



Georg-August-Universität  
Göttingen  
Zentrum für Informatik

ISSN 1612-6793  
Nummer ZFI-BM-2008-02

## **Masterarbeit**

im Studiengang "Angewandte Informatik"

# **Implementierung eines iterativen Druckkorrektur-Verfahrens beim zweidimensionalen Driven-Cavity-Problem**

Irina Aab

am Institut für

Numerische und Angewandte Mathematik

Bachelor- und Masterarbeiten  
des Zentrums für Informatik  
an der Georg-August-Universität Göttingen

5. Februar 2008

Georg-August-Universität Göttingen  
Zentrum für Informatik

Lotzestraße 16-18  
37083 Göttingen  
Germany

Tel. +49 (5 51) 39-1 44 14

Fax +49 (5 51) 39-1 44 15

Email [office@informatik.uni-goettingen.de](mailto:office@informatik.uni-goettingen.de)

WWW [www.informatik.uni-goettingen.de](http://www.informatik.uni-goettingen.de)

---

Ich erkläre hiermit, dass ich die vorliegende Arbeit selbständig verfasst und keine anderen als die angegebenen Quellen und Hilfsmittel verwendet habe.

Göttingen, den 5. Februar 2008



Masterarbeit

**Implementierung eines iterativen  
Druckkorrektur-Verfahrens beim  
zweidimensionalen Driven-Cavity-Problem**

Irina Aab

5. Februar 2008

Betreut durch Prof. Dr. Robert Schaback  
Institut für Numerische und Angewandte Mathematik (NAM)  
Georg-August-Universität Göttingen



# Danksagung

An dieser Stelle möchte ich ganz herzlich meinem Hauptbetreuer, Herrn Prof. Dr. Robert Schaback vom Institut für Numerische und Angewandte Mathematik der Georg-August-Universität Göttingen, für die interessante Aufgabenstellung und die hervorragende Betreuung danken. Seine Ideen und Vorschläge waren bei der Anfertigung dieser Arbeit immer eine wesentliche Hilfe.

Des Weiteren bedanke ich mich bei Herrn Prof. Dr. Holger Wendland von der Universität Sussex, England für die Zweitkorrektur dieser Arbeit.

Ein besonderer Dank gilt meiner Familie, insbesondere meinem Mann Sergej, meinem Sohn Nikita und meinen Eltern Waldemar und Ljuba für die wertvolle seelische Unterstützung während meines gesamten Studiums, was mit der Abgabe dieser Arbeit nun abgeschlossen ist.

Anschließend möchte ich mich bei meiner Schwester Anastasia für das gewissenhafte Korrekturlesen dieser Arbeit bedanken.

---



# Abstract

Strömungssimulationen sind zu einer wichtigen Aufgabe bei Entwicklungsarbeiten in Industrie und Wissenschaft geworden. Die Strömungsmechanik oder auch Fluidmechanik ist die Lehre des physikalischen Verhaltens von Fluiden. Unter dem Begriff Fluid versteht man Medien, welche sich unter dem Einfluss von Scherspannungen unbegrenzt verformen (also Flüssigkeiten oder Gase). Die Strömungslehre wird in Fachgebiete aufgeteilt, die sich mit verschiedenen Teilaspekten von Fluiden auseinandersetzen.

Zwei große Gebiete der Strömungslehre sind die Fluidstatik und die Fluiddynamik. Die Fluidstatik ist die Lehre von (annähernd) ruhenden Fluiden. Ein Teilgebiet der Fluidstatik ist z.B. die Aerostatik, die sich mit der Schichtung der ruhenden Atmosphäre bzw. Erdatmosphäre beschäftigt. Ein anderer Teilaspekt wäre die Hydrostatik. Diese betrachtet unter anderem die Druckverteilung in ruhenden Flüssigkeiten und die Schwimmstabilität von Körpern. Die Fluiddynamik ist die Lehre von bewegten Fluiden. Deren Teilgebiete sind z.B. die Aerodynamik, die speziell das Verhalten von Körpern in Gasen betrachtet, und die Hydrodynamik, die die Bewegung unter Wirkung der vom Medium selbst erzeugten Felder untersucht.

Innerhalb der Fluiddynamik werden einige Eigenschaften beschrieben, die verschiedene Aspekte der Fluide abbilden. Zu den Eigenschaften gehören z.B.

- Das Verhalten des Fluids
    - inkompressible Strömung (Strömung, die ihre Dichte bei Druck von Außen nicht ändert)
    - kompressible Strömung
  - Die Strömungsart
    - instationäre Strömung
    - stationäre Strömung
  - Die Strömungsform
    - laminare Strömung
    - turbulente Strömung
-

- 
- Die Art des Fluids
    - reibungsfreie Strömung
    - viskose Strömung

Eine etablierte Methode der Strömungsmechanik ist die numerische Strömungsmechanik. Sie hat das Ziel, strömungsmechanische Probleme approximativ mit numerischen Methoden zu lösen. Die benutzten Modellgleichungen sind unter anderem die Navier-Stokes-Gleichungen. Es handelt sich hierbei um ein System von nichtlinearen partiellen Differentialgleichungen zweiter Ordnung, die ein newtonsches Fluid komplett beschreiben.

Die vorliegende Masterarbeit beschäftigt sich mit der numerischen Lösung der Navier-Stokes-Gleichung auf dem sogenannten „driven-cavity-problem“. Eine Schwierigkeit bei der Behandlung dieser Gleichung besteht in der Kopplung zweier Größen - dem Druck und dem Strömungsfluss. Der in dieser Arbeit implementierte Algorithmus basiert auf zwei Iterationsschleifen. In der ersten Schleife wird eine erste Lösung für den Strömungsfluss berechnet, der auf konstantem Druck basiert. In der zweiten Schleife wird zuerst immer die Druckfunktion berechnet. Danach folgt eine Interpolation des Strömungsflusses. Die Druckfunktion wird immer wieder neu erzeugt, bis die zweite Schleife mit der Berechnung vom Strömungsfluss deren stabilen Zustand erreicht hat. Die Ergebnisse werden auf dem Standardproblem der Strömungsmechanik, dem zweidimensionalen „driven-cavity-problem“, dargestellt. Mit Hilfe der Schrittweitensteuerung wird das Gitter von  $10 \times 10$  auf  $20 \times 20$  und  $30 \times 30$  Punkte auf dem  $\Omega = [0, 1] \times [0, 1]$  verfeinert.

---

# Inhaltsverzeichnis

<b>Symbolverzeichnis</b>	<b>II</b>
<b>Operatorverzeichnis</b>	<b>IV</b>
<b>Abbildungsverzeichnis</b>	<b>VI</b>
<b>Tabellenverzeichnis</b>	<b>VIII</b>
<b>1 Einleitung</b>	<b>1</b>
1.1 Aufbau der vorliegenden Arbeit . . . . .	3
<b>2 Einführung in die Theorie</b>	<b>5</b>
2.1 Navier-Stokes-Gleichungen . . . . .	5
2.1.1 Existenz und Eindeutigkeit der Lösung der Navier-Stokes-Gleichungen	6
2.2 Das zweidimensionale Driven-Cavity-Problem . . . . .	6
2.3 Anwendung der radialen Basisfunktionen . . . . .	7
2.3.1 Der konstante Druck . . . . .	8
<b>3 Numerische Grundlagen des Verfahrens</b>	<b>11</b>
3.1 Radiale Basisfunktionen . . . . .	11
3.1.1 Wendland-Funktionen . . . . .	12
3.2 Interpolation mit radialen Basisfunktionen . . . . .	12
3.3 Die grundlegende numerische Methode . . . . .	15
3.3.1 Der normale Druck . . . . .	17
3.4 Algorithmus . . . . .	19
3.4.1 Algorithmus I . . . . .	19
3.4.2 Algorithmus II . . . . .	20
<b>4 Implementierung</b>	<b>23</b>
4.1 Modulaufbau und Initialisierung . . . . .	23
4.2 Die einzelnen Schritte der Algorithmen . . . . .	24
4.2.1 Der erste Schritt . . . . .	24
4.2.2 Der zweite Schritt . . . . .	24

4.2.3	Der dritte Schritt . . . . .	26
4.2.4	Der vierte Schritt . . . . .	29
4.3	Die verwendeten Funktionen . . . . .	30
4.3.1	Die Wendland-Funktion . . . . .	30
4.3.2	Die Gauss-Funktion . . . . .	31
4.4	Die Unterschiede zwischen der Doktorarbeit von Svenja Lowitzsch und der vorliegenden Masterarbeit . . . . .	33
4.4.1	Grenzen der Randpunkte . . . . .	33
4.4.2	Definition der Druckfunktion . . . . .	33
<b>5</b>	<b>Aufbau und Ergebnisse</b>	<b>35</b>
5.1	Ergebnisse der ersten Lösungsmöglichkeit . . . . .	35
5.1.1	Das Gitter der Größe $10 \times 10$ . . . . .	35
5.1.2	Das Gitter der Größe $20 \times 20$ . . . . .	36
5.1.3	Das Gitter der Größe $30 \times 30$ . . . . .	38
5.2	Ergebnisse der zweiten Lösungsmöglichkeit . . . . .	43
5.2.1	Das Gitter der Größe $10 \times 10$ . . . . .	43
5.2.2	Das Gitter der Größe $20 \times 20$ . . . . .	46
5.2.3	Das Gitter der Größe $30 \times 30$ . . . . .	48
<b>6</b>	<b>Zusammenfassung und Ausblick</b>	<b>51</b>
<b>7</b>	<b>Literaturverzeichnis</b>	<b>53</b>

# Symbolverzeichnis

$\alpha$	Koeffizientenvektor
$A$	Interpolationsmatrix
$\beta$	Koeffizientenvektor
$B = \{B_1, \dots, B_4\}$	Menge der Randpunkte
$d$	Datensätze, die aus der Funktion $f$ an gegebenen Punkten $(X, Y)$ stammen
$D^B$	Datensätze auf dem Rand
$e$	Exponentialfunktion
$f$	vektorielle Funktion, $f : \mathbb{R}^s \rightarrow \mathbb{R}$
$g$	Funktion der Gaussglocke
$I$	$s$ -dimensionale Einheitsmatrix
$I$	Menge der inneren Punkte
$\Omega$	kompakte Teilmenge von $\mathbb{R}^s$
$\nu$	Viskositätskoeffizient
$N^B$	Anzahl aller Interpolationspunkte auf dem Rand von $\Omega$
$N^h$	Anzahl der Interpolationspunkte auf den horizontalen Rändern $B_1$ und $B_4$ von $\Omega$
$N^v$	Anzahl der Interpolationspunkte auf den vertikalen Rändern $B_2$ und $B_3$ von $\Omega$
$N^I$	Anzahl der Interpolationspunkte im Inneren von $\Omega$
$\Phi$	matrixwertige radiale Basisfunktion
$\psi$	skalarwertige radiale Basisfunktion mit kompaktem Träger
$p$	Druck
$s$	Dimension des Vektorfeldes
$s_{f,X}$	Interpolante
$t$	Zeit
$\mathbf{u} = (u, v)$	Geschwindigkeit der Strömung
$X = \{x_1, \dots, x_N\}$	paarweise verschiedene Punkte, $X \subset \Omega \subset \mathbb{R}^s$
$Y = \{y_1, \dots, y_N\}$	paarweise verschiedene Punkte, $Y \subset \Omega \subset \mathbb{R}^s$
$(X, Y)$	gegebene Punkte: $(X, Y) \in \Omega$



# Operatorverzeichnis

$A^{-1}$	Inverse einer Matrix
$\Delta$	Laplace-Operator
$\nabla$	Gradient



# Abbildungsverzeichnis

2.1	Die Randbedingungen beim zweidimensionalen „driven-cavity-problem“ . . . . .	7
3.1	$\psi_{4,2}(r) = (1 - r)_+^6 (35r^2 + 18r + 3)$ . . . . .	13
5.1	Das Strömungsfeld $s_r$ nach dem Ausführen von Schritt 2 für $h = 0,1$ . . . . .	36
5.2	Das stabilisierte Druckfeld $p$ nach dem Ausführen von Schritt 3 für $h = 0,1$ . . . . .	37
5.3	Das stabilisierte Strömungsfeld $s_r$ nach dem Ausführen von Schritt 4 für $h = 0,1$ . . . . .	37
5.4	Das Strömungsfeld $s_r$ nach dem Ausführen von Schritt 2 für $h = 0,05$ . . . . .	39
5.5	Das stabilisierte Druckfeld $p$ nach dem Ausführen von Schritt 3 für $h = 0,05$ . . . . .	39
5.6	Das stabilisierte Strömungsfeld $s_r$ nach dem Ausführen von Schritt 4 für $h = 0,05$ . . . . .	40
5.7	Das Strömungsfeld $s_r$ nach dem Ausführen von Schritt 2 für $h = 0,0333$ . . . . .	41
5.8	Das stabilisierte Druckfeld $p$ nach dem Ausführen von Schritt 3 für $h = 0,0333$ . . . . .	42
5.9	Das stabilisierte Strömungsfeld $s_r$ nach dem Ausführen von Schritt 4 für $h = 0,0333$ . . . . .	42
5.10	Das Strömungsfeld $s_r$ nach dem Ausführen von Schritt 2 für $h = 0,1$ . . . . .	44
5.11	Das stabilisierte Druckfeld $p$ nach dem Ausführen von Schritt 3 für $h = 0,1$ . . . . .	44
5.12	Das stabilisierte Strömungsfeld $s_r$ nach dem Ausführen von Schritt 4 für $h = 0,1$ . . . . .	45
5.13	Das Strömungsfeld $s_r$ nach dem Ausführen von Schritt 2 für $h = 0,05$ . . . . .	46
5.14	Das stabilisierte Druckfeld $p$ nach dem Ausführen von Schritt 3 für $h = 0,05$ . . . . .	47
5.15	Das stabilisierte Strömungsfeld $s_r$ nach dem Ausführen von Schritt 4 für $h = 0,05$ . . . . .	47
5.16	Das Strömungsfeld $s_r$ nach dem Ausführen von Schritt 2 für $h = 0,0333$ . . . . .	49
5.17	Das stabilisierte Druckfeld $p$ nach dem Ausführen von Schritt 3 für $h = 0,0333$ . . . . .	49
5.18	Das stabilisierte Strömungsfeld $s_r$ nach dem Ausführen von Schritt 4 für $h = 0,0333$ . . . . .	50



