. Zur Visualisierung stehen verschiedene Verfahren und Programme zur Verfügung. Durch die grafische Auswertung der Daten können die Versuchsabläufe in Simulationen anschaulich dargestellt werden. Dies wird in Kapitel 4.2 am Beispiel demonstriert. Im *Postprocessing* werden die Daten weiterverarbeitet, um zum Beispiel Zielfunktionale oder abgeleitete Größen zu berechnen.

Zur numerischen Lösung von PDEs sind verschiedene kommerzielle und freie Softwarepakete verfügbar. Die freien Softwarepakete sind oft auf ein spezielles Problem zugeschnitten und bedürfen einiger Einarbeitungs- und Anpassungszeit. Sie sind dadurch aber auch auf die eigenen Bedürfnisse sehr gut anpassbar. Kommerzielle Software deckt meist größere Aufgabenbereiche ab, ist dadurch aber auch nur schwer erweiterbar.

Diese Arbeit beschäftigt sich ausschließlich mit der Software NAVIER, die nun vorgestellt wird.

### 4.1 Umsetzung des Modells in NAVIER

Zur numerischen Berechnung von inkompressiblen Strömungsproblemen entwickelte Prof. E. Bänsch von der FAU Erlangen das Programm NAVIER, dessen Implementierung u.a. in [BAEN] beschrieben ist. Es handelt sich hierbei um ein auf der Finite-Elemente-Methode (FEM) im zweidimensionalen Fall basierendes Programm. Die erste Version wurde 1991 in der Programmiersprache FORTRAN77 geschrieben. Seither wurde NAVIER ständig weiterentwickelt. Die aktuelle Version ist in FORTRAN90 implementiert. Im Rahmen eines Kooperationsvertrages steht uns NAVIER zur Nutzung und Modifikation zur Verfügung. Auf die mathematischen Hintergründe und die Implementierung unseres Problems wird in den folgenden Teilabschnitten eingegangen.

### 4.1.1 Mathematische Grundlagen für NAVIER

In diesem Kapitel wird ein kurzer Einblick in die mathematischen Hintergründe von NAVIER gegeben. Grundlage dieser Betrachtungen liefert Kapitel 4 aus [HOEH]. Hier werden die Lösungsansätze für den Fall der Navier-Stokes Gleichung mit Dirichlet-Randbedingung hergeleitet.

Die NSE ist ein nichtlineares, parabolisches System von Differentialgleichungen. Man betrachtet also eine Anfangsrandwertaufgabe, dies bedeutet, dass zur Lösung nicht nur Randbedingungen, sondern auch Anfangsbedingungen benötigt werden.

Die zeitliche Diskretisierung erfolgt durch das *Zwischenschritt- $\theta$ -Verfahren* in der Variante des Operatorsplittings. Hierbei wird jedes Zeitintervall in drei Teilschritte unterteilt, wodurch die Inkompressibilität und die Nichtlinearität entkoppelt wird. Schematisch ist dies in Abbildung 6 dargestellt.

Für  $\theta := 1 - \frac{1}{2}\sqrt{2}$  und  $\theta' := 1 - 2\theta$  wurde in [HOEH] gezeigt dass das Verfahren *stark A-stabil* und von *2. Ordnung* ist. Diese Parameter werden in NAVIER verwendet.

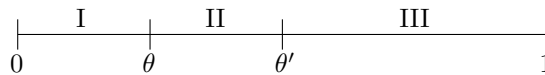


Abbildung 6: Intervallteilung für Zwischenschritt- $\theta$ -Verfahren

Im ersten und dritten Teilschritt wird ein lineares Sattelpunktproblem gelöst, bei dem die Nichtlinearität nur in die rechte Seite eingeht und somit wie in einem expliziten Zeitschrittverfahren behandelt wird. Im Teilschritt II wird die Inkompressibilitätsbedingung außer Acht gelassen und es wird ein nichtlineares Problem gelöst.

Im Ort wird die PDE durch *klassische Taylor-Hood-Elemente* diskretisiert. Dies sind „global stetige, stückweise quadratische Finite Elemente für die Geschwindigkeit und global stetige, stückweise lineare Finite Elemente für den Druck“ [HOEH].

Das Sattelpunktproblem aus Teilschritt I und III wird durch eine *Schurkomplement-Methode* auf ein selbstadjungiertes, positiv definites lineares Problem reduziert. Dieses wird mit einem *Verfahren der konjugierten Gradienten* (CG) gelöst. Das nichtlineare Problem im mittleren Teilschritt wird durch ein Iterationsverfahren gelöst. Dabei wird in jedem Teilschritt ein lineares Problem mittels Krylovraumverfahren GMRES (Generalized minimal residual method) gelöst. Details zu den numerischen Verfahren für lineare Gleichungssysteme sind in [NLGS] beschrieben.

In der aktuellen Version von NAVIER sind noch alternative Lösungsverfahren implementiert. Die Wahl dieser ist problemabhängig und wird für unseren Sachverhalt im nächsten Abschnitt erläutert.

Bei stark dominanter Konvektion im Vergleich zur Diffusion, d.h. der Faktor  $\varepsilon$  vor dem Term höchster Ordnung in der PDE ist viel kleiner als Eins, kann es zu numerischen Oszillationen kommen, wenn das Gitter die wechselnd steilen Abschnitte nicht fein genug auflösen kann. Ein Verfahren diese zu verringern ist die *Streamline-Diffusion-Method*. Dabei werden zusätzliche Ansatzfunktionen, deren Träger komplett im Inneren des Intervalls liegen, gewählt, die die Qualität der Approximation in den Bereichen der Oszillation verbessern. In [SLD]

wird dies in Abschnitt 3.3.2 hergeleitet. Die veränderten Ansatzfunktionen werden dabei um einen mit dem Strömungsfeld  $\mathbf{v}$  gewichteten Anteil des Gradienten der ursprünglichen Ansatzfunktion ergänzt. Dadurch ist der Einfluss nur in Stromlinienrichtung markant. Die Wahl des Gewichtungskoeffizienten ist nicht allgemein gültig, wird aber in [SLD] diskutiert. Er sollte sich in einem Bereich der maximalen Geschwindigkeit bewegen. Die Definition des Koeffizienten in NAVIER ist in Abschnitt 4.2.2 erläutert. Intern wird der Koeffizient noch mit der Gitterschrittweite skaliert.

#### 4.1.2 Problemimplementation

Die Grundstruktur des Programms ist in zwei Blöcke unterteilt. Der erste Block bildet das Programm an sich, welches vom konkret zu simulierenden Modell relativ unabhängig ist. Hier sind die Lösungsstrukturen für NSE und DCE wie auch Temperaturverteilungen implementiert. Dafür sind verschiedene Gleichungssystemlöser und Vorkonditionierer implementiert bzw. aus externen Bibliotheken integriert. Anpassungen dieses Programmteils waren zur Integration der Feedbacksteuerung jedoch auch notwendig, diese sind im Abschnitt 4.3 vorgestellt.

Am prinzipiellen Ablauf der Berechnung wurde jedoch nichts verändert. Dieser Teil wird zu einer dynamischen Bibliothek **libnavier.so** kompiliert und in die weitere Berechnung eingebunden. Der zweite Block bildet die konkrete Problemdefinition. Als Input dienen die Dateien, die in Tabelle 1 kurz erklärt sind. Zwei dieser Dateien werden nun noch etwas näher betrachtet.