# Tabellenverzeichnis

3.1	Verschiedene Wendland-Funktionen . . . . .	13
5.1	$\ell_2$ -Fehler der iterativen Interpolation mit der ersten Lösungsmöglichkeit bei einer Auflösung von $10 \times 10$ . . . . .	38
5.2	$\ell_2$ -Fehler der iterativen Interpolation mit der ersten Lösungsmöglichkeit bei einer Auflösung von $20 \times 20$ . . . . .	40
5.3	$\ell_2$ -Fehler der iterativen Interpolation mit der ersten Lösungsmöglichkeit bei einer Auflösung von $30 \times 30$ . . . . .	43
5.4	$\ell_2$ -Fehler der iterativen Interpolation mit der zweiten Lösungsmöglichkeit bei einer Auflösung von $10 \times 10$ . . . . .	45
5.5	$\ell_2$ -Fehler der iterativen Interpolation mit der zweiten Lösungsmöglichkeit bei einer Auflösung von $20 \times 20$ . . . . .	48
5.6	$\ell_2$ -Fehler der iterativen Interpolation mit der zweiten Lösungsmöglichkeit bei einer Auflösung von $30 \times 30$ . . . . .	50



# 1 Einleitung

Die Strömungsmechanik (auch Fluidmechanik oder Fluidodynamik genannt) befasst sich mit dem Verhalten von Fluiden. Die Strömungsvorgänge der Fluide werden allgemein durch die Geschwindigkeit, den Druck, die Dichte, die Viskosität und die Temperatur als Funktion von Ort und Zeit beschrieben. Die Navier-Stokes-Gleichungen sind die Grundgleichungen der Strömungsmechanik. Sie sind die am besten untersuchten und in der Praxis am häufigsten benutzten Gleichungen.

Die Navier-Stokes-Gleichungen verkörpern ein System von nichtlinearen partiellen Differentialgleichungen zweiter Ordnung. Das System besteht aus einem Impulssatz und der Bedingung der Divergenzfreiheit des Geschwindigkeitsfeldes. Eine Grundannahme der vorliegenden Arbeit ist die Divergenzfreiheit des Ansatzraumes.

Die Interpolation und Approximation mittels radialer Basisfunktionen (RBFs) ist interessant in ihren Anwendungen in der Signalverarbeitung, in der Computertomografie, sowie in vielen anderen Bereichen. In der vorliegenden Masterarbeit wird ein numerisches Verfahren zur Lösung von Strömungsaufgaben mit Hilfe neuartiger kernbasierter Methoden eingeführt.

In der Doktorarbeit von Frau Svenja Lowitzsch, auf der die Arbeit basiert, wurde eine neue Klasse von divergenzfreen RBFs vorgestellt, die auf skalarwertigen RBFs mit kompakten Trägern basieren und folglich viel effizienter für Rechenzwecke sind, weil schnelle Löser gefunden werden können.

Die allgemeine Form der  $s \times s$  matrixwertigen RBF wird definiert als

$$\Phi(x) = -\Delta I + \nabla \nabla^T \psi(x). \quad (1.1)$$

Dabei ist  $\Delta$  der Laplace-Operator,  $I$  die  $s$ -dimensionale Einheitsmatrix,  $\nabla$  der Gradient und  $\psi$  eine skalarwertige RBF mit einem kompakten Träger. Dann ist  $\Phi$  divergenzfrei und positiv definit. In dieser Arbeit werden als  $\psi$  die Wendland Funktionen benutzt.

Sei  $\Omega$  eine kompakte Teilmenge von  $\mathbb{R}^s$ ,  $X = \{x_1, \dots, x_N\} \subset \Omega \subset \mathbb{R}^s$  paarweise verschiedene Punkte und  $f$  eine vektorwertige Funktion  $f : \mathbb{R}^s \rightarrow \mathbb{R}$ . In diesem Fall, wenn die zu interpolierenden Daten die Form  $\{x_j, f_j\}_{j=1}^N$  annehmen, und wenn  $\{f_j\}_{j=1}^N \in \mathbb{R}^N$  Datensätze sind, die aus der Funktion  $f$  an gegebenen Punkten  $X$  stammen, also  $f_j := f(x_j)$ , hat die divergenzfreie Interpolationsfunktion  $s_{f,X}$  die Form

$$s_{f,X}(x) = \sum_{j=1}^N \alpha_j \Phi(x - x_j). \quad (1.2)$$

Das Interpolationsproblem besteht darin, ein  $\{\alpha_j\}_{j=1}^N$  zu finden, so dass

$$s_{f,X}(x_j) = f_j \quad (1.3)$$

für alle  $1 \leq j \leq N$ .

Dieses Problem kann umformuliert werden zu einem linearen  $2N \times 2N$  - Gleichungssystem

$$\sum_{j=1}^N \alpha_j \Phi(x_k - x_j) = f_k, \quad (1.4)$$

für alle  $1 \leq j, k \leq N$ . Die entstehende Interpolationsmatrix  $A$  ist positiv definit, symmetrisch und schwach besetzt. Deswegen können schnelle Löser eingesetzt werden, um ein einheitliches Ergebnis zu bekommen.

In der Doktorarbeit von Frau Svenja Lowitzsch wird der Fall für einen konstanten Druck behandelt. D.h. es werden zwei folgende Annahmen gemacht:

1. Das Strömungsproblem ist zeitunabhängig

$$\frac{\partial \mathbf{u}}{\partial t} = 0,$$

2. Der Strömungsfluss hat einen konstanten Druck

$$\nabla p = 0.$$

Mit diesen Annahmen kann die Navier-Stokes-Gleichung der inkompressiblen Fluidströmung

$$\begin{cases} \frac{\partial \mathbf{u}}{\partial t} + (\mathbf{u} \cdot \nabla) \mathbf{u} = -\nabla p + \nu \Delta \mathbf{u}, \\ \nabla \cdot \mathbf{u} = 0 \end{cases} \quad (1.5)$$

zum folgenden Problem umgeschrieben werden:

Finde  $\mathbf{u} = (u, v)^T$  aus dem divergenzfreien Ansatzraum, so dass

$$\begin{cases} \nu \Delta \mathbf{u} = (\mathbf{u} \cdot \nabla) \mathbf{u} & \text{in } X^I, \\ \mathbf{u} = (1, 0)^T & \text{in } B_1, \\ \mathbf{u} = (0, 0)^T & \text{in } B_2, B_3, B_4, \end{cases} \quad (1.6)$$

wobei  $\nu$  der Viskositätskoeffizient ist.  $X^I$  steht für die Menge der inneren Punkte von dem gegebenen Raum,  $B_1$  bis  $B_4$  beschreiben die Mengen der Punkte an den jeweiligen Rändern.

In diesem Fall braucht man die Voraussetzung

$$\nabla \cdot \mathbf{u} = 0$$

nicht, da die Interpolante diese Bedingung selbstverständlich erfüllt.

In der vorliegenden Arbeit wird zum ersten Mal auf den Fall eines nichtkonstanten Druckes eingegangen. Folglich bleibt nur die Annahme:

1. Das Strömungsproblem ist zeitunabhängig.

Damit ergibt sich aus der Navier-Stokes-Gleichung das Problem:

Finde  $\mathbf{u} = (u, v)^T$ , so dass

$$\begin{cases} (v \Delta - (\mathbf{u} \cdot \nabla))\mathbf{u} = \nabla p & \text{in } X^I, \\ \mathbf{u} = (1, 0)^T & \text{in } B1, \\ \mathbf{u} = (0, 0)^T & \text{in } B2, B3, B4. \end{cases} \quad (1.7)$$

Die Neumann-Bedingungen an den vier Rändern  $B1$  bis  $B4$  sind gegeben als

$$\begin{cases} \partial_2 p = v \partial_2^2 v & \text{in } B1, \\ \partial_1 p = v \partial_1^2 u & \text{in } B2, \\ \partial_1 p = -v \partial_1^2 u & \text{in } B3, \\ \partial_2 p = -v \partial_2^2 v & \text{in } B4. \end{cases} \quad (1.8)$$

Die Interpolante zur Berechnung des Druckes wird definiert als

$$p(x) := \sum_{s=1}^{N^h} \beta_{r,s}^h \partial_2 g(x - x_s^h) + \sum_{t=1}^{N^v} \beta_{r,t}^v \partial_1 g(x - x_t^v) + \sum_{k=1}^{N^l} \beta_{r,k}^l \Delta g(x - x_k^l), \quad (1.9)$$

wobei  $x^h \in B1 \cup B4$  und  $x^v \in B2 \cup B3$  den horizontalen und vertikalen Rändern entsprechen. Daher liegt die eigentliche numerische Methode wie folgt vor:

Gegeben sind  $s_{r-1}$  und  $s_r = (u_r, v_r)^T$ . Finde  $p$ , so dass

$$\begin{cases} (v \Delta - (s_{r-1} \cdot \nabla))s_r = \nabla p & \text{in } X^I, \\ \partial_2 p = \pm v \partial_2^2 v_r & \text{in } B1, B4, \\ \partial_1 p = \pm v \partial_1^2 u_r & \text{in } B2, B3. \end{cases} \quad (1.10)$$

## 1.1 Aufbau der vorliegenden Arbeit

Die Arbeit beginnt in Kapitel 2 mit der Einführung in die Theorie. Es werden die grundlegenden mathematischen Bausteine, die dem implementierten Druckkorrektur-Verfahren zugrunde liegen, erläutert. Zuerst werden im Kapitel 2.1 die Navier-Stokes-Gleichungen eingeführt. Die Existenz und Eindeutigkeit der Lösung von ihnen wird im Unterkapitel 2.1.1 dargestellt. Danach wird im Kapitel 2.2 das zweidimensionale „driven-cavity-problem“ genau erläutert. Im Kapitel 2.3 folgt die Anwendung bei radialen Basisfunktionen. Als Unterkapitel 2.3.1 wird der Begriff des konstanten Druckes erklärt.

Im dritten Kapitel werden die numerischen Grundlagen des Verfahrens dargestellt. In Kapitel 3.1 werden die radialen Basisfunktionen eingeführt. Als ein Spezialfall der radialen Basisfunktionen werden in Unterkapitel 3.1.1 die Wendland-Funktionen beschrieben. In Kapitel 3.2 wird die Interpolation mit radialen Basisfunktionen erklärt. Im Weiteren wird in Kapitel 3.3 die eigentliche Problemstellung beschrieben. Die Umstellung vom konstanten auf nicht mehr konstanten Druck wird in Unterkapitel 3.3.1 eingebracht. Abschließend wird in Kapitel 3.4 der Lösungsalgorithmus von Frau Svenja Lowitzsch aufgegriffen, welcher den Kern der vorliegenden Arbeit, sowie den Ausgangspunkt für alle nachfolgenden Berechnungen bildet. Darüber hinaus wird in Unterkapitel 3.4.2 ein zweiter Algorithmus aufgestellt, dessen Unterschied zu dem ersten Algorithmus in den folgenden Kapiteln genau erläutert wird.

Die Implementierung von der in Kapitel 3.4 präsentierten Algorithmen erfolgt in Kapitel 4. Die Darstellung der Implementierung beginnt mit der Beschreibung vom Modulbau und der Initialisierung (Kapitel 4.1). In Kapitel 4.2 wird die Implementierung der einzelnen Schritte von dem Code genau erklärt. Danach wird in Kapitel 4.3 gezeigt, wie die beiden bei den Algorithmen verwendeten Funktionen, die Wendland-Funktion und die Gauss-Funktion, analytisch berechnet werden. Anschließend werden in Kapitel 4.4 die Unterschiede zwischen der Doktorarbeit von Svenja Lowitzsch und der vorliegenden Masterarbeit herausgearbeitet.

In Kapitel 5 wird der Aufbau des Rechengitters und der Eingabeparameter erklärt und die Ergebnisse vorgestellt. Das Rechengitter wird von der Größe  $10 \times 10$  auf die Größen  $20 \times 20$  und  $30 \times 30$  verfeinert.

Um einen Überblick gewinnen zu können, werden in Kapitel 6 die wichtigsten Punkte der Arbeit zusammengefasst. Auch wird ein Ausblick auf die möglichen weiterführenden Arbeiten gegeben.

Die Arbeit endet mit der Literaturliste der zur Entwicklung des Verfahrens und Beschreibung der Arbeit benutzten Quellen.

## 2 Einführung in die Theorie

Die Untersuchung des Verhaltens von Strömungen spielt in der numerischen Simulation die entscheidende Rolle. Strömungen tauchen in fast allen Lebensbereichen auf. Beispiele sind der aufsteigende Rauch einer Zigarette oder das Ein- und Auslaufen von Wasser in einer Badewanne [11].