<b>Dateiname</b>	<b>Kurzbeschreibung</b>
navier.dat	Beinhaltet die Konfigurationsparameter für NAVIER (siehe Tabelle 2)
physics.dat	Beinhaltet physikalische Konstanten zur Entdimensionierung
tolerance.dat	Beinhaltet Genauigkeitstoleranzen und maximale Schrittweiten für die verschiedenen Problem-Solver
triang.dat	Triangulationsdatei (siehe Tabelle 3)
problem.f90	Problemdefinition mit Implementation von Rand- und Anfangsbedingungen, sowie Integration der Feedbacksteuerung
problem.inc	Definition einiger Abkürzungen und Bezeichnungen

Tabelle 1: Problem-Input-Dateien

---

```

1 F      #Start values (new/restart)
2 dce.dat #Macro triangulation
3 '<dir>' #Output directory
4 20     #maximal Refinement level
5 1     #update (1=full, 2=partial, 3=full with smoothing)
6 1d-1  #Time step
7 1d2   #End time
8 1     #graphics output step
9 0     #checkpoint outstep step
10 1    #time discretization (0=Glowinski,1=BDF2,2=fixpoint,3=stationary)
11 0    #grid update (0=every substep, 1=every step)
12 0    #element order for temperature
13 1    #element order for concentration
14 T    #solve Navier-Stokes equation
15 F    #solve only a Stokes equation
16 F    #solve temperature equation
17 T    #solve concentration equation
18 T    #use convective term for temperature and concentration
19 1    #preconditioner for heat eq. (0=no,1=diag,2=ILU,3=ICC,4=ICCM,5=SSOR)
20 2    #preconditioner for conc eq. (as above)
21 1    #preconditioner for Quasi-Stokes eq. (as above)
22 2    #preconditioner for Burgers eq. (as above)
23 3    #preconditioner for pressure-space Mass-matrix problems
24 3    #preconditioner for pressure-space Neumann/Dirichlet-problems
25 2    #Solver for nonsym. problems in T,c (0=GMRES,1=BICGSTAB,2=PARDISO)
26 2    #Solver for Burgers-Step (0=GMRES,1=BICGSTAB,2=PARDISO)
27 2    #treatment of U* (0=explicit,1=implicit,2=extrapolate)
28 0d0  #streamline diffusion coefficient for temperature
29 1d0  #streamline diffusion coefficient for concentration
30 25   #Krylov space dimension to restart for GMRES
31

```

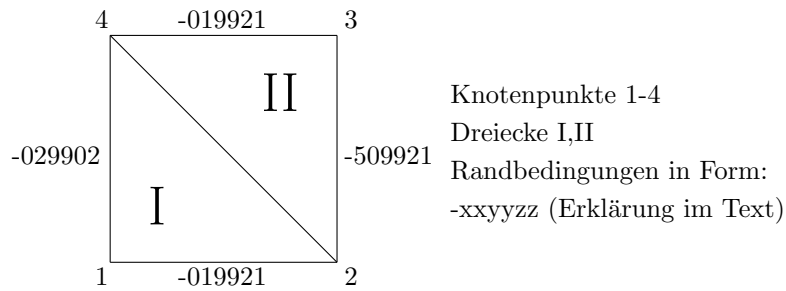
---

Tabelle 2: Konfigurationsdatei

**navier.dat** Die Hauptkonfiguration für die Berechnung erfolgt in dieser Datei. Ein Beispiel ist in Tabelle 2 zu sehen. Die meisten Zeilen sind hierbei selbst erklärend. Zu Anfang gibt es die Option *RESTART*, um ein neues Problem zu rechnen, bzw. ein bestehendes weiter zu rechnen (Zeile 1). Die maximal mögliche Verfeinerung wird in Zeile 4 beschrieben. Nachdem Endzeit und Zeitschrittweite festgelegt sind, folgen einige Einstellungen für Ausgabe und Gitterupdate. Mit der Einstellung von Zeile 10 kann man das Diskretisierungsverfahren für die Zeit auswählen. In Zeile 12, 13 wird die Ordnung der Temperatur- bzw. Konzentrationsansatzfunktionen bestimmt, dabei kann zwischen linearen und quadratischen gewählt werden. In den folgenden Zeilen legt man fest, welche Art von PDE gelöst werden soll. Zeile 18 bestimmt, ob der konvektive Anteil in Temperatur- oder Konzentrationsgleichung mit berücksichtigt werden soll. Die Zeilen 19-24 geben die Möglichkeit, *Vorkonditionierer* für die verschiedenen Probleme zu wählen. Zeile 25 bestimmt die Lösungsmethoden für den unsymmetrischen Anteil bei Temperatur- und Konzentrationsgleichung. In Zeile 26 wird die Methode zur Lösung des nichtlinearen Teilproblems gewählt. Auf den *Streamline-Diffusion-Coefficient* in Zeile 28/29, wurde am Ende von Abschnitt 4.1.1 eingegangen. Zeile 30 gibt die maximale Dimension des Krylovraumverfahrens GMRES vor. Eine leere Zeile schließt die Datei ab.

**triang.dat** Zur Diskretisierung im Ort wird eine Triangulation benötigt. Diese muss die Struktur wie in Tabelle 3, beispielhaft für eine triviale Anordnung gezeigt, aufweisen. Man betrachtet hierbei ein Einheitsquadrat, das durch zwei Dreiecke diskretisiert wird.

Die Randbedingungen sind dabei kodiert, werden durch ein '-' eingeleitet und haben die Form '-xxyzz'. Die Unterteilung in die drei Segmente unterscheidet dabei die Randbedingungen für Geschwindigkeit (x), Temperatur (y) und Konzentration (z).



triang.dat:			Erklärungen
2	4		# Dreiecke, #Knotenpunkte
1	2	4	Dreieck I aus Punkten 1,2,4
2	3	4	Dreieck II aus Punkten 2,3,4
2	-029902	-019921	I: gegenüberliegende Objekte der Punkte
-019921	1	-509921	II: gegenüberliegende Objekte der Punkte
1			I: lok. Punkt gegenüber der längsten Seite
2			II: lok. Punkt gegenüber der längsten Seite
0.0E+00	0.0E+00		Koordinaten von Punkt 1
0.1E+01	0.0E+00		Koordinaten von Punkt 2
0.1E+01	0.1E+01		Koordinaten von Punkt 3
0.0E+00	0.1E+01		Koordinaten von Punkt 4
			Leerzeile

Tabelle 3: Schematischer Aufbau der Triangulationsdatei

Die genauen Kodierungen für die Randbedingungen sind in der Datei **problem.inc** definiert. Es können jeweils zehn verschiedene Dirichlet- (01-10), Neumann- (21-30) und Robin-Randbedingungen (31-40) vorgegeben werden. Zusätzlich gibt es für die Geschwindigkeit noch einige Spezialfälle wie DO-NOTHING (40), oder NO-BOUNDARY (50). Falls keine Vorgaben vorhanden sind, wird (99) gesetzt. Die Definition der Randbedingungen erfolgt dann in der Datei **problem.f90**. Die in Tabelle 3 benutzten Randbedingungen und deren Bedeutung sind in Tabelle 4 erläutert.

Kodierung	Geschwindigkeit	Temperatur	Konzentration
-01 99 21	1. Dirichletrand	nicht definiert	1. Neumann-Rand
-02 99 02	2. Dirichletrand	nicht definiert	2. Dirichletrand
-50 99 21	NO-BOUNDARY	nicht definiert	1. Neumann-Rand

Tabelle 4: Kodierung der Randbedingungen

Die Dateien **problem.f90** und **problem.inc** werden nun zu der ausführbaren Datei **navier** kompiliert, wobei die dynamische Bibliothek **libnavier.so** mit eingebunden wird. Beim Ausführen von **navier** werden die restlichen '\*.dat'-Dateien aus Tabelle 1 eingelesen. Da diese beim Kompilieren noch keine Rolle spielen, können sie jederzeit modifiziert werden, ohne dass das Programm verändert werden muss. Eine Variante, die Datei **problem.f90** ebenfalls ohne Neukompilierung verändern zu können, ist ebenfalls vorgesehen, wurde aber für

diese Arbeit noch nicht genutzt.

Eine Beispielsimulation und deren Auswertung wird im nächsten Unterabschnitt beschrieben.

## 4.2 Beispielsimulation

Eine konkrete Implementierung und die Auswertung der resultierenden Simulation wird nun näher betrachtet. Sämtliche Berechnungen wurden dafür auf *LEIBNIZ*, einem Compute-Server des mathematischen Rechenzentrums der TU Chemnitz [MRZ], durchgeführt. *LEIBNIZ* besitzt einen Intel<sup>®</sup>Xeon<sup>®</sup>Quadcore Prozessor mit 4 Kernen, der *Hyper-Threading* [HPT] unterstützt und dadurch theoretisch 8 Kerne á 2.6 GHz für Rechnungen nutzen kann. Für numerische Rechnungen ist die Verwendung von *Hyper-Threading* jedoch nicht empfehlenswert. Der Cache des Prozessors ist mit 4 MB ausgerüstet, der Arbeitsspeicher umfasst 16 GB und als Betriebssystem läuft das 64-Bit *openSUSE Linux 11.1 (x86\_64)*.

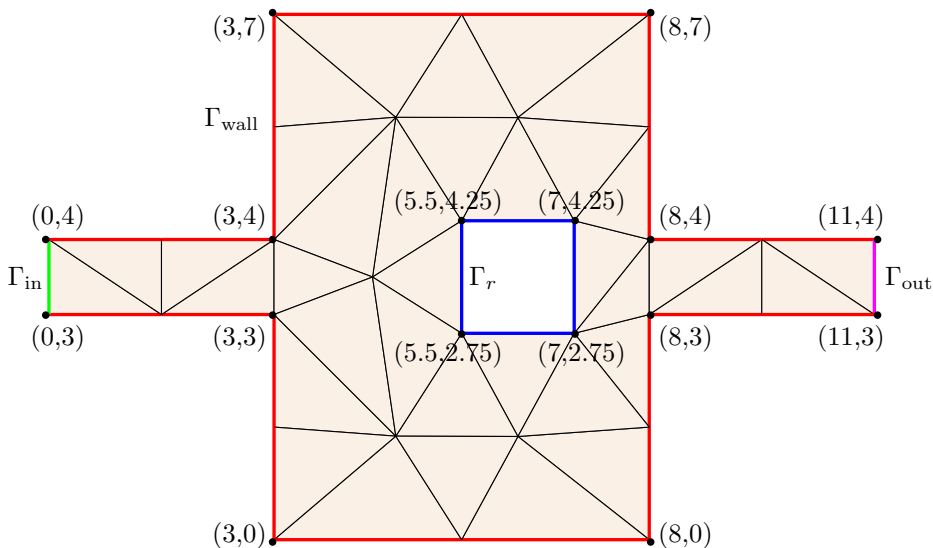


Abbildung 7: Ausgangstriangulation

### 4.2.1 Gebietstriangulation

Zur numerischen Lösung der PDEs wird, wie zu Beginn von Abschnitt 4 erwähnt, eine Triangulation des Gebietes benötigt. In unserem Beispiel geschieht dies durch Dreiecke. Die Triangulation kann auf verschiedene Weise erzeugt werden. Zum einen kann für einfache Gebiete die Grundstruktur von Hand definiert werden, wie in Tabelle 3 dargestellt. Zum Anderen können Netzgeneratoren, wie in der MATLAB<sup>®</sup>-Toolbox PDE enthalten, benutzt werden.

Die Genauigkeit der Berechnung hängt vom Gitter und dessen Verfeinerung ab, die vorher durch den Netzgenerator festgelegt werden kann. NAVIER selbst



verfügt aber auch über Funktionen zur Verfeinerung des Netzes, die global oder in gewissen Gebieten angewandt werden können. Der Vorteil dabei liegt darin, dass keine externen Programme genutzt werden müssen, um verschiedene Verfeinerungslevel einzustellen. Um diese Funktion für diese Arbeit bestmöglich zu nutzen, wurde die Ausgangstriangulation so grob wie möglich erstellt und ist in Abbildung 7 dargestellt. Die Verfeinerung in NAVIER erfolgt dabei in drei unterschiedlichen Bereichen. Neben einer globalen Verfeinerung, die auf dem ganze Gebiet  $\Omega$  wirkt, wird an lokalen Stellen noch zusätzlich verfeinert. Dazu zählen die einspringenden Ecken bei  $(3, 3)$  und  $(3, 4)$  sowie der Rand  $\Gamma_r$  des Reaktionsgebietes. Das Problem der Grenzschichten, wie in Abschnitt 3.4 beschrieben, bedingt ein höheres Verfeinerungslevel am Rand  $\Gamma_r$ . Ein höheres Level ergibt dabei mehr Freiheitsgrade in der Diskretisierung, was zu längeren Rechenzeiten führt. Hierfür muss ein Kompromiss zwischen Rechengenauigkeit und Aufwand gefunden werden. Die Wahl der Verfeinerungsgebiete sowie des Verfeinerungslevels wird in **problem.f90** definiert. Ergebnisse zu den verschiedenen Verfeinerungskombinationen werden im Abschnitt 4.2.3 verglichen.

<b>Time-Discret. (10)</b>	Glowinski	BDF2	Glowinski	BDF2
<b>Nonsym. Probl. (25)</b>	GMRES	GMRES	PARDISO	PARDISO
<b>Burger-Step (26)</b>	GMRES	GMRES	PARDISO	PARDISO
<b>CPU-Time (sec.)</b>	3645	1136	2323	644
<b>Real-Time (sec.)</b>	3646	1137	889	509
<b>Speed-Up</b>	1.0	1.0	2.6	1.3

Tabelle 5: Konfigurationsvergleich

#### 4.2.2 Konfigurationsdaten sowie Anfangs- und Randbedingungen

Die in Tabelle 2 dargestellte Konfigurationsdatei beschreibt die besten Einstellungen, die für dieses Problem gefunden wurden. Tabelle 5 vergleicht die Rechenzeiten verschiedener Kombinationen an einem einheitlichen Beispiel. CPU-Time beschreibt die Summe der Rechenzeiten der einzelnen Prozessoren. Real-Time hingegen ist die reale Rechenzeit, die zur Berechnung des Problems benötigt wird. CPU- und Real-Time unterscheiden sich dabei, wenn Algorithmen die Parallelstruktur der Rechenmaschine ausnutzen konnten. Dies spiegelt sich im Speed-Up Faktor, der das Verhältnis von CPU- zu Real-Time angibt, wider.

Auf eine ausführliche Erklärung aller auswählbaren Verfahren wird im Folgenden verzichtet.

Man erkennt eine deutliche Zeitersparnis bei der Wahl von PARDISO für die Punkte 25 und 26 in Tabelle 2. PARDISO ist ein robuster, speicheroptimierter, und leicht zu verwendender Hochleistungs-Löser für große dünnbesetzte symmetrische und unsymmetrische lineare Gleichungssysteme, der auf *shared-memory-multiprocessors* arbeiten kann. Nähere Informationen zu dem Paket sowie die Lizenzbedingungen sind unter [PARD] zu erhalten.

Als Anfangsbedingung für Geschwindigkeit und Konzentration wird

$$\mathbf{AB}: \quad \begin{aligned} \mathbf{v}(0, \mathbf{x}) &= 0 \\ c(0, \mathbf{x}) &= 0 \end{aligned} \quad \text{auf } \Omega \quad (40)$$

definiert. Das Fluid ist also in Ruhe und es befindet sich kein gelöster Stoff im Gebiet  $\Omega$ . Die Randbedingungen sind wie in Abschnitt 3.3 beschrieben.