Flüssigkeiten und Gase werden unter dem Oberbegriff *Fluide* zusammengefasst. Während einerseits die Kräfte zwischen einem strömenden Fluid und einem ruhenden Körper eine Rolle spielen, ist das unterschiedliche Verhalten von Fluiden andererseits auf die Wechselwirkung zwischen den Fluidpartikeln untereinander zurückzuführen. Der Grund für diese Kräfte besteht in der sogenannten Viskosität, der Zähigkeit der Fluide. Unter Zähigkeit versteht man die Eigenschaft eines Fluids, gegen Veränderungen seiner Form Widerstand zu leisten. Das bedeutet, dass die Zähigkeit eines Fluids innere Reibung verursacht, was zur Folge hat, dass ein sich bewegendes Fluid auch ohne Einfluß äußerer Kräfte zur Ruhe kommen kann.

Es ist klar, dass bei sehr zähen Fluiden, wie z.B. Honig, die innere Reibungskraft sehr stark ist. Das bedeutet, dass der Widerstand gegen Verformung sehr viel höher ist als bei weniger zähen Fluiden, wie z.B. Wasser.

Eine weitere Unterscheidung betrifft kompressible und inkompressible Fluide. Kompressibel bedeutet, dass das Fluid zusammengedrückt werden kann. Insbesondere muss somit die gleiche Masse des Fluids nicht immer das gleiche Volumen haben, denn das Volumen hängt direkt vom Druck ab. In dieser Arbeit werden nur inkompressible Fluide betrachtet, also solche, bei denen die Fluidmasse immer das gleiche Volumen einnimmt, d.h. die Dichte des Fluids ist konstant.

### 2.1 Navier-Stokes-Gleichungen

Allgemein wird die Strömung eines Fluids in einem Gebiet  $\Omega \subset \mathbb{R}^N$  ( $N \in \{2, 3\}$ ) über der Zeit  $t \in [0, t_{end}]$  durch folgende Größen charakterisiert:

- das Geschwindigkeitsfeld  $\mathbf{u} : \Omega \times [0, t_{end}] \rightarrow \mathbb{R}^N$ ,
- und den Druck  $p : \Omega \times [0, t_{end}] \rightarrow \mathbb{R}$ .

Bei kompressiblen Fluiden kommt die Dichte  $\rho : \Omega \times [0, t_{end}] \rightarrow \mathbb{R}$  als Zustandsgröße hinzu.

Die inkompressible Strömung kann nun durch die *Impulsgleichung*

$$\frac{d\mathbf{u}}{dt} + (\mathbf{u} \cdot \nabla)\mathbf{u} + \nabla p - \nu \Delta \mathbf{u} = f, \quad (2.1)$$

und die *Kontinuitätsgleichung*

$$\nabla \cdot \mathbf{u} = 0 \quad (2.2)$$

in  $\Omega, t \in [0, t_{end}]$  beschrieben werden. Das System der partiellen Differentialgleichungen (2.1) und (2.2) wird als *Navier-Stokes-Gleichungen* bezeichnet.

Dabei ist  $f : \Omega \rightarrow \mathbb{R}^2$  eine vorgeschriebene vektorwertige Funktion, die die äußeren Kräfte beschreibt, wie z.B. die Erdanziehung, denen das physikalische System unterliegt. Der Druck  $p$  ist bis auf eine Konstante bestimmt. Die reelle Zahl  $\nu$  ist der Viskositätskoeffizient, der im dimensionslosen Fall gerade dem Kehrwert der in der Strömungsmechanik wichtigen Reynoldszahl  $Re$  entspricht. Die Reynoldszahl wird wie folgt definiert:

$$Re \sim \frac{1}{\nu},$$

wobei  $\nu$  die Zähigkeit des Fluides ist.

### 2.1.1 Existenz und Eindeutigkeit der Lösung der Navier-Stokes-Gleichungen

Abschließend bleibt die Frage nach der Existenz und Eindeutigkeit der Lösung der Navier-Stokes-Gleichung zu klären. Es wurden mehrere Wege gefunden, die Existenz einer schwachen Lösung sowohl in zwei, als auch in drei Raumdimensionen zu beweisen [12]; so zum Beispiel mit dem Beweis aus den 1980er Jahren von Sohr [13].

Für die schwache Lösung in zwei Raumdimensionen konnte die sogenannte Serrin's Bedingung aus [14] gezeigt werden, was deren Eindeutigkeit garantiert. In drei Raumdimensionen konnte die Eindeutigkeit noch nicht gezeigt werden.

Sowohl in zwei, als auch in drei Raumdimensionen gilt, dass existierende klassische Lösungen auch eindeutig sind.

## 2.2 Das zweidimensionale Driven-Cavity-Problem

Das „driven-cavity-problem“ ist ein klassisches Test-Beispiel der numerischen Strömungssimulation. Dieses Problem wird vor allem benutzt um numerische Methoden zu testen, die auf den Navier-Stokes Gleichungen basieren [10].

Als Modell dient dabei ein mit einem Fluid gefüllter rechteckiger Behälter, über dessen obere Kante ein Antriebsband mit einer vorgegebenen Geschwindigkeit gezogen wird, wodurch sich das Fluid in Bewegung setzt. Die Abbildung 2.1 zeigt eine schematische Darstellung des Modells [4].

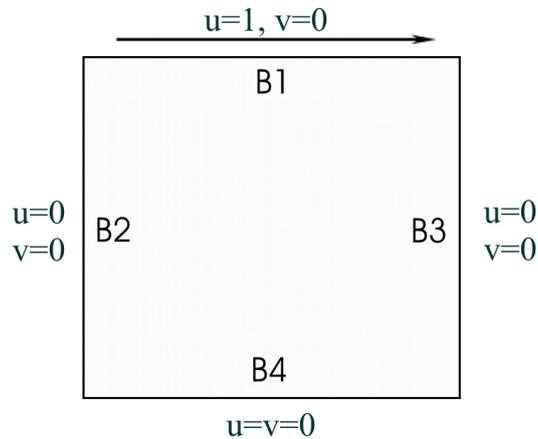


Abbildung 2.1: Die Randbedingungen beim zweidimensionalen „driven-cavity-problem“

Die Randbedingungen auf der unteren, der linken und der rechten Seite sind stationär, also für  $\mathbf{u} = (u, v)$  gesetzt auf  $u = v = 0$ . An der oberen Seite wird das Antriebsband in  $x$ -Richtung mit der Geschwindigkeit  $u = 1$  gezogen. Die vertikale Geschwindigkeit ist auch auf dem oberen Rand gleich 0.

Als Kantenlänge wählt man 1, wodurch sich für  $\Omega := [0, 1] \times [0, 1]$  ergibt, also ein Einheitsquadrat entsteht.

## 2.3 Anwendung der radialen Basisfunktionen

In diesem Kapitel werden die Anwendungen der divergenzfreien radialen Basisfunktionen basierend auf dem zweidimensionalen „driven-cavity-problem“ untersucht.

Gegeben ist ein rechteckiger Raum mit der horizontalen Strömung am oberen Ende B1, und einer Nullströmung an seinen drei restlichen Seiten B2, B3 und B4. Die Aufgabe ist, die Strömung in dem Raum zu modellieren, nachdem es seinen eingeschwungenen Zustand erreicht hat. Um die Ergebnisse visualisieren zu können, wird MATLAB benutzt.

Man nehme an, die Strömung sei inkompressibel, d.h. sie verändert ihre Dichte beim Druck von Außen nicht. Die Differentialgleichung, welche die inkompressible Materialströmung beschreibt, ist eine Navier-Stokes-Gleichung. Das Material, mit dem gearbeitet wird, ist Luft. Es soll eine numerische Lösung für das „driven-cavity-problem“ gefunden werden, welche die Geschwindigkeit  $\mathbf{u}$  der Navier-Stokes-Gleichung der inkompressiblen

Luftströmung beschreibt. Das führt zum System

$$\begin{cases} \frac{D\mathbf{u}}{Dt} = -\nabla p + \nu \Delta \mathbf{u}, \\ \nabla \cdot \mathbf{u} = 0 \end{cases} \quad (2.3)$$

im Inneren,  $X^I$ , von dem Raum, wo

$$\frac{D\mathbf{u}}{Dt} := \frac{\partial \mathbf{u}}{\partial t} + (\mathbf{u} \cdot \nabla) \mathbf{u}$$

die Materialableitung,  $p$  der Druck und  $\nu$  der Viskositätskoeffizient ist.

Die zweite Gleichung in (2.3) bedeutet, dass der Fluss divergenzfrei ist. Daher wird eine Interpolationsfunktion gesucht, die auf einer divergenzfreien matrixwertigen RBF basiert. Es werden streng positiv definite Funktionen mit kompakten Träger benutzt, weil diese eine schwachbesetzte, symmetrische und positiv definite Interpolationsmatrix liefern, und folglich den Algorithmus beschleunigen.

Um die Eindeutigkeit der Lösung sicherzustellen, sind Randbedingungen erforderlich. Solche sind gegeben als

$$\begin{cases} \mathbf{u} = (1, 0)^T & \text{in } B_1, \\ \mathbf{u} = (0, 0)^T & \text{in } B_2, B_3, B_4, \end{cases} \quad (2.4)$$

welche bedeuten, dass man eine horizontale Strömung im oberen Bereich des Raumes hat, und keine Strömung an den drei übrigen Rändern.

### 2.3.1 Der konstante Druck

Man betrachte zuerst die vereinfachte Version der Navier-Stokes-Gleichung in zwei Dimensionen.

Folgende zwei Annahmen werden gemacht:

1. Das Strömungsproblem ist zeitunabhängig,
2. Der Strömungsfluss hat einen konstanten Druck.

Die Annahmen 1 und 2 setzen die Gleichungen  $\frac{\partial \mathbf{u}}{\partial t} = 0$  und  $\nabla p = 0$  voraus. Eine gemischte Formulierung der inkompressiblen Navier-Stokes-Gleichung führt zu einer verschwindenden Druckgröße.

Mit den Annahmen 1 und 2 kann das Problem (2.3) folgendermaßen umformuliert werden:

Finde  $\mathbf{u} = (u, v)^T$  aus dem divergenzfreien Ansatzraum, so dass

$$\begin{cases} \nu \Delta \mathbf{u} = (\mathbf{u} \cdot \nabla) \mathbf{u} & \text{in } X^I, \\ \mathbf{u} = (1, 0)^T & \text{in } B_1, \\ \mathbf{u} = (0, 0)^T & \text{in } B_2, B_3, B_4, \end{cases} \quad (2.5)$$

wobei  $\nu$  der Viskositätskoeffizient ist.

In (2.5) braucht man dabei die Voraussetzung

$$\nabla \cdot \mathbf{u} = 0$$

nicht, da die Interpolante diese Bedingung selbstverständlich erfüllt.

Das System (2.5) basiert nicht-linear auf dem Ausdruck  $(\mathbf{u} \cdot \nabla)\mathbf{u}$  aus der ersten Gleichung. Daher wird eine iterative Methode benutzt, um das „driven-cavity-problem“ numerisch zu lösen.



## 3 Numerische Grundlagen des Verfahrens

Es gibt viele lokale und globale Methoden, um eine Funktion mit mehreren Variablen zu approximieren [8]: multivariate Polynome, Splines oder Tensorproduktmethoden. Alle diese Ansätze haben viele Vorteile und auch einige Nachteile. Wenn die Anzahl der Variablen aber groß ist, was in vielen Anwendungen von der Statistik bis hin zu neuronalen Netzwerken oft der Fall ist, reduzieren sich die Auswahlmöglichkeiten schnell, bis nur noch die Tensorproduktmethoden zur Verfügung stehen. Die Daten können manchmal auch verstreut sein, was eine Anwendung der Tensorproduktmethoden unmittelbar ausschließt. Außerdem benötigen Tensorproduktmethoden in höheren Dimensionen immer viele, manchmal zu viele Informationen. In solchen Situationen ist die Wahl der Methode mit einer Annäherung der radialen Basisfunktion eine sehr gute Lösung. Die Methode der radialen Basisfunktionen liefert sowohl bei niedrigeren, als auch bei höheren Dimensionen immer gute Ergebnisse. Aufgrund ihrer ausgezeichneten Approximationseigenschaften ist sie auch eine gute Alternative zu bruchstückhaften Polynomen.

### 3.1 Radiale Basisfunktionen

Methoden, die auf radialen Basisfunktionen basieren, werden oft zur Approximation von multivariaten Funktionen benutzt, besonders, wenn Informationen über das Gitter fehlen. Radiale Basisfunktionen werden bereits seit mehreren Jahren untersucht und getestet.

**Definition 3.1** Eine stetige Funktion  $\Phi : \mathbb{R}^d \rightarrow \mathbb{R}$  mit höchstem polynomialem Wachstum im Unendlichen heißt *radiale Basisfunktion*, in Zeichen RBF, falls  $\Phi$  im Sinne von  $\Phi(x) = \phi(\|x\|)$  mit einer reellwertigen Funktion  $\phi : \mathbb{R}_{\geq 0} \rightarrow \mathbb{R}$  erklärt ist.

Die am häufigsten betrachteten radialen Basisfunktionen [7] sind

$$\begin{aligned}\Phi_1(x) &= e^{-\lambda\|x\|^2}, \lambda > 0, && \text{(Gaussglocken),} \\ \Phi_2(x) &= (-1)^{(k+1)}\|x\|^{2k}\log\|x\|, k \in \mathbb{N}, && \text{(Thin – Plate – Splines),} \\ \Phi_3(x) &= (-1)^{\lceil \frac{\nu}{2} \rceil}\|x\|^\nu, \nu > 0, \nu \notin 2\mathbb{N}, && \text{(3.1)} \\ \Phi_4(x) &= (-1)^{\lceil \frac{\nu}{2} \rceil}(c^2 + \|x\|^2)^{\frac{\nu}{2}}, \nu, c > 0, \nu \notin 2\mathbb{N}, && \text{(Multiquadrics),} \\ \Phi_5(x) &= (c^2 + \|x\|^2)^{(-\frac{\nu}{2})}, \nu, c > 0, && \text{(Inverse Multiquadrics).}\end{aligned}$$

### 3.1.1 Wendland-Funktionen

Wendland-Funktionen [3] sind neue radiale Basisfunktionen mit einem kompakten Träger. Diese Funktionen sind bruchstückhaft polynomial, positiv definit und besitzen einen minimalen Grad in Abhängigkeit von Glätte und Raumdimension.

Für den Raum aller positiven definiten Funktionen auf dem  $\mathbb{R}^d$  schreibe man  $PD_d$ . Mit

$$\psi_\ell(r) := (1 - r)_+^\ell$$

bezeichnet man die bekannte abgeschnittene Potenzfunktion. Sie ist definiert als

$$(1 - r)_+^\ell := \begin{cases} (1 - r)^\ell, & 0 \leq r \leq 1 \\ 0, & r \geq 1. \end{cases}$$

Beachte, dass für alle Wendland-Funktionen,  $r = \|x\|$  gilt. Nach dem Theorem 3.1 aus [3] existiert für jede Raumdimension  $d \in \mathbb{N}$  und jedes  $k \in \mathbb{N}$  eine Funktion  $\psi_{\ell,k}$  mit  $\ell = \lfloor d/2 \rfloor + 3k + 1$  und  $\psi_{\ell,k} \in PD_d \cap C^{2k}$ .

Der Operator  $\mathcal{I}$  sei definiert durch

$$\mathcal{I}\psi(r) := \begin{cases} \int_r^\infty s\psi(s)ds, & r > 0 \\ \mathcal{I}\psi(-r), & r < 0. \end{cases}$$

Eine Wendland-Funktion wird durch

$$\psi_{\ell,k}(r) := \mathcal{I}^k \psi_\ell(r)$$

beschrieben, wobei mit  $k \in \mathbb{N}_0$  die Glätte, und mit  $\ell \in \mathbb{R}_{>0}$  der Grad der Wendland-Funktion gemeint sind.

Tabelle 3.1 beinhaltet eine Liste von verschiedenen Wendland-Funktionen. Sie ist aus [3] entnommen. Dabei bedeutet  $\doteq$  Gleichheit bis auf Konstanten. In der Abbildung (3.1) ist als Beispiel die Wendland-Funktion  $\psi_{4,2}(r) = (1 - r)_+^6 (35r^2 + 18r + 3)$  dargestellt.

### 3.2 Interpolation mit radialen Basisfunktionen

Ein Teilgebiet der multivariaten Interpolation ist die Interpolation mit radialen Basisfunktionen [9]. Dass die Basisfunktionen radial sind, bedeutet, ihr Wert hängt an einer vorgegebenen Stelle  $x$  nur von der euklidischen Norm  $\|x\|$  von  $x$  ab. Das weist vor allem darauf hin, dass eine univariate Funktion als Funktion in der Norm, als radiale Basisfunktion in jeder vorgegebenen Raumdimension dienen kann. Der folgende Abschnitt beschreibt die multivariate Interpolation mit radialen Basisfunktionen. Die hier benutzten radialen Basisfunktionen sind quadratisch und matrixwertig. Betrachtet wird ein Raum der Dimension  $d = s$ .

d=1	$\psi_{1,0}(r) = (1-r)_+$	$C^0$
	$\psi_{2,1}(r) \doteq (1-r)_+^3(3r+1)$	$C^2$
	$\psi_{3,2}(r) \doteq (1-r)_+^5(8r^2+5r+1)$	$C^4$
d=3	$\psi_{2,0}(r) = (1-r)_+^2$	$C^0$
	$\psi_{3,1}(r) \doteq (1-r)_+^4(4r+1)$	$C^2$
	$\psi_{4,2}(r) \doteq (1-r)_+^6(35r^2+18r+3)$	$C^4$
	$\psi_{5,3}(r) \doteq (1-r)_+^8(32r^3+25r^2+8r+1)$	$C^6$
d=5	$\psi_{3,0}(r) = (1-r)_+^3$	$C^0$
	$\psi_{4,1}(r) \doteq (1-r)_+^5(5r+1)$	$C^2$
	$\psi_{5,2}(r) \doteq (1-r)_+^7(16r^2+7r+1)$	$C^4$

Tabelle 3.1: Verschiedene Wendland-Funktionen

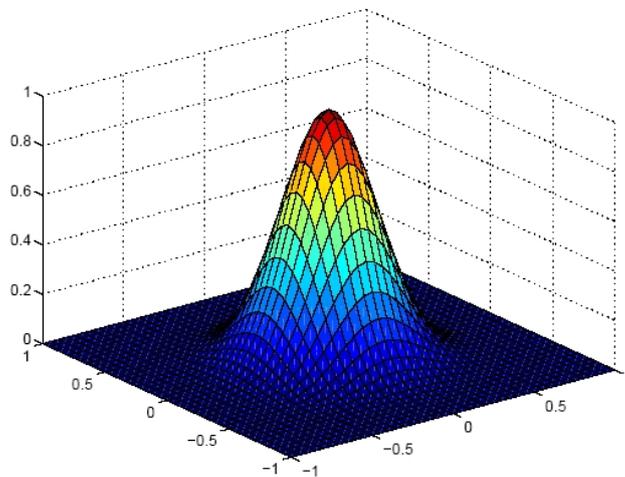


Abbildung 3.1:  $\psi_{4,2}(r) = (1-r)_+^6(35r^2+18r+3)$

Die allgemeine Form der  $s \times s$ -matrixwertigen radialen Basisfunktion  $\Phi_{\ell,k}$  kann geschrieben werden als

$$\Phi_{\ell,k}(x) = \{-\Delta I + \nabla \nabla^T\} \psi_{\ell,k}(x),$$

wobei  $\psi$  eine skalarwertige RBF mit kompakten Träger,  $\Delta$  der Laplace-Operator,  $\nabla$  der Gradient und  $I$  die  $s$ -dimensionale Einheitsmatrix ist. Wie in Lemma 3.5 aus [1] bewiesen wurde, sind die Spalten der Funktionen  $\Phi_{\ell,k} = \{-\Delta I + \nabla \nabla^T\} \psi_{\ell,k}$  divergenzfrei, wobei  $\psi_{\ell,k}$  eine skalarwertige Wendland-Funktion mit  $k \geq 2$  ist.

**Definition 3.2** Eine stetige Funktion  $\psi : \mathbb{R}^d \rightarrow \mathbb{R}$  heißt *bedingt positiv definit der Ordnung  $m$* , in Zeichen  $\Phi \in BPD(m)$ , falls für jede endliche Menge  $X = \{x_1, \dots, x_N\}$ , bestehend aus paarweise verschiedenen Punkten  $x_j$  im  $\mathbb{R}^d$ , und beliebige reelle Zahlen  $\alpha_1, \dots, \alpha_N \in \mathbb{R}$  gilt:

$$\sum_{j=1}^N \alpha_j \psi(x - x_j) \geq 0.$$

Dabei soll Gleichheit nur im Fall  $\alpha_1 = \dots = \alpha_N = 0$  auftreten. Eine bedingt positiv definite Funktion der Ordnung Null nennt man auch eine *positiv definite Funktion*.

Alle in (3.1) vorgestellten Funktionen sind (bedingt) positiv definit.

Sei  $\Omega$  eine kompakte Teilmenge von  $\mathbb{R}^s$ ,  $X = \{x_1, \dots, x_N\} \subset \Omega \subset \mathbb{R}^s$  paarweise verschiedene Punkte und  $f$  eine Funktion,  $f : \mathbb{R}^s \rightarrow \mathbb{R}$ . In diesem Fall, wenn die zu interpolierenden Daten die Form  $\{x_j, f_j\}_{j=1}^N$  annehmen, und wenn  $\{f_j\}_{j=1}^N \in \mathbb{R}^N$  Datensätze sind, die aus der Funktion  $f$  an gegebenen Punkten  $X$  stammen, also  $f_j := f(x_j)$ , hat die divergenzfreie Interpolationsfunktion  $s_{f,X}$  die Form [6]

$$s_{f,X}(x) = \sum_{j=1}^N \alpha_j \psi(x - x_j).$$

Dabei nennt man  $\psi$  Basisfunktion. Lässt sich  $\psi(x) = \phi(\|x\|)$  schreiben mit einem  $\phi : \mathbb{R}_{\geq 0} \rightarrow \mathbb{R}$  und der euklidischen Norm  $\|\cdot\|$ , dann ist  $\psi$  eine radiale Basisfunktion.

Das Interpolationsproblem besteht darin, ein  $\{\alpha_j\}_{j=1}^N$  zu finden, so dass

$$s_{f,X}(x_j) = f_j$$

für alle  $1 \leq j \leq N$ .

Dieses Problem kann umformuliert werden zu einem linearen  $2N \times 2N$  - Gleichungssystem

$$\sum_{j=1}^N \alpha_j \psi(x_k - x_j) = f_k,$$

für alle  $1 \leq j, k \leq N$ . Die entstehende Interpolationsmatrix  $A$  ist positiv definit, symmetrisch und schwach besetzt [1]. Deswegen können schnelle Löser eingesetzt werden, um eine einheitliche Lösung zu bekommen.

Das Gleichungssystem sieht in der Matrixschreibweise dargestellt folgendermaßen aus:

$$A\alpha = f,$$

mit der Interpolationsmatrix

$$A_{\psi, X} = A = \begin{pmatrix} \psi(x_1 - x_1) & \psi(x_1 - x_2) & \dots & \psi(x_1 - x_N) \\ \psi(x_2 - x_1) & \psi(x_2 - x_2) & \dots & \psi(x_2 - x_N) \\ \vdots & \vdots & \ddots & \vdots \\ \psi(x_N - x_1) & \psi(x_N - x_2) & \dots & \psi(x_N - x_N) \end{pmatrix}$$

und  $\alpha = (\alpha_1, \dots, \alpha_N)^T, f = (f_1, \dots, f_N)^T$ . Das Gleichungssystem  $A\alpha = f$  ist für beliebige  $f = (f_1, \dots, f_N)^T$  lösbar, wenn die Matrix  $A$  nicht singular ist, d.h. wenn die Determinante von  $A$  nicht verschwindet. Dieses Kriterium ist erfüllt, da die Matrix  $A$  positiv definit ist.

In dieser Arbeit werden nur Basisfunktionen mit einem kompakten Träger benutzt. Deren Vorteil ist, dass sie schwachbesetzte Interpolationsmatrizen produzieren und dadurch viel schnellere Rechnungen ermöglichen.

### 3.3 Die grundlegende numerische Methode

Im folgenden Unterkapitel soll das eigentliche numerische Problem dargestellt werden. Zuerst konstruiert man eine Eingangsinterpolante  $s_0$ , welche die Strömung nur an den vier Rändern  $B_1$  bis  $B_4$  interpoliert. Um den Raum zu definieren wählt man als Interpolationspunkte ein regelmäßiges Gitter auf dem Raum. Dass die Daten auf dem Rand von einer einfachen Punktauswertung stammen, spiegelt sich in der Interpolanten  $s_0$  wieder. Die Anfangsinterpolante hat die Form

$$s_0(x) = \sum_{k=1}^{N^B} \alpha_{0,k}^B \Phi(x - x_k^B).$$

Daraus soll  $\alpha_0$  so bestimmt werden, dass

$$s_0(x_l^B) = d_l^B$$

für  $x_l^B \in X^B$  der Randdaten  $D^B = \{d_l^B\}_{l=1}^{N^B}$  auf den Rändern  $X^B = \{x_l^B\}_{l=1}^{N^B}$  des Raumes gilt, wie in (2.5) dargestellt. Dabei ist  $B$  die Vereinigung aller vier Ränder,  $N^B$  die Anzahl der Interpolationspunkte auf dem Rand.

Die Interpolante  $s_0$  besteht nur aus den Umrechnungen der Ausdrücke der radialen Basisfunktionen. Daher kann das umgeschrieben werden als eine Linearkombination der Faltungen der Deltafunktionale mit radialen Basisfunktionen, d.h.

$$s_0(x) = \sum_{k=1}^{N^B} \alpha_{0,k}^B (\delta_{x_k^B} * \Phi)(x),$$

weil das Anfangsinterpolationsproblem nur aus Daten besteht, die aus Punktauswertungen entstanden sind.

Als nächstes definiert man eine Interpolante  $s_r$ , welche die Strömung sowohl an den Rändern, als auch im Inneren interpoliert. Im Inneren sind nicht nur die Daten, die von Punktauswertungen stammen, sondern auch die Information über Ableitungen gegeben, was sich auch in der Interpolante  $s_r$  widerspiegelt.