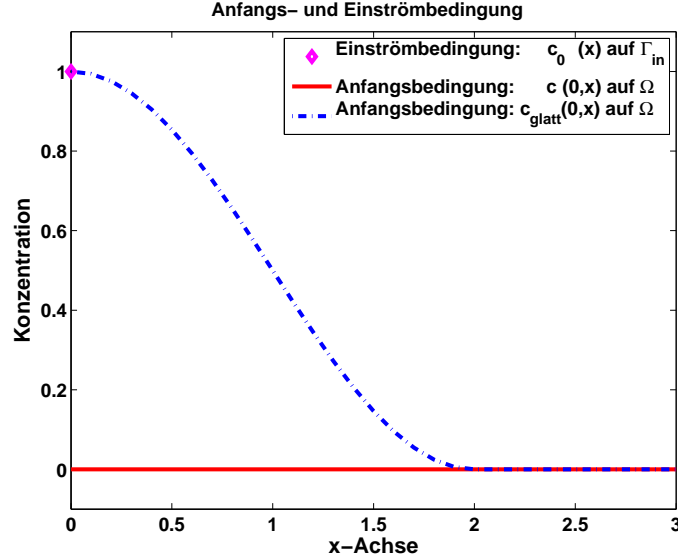


Abbildung 8: Anfangs- und Einströmbedingung für Konzentration

Die Einströmbedingungen bei  $\Gamma_{\text{in}}$  sind definiert mit

$$\left. \begin{aligned} \mathbf{v}_0(\mathbf{x}) &= \begin{bmatrix} -4 \cdot (x_2 - 3) \cdot (x_2 - 4) \\ 0 \end{bmatrix} \\ c_0(\mathbf{x}) &= 1 \end{aligned} \right\} \text{ auf } \Gamma_{\text{in}}. \quad (41)$$

Für die Geschwindigkeit wird eine parabelförmige Einströmbedingung, die konsistent mit den Randbedingungen auf  $\Gamma_{\text{wall}}$  ist und Eins als Maximum auf der Mitte von  $\Gamma_{\text{in}}$  annimmt, benutzt. Die Konzentration wird als konstant Eins gesetzt.

Die Bedingungen (40) und (41) für die Konzentration sind, wie in Abbildung 8 dargestellt, inkonsistent. Dies führte bei unseren Tests zu numerischen Oszillationen zu Beginn der Rechnung. Um dem vorzubeugen, wurde eine  $C^1(\Omega)$  Funktion  $c_{\text{glatt}}$  als Anfangsbedingung für die Konzentration eingesetzt, die einen glatten Übergang zwischen der Einströmbedingung  $c_0(\mathbf{x})$  und der geforderten Anfangsbedingung für  $\Omega$  realisiert.  $c(0, \mathbf{x})$  in (40) wird durch die Funktion

$$c_{\text{glatt}}(0, \mathbf{x}) = \begin{cases} \frac{\cos(\frac{\pi}{2} \cdot x_1) + 1}{2} & \forall 0 \leq x_1 \leq 2 \\ 0 & \forall x_1 > 2 \end{cases} \text{ auf } \Omega, \quad (42)$$

ersetzt, die ebenfalls in Abbildung 8 dargestellt ist.

Die Abbildungen 9a-9d zeigen den zeitlichen Verlauf der Konzentration als *LinePlot* zwischen den Punkten  $P_1(0, 3.5)$  und  $P_2(6, 3.5)$  im Vergleich für die Anfangsbedingungen  $c(0, \mathbf{x})$  bzw.  $c_{glatt}(0, \mathbf{x})$ .

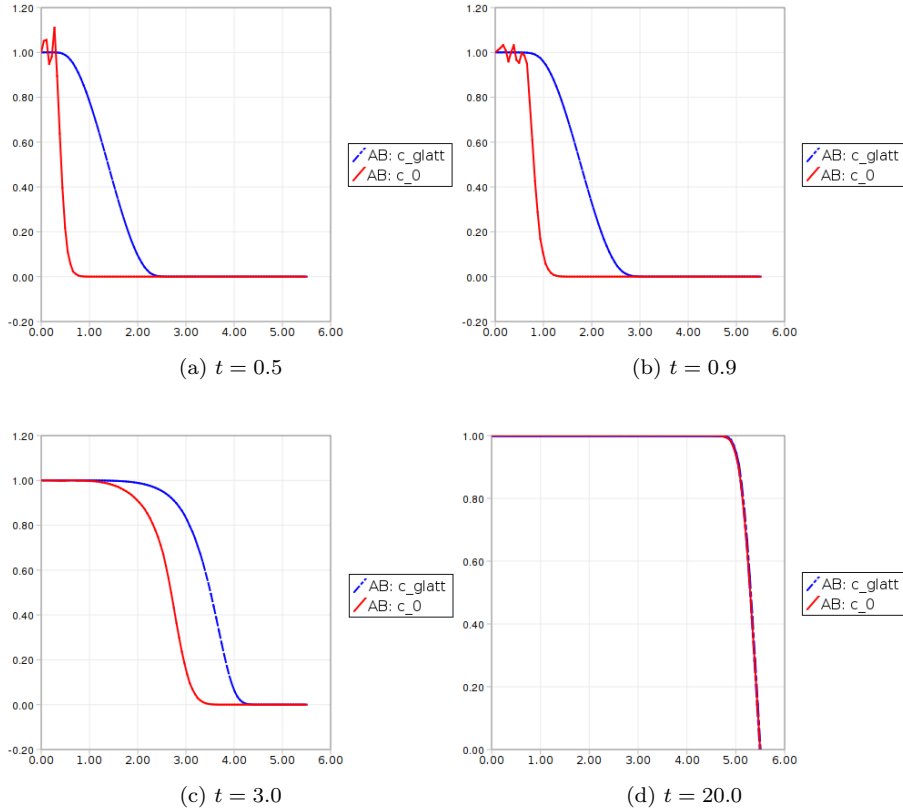


Abbildung 9: Zeitlicher Verlauf für verschiedene AB

Man erkennt dabei die Oszillationen zu Beginn der Rechnung. Bei  $t = 3.0$  sind diese schon fast verschwunden. Bei  $t = 20.0$  gleichen sich beide Verläufe in etwa. Da man sich bei der Regelung mit langen Rechenzeiträumen beschäftigt, macht die Auswahl der Anfangsbedingung wahrscheinlich keinen Unterschied. Im weiteren wurden alle Rechnungen mit  $c_{glatt}(0, \mathbf{x})$  durchgeführt, um von Beginn an die Oszillationen zu unterdrücken.

Zusammenfassend erhält man die **PDE**

$$\left. \begin{aligned}
 & \left. \begin{aligned}
 \frac{\partial \mathbf{v}(t, \mathbf{x})}{\partial t} - \frac{1}{\text{Re}} \Delta \mathbf{v}(t, \mathbf{x}) + \mathbf{v}(t, \mathbf{x}) \cdot \nabla \mathbf{v}(t, \mathbf{x}) + \nabla p(t, \mathbf{x}) = 0 \\
 \nabla \cdot \mathbf{v}(t, \mathbf{x}) = 0
 \end{aligned} \right\} \text{auf } \Omega, \\
 & \left. \begin{aligned}
 \frac{\partial c(t, \mathbf{x})}{\partial t} - \frac{1}{\text{Re Sc}} \Delta c(t, \mathbf{x}) + \mathbf{v}(t, \mathbf{x}) \cdot \nabla c(t, \mathbf{x}) = 0
 \end{aligned} \right\} \\
 & \text{mit den **Randbedingungen**} \\
 & \left. \begin{aligned}
 \mathbf{v}(t, \mathbf{x}) = u_v(t) \cdot \begin{bmatrix} -4 \cdot (x_2 - 3) \cdot (x_2 - 4) \\ 0 \end{bmatrix} \\
 c(t, \mathbf{x}) = u_c(t) \cdot 1
 \end{aligned} \right\} \text{auf } \Gamma_{\text{in}}, \\
 & \left. \begin{aligned}
 \mathbf{v}(t, \mathbf{x}) = 0 \\
 \partial_{\mathbf{n}} c(t, \mathbf{x}) = 0
 \end{aligned} \right\} \text{auf } \Gamma_{\text{wall}}, \\
 & \left. \begin{aligned}
 \mathbf{v}(t, \mathbf{x}) = 0 \\
 c(t, \mathbf{x}) = 0
 \end{aligned} \right\} \text{auf } \Gamma_r, \\
 & \text{und den **Anfangsbedingungen**} \\
 & \left. \begin{aligned}
 \mathbf{v}(0, \mathbf{x}) = 0 \\
 c_{\text{glatt}}(0, \mathbf{x}) = \begin{cases} \frac{\cos(\frac{\pi}{2} \cdot x_1) + 1}{2} & \forall 0 \leq x_1 \leq 2 \\ 0 & \forall x > 2 \end{cases}
 \end{aligned} \right\} \text{auf } \Omega.
 \end{aligned} \right\} (\star)$$

Das System  $(\star)$  wurde für verschiedene Konfigurationen und Verfeinerungslevel gelöst. Die Auswertung folgt im nächsten Abschnitt.