Die iterative Interpolation der Navier-Stokes-Gleichung kann folgendermaßen beschrieben werden:

Für  $r = 1 : nsteps$

Nimm die Interpolante

$$s_r(x) = \sum_{i=1}^{N^I} \alpha_{r,i}^I \Delta \Phi(x - x_i^I) + \sum_{k=1}^{N^B} \alpha_{r,k}^B \Phi(x - x_k^B),$$

finde den Koeffizienten  $\alpha_r$ , so dass

$$\begin{cases} (v\Delta - (s_{r-1}(x_j^I) \cdot \nabla))s_r(x_j^I) = 0 & \text{for } x_j^I \in X^I, \\ s_r(x_\ell^B) = d_\ell^B & \text{for } x_\ell^B \in X^B, \end{cases}$$

wobei  $d_\ell^B$  in (2.5) gegeben ist.

end

Da die iterative Interpolante aus Ausdrücken besteht, die einerseits vom Laplace-Operator stammen, der auf radiale Basisfunktionen angewandt wurde, und andererseits aus Ausdrücken, die aus Übertragungen der radialen Basisfunktionen entstanden sind, kann sie umgeschrieben werden als

$$s_r(x) = \sum_{i=1}^{N^I} \alpha_{r,i}^I (\Delta \delta_{x_i^I} * \Phi)(x) + \sum_{k=1}^{N^B} \alpha_{r,k}^B (\delta_{x_k^B} * \Phi)(x).$$

Der Grund für die Wahl der Interpolante ist, dass beim Problem der iterativen Interpolation Daten interpoliert werden, die sowohl aus den Punktauswertungen am Rand stammen, als auch Laplace-Daten aus dem Inneren des Raumes. Die Iteration ist vollendet, wenn ein stabiler Zustand erreicht wird, d.h. wenn die Differenz von zwei aufeinander folgenden Interpolanten sehr klein wird.

### 3.3.1 Der normale Druck

Es wird jetzt die normale Form der Navier-Stokes-Gleichung untersucht. Man leitet eine Interpolationsmethode her für den Fall, dass der Druck nicht konstant ist. Es wird immer noch angenommen, dass das Problem zeitunabhängig ist. Folglich hat man nur eine Annahme:

1. Das Strömungsproblem ist zeitunabhängig,

d.h. es wird auf die Annahme vom konstanten Druck aus dem vorherigen Kapitel verzichtet. Das Problem, das sich aus der Navier-Stokes Gleichung ergibt, ist das folgende:

Finde  $\mathbf{u} = (u, v)^T$ , so dass

$$\begin{cases} (v \Delta - (\mathbf{u} \cdot \nabla))\mathbf{u} = \nabla p & \text{in } X^I, \\ \mathbf{u} = (1, 0)^T & \text{in } B_1, \\ \mathbf{u} = (0, 0)^T & \text{in } B_2, B_3, B_4. \end{cases} \quad (3.2)$$

Dabei ist der Laplace-Operator definiert als [5]

$$\Delta := \sum_{i=1}^s \partial_{x^{(i)}}^2$$

und der Gradient ist gegeben als

$$\nabla := (\partial_{x^{(1)}}, \dots, \partial_{x^{(s)}})^T. \quad (3.3)$$

Die Anwendung des Gradienten als Skalarprodukt auf die erste Gleichung von (3.2) ergibt

$$\begin{aligned} \Delta p &= \nabla \cdot [v \Delta \mathbf{u} - (\mathbf{u} \cdot \nabla)\mathbf{u}] \\ &= \underbrace{\nabla \cdot [v \Delta \mathbf{u}]}_0 - \nabla \cdot [(\mathbf{u} \cdot \nabla)\mathbf{u}] \\ &= -\nabla \cdot [(\mathbf{u} \cdot \nabla)\mathbf{u}] \end{aligned} \quad (3.4)$$

im Inneren des Raumes. Der erste Ausdruck verschwindet, weil  $\mathbf{u}$  divergenzfrei ist. Um eine eindeutige Lösung für den Druck zu bekommen, leitet man Randbedingungen her. Dabei ist die erste Gleichung von (3.2) äquivalent zu

$$\begin{pmatrix} \partial_1 p \\ \partial_2 p \end{pmatrix} = \begin{pmatrix} v(\partial_1^2 + \partial_2^2)u - (u\partial_1 u + v\partial_2 u) \\ v(\partial_1^2 + \partial_2^2)v - (u\partial_1 v + v\partial_2 v) \end{pmatrix} \quad (3.5)$$

in zwei Dimensionen, wobei  $\mathbf{u} = (u, v)^T$ .

Deshalb sind Neumann-Bedingungen an den vier Rändern gegeben als

$$\begin{cases} \partial_2 p = v \partial_2^2 v & \text{in } B1, \\ \partial_1 p = v \partial_1^2 u & \text{in } B2, \\ \partial_1 p = -v \partial_1^2 u & \text{in } B3, \\ \partial_2 p = -v \partial_2^2 v & \text{in } B4, \end{cases} \quad (3.6)$$

basierend auf den Randwerten, die in (3.2) formuliert sind. Die numerische Methode ist daher für die Lösung des Druckes  $p$  wie folgt gegeben:

Gegeben sind  $s_{r-1}$  und  $s_r = (u_r, v_r)^T$ . Finde  $p$ , so dass

$$\begin{cases} (v \Delta - (s_{r-1} \cdot \nabla)) s_r = \nabla p & \text{in } X^I, \\ \partial_2 p = \pm v \partial_2^2 v_r & \text{in } B1, B4, \\ \partial_1 p = \pm v \partial_1^2 u_r & \text{in } B2, B3. \end{cases} \quad (3.7)$$

Die skalarwertige Interpolante zur Berechnung des Druckes wird in der Doktorarbeit von Frau Svenja Lowitzsch [1] definiert als

$$p(x) := \sum_{s=1}^{N^h} \beta_{r,s}^h \partial_2 g(x - x_s^h) + \sum_{t=1}^{N^v} \beta_{r,t}^v \partial_1 g(x - x_t^v) + \sum_{k=1}^{N^I} \beta_{r,k}^I \Delta g(x - x_k^I), \quad (3.8)$$

wobei  $x^h \in B1 \cup B4$  und  $x^v \in B2 \cup B3$  den horizontalen und vertikalen Rändern entsprechen.

Es gibt aber auch eine andere Möglichkeit die Druckberechnung zu definieren. Diese besteht darin, dass man den Laplace-Operator von der Summe

$$\sum_{k=1}^{N^I} \beta_{r,k}^I \Delta g(x - x_k^I)$$

zur Berechnung von  $p(x)$  an den inneren Punkten von  $X$  ersetzt durch

$$\sum_{k=1}^{N^I} \beta_{r,k}^I \partial_1 g(x - x_k^I) + \sum_{k=1}^{N^I} \beta_{r,k}^I \partial_2 g(x - x_k^I).$$

Also wird die Interpolante zur Berechnung des Druckes definiert als

$$p(x) := \sum_{s=1}^{N^h} \beta_{r,s}^h \partial_2 g(x - x_s^h) + \sum_{t=1}^{N^v} \beta_{r,t}^v \partial_1 g(x - x_t^v) + \sum_{k=1}^{N^I} \beta_{r,k}^I \partial_1 g(x - x_k^I) + \sum_{k=1}^{N^I} \beta_{r,k}^I \partial_2 g(x - x_k^I), \quad (3.9)$$

wobei  $x^h \in B1 \cup B4$  und  $x^v \in B2 \cup B3$  den horizontalen und vertikalen Rändern entsprechen. Dieser Ansatz ist so gewählt, dass sich im Folgenden ein symmetrisches Hermite-Birkhoff-Interpolationsproblem [15] ergibt.

### 3.4 Algorithmus

Es soll eine numerische Methode entwickelt werden, um die Navier-Stokes-Gleichung gleichzeitig für den Druck und den Strömungsfluss zu lösen. Sie basiert auf zwei Iterationsschleifen. Die erste Schleife wird herangezogen, um eine erste Lösung für die Strömung zu bekommen, die auf einem konstanten Druck basiert, eine zweite Schleife, um eine Lösung für den Druck zu erhalten, gefolgt von einer Interpolation der Strömung, die diese Druckfunktion erzeugt, bis die zweite Schleife ihren stabilen Zustand erreicht. Dabei wählt man die skalarwertige  $C^8$ -Wendland-RBF, da die Implementierung des Druckes in der Interpolationsmethode damit endet, dass zwei weitere Ableitungen verwendet werden müssen.

Wie im vorherigen Abschnitt erläutert, kann die Berechnung der Druckfunktion auf zwei unterschiedliche Weisen erfolgen. In der vorliegenden Arbeit werden beide Möglichkeiten implementiert. Der Algorithmus I beinhaltet die ursprünglich vorgesehene Definition, die in der Gleichung (3.8) gegeben ist. Beim Algorithmus II ist die Druckfunktion  $p(x)$  wie in der Gleichung (3.9) definiert.

#### 3.4.1 Algorithmus I

i) Finde den Koeffizientenvektor  $\alpha_0$ , so dass

$$s_0(x) = \sum_{k=1}^{N^B} \alpha_{0,k}^B \Phi(x - x_k^B)$$

löst

$$s_0(x_\ell^B) = d_\ell^B$$

für  $x_\ell^B \in X^B$  für die Randdaten  $D^B = \{d_\ell^B\}_{\ell=1}^{N^B}$ , wie in dem System (2.5)

ii) Für  $r = 1 : nsteps$

Finde den Koeffizientenvektor  $\alpha_r$ , so dass

$$s_r(x) = \sum_{i=1}^{N^I} \alpha_{r,i}^I \Delta \Phi(x - x_i^I) + \sum_{k=1}^{N^B} \alpha_{r,k}^B \Phi(x - x_k^B)$$

löst

$$\begin{cases} (v \Delta - (s_{r-1}(x_j^I) \cdot \nabla)) s_r(x_j^I) = 0 & \text{for } x_j^I \in X^I, \\ s_r(x_\ell^B) = d_\ell^B & \text{for } x_\ell^B \in X^B, \end{cases}$$

wobei  $d_\ell^B$  in (2.5) gegeben ist.

**end**

iii) Finde den Koeffizientenvektor  $\beta_r$ , so dass

$$p(x) := \sum_{s=1}^{N^h} \beta_{m,s}^h \partial_2 g(x - x_s^h) + \sum_{t=1}^{N^v} \beta_{m,t}^v \partial_1 g(x - x_t^v) + \sum_{k=1}^{N^I} \beta_{m,k}^I \Delta g(x - x_k^I)$$

löst

$$\begin{cases} \nabla p(x_j^I) = (v \Delta - (s_{nsteps-1}(x_j^I) \cdot \nabla)) s_{nsteps}(x_j^I) & \text{for } x_j^I \in X^I, \\ \partial_2 p(x_\ell^h) = \pm v \partial_2^2 v_{nsteps}(x_\ell^h) & \text{for } x_\ell^h \in B1, B4, \\ \partial_1 p(x_i^v) = \pm v \partial_1^2 u_{nsteps}(x_i^v) & \text{for } x_i^v \in B2, B3, \end{cases}$$

wobei  $x^h$  und  $x^v$  den horizontalen und vertikalen Rändern entsprechen.

iv) Für  $r = 1 : nsteps$

Finde den Koeffizientenvektor  $\alpha_r$ , so dass  $s_r(x)$ , das definiert ist wie in Schritt ii), löst

$$\begin{cases} (v \Delta - (s_{r-1}(x_j^I) \cdot \nabla)) s_r(x_j^I) = \nabla p(x_j^I) & \text{for } x_j^I \in X^I, \\ s_r(x_\ell^B) = d_\ell^B & \text{for } x_\ell^B \in X^B, \end{cases}$$

wobei  $d_\ell^B$  in (2.5) gegeben ist.

end

v) Wiederhole Schritte iii) und iv), bis sowohl für den Strömungsfluss, als auch für den Druck eine stabile Lösung gefunden ist.

### 3.4.2 Algorithmus II

i) Finde den Koeffizientenvektor  $\alpha_0$ , so dass

$$s_0(x) = \sum_{k=1}^{N^B} \alpha_{0,k}^B \Phi(x - x_k^B)$$

löst

$$s_0(x_\ell^B) = d_\ell^B$$

für  $x_\ell^B \in X^B$  für die Randdaten  $D^B = \{d_\ell^B\}_{\ell=1}^{N^B}$ , wie in dem System (2.5)

ii) Für  $r = 1 : nsteps$

Finde den Koeffizientenvektor  $\alpha_r$ , so dass

$$s_r(x) = \sum_{i=1}^{N^I} \alpha_{r,i}^I \Delta \Phi(x - x_i^I) + \sum_{k=1}^{N^B} \alpha_{r,k}^B \Phi(x - x_k^B)$$

löst

$$\begin{cases} (v \Delta - (s_{r-1}(x_j^I) \cdot \nabla))s_r(x_j^I) = 0 & \text{for } x_j^I \in X^I, \\ s_r(x_\ell^B) = d_\ell^B & \text{for } x_\ell^B \in X^B, \end{cases}$$

wobei  $d_\ell^B$  in (2.5) gegeben ist.

**end**

**iii)** Finde den Koeffizientenvektor  $\beta_r$ , so dass

$$p(x) := \sum_{s=1}^{N^h} \beta_{m,s}^h \partial_2 g(x - x_s^h) + \sum_{t=1}^{N^v} \beta_{m,t}^v \partial_1 g(x - x_t^v) + \sum_{k=1}^{N^I} \beta_{m,k}^I \partial_1 g(x - x_k^I) + \sum_{k=1}^{N^I} \beta_{m,k}^I \partial_2 g(x - x_k^I)$$

löst

$$\begin{cases} \nabla p(x_j^I) = (v \Delta - (s_{nsteps-1}(x_j^I) \cdot \nabla))s_{nsteps}(x_j^I) & \text{for } x_j^I \in X^I, \\ \partial_2 p(x_\ell^h) = \pm v \partial_2^2 v_{nsteps}(x_\ell^h) & \text{for } x_\ell^h \in B1, B4, \\ \partial_1 p(x_i^v) = \pm v \partial_1^2 u_{nsteps}(x_i^v) & \text{for } x_i^v \in B2, B3, \end{cases}$$

wobei  $x^h$  und  $x^v$  den horizontalen und vertikalen Rändern entsprechen.

Hier wirkt sich der Ansatz (3.9) so aus, dass sich ein symmetrisches System ergibt.

**iv) Für  $r = 1 : nsteps$**

Finde den Koeffizientenvektor  $\alpha_r$ , so dass  $s_r(x)$ , das definiert ist wie in Schritt ii), löst

$$\begin{cases} (v \Delta - (s_{r-1}(x_j^I) \cdot \nabla))s_r(x_j^I) = \nabla p(x_j^I) & \text{for } x_j^I \in X^I, \\ s_r(x_\ell^B) = d_\ell^B & \text{for } x_\ell^B \in X^B, \end{cases}$$

wobei  $d_\ell^B$  in (2.5) gegeben ist.

**end**

**v)** Wiederhole Schritte iii) und iv), bis sowohl für den Strömungsfluss, als auch für den Druck eine stabile Lösung gefunden ist.



## 4 Implementierung

In diesem Kapitel soll die Implementierung der im vorherigen Kapitel vorgestellten Algorithmen erläutert werden. Es wird auch erklärt, welche Möglichkeiten dem Benutzer bei der Bedienung des Programms zur Verfügung stehen.

### 4.1 Modulaufbau und Initialisierung

Das implementierte Programm ist ein numerisches Verfahren zur Lösung von Strömungsaufgaben mit divergenzfreien Ansatzräumen. Das Programm besteht, wie üblich, aus einem Hauptmodul und mehreren kleinen Modulen, welche ausgelagerte Hilfsfunktionen enthalten. Weiterhin beinhaltet es einen Input-File, in dem alle Eingabeparameter gespeichert werden. Der Benutzer hat damit die Möglichkeit, die einzelnen Parameter seinen Wünschen entsprechend zu setzen. Das ist eine Auflistung aller Parameter:

1.  $\nu$  - der Viskositätskoeffizient des Fluids,
2.  $\ell$  - die Länge des Raumes,
3.  $d$  - die Tiefe des Raumes,
4.  $hsteps_x$  - die Anzahl der Schritte in x-Richtung,
5.  $hsteps_y$  - die Anzahl der Schritte in y-Richtung,
6.  $nsteps$  - die Anzahl der Iterationsschritte.

Mit der Anzahl der Schritte in  $x$ - und  $y$ -Richtung ist gemeint, in wieviele Teile die Länge und die Tiefe des Raumes jeweils aufgeteilt werden soll. Die Schrittgröße ist dann

$$\begin{aligned} dx &= \ell / hsteps_x \\ dy &= d / hsteps_y \end{aligned}$$

Mit diesen Eingabeparametern werden anschließend in den Hauptprogrammen regelmäßige Gitter  $[X, Y]$  mit den Eckpunkten  $(0, 0)$ ,  $(\ell, 0)$ ,  $(0, -d)$  und  $(\ell, -d)$  angelegt. Die Größe der beiden Gitter ist jeweils  $m \times n$ . Weiterhin werden zwei Gitter  $XI$  und  $YI$  erzeugt, mit den jeweils inneren Punkten von  $X$  und  $Y$ . Diese haben dann dementsprechend die Größe  $(m - 2) \times (n - 2)$ . Die Randwerte der Ränder  $B1$  bis  $B4$  werden zu den Vektoren  $XB_{vec}$  und  $YB_{vec}$  zusammengefasst. Deren Länge ist jeweils  $(2m) + (2n - 2)$ .

## 4.2 Die einzelnen Schritte der Algorithmen

Die implementierten Algorithmen, die im vorherigen Kapitel vorgestellt wurden, bestehen jeweils aus vier wichtigen Schritten. Was in den einzelnen Schritten genau passiert, wird in den folgenden Kapiteln genauer dargestellt.

### 4.2.1 Der erste Schritt

Im ersten Schritt wird der Startwert  $s_0(x)$  für alle Randwerte berechnet. D.h. es wird zuerst der Koeffizientenvektor  $\alpha_0$  bestimmt:

$$\alpha_{0,k}^B = \Phi(x - x_k^B) \setminus D^B$$

für alle  $k = 1, \dots, N^B$ , wobei  $N^B$  für die Menge aller Randpunkte steht.

Die Menge der Randwerte  $D^B$  ist definiert als  $D^B = \{d_\ell^B\}_{\ell=1}^{N^B}$ , wie in dem System (2.5) für die Randwerte B1 bis B4, d.h.  $\mathbf{u} = (u, v)^T$ , so dass

$$\begin{cases} \mathbf{u} = (1, 0)^T & \text{in B1 und} \\ \mathbf{u} = (0, 0)^T & \text{in B2, B3, B4} \end{cases}$$

Nachdem der Vektor  $\alpha_0$  bestimmt wurde, kann nun auch der Strömungsvektor  $s_0^B(x)$  für die Randpunkte berechnet werden, also

$$s_0^B(x) = \sum_{k=1}^{N^B} \alpha_{0,k}^B \Phi(x - x_k^B).$$

Für die Berechnung der weiteren Schritte muss auch noch

$$s_0(x) = \sum_{k=1}^N \alpha_{0,k} \Phi(x - x_k)$$

berechnet werden. Dadurch erhält man die nötigen Startvektoren  $\alpha_0$  und  $s_0(x)$  für die nachfolgende Interpolation.

### 4.2.2 Der zweite Schritt

Im zweiten Schritt wird eine *for*-Schleife *nsteps*-mal durchlaufen. Dabei wird für jedes  $r = 1, \dots, nsteps$  der Koeffizientenvektor  $\alpha_r$ , und daraufhin der Strömungsvektor  $s_r(x)$  berechnet. Es wird über alle Gitterpunkte interpoliert. Um den Koeffizientenvektor  $\alpha_r$  zu bestimmen, muss das Gleichungssystem  $A\alpha_r = b$  gelöst werden. Die Matrix  $A$  stellt die

linke Seite der beiden Bedingungen, für die inneren und äußeren Werte von  $X$ , dar. Die Bedingung für die inneren Punkte ist definiert als

$$(v\Delta - (s_{r-1}(x_j^I) \cdot \nabla))s_r(x_j^I) = v\Delta s_r(x_j^I) - s_{r-1}(x_j^I) \cdot \nabla s_r(x_j^I). \quad (4.1)$$

Dabei ist der Gradient nach (3.3) definiert als

$$\nabla := (\partial_{x(1)}, \dots, \partial_{x(s)})^T.$$

Der Strömungsvektor  $s_r(x)$  ist gegeben als

$$s_r(x) = \sum_{i=1}^{N^I} \alpha_{r,i}^I \Delta \Phi(x - x_i^I) + \sum_{k=1}^{N^B} \alpha_{r,k}^B \Phi(x - x_k^B)$$

Zuerst wird die erste Summe für die inneren Punkte zusammengestellt, d.h.

$$\begin{aligned} A_1 &= v\Delta s_r(x_j^I) - (s_{r-1}(x_j^I) \cdot \nabla s_r(x_j^I)) \\ &= v\Delta^2 \Phi(x_j^I) - (s_{r-1}(x_j^I) \cdot \begin{pmatrix} \partial_1 s_r(x_j^I) \\ \partial_2 s_r(x_j^I) \end{pmatrix}) \\ &= v\Delta^2 \Phi(x_j^I) - (u_{r-1}(x_j^I) \cdot \partial_1 \Delta \Phi(x_j^I) + v_{r-1}(x_j^I) \cdot \partial_2 \Delta \Phi(x_j^I)). \end{aligned}$$

Und dann für die Randpunkte

$$\begin{aligned} A_2 &= v\Delta s_r(x_j^I) - (s_{r-1}(x_j^I) \cdot \nabla s_r(x_j^I)) \\ &= v\Delta \Phi(x_j^I - x_k^B) - (s_{r-1}(x_j^I) \cdot \begin{pmatrix} \partial_1 s_r(x_j^I) \\ \partial_2 s_r(x_j^I) \end{pmatrix}) \\ &= v\Delta \Phi(x_j^I - x_k^B) - (u_{r-1}(x_j^I - x_k^B) \cdot \partial_1 \Phi(x_j^I - x_k^B) + v_{r-1}(x_j^I - x_k^B) \cdot \partial_2 \Phi(x_j^I - x_k^B)). \end{aligned}$$

Als Bedingung für die äußeren Punkte von  $X$  ist das Strömungsfeld  $s_r(x_\ell^B)$  gegeben. Auch dieses Feld wird in zwei Matrizen aufgeteilt. Die erste Matrix stellt die Summe von den inneren Punkten von  $s_r(x)$  dar:

$$A_3 = \Delta \Phi(x_\ell^B - x_i^I),$$

wobei  $\ell = 1, \dots, N^B$  und  $i = 1, \dots, N^I$ . Die zweite Matrix stellt die zweite Summe von den Randpunkten von  $s_r(x)$  dar:

$$A_4 = \Phi(x_\ell^B), \quad \ell = 1, \dots, N^B.$$

Die vier einzelnen Matrizen  $A_1$  bis  $A_4$  werden zum Schluß zu der gemeinsamen Matrix  $A$  zusammengefasst:

$$A = \begin{pmatrix} A_1 & A_2 \\ A_3 & A_4 \end{pmatrix}.$$

Jetzt wird die rechte Seite  $b$  des Gleichungssystems  $A\alpha_r = b$  implementiert. Für alle inneren Punkte wird ein reiner Nullvektor eingetragen. Die Bedingung für die Randpunkte ist gleich der Bedingung aus dem ersten Schritt. Die Ergebnisse werden zum Vektor  $b$  zusammengefasst.

Nun kann das Gleichungssystem  $A\alpha_r = b$  nach dem Koeffizientenvektor  $\alpha_r$  aufgelöst werden. Und wurde  $\alpha_r$  bestimmt, so kann auch der Strömungsvektor  $s_r(x)$  durch das Ausrechnen der Summen bestimmt werden.

### 4.2.3 Der dritte Schritt

Vor dem Beginn der großen äußeren Schleife wird eine Matrix  $A$  gebildet, welche die linke Seite der Bedingungen beschreibt. Diese sieht folgendermaßen aus:

$$A = \begin{pmatrix} \nabla p(x_j^I) \\ \partial_2 p(x_\ell^{B1}) \\ \partial_1 p(x_t^{B2}) \\ \partial_2 p(x_s^{B3}) \\ \partial_2 p(x_t^{B4}) \end{pmatrix},$$

für alle  $x_j^I \in X^I, x_\ell^{B1} \in X^{B1}, x_t^{B2} \in X^{B2}, x_s^{B3} \in X^{B3}$  und  $x_t^{B4} \in X^{B4}$ .

Im Weiteren kann der Gradient von  $p$  in seine beiden Bestandteile zerlegt werden, d.h.

$$A = \begin{pmatrix} \partial_1 p(x_j^I) \\ \partial_2 p(x_j^I) \\ \partial_2 p(x_\ell^{B1}) \\ \partial_1 p(x_t^{B2}) \\ \partial_2 p(x_s^{B3}) \\ \partial_2 p(x_t^{B4}) \end{pmatrix}. \quad (4.2)$$

Jetzt muss die Berechnung der Druckfunktion implementiert werden. Diese Berechnung unterscheidet, wie bereits erwähnt, die Algorithmen I und II.

#### Algorithmus I

Beim Algorithmus I ist die Interpolante  $p(x)$ , welche den Druck darstellt, definiert als

$$\begin{aligned} p(x) &= \sum_{k=1}^{N^I} \beta_k^I \Delta g(x - x_k^I) + \sum_{s=1}^{N^{B1}} \beta_s^{B1} \partial_2 g(x - x_s^{B1}) + \sum_{t=1}^{N^{B2}} \beta_t^{B2} \partial_1 g(x - x_t^{B2}) \\ &+ \sum_{w=1}^{N^{B3}} \beta_w^{B3} \partial_1 g(x - x_w^{B3}) + \sum_{z=1}^{N^{B4}} \beta_z^{B4} \partial_2 g(x - x_z^{B4}). \end{aligned} \quad (4.3)$$

Die Matrix  $A$ , die daraus entsteht, ist nicht symmetrisch.

$$\begin{pmatrix} \partial_1 \Delta g(x^I - x^I) & \partial_1 \partial_2 g(x^I - x^{B1}) & \partial_1 \partial_1 g(x^I - x^{B2}) & \partial_1 \partial_1 g(x^I - x^{B3}) & \partial_1 \partial_2 g(x^I - x^{B4}) \\ \partial_2 \Delta g(x^I - x^I) & \partial_2 \partial_2 g(x^I - x^{B1}) & \partial_2 \partial_1 g(x^I - x^{B2}) & \partial_2 \partial_1 g(x^I - x^{B3}) & \partial_2 \partial_2 g(x^I - x^{B4}) \\ \partial_2 \Delta g(x^{B1} - x^I) & \partial_2 \partial_2 g(x^{B1} - x^{B1}) & \partial_2 \partial_1 g(x^{B1} - x^{B2}) & \partial_2 \partial_1 g(x^{B1} - x^{B3}) & \partial_2 \partial_2 g(x^{B1} - x^{B4}) \\ \partial_1 \Delta g(x^{B2} - x^I) & \partial_1 \partial_2 g(x^{B2} - x^{B1}) & \partial_1 \partial_1 g(x^{B2} - x^{B2}) & \partial_1 \partial_1 g(x^{B2} - x^{B3}) & \partial_1 \partial_2 g(x^{B2} - x^{B4}) \\ \partial_1 \Delta g(x^{B3} - x^I) & \partial_1 \partial_2 g(x^{B3} - x^{B1}) & \partial_1 \partial_1 g(x^{B3} - x^{B2}) & \partial_1 \partial_1 g(x^{B3} - x^{B3}) & \partial_1 \partial_2 g(x^{B3} - x^{B4}) \\ \partial_2 \Delta g(x^{B4} - x^I) & \partial_2 \partial_2 g(x^{B4} - x^{B1}) & \partial_2 \partial_1 g(x^{B4} - x^{B2}) & \partial_2 \partial_1 g(x^{B4} - x^{B3}) & \partial_2 \partial_2 g(x^{B4} - x^{B4}) \end{pmatrix}.$$