Bezeichnung	$\mathbf{V}_{\Omega}$	$\mathbf{V}_{\Gamma_{\text{E}}}$	$\mathbf{V}_{\Gamma_r}$	$N_v$	$N_p$	$N_T$	$t_{\text{CPU}}$
level-0	7	8	11	18371	4960	8991	626 sec
level-1	8	9	12	35800	9135	17530	1325 sec
level-2	9	10	13	71779	18140	35499	2930 sec
level-3	10	11	14	141820	35828	70164	6756 sec
level-4	11	12	15	284163	71432	141299	16509 sec

Tabelle 6: Gitterparameter und Rechenzeit

### 4.2.3 Ergebnisauswertung

Die verschiedenen Verfeinerungslevel, für die das System  $(\star)$  gelöst wurde, sind in Tabelle 6 aufgelistet.  $\mathbf{V}_{\Omega}$  ist das Level der globalen Verfeinerung. Mit  $\mathbf{V}_{\Gamma_{\text{E}}}$  wird das Level an den einspringenden Ecken bei  $(3, 3)$  und  $(3, 4)$  beschrieben.  $\mathbf{V}_{\Gamma_r}$  steht für das Level der Verfeinerung am Reaktionsrand  $\Gamma_r$ . Die Werte  $N_i$  stehen für die Anzahl der Geschwindigkeitsknoten ( $v$ ), Druckknoten ( $p$ ) sowie der Dreiecke ( $T$ ) in der Triangulation. Die benötigte CPU-Rechenzeit  $t_{\text{CPU}}$  ist in der letzten Spalte dargestellt. level-0 ist die Ausgangsverfeinerung, für die das Modell ordentliche Lösungen im erwarteten Bereich lieferte, da die Grenzschichten ausreichend fein aufgelöst werden. Dabei wurden die Parameter  $\text{Re} = 10$  und  $\text{Sc} = 10$ , sowie  $t_{\text{end}} = 100$  gewählt. Für größere Werte der Parameter und die daraus folgenden dünneren Grenzschichten wäre die globale Verfeinerung, die

nötig ist, um die inneren Grenzschichten aufzulösen, zu aufwändig geworden. Deshalb wurden nach anfänglichen Tests alle weiteren Rechnungen mit diesen Parametern durchgeführt. Eine weitere Verfeinerung auf level-5 war nicht mehr möglich.

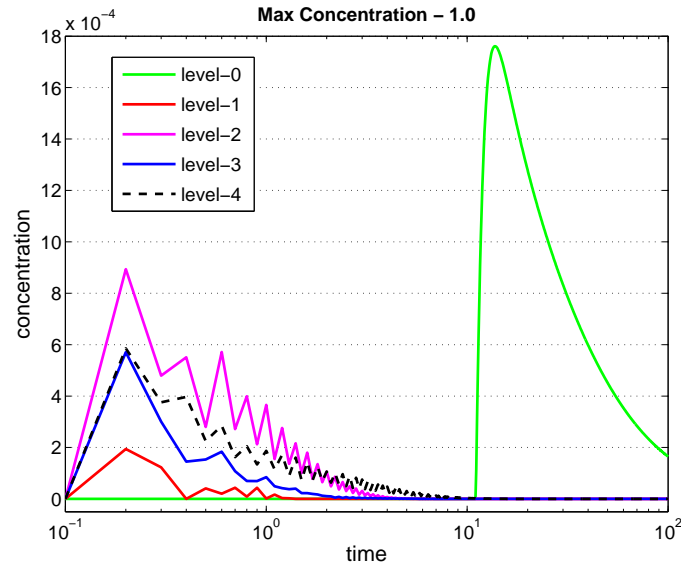


Abbildung 10: Maximale Konzentration

Um nun die Konvergenz des Verfahrens zu untersuchen, wurde das Netz ausgehend von level-0 zweimal uniform verfeinert (level-0 → level-2 & level-2 → level-4). In NAVIER wird die *Bänsch-Grün-Verfeinerung* angewandt, bei der ein Dreieck in zwei Dreiecke zerlegt wird, indem auf der Mitte der längsten Seite ein neuer Knotenpunkt definiert wird. Dadurch benötigt man zwei globale *Bänsch-Grün-Verfeinerungen* für eine uniforme Verfeinerung, bei der ein Dreieck in 4 Dreiecke zerlegt wird, wobei die Seiten jeweils halbiert werden. Hierbei entstehen die Zwischenschritte level-1 und level-3.

Bei jeder Rechnung wird ein *Log-File* herausgeschrieben. Dieses beinhaltet die benutzten Parameter, die Netzgrößen, sowie für jeden Zeitschritt die maximale Konzentration, die minimale Konzentration, sowie die Menge an Stoff, die über den Rand  $\Gamma_r$  aufgenommen wurde. Diese Menge wird, wie in Gleichung (37) eingeführt, mit *outflow* bezeichnet. Die Werte werden anschließend durch ein MATLAB-Skript aufgearbeitet. Die maximale Konzentration wird in Abbildung 10 dargestellt. Diese sollte den Wert Eins nicht überschreiten, deshalb ist in der Abbildung  $|c_{max} - 1.0|$  dargestellt. Die Werte schachteln sich dabei in den verschiedenen Level ineinander.

In Abbildung 11 ist die minimale Konzentration dargestellt. Dabei sieht man, dass ab Level-3 die Abweichung von Null fast verschwindet. Beide Grafiken sind logarithmisch in der Zeit aufgetragen, da sich nach einer gewissen Zeit beide Werte Null nähern, bzw. sogar Null als Wert annehmen.