An dieser Stelle beginnt die äußere Schleife, welche auch den vierten Schritt einschließt. Es soll die rechte Seite der Bedingungsgleichungen beschrieben werden; Zuerst die Bedingung für die inneren Punkte:

$$(v \Delta - (s_{nsteps-1}(x_j^I) \cdot \nabla)) s_{nsteps}(x_j^I) = v \Delta s_{nsteps}(x_j^I) - s_{nsteps-1}(x_j^I) \cdot \nabla s_{nsteps}(x_j^I). \quad (4.4)$$

Da der Gradient definiert ist als

$$\nabla := (\partial_{x(1)}, \dots, \partial_{x(s)})^T,$$

kann die Bedingung (4.6) der rechten Seite umgeschrieben werden als

$$\begin{aligned} & v \Delta s_{nsteps}(x_j^I) - s_{nsteps-1}(x_j^I) \cdot \nabla s_{nsteps}(x_j^I) \\ &= \begin{pmatrix} v \Delta u_{nsteps}(x_j^I) - (u_{nsteps-1} \cdot \partial_1 u_{nsteps}(x_j^I) + u_{nsteps-1} \cdot \partial_2 u_{nsteps}(x_j^I)) \\ v \Delta v_{nsteps}(x_j^I) - (v_{nsteps-1} \cdot \partial_1 v_{nsteps}(x_j^I) + v_{nsteps-1} \cdot \partial_2 v_{nsteps}(x_j^I)) \end{pmatrix}, \end{aligned}$$

wobei die Strömungsflussinterpolanten  $s_{nsteps}$  und  $s_{nsteps-1}$  aus dem zweiten Schritt bekannt sind.

Nun müssen die Bedingungen 2 und 3 der rechten Seite für die Randpunkte definiert werden. Diese sind, wie in (3.6) angegeben, folgendermaßen festgesetzt

$$\begin{cases} v \partial_2^2 v_{nsteps}(x_\ell^{B1}) \\ v \partial_1^2 u_{nsteps}(x_i^{B2}) \\ -v \partial_1^2 u_{nsteps}(x_s^{B3}) \\ -v \partial_2^2 v_{nsteps}(x_t^{B4}). \end{cases} \quad (4.5)$$

Diese werden, der Reihe nach, alle in einem Vektor  $z$  abgespeichert:

$$z = [v \partial_2^2 v_{nsteps}(x_\ell^{B1}), v \partial_1^2 u_{nsteps}(x_i^{B2}), -v \partial_1^2 u_{nsteps}(x_s^{B3}), -v \partial_2^2 v_{nsteps}(x_t^{B4})].$$

Anschließend werden die Vektoren, welche die erste Bedingung enthalten, und der Vektor  $z$  mit den Randbedingungen ebenso der Reihe nach in einem gemeinsamen Vektor  $b$  vereint.

Danach kann auch der Ergebnisvektor  $\beta_r$  als Lösungsvektor des Gleichungssystems  $A\beta_r = b$  berechnet werden. Und wenn  $\beta$  bekannt ist, kann auch der Druck  $p$  berechnet werden als

$$p(x) := \sum_{k=1}^{N^I} \beta_{m,k}^I \Delta g(x - x_k^I) + \sum_{s=1}^{N^h} \beta_{m,s}^h \partial_2 g(x - x_s^h) + \sum_{t=1}^{N^v} \beta_{m,t}^v \partial_1 g(x - x_t^v).$$

### Algorithmus II

Die Interpolante  $p(x)$ , welche den Druck darstellt, ist nach (3.9) definiert als

$$\begin{aligned} p(x) = & \sum_{k=1}^{N^I} \beta_k^I \partial_1 g(x - x_k^I) + \sum_{k=1}^{N^I} \beta_k^I \partial_2 g(x - x_k^I) + \sum_{s=1}^{N^{B1}} \beta_s^{B1} \partial_2 g(x - x_s^{B1}) \\ & + \sum_{t=1}^{N^{B2}} \beta_t^{B2} \partial_1 g(x - x_t^{B2}) + \sum_{w=1}^{N^{B3}} \beta_w^{B3} \partial_1 g(x - x_w^{B3}) + \sum_{z=1}^{N^{B4}} \beta_z^{B4} \partial_2 g(x - x_z^{B4}). \end{aligned}$$

Das bedeutet, dass die symmetrische Matrix  $A$  folgendermaßen aussieht:

$$\begin{pmatrix} \partial_1 \partial_1 g(x^I - x^I) & \partial_1 \partial_2 g(x^I - x^I) & \partial_1 \partial_2 g(x^I - x^{B1}) & \partial_1 \partial_1 g(x^I - x^{B2}) & \partial_1 \partial_1 g(x^I - x^{B3}) & \partial_1 \partial_2 g(x^I - x^{B4}) \\ \partial_2 \partial_1 g(x^I - x^I) & \partial_2 \partial_2 g(x^I - x^I) & \partial_2 \partial_2 g(x^I - x^{B1}) & \partial_2 \partial_1 g(x^I - x^{B2}) & \partial_2 \partial_1 g(x^I - x^{B3}) & \partial_2 \partial_2 g(x^I - x^{B4}) \\ \partial_2 \partial_1 g(x^{B1} - x^I) & \partial_2 \partial_2 g(x^{B1} - x^I) & \partial_2 \partial_2 g(x^{B1} - x^{B1}) & \partial_2 \partial_1 g(x^{B1} - x^{B2}) & \partial_2 \partial_1 g(x^{B1} - x^{B3}) & \partial_2 \partial_2 g(x^{B1} - x^{B4}) \\ \partial_1 \partial_1 g(x^{B2} - x^I) & \partial_1 \partial_2 g(x^{B2} - x^I) & \partial_1 \partial_2 g(x^{B2} - x^{B1}) & \partial_1 \partial_1 g(x^{B2} - x^{B2}) & \partial_1 \partial_1 g(x^{B2} - x^{B3}) & \partial_1 \partial_2 g(x^{B2} - x^{B4}) \\ \partial_1 \partial_1 g(x^{B3} - x^I) & \partial_1 \partial_2 g(x^{B3} - x^I) & \partial_1 \partial_2 g(x^{B3} - x^{B1}) & \partial_1 \partial_1 g(x^{B3} - x^{B2}) & \partial_1 \partial_1 g(x^{B3} - x^{B3}) & \partial_1 \partial_2 g(x^{B3} - x^{B4}) \\ \partial_2 \partial_1 g(x^{B4} - x^I) & \partial_2 \partial_2 g(x^{B4} - x^I) & \partial_2 \partial_2 g(x^{B4} - x^{B1}) & \partial_2 \partial_1 g(x^{B4} - x^{B2}) & \partial_2 \partial_1 g(x^{B4} - x^{B3}) & \partial_2 \partial_2 g(x^{B4} - x^{B4}) \end{pmatrix}.$$

Auch beim zweiten Algorithmus fängt an dieser Stelle die äußere Schleife an. Es werden wieder die Bedingungsgleichungen der rechten Seite beschrieben; Zuerst die Bedingung für die inneren Punkte:

$$(v \Delta - (s_{nsteps-1}(x_j^I) \cdot \nabla)) s_{nsteps}(x_j^I) = v \Delta s_{nsteps}(x_j^I) - s_{nsteps-1}(x_j^I) \cdot \nabla s_{nsteps}(x_j^I). \quad (4.6)$$

Da der Gradient definiert ist als

$$\nabla := (\partial_{x(1)}, \dots, \partial_{x(s)})^T,$$

kann die Bedingung (4.6) der rechten Seite umgeschrieben werden als

$$\begin{aligned} & v \Delta s_{nsteps}(x_j^I) - s_{nsteps-1}(x_j^I) \cdot \nabla s_{nsteps}(x_j^I) \\ & = \begin{pmatrix} v \Delta u_{nsteps}(x_j^I) - (u_{nsteps-1} \cdot \partial_1 u_{nsteps}(x_j^I) + u_{nsteps-1} \cdot \partial_2 u_{nsteps}(x_j^I)) \\ v \Delta v_{nsteps}(x_j^I) - (v_{nsteps-1} \cdot \partial_1 v_{nsteps}(x_j^I) + v_{nsteps-1} \cdot \partial_2 v_{nsteps}(x_j^I)) \end{pmatrix}, \end{aligned}$$

wobei die Strömungsflussinterpolanten  $s_{nsteps}$  und  $s_{nsteps-1}$  aus dem zweitem Schritt bekannt sind.

Jetzt müssen die Bedingungen 2 und 3 der rechten Seite für die Randpunkte definiert werden. Diese sind, wie in (3.6) angegeben, folgendermaßen festgesetzt

$$\begin{cases} v\partial_2^2 v_{nsteps}(x_\ell^{B1}) \\ v\partial_1^2 u_{nsteps}(x_i^{B2}) \\ -v\partial_1^2 u_{nsteps}(x_s^{B3}) \\ -v\partial_2^2 v_{nsteps}(x_t^{B4}). \end{cases} \quad (4.7)$$

Diese werden nun, der Reihe nach, alle in einem Vektor  $z$  abgespeichert:

$$z = [v\partial_2^2 v_{nsteps}(x_\ell^{B1}), v\partial_1^2 u_{nsteps}(x_i^{B2}), -v\partial_1^2 u_{nsteps}(x_s^{B3}), -v\partial_2^2 v_{nsteps}(x_t^{B4})].$$

Anschließend werden die Vektoren, welche die erste Bedingung enthalten, und der Vektor  $z$  mit den Randbedingungen ebenso der Reihe nach in einem gemeinsamen Vektor  $b$  vereint.

Danach kann auch der Ergebnisvektor  $\beta_r$  als Lösungsvektor des Gleichungssystems  $A\beta_r = b$  berechnet werden. Und wenn  $\beta$  bekannt ist, kann auch der Druck  $p$  berechnet werden als

$$p(x) := \sum_{k=1}^{N^l} \beta_{m,k}^l \partial_1 g(x - x_k^l) + \sum_{k=1}^{N^l} \beta_{m,k}^l \partial_2 g(x - x_k^l) + \sum_{s=1}^{N^h} \beta_{m,s}^h \partial_2 g(x - x_s^h) + \sum_{t=1}^{N^v} \beta_{m,t}^v \partial_1 g(x - x_t^v).$$

#### 4.2.4 Der vierte Schritt

Der vierte Schritt von dem Algorithmus ist vom Prinzip her gleich dem zweiten Schritt. Unterschiedlich ist nur, dass im vierten Schritt für die inneren Punkte nun der Druck mitberücksichtigt wird. Die linke Seite der Bedingungen ist identisch mit der linken Seite aus Schritt zwei. Auf der rechten Seite wird für die inneren Punkte anstelle des Nullvektors aus dem zweiten Schritt nun der Gradient des Druckes eingesetzt:

$$\nabla p(x_j^l) = (\partial_1 p(x_j^l), \partial_2 p(x_j^l)) \quad , x_j^l \in X^l.$$

Die Interpolationsschleife wird im vierten Schritt, genauso wie im zweiten Schritt,  $nsteps$ -mal durchlaufen.

Die äußere Schleife umfasst den dritten und den vierten Schritt. Die Anzahl der Abläufe kann beliebig festgelegt werden. Im Endergebnis sollten sowohl für den Strömungsfluss, als auch für den Druck stabile Lösungen erzielt werden.

## 4.3 Die verwendeten Funktionen

### 4.3.1 Die Wendland-Funktion

Die divergenzfreie radiale Basisfunktion  $\Phi$  basiert auf der folgenden zweidimensionalen skalarwertigen Wendland-Funktion:

$$\psi_{6,4}(r) = (1-r)_+^{10}(85, 8r^4 + 90r^3 + 42r^2 + 10r + 1) \in C^8.$$

Man wählt diese Funktion, weil bei dem geschilderten Problem Ableitungen zweiter Ordnung vorkommen, und auch die Interpolante aus den Informationen der Ableitungen zweiter Ordnung besteht. Die entstehende divergenzfreie, matrixwertige radiale Basisfunktion  $\Phi = \Phi_{6,4}$  hat die Form

$$\begin{aligned} \Phi &= \{-\Delta I + \nabla \nabla^T\} \psi(r) \\ &= (-\partial_1^2 + \partial_2^2)^I + \begin{pmatrix} \partial_1 & \\ & \partial_2 \end{pmatrix} (\partial_1 \ \partial_2) \psi(r) \\ &= \left( - \begin{pmatrix} \partial_1^2 + \partial_2^2 & 0 \\ 0 & \partial_1^2 + \partial_2^2 \end{pmatrix} + \begin{pmatrix} \partial_1 \partial_1 & \partial_1 \partial_2 \\ \partial_2 \partial_1 & \partial_2 \partial_2 \end{pmatrix} \right) \psi(r) \\ &= \begin{pmatrix} \partial_1^2 - \partial_1^2 - \partial_2^2 & \partial_1 \partial_2 \\ \partial_2 \partial_1 & \partial_2^2 - \partial_1^2 - \partial_2^2 \end{pmatrix} \psi(r) \\ &= \begin{pmatrix} -\partial_2^2 & \partial_1 \partial_2 \\ \partial_2 \partial_1 & -\partial_1^2 \end{pmatrix} \psi(r) \\ &= \begin{pmatrix} \phi_{1,1}(r) & \phi_{1,2}(r) \\ \phi_{2,1}(r) & \phi_{2,2}(r) \end{pmatrix}, \end{aligned}$$

mit

$$\begin{aligned} \phi_{1,1}(r) &= -\partial_2^2 \psi(r) \\ &= \frac{26}{5} (1-r)^8 (5 + 40r + 114x^2 - 18y^2 + 24r(3x^2 - 41y^2) - 231r^2(x^2 + 13y^2)), \\ \phi_{1,2}(r) &= \phi_{2,1}(r) = \partial_1 \partial_2 \psi(r) = 686.4 \cdot (1-r)^8 (1 + 8r + 21r^2) xy, \\ \phi_{2,2}(r) &= -\partial_1^2 \psi(r) \\ &= \frac{26}{5} (1-r)^8 (5 + 40r + 114y^2 - 18x^2 + 24r(3y^2 - 41x^2) - 231r^2(13x^2 + y^2)), \end{aligned}$$

wenn  $r = \sqrt{(x^2 + y^2)} \leq 1$ , und 0 sonst.