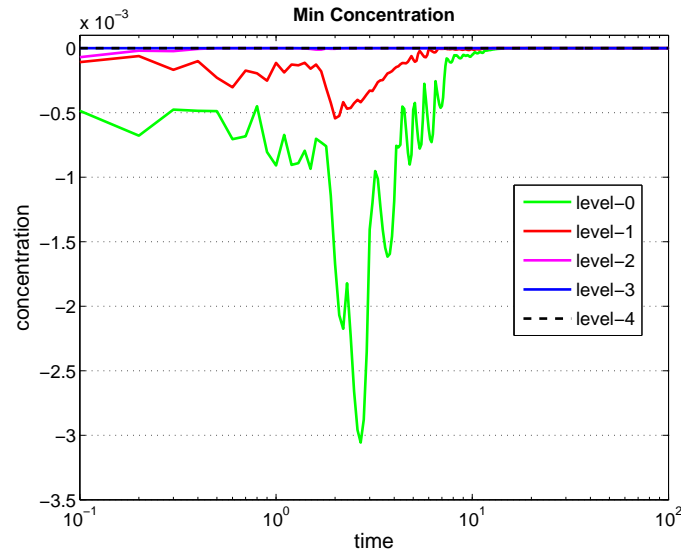


Abbildung 11: Minimale Konzentration

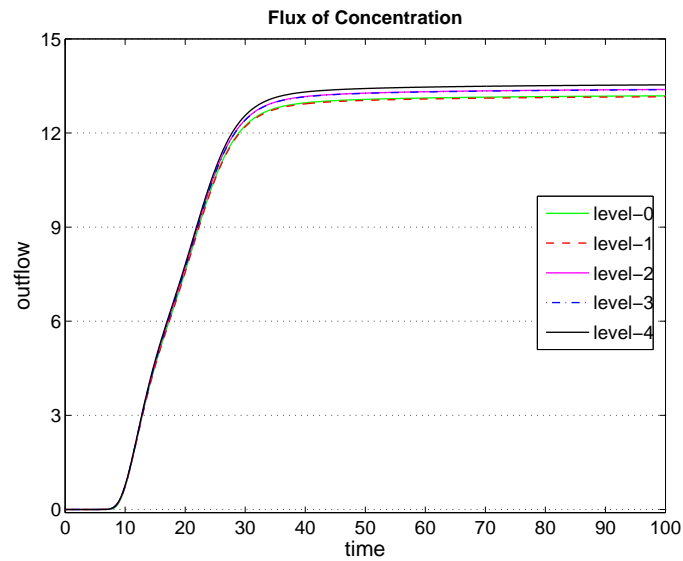


Abbildung 12: Konzentration, die pro Zeiteinheit aufgebraucht wird

In Abbildung 12 ist der Wert von *outflow* über der Zeit abgetragen. Man sieht, dass nach der Zeit  $t > 6$  gelöster Stoff den Reaktionsrand  $\Gamma_r$  erreicht und durch die Reaktion aufgebraucht wird. Der Wert steigt bis zu einem gewissen Niveau und bleibt dann annähernd konstant. Zwei Bereiche werden dabei näher betrachtet. Der Zeitpunkt, zu dem gelöster Stoff den Rand  $\Gamma_r$  erreicht, ist in Abbildung 13 dargestellt. Bei Level-0 fällt der Wert noch etwas unter Null, ehe er ansteigt. Dieser Fehler ist ab Level-1 jedoch behoben.

Abbildung 14 zeigt einen Zoom auf den oberen Bogen, wo sich der Graph

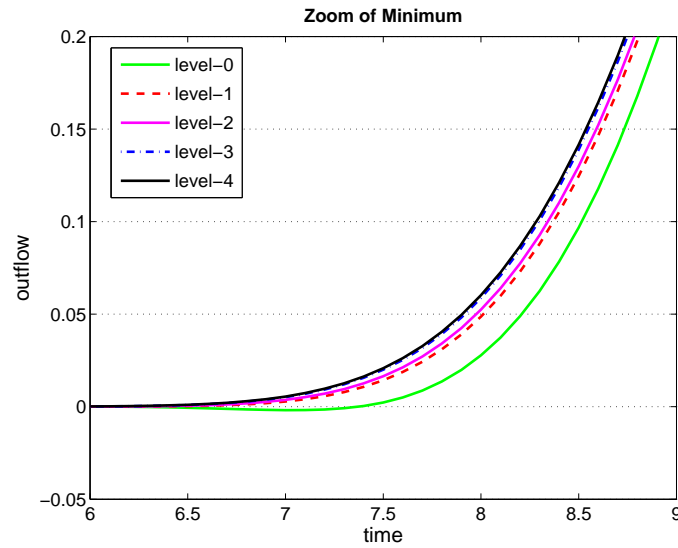


Abbildung 13: Zoom des Minimums

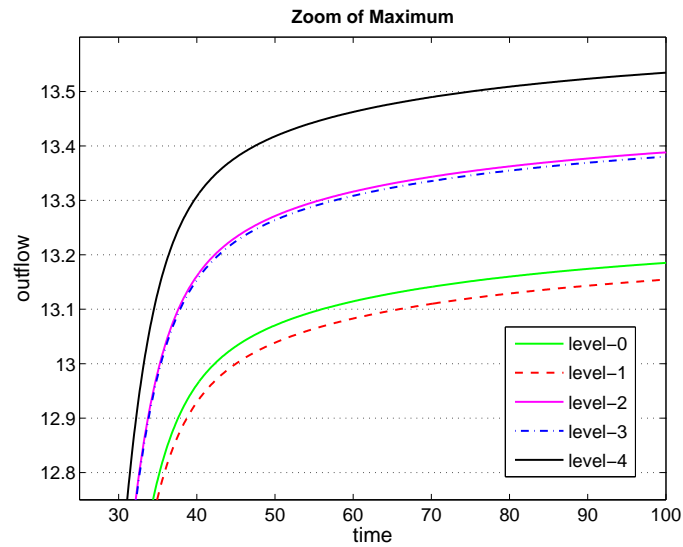


Abbildung 14: Zoom des Maximums

seiner oberen Asymptote annähert.

Bei beiden Abbildungen erkennt man, dass sich bei einer uniformen Verfeinerung die Funktion markant sprunghaft erhöht.

Die Abstände wurden für die Zeit  $t = 100$  in Tabelle 7 dargestellt. Die Sprunghöhen werden dabei mit höherem Level kleiner und bewegen sich im einstelligen Prozentbereich. Bei den Zwischenschritten, also einer 1-fachen globalen Verfeinerung, werden die Funktionsverläufe geringfügig kleiner. Der Abstand wird mit zunehmendem Verfeinerungslevel auch hier kleiner. Die Veränderungen

bewegen sich dabei in Bereichen von Zehntel- bis Hundertstelprozenten.

	Level	outflow bei $t = 100$	
-0.2%	0	13.1853	+1.54%
	1	13.1546	
-0.0575%	2	13.3882	+1.11%
	3	13.3805	
	4	13.5347	

Tabelle 7: Level-Vergleich

Zur Visualisierung der Ergebnisse wurde das Programm ParaView [PARA] genutzt. Dies ist eine *open-source* Software zur qualitativen oder quantitativen Darstellung von sehr großen Datenmengen in 2D oder 3D. Die Ergebnisse können dabei als Bild oder Video direkt durch die Software gespeichert werden.

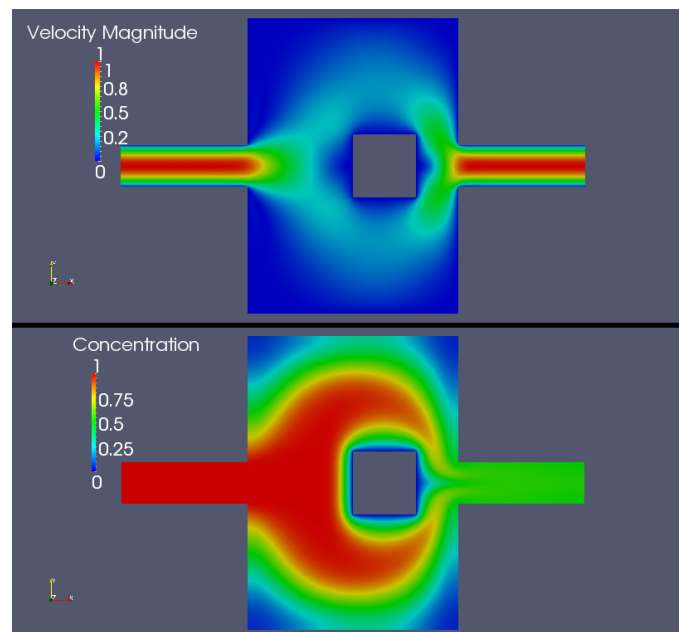


Abbildung 15: 2D für  $t = 50$

Die Daten werden dafür durch NAVIER im VTK-Format [VTK] abgespeichert, dies ist ebenfalls ein *open-source* System, um 3D-Computer-Grafiken darzustellen und abzuspeichern.

In Abbildung 15 sind Geschwindigkeit und Konzentration für  $t = 50$  des Verfeinerungslevels 4 dargestellt.



Eine 3D-Darstellung für dieselbe Konfiguration, wobei die Konzentration in Richtung der z-Achse dargestellt wird, ist in Abbildung 16 zu sehen.

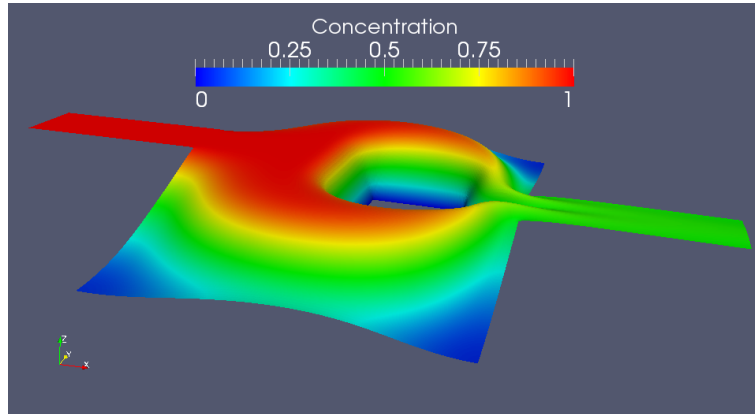


Abbildung 16: 3D für  $t = 50$

In Abbildung 17 sind zusätzlich noch Konturlinien dargestellt. Weitere Darstellungen sowie Videos verschiedener Simulationen sind unter [MOD] verfügbar.

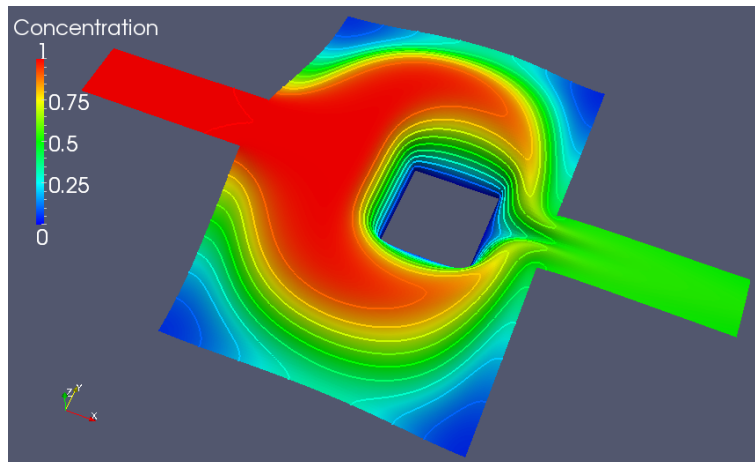


Abbildung 17: 3D mit Konturlinien für  $t = 50$

### 4.3 Zusatzfunktionen für die Regelung

Um NAVIER an unser Szenario hinsichtlich der DCE und deren Auswertung, sowie der späteren Regelung, anzupassen, wurden einige zusätzliche Funktionen in NAVIER integriert. Außerdem sind einige Skripts für MATLAB entworfen, die zum *Postprocessing* genutzt werden können. Diese werden nun kurz erläutert.

#### 4.3.1 outflow-Funktion

Die in Abschnitt 3.5 unter (37) eingeführte Funktion *outflow* beschreibt als Integral über dem Rand  $\Gamma_r$  die Menge an Stoff, der auf Grund der Reaktion am Rand aufgebraucht wird. Die Funktion ist in der Datei **fe\_lib.f90** aus dem Hauptblock von NAVIER implementiert. Dabei werden alle Dreiecke auf Zugehörigkeit zum Rand  $\Gamma_r$  geprüft. Anschließend werden diese auf ein Referenzdreieck transformiert. Über dem entsprechenden Randstück wird dann mittels numerischer Integration mit der Rechteckregel das Integral approximativ bestimmt.

#### 4.3.2 Systemmatrizen

Hauptzielstellung von Szenario 1 ist die Regelung der Menge an gelöstem Stoff, die am Rand  $\Gamma_r$  durch die Reaktion verbraucht wird. Wie in Abschnitt 3.5 beschrieben, handelt es sich um eine DAE, das die Grundlage des Steuerproblems liefert. Um später eine Regelung berechnen zu können, benötigt man die Systemmatrizen, die das zu steuernde System beschreiben. Hierfür existiert in NAVIER die Funktion *write\_matlab\_matrix*, die die innerhalb von NAVIER erzeugten Matrizen in eine für MATLAB lesbare Form herausschreibt. Da in NAVIER der Einsatz direkter Gleichungssystemlöser zur Auswahl steht, werden die Systemmatrizen bei Auswahl explizit erzeugt und im *Compressed-Row-Storage-Format* [CRS] abgespeichert.

Diese Matrizen können mit der MATLAB-Funktion *spconvert* in ein für MATLAB lesbares Format umgewandelt werden.

#### 4.3.3 Feedback-Funktion

Um die Funktion  $\mathcal{U}(t)$  aus Abschnitt 3.5 in NAVIER zu integrieren, wurde in **problem.f90** die Funktion *apply\_feedback* implementiert. In Abhängigkeit der Zeit verändert diese Funktion die Dirichlet-Randbedingungen bei  $\Gamma_{in}$ . Der Funktion stehen dabei die Werte der Zustandsfunktion  $\mathcal{X}(t)$ , ebenfalls in Abschnitt 3.5 definiert, zur Verfügung.

Um die Knotennummer eines Gitterpunktes anhand seiner räumlichen Lage in  $\Omega$  bestimmen zu können, wurde ein MATLAB-Skript *feedback\_point* implementiert. Dies kann sinnvoll sein, falls ein bestimmter Knotenpunkt die Funktion eines Sensors übernehmen soll und die aktuellen Werte dessen in die Regelung mit eingehen sollen.

## 5 Zusammenfassung/Ausblick

### 5.1 Zusammenfassung

**PDEs für Szenario 1** Die als Ansatz für das Szenario 1 aus Abschnitt 1.2 angedachten PDEs wurden in dieser Arbeit erläutert. Auf die Herleitung der NSE wurde dabei verzichtet. Hingegen wurde die DCE und die daraus folgende *outflow*-Funktion hergeleitet. Gemeinsam sind diese in NAVIER implementiert und wurden für verschiedene Konfigurationen getestet. Dabei wurden unterschiedliche Ergebnisse beobachtet.

**Gitterauflösung** Wir haben bei den Tests sehen können, dass sich für die Geschwindigkeit auch für größere Reynoldszahlen ( $Re > 100$ ) gute numerische Ergebnisse erzielen ließen.

Die dabei entstehenden Wirbel stellten jedoch Probleme bei der Berechnung der DCE dar. Am Rand der Wirbel entstehen für die Konzentration innere Schichten, die fein genug aufgelöst werden müssen. Da die Auflösung dieser für die DCE um den Faktor der Schmidtzahl  $Sc$  feiner sein muss, konnten teilweise die Wirbel im Strömungsfeld noch stabil berechnet werden, die Ausbreitung des gelösten Stoffes wurde jedoch zunehmend durch numerische Oszillation verfälscht.

Eine Möglichkeit, die dabei entstehenden numerischen Oszillationen zu verringern, ist die *Streamline-Diffusion-Method*, die in Abschnitt 4.1.1 eingeführt wurde. Leider konnte dieses Verfahren für die Wirbel keine befriedigenden Ergebnisse liefern, wodurch wir zunächst moderate Werte  $Re = 10$  und  $Sc = 10$  als Grundlage für alle Berechnungen betrachteten.

Bei der am Ende verwendeten Gitterauflösung konnte festgestellt werden, dass der Koeffizient, der die Methode in ihrer Intensität steuert, keinen bzw. nur einen geringen Einfluss auf die Ergebnisse hatte.

Eine weitere Möglichkeit zur Vermeidung von Oszillationen ist die Veränderung der Anfangsbedingung, wie in Abschnitt 4.2.2 erläutert.

Für weitere Untersuchungen und die angedachte Regelung muss man einen Kompromiss zwischen Rechengenauigkeit, der dadurch notwendigen Gitterauflösung und dem Aufwand der Rechnungen finden.

**Performancetest** Seit Januar 2010 wurde ein Teil der Compute-Server des MRZ der TU Chemnitz mit neuer Technik ausgerüstet. Ein Vergleich zeigte hierbei gute Ergebnisse, die in Tabelle 8 dargestellt sind.

Bezeichnung	CPU-Time (sec.)	Real-Time (sec.)	Speed-Up
<b>LEIBNIZ</b>	<b>2907</b>	<b>2117</b>	<b>1.37</b>
<i>CASE9</i>	2398	1660	1.45
vgl. zu <i>LEIBNIZ</i>	-17.4%	-21%	+5%
<i>lenovo</i>	2409	1933	1.25
vgl. zu <i>LEIBNIZ</i>	-17.1%	-12.8%	-8.8%

Tabelle 8: Performancetest

Als Vergleichsrechner wurde *CASE9* gewählt. Dieser arbeitet mit einem Intel®Core™i7 CPU 920 mit 4 Kernen, der ebenfalls *Hyper Threading* unterstützt und somit 8 Kerne mit jeweils 2.67 GHz nutzt. Weiterhin ist er mit 8 MB Cache und 12 GB Arbeitsspeicher ausgestattet. Als Betriebssystem ist *openSUSE Linux 11.1 (x86\_64)* in der 64-Bit Version installiert. Für zukünftige Rechnungen werden wir auf diese Rechner zurückgreifen.

Bei *lenovo* handelt es sich um mein aktuelles privates Notebook. Dieses arbeitet mit einem Intel®Core™2 P8700 Prozessor mit 2 Kernen á 2.53 GHz Taktfrequenz und 3 MB Cache. Verbaut sind 4 GB Arbeitsspeicher und 64-Bit *Ubuntu 9.10-Karmic Koala* ist als Betriebssystem installiert. Wie der Test zeigt, konnten auch hier gute Rechenleistungen erreicht werden.

## 5.2 Ausblick

Die Modelle für Szenario 1 sind überprüft und die Berechnungen derer Lösungen in NAVIER wurde getestet. Als nächster Schritt muss nun eine konkrete Regelung berechnet und implementiert werden. NAVIER ist dafür, wie zu Beginn dieses Abschnittes beschrieben, ausreichend vorbereitet. Es gilt nun also, sich detailliert mit der auf dieses Problem bezogenen Regelung für DAEs zu beschäftigen, Lösungsansätze zu implementieren und auf ihre Funktionsfähigkeit zu überprüfen.

Dieser Aufgabe möchte ich mich in meiner Diplomarbeit widmen.

## **6 Schlusswort**

Mein besonderer Dank für die Unterstützung bei meiner Arbeit gilt Prof. Peter Benner, Dr. René Schneider, Dr. Jens Saak und Martin Köhler von der TU Chemnitz sowie Prof. Eberhard Bänsch, Dipl. math. Stephan Weller und Dipl. Ing. Rodolphe Prigniz von der FAU Erlangen. Außerdem danke ich allen fleißigen Korrekturlesern für das Ausbügeln der Fehler in Sprache und Text.

## 7 Quellenverzeichnis

### Literatur

- [BAEN] **An adaptive finite-elemente-strategy for the three-dimensional time dependent Navier-Stokes equations**  
Autor: Eberhard Bänsch  
J. Comp. Appl. Math. 36, 3-28 (1991)
- [CRS] [http://de.wikipedia.org/wiki/Compressed\\_Row\\_Storage](http://de.wikipedia.org/wiki/Compressed_Row_Storage)
- [DFG] [http://www.tu-chemnitz.de/mathematik/industrie\\_technik/projekte/optconfeestabmultiflow](http://www.tu-chemnitz.de/mathematik/industrie_technik/projekte/optconfeestabmultiflow)
- [HOEH] **Numerik für die Marangoni-Konvektion beim Floating-Zone Verfahren**  
Autor: Burkhard Höhn  
Dissertation Mathematische Fakultät Albert-Ludwigs-Universität Freiburg i. B. 1999
- [HPT] <http://de.wikipedia.org/wiki/Hyper-Threading>
- [MOD] [http://www.tu-chemnitz.de/~wehei/html/mod\\_sem.php](http://www.tu-chemnitz.de/~wehei/html/mod_sem.php)
- [MRZ] <http://www.tu-chemnitz.de/mathematik/mrz/computer.php>
- [NLGS] **Numerik linearer Gleichungssysteme:**  
Eine Einführung in moderne Verfahren  
Autor: Andreas Meister  
Vieweg Braunschweig, Wiesbaden 1999
- [NPDE] **Numerische Behandlung partieller Differentialgleichungen**  
Autoren: Christian Großmann, Hans-Görg Roos  
Teubner Verlag 1994
- [PARA] <http://www.paraview.org>
- [PARD] <http://www.pardiso-project.org>
- [RENS] **Applications of the Discrete Adjoint Method in Computational Fluid Dynamics**  
Autor: René Schneider  
Dissertation School of Computing, University of Leeds, April 2006
- [ROOS] **Layer-Adapted Grids for Singular Perturbation Problems**  
Autor: H.-G. Roos  
Zeitschrift für Angewandte Mathematik und Technik (ZAMM), 78(5):291-309, 1998
- [SLD] **Finite Elements and Fast Iterative Solvers**  
with applications in incompressible fluid dynamics  
Autoren: Howard Elman, David Silvester, Andy Wathen  
Oxford University Press 2005

- [SRT] **Regelungstechnik:**  
 mathematische Grundlagen, Entwurfsmethoden, Beispiele  
 Autor: Hans Peter Geering  
 Springer Verlag 2004
- [STM] **Strömungsmechanik A-Z:**  
 Eine systematische Einordnung von Begriffen und Konzepten der  
 Strömungsmechanik  
 Autor: Heinz Herwig  
 Vieweg Praxiswissen 2004
- [VTK] <http://www.vtk.org>
- [WST] **Wärme- und Stofftransport**  
 Autoren: Häfner, Sames, Voigt  
 Springer-Lehrbuch 1992

## 8 Verzeichnisse

### Abbildungsverzeichnis

1	Gebietsdefinition . . . . .	5
2	Geschlossener Regelkreis . . . . .	7
3	Konzentrationsveränderung in Kontrollvolumen . . . . .	10
4	Schematische Darstellung der Grenzschicht . . . . .	14
5	Kontrollvolumen $S_E$ am Rand $\Gamma_r$ . . . . .	16
6	Intervallteilung für Zwischenschritt- $\theta$ -Verfahren . . . . .	20
7	Ausgangstriangulation . . . . .	24
8	Anfangs- und Einströmbedingung für Konzentration . . . . .	26
9	Zeitlicher Verlauf für verschiedene AB . . . . .	27
10	Maximale Konzentration . . . . .	29
11	Minimale Konzentration . . . . .	30
12	Konzentration, die pro Zeiteinheit aufgebraucht wird . . . . .	30
13	Zoom des Minimums . . . . .	31
14	Zoom des Maximums . . . . .	31
15	2D für $t = 50$ . . . . .	32
16	3D für $t = 50$ . . . . .	33
17	3D mit Konturlinien für $t = 50$ . . . . .	33

### Tabellenverzeichnis

1	Problem-Input-Dateien . . . . .	21
2	Konfigurationsdatei . . . . .	22
3	Schematischer Aufbau der Triangulationsdatei . . . . .	23
4	Kodierung der Randbedingungen . . . . .	23
5	Konfigurationsvergleich . . . . .	25
6	Gitterparameter und Rechenzeit . . . . .	28
7	Level-Vergleich . . . . .	32
8	Performancetest . . . . .	35