Für die Berechnung des Strömungsflusses  $s_r$  aus dem zweiten Schritt werden Ableitungen der  $\Phi$ -Funktion erster Ordnung nach der ersten und der zweiten Unbekannten benötigt. Diese Ableitungen werden mit MuPAD berechnet. Dabei wird die Funktion  $\Phi$  analytisch abgeleitet, d.h. jedes Bestandteil von  $\Phi$  wird extra differenziert.

Die erste Ableitung nach der ersten Unbekannten sieht dann folgendermaßen aus

$$\partial_1 \Phi = \begin{pmatrix} \partial_1 \phi_{1,1}(r) & \partial_1 \phi_{1,2}(r) \\ \partial_1 \phi_{2,1}(r) & \partial_1 \phi_{2,2}(r) \end{pmatrix},$$

wobei

$$\begin{aligned} \partial_1 \phi_{1,1}(r) &= -686,4(1-r)^7(21(x^2 + 11y^2)r - 13x^2 + 17y^2 - 7r - 1)x, \\ \partial_1 \phi_{1,2}(r) &= \partial_1 \phi_{2,1}(r) \\ &= 686,4(1-r)^7(21(y^2 + 11x^2)r + 17x^2 - 13y^2 - 7r - 1)y, \\ \partial_1 \phi_{2,2}(r) &= -2059,2(1-r)^7(7(3y^2 + 13x^2)r - 3x^2 - 13y^2 - 7r - 1)x, \end{aligned}$$

wenn  $r = \sqrt{(x^2 + y^2)} \leq 1$ , und 0 sonst.

Und die erste Ableitung nach der zweiten Unbekannten

$$\partial_2 \Phi = \begin{pmatrix} \partial_2 \phi_{1,1}(r) & \partial_2 \phi_{1,2}(r) \\ \partial_2 \phi_{2,1}(r) & \partial_2 \phi_{2,2}(r) \end{pmatrix},$$

wobei

$$\begin{aligned} \partial_2 \phi_{1,1}(r) &= -2059,2(1-r)^7(7(13y^2 + 3x^2)r - 13x^2 - 3y^2 - 7r - 1)y, \\ \partial_2 \phi_{1,2}(r) &= \partial_2 \phi_{2,1}(r) \\ &= 686,4(1-r)^7(21(x^2 + 11y^2)r - 13x^2 + 17y^2 - 7r - 1)x, \\ \partial_2 \phi_{2,2}(r) &= -686,4(1-r)^7(21(11x^2 + y^2)r + 17x^2 - 13y^2 - 7r - 1)y, \end{aligned}$$

wenn  $r = \sqrt{(x^2 + y^2)} \leq 1$ , und 0 sonst.

Die anderen benötigten Berechnungen, wie  $\Delta \Phi$ ,  $\Delta^2 \Phi$ ,  $\partial_1 \Delta \Phi$  und  $\partial_2 \Delta \Phi$  werden ebenfalls jeweils komponentenweise mit MuPAD bestimmt.

### 4.3.2 Die Gauss-Funktion

Für den dritten Schritt der Algorithmen wird die Gauss-Funktion

$$g = \frac{1}{2\pi\sigma^2} \exp\left(-\frac{x^2+y^2}{2\sigma^2}\right)$$

benötigt. Um den Druck  $p$  berechnen zu können, muss diese Gauss-Funktion zweimal differenziert werden. Bei der Implementierung von Algorithmus I werden diese Ableitungen mit Hilfe von MuPAD berechnet.  $x$  und  $y$  sind bei diesem Algorithmus bereits Differenzmatrizen.

Für die erste Ableitung nach der ersten und der zweiten Unbekannten der Gauss-Funktion ergibt sich das Folgende

$$\begin{aligned}\partial_1 g &= -0,1591550775 \cdot x \cdot \exp^{-\frac{x^2}{2} - \frac{y^2}{2}}, \\ \partial_2 g &= -0,1591550775 \cdot y \cdot \exp^{-\frac{x^2}{2} - \frac{y^2}{2}}.\end{aligned}$$

Dementsprechend sehen die zweiten Ableitungen nach den beiden Unbekannten wie folgt aus

$$\begin{aligned}\partial_1 \partial_1 g &= 0,1591550775 \cdot x^2 \cdot \exp^{-\frac{x^2}{2} - \frac{y^2}{2}} - 0,1591550775 \cdot \exp^{-\frac{x^2}{2} - \frac{y^2}{2}}, \\ \partial_1 \partial_2 g &= 0,1591550775 \cdot x \cdot y \cdot \exp^{-\frac{x^2}{2} - \frac{y^2}{2}}, \\ \partial_2 \partial_1 g &= 0,1591550775 \cdot x \cdot y \cdot \exp^{-\frac{x^2}{2} - \frac{y^2}{2}}, \\ \partial_2 \partial_2 g &= 0,1591550775 \cdot y^2 \cdot \exp^{-\frac{x^2}{2} - \frac{y^2}{2}} - 0,1591550775 \cdot \exp^{-\frac{x^2}{2} - \frac{y^2}{2}},\end{aligned}$$

Desweiteren werden  $\Delta g$ ,  $\partial_1 \Delta g$  und  $\partial_2 \Delta g$  auch mit Hilfe von MuPAD berechnet.

Bei der Implementierung des zweiten Algorithmus werden die Ableitungen der Gauss-Funktion selbst von Hand berechnet. Die Gaussfunktion ist also definiert als

$$g = \frac{1}{2\pi\sigma^2} \exp^{-\frac{(x-u)^2 + (y-v)^2}{2\sigma^2}}.$$

Die ersten Ableitungen sind

$$\begin{aligned}\partial_x g &= g \cdot \left(-\frac{x-u}{\sigma^2}\right), \\ \partial_y g &= g \cdot \left(-\frac{y-v}{\sigma^2}\right), \\ \partial_u g &= g \cdot \left(\frac{x-u}{\sigma^2}\right), \\ \partial_v g &= g \cdot \left(\frac{y-v}{\sigma^2}\right).\end{aligned}$$

Für die zweiten Ableitungen ergibt sich

$$\begin{aligned}\partial_x \partial_u g &= g \cdot \left(\frac{1}{\sigma^2}\right) + \left(-\frac{(x-u)}{\sigma^2}\right) \cdot g \cdot \left(\frac{(x-u)}{\sigma^2}\right) = g \cdot \left(\frac{1}{\sigma^2} - \frac{(x-u)^2}{\sigma^2}\right), \\ \partial_u \partial_y g &= g \cdot \left(-\frac{(y-v)}{\sigma^2}\right) \cdot \left(\frac{(x-u)}{\sigma^2}\right), \\ \partial_v \partial_x g &= g \cdot \left(-\frac{(x-u)}{\sigma^2}\right) \cdot \left(\frac{(y-v)}{\sigma^2}\right), \\ \partial_v \partial_y g &= g \cdot \left(\frac{1}{\sigma^2}\right) + \left(-\frac{(y-v)}{\sigma^2}\right) \cdot g \cdot \left(\frac{(y-v)}{\sigma^2}\right) = g \cdot \left(\frac{1}{\sigma^2} - \frac{(y-v)^2}{\sigma^2}\right).\end{aligned}$$

## 4.4 Die Unterschiede zwischen der Doktorarbeit von Svenja Lowitzsch und der vorliegenden Masterarbeit

Der Algorithmus, der in Kapitel 3.4 vorgestellt und in der vorliegenden Masterarbeit implementiert wurde, ist von der Frau Svenja Lowitzsch in [1] eingeführt worden. Sie hat den Fall mit dem noch konstanten Druck behandelt. Von ihr sind die Schritte 1 und 2 von dem Algorithmus bereits implementiert worden. In der vorliegenden Arbeit wird zum ersten Mal der Fall bearbeitet, wenn der Druck nicht konstant ist. D.h. die Schritte 3 und 4 werden implementiert.

### 4.4.1 Grenzen der Randpunkte

Bei der Festlegung der Grenzen für die Abspeicherung der Randpunkte geht Frau Lowitzsch so vor, dass sie die Anzahl  $(2n - 1) + (2m - 1)$  der Gitterpunkte an allen vier Rändern gleichmäßig in vier gleichgroßen Vektoren ablegt ( $n$  ist dabei die Anzahl aller Gitterpunkte auf der  $x$ -Achse, und  $m$  die Anzahl aller Gitterpunkte auf der  $y$ -Achse). In dem von ihr vorgelegten MATLAB-Code wird das  $XB\_vec = [XB1, XB2', XB3', XB4]$  folgendermaßen beschrieben:

$$\begin{aligned} XB1 &= X(1, 1 : n - 1), \\ XB2 &= X(2 : m, 1), \\ XB3 &= X(1 : m - 1, n), \\ XB4 &= X(m, 2 : n). \end{aligned}$$

Bei der in dieser Masterarbeit vorgestellten Implementierung werden die einzelnen Komponenten  $XB1$  bis  $XB4$  des  $XB\_vec$  wie folgt definiert

$$\begin{aligned} XB1 &= X(1, 1 : n), \\ XB2 &= X(2 : m - 1, 1), \\ XB3 &= X(2 : m - 1, n), \\ XB4 &= X(m, 1 : n). \end{aligned}$$

Die Länge der Randvektoren ist auf diese Weise definiert worden, damit der Fall ausgeschlossen wird, dass der letzte  $n$ -te Punkt auf  $B1$  bereits zu dem  $B3$ -Randvektor gehört.

### 4.4.2 Definition der Druckfunktion

Bei dem von der Frau Lowitzsch in [1] eingeführten Algorithmus wird die Berechnung der Druckfunktion definiert als

$$p(x) := \sum_{s=1}^{N^h} \beta_{r,s}^h \partial_2 g(x - x_s^h) + \sum_{t=1}^{N^v} \beta_{r,t}^v \partial_1 g(x - x_t^v) + \sum_{k=1}^{N^I} \beta_{r,k}^I \Delta g(x - x_k^I),$$

wobei  $x^h \in B1 \cup B4$  und  $x^v \in B2 \cup B3$  den horizontalen und vertikalen Rändern entsprechen. Dieser Ansatz ergibt jedoch keine symmetrische Matrix.

In der vorliegenden Arbeit wird der Vorschlag definiert und implementiert, die Druckfunktion  $p(x)$  zu berechnen als

$$p(x) := \sum_{s=1}^{N^h} \beta_{r,s}^h \partial_2 g(x - x_s^h) + \sum_{t=1}^{N^v} \beta_{r,t}^v \partial_1 g(x - x_t^v) + \sum_{k=1}^{N^I} \beta_{r,k}^I \partial_1 g(x - x_k^I) + \sum_{k=1}^{N^I} \beta_{r,k}^I \partial_2 g(x - x_k^I).$$

D.h. bei der Berechnung des Druckes an den inneren Punkten des Feldes wird der Laplace-Operator von der Gaussfunktion ersetzt mit den Summen ihrer beiden Ableitungen. Das führt dazu, dass die in Schritt iii) benötigte Matrix  $A$  symmetrisch wird.

## 5 Aufbau und Ergebnisse

Wie bereits erwähnt, wird zur Darstellung der Ergebnisse, die das implementierte Programm liefert, das Programmpaket MATLAB/SIMULINK benutzt. Es wird ein regelmäßiges Gitter verwendet. Als Größe des Gitters wird ein Rechteck gewählt, dessen Eckpunkte bei  $(0,0)$ ,  $(\ell,0)$ ,  $(0,-d)$  und  $(\ell,-d)$  liegen. Die Parameter  $\ell$  und  $d$  sind frei wählbar. In dieser Arbeit wird nur der Fall eines Quadrates betrachtet, d.h.  $\ell = d = 1$ , also haben alle vier Kanten die Länge 1. Das Programm wird auf drei unterschiedlichen Feinheitsstufen des Gitters getestet. Auf den dargestellten Bildern ist immer die Strömung im stabilen Endzustand zu sehen.

Nach der Visualisierung der Ergebnisse sollen für jede Lösungsmöglichkeit und jede Gitterfeinheit die wichtigsten Fehler analysiert werden. Es werden vier bestimmte  $\ell_2$ -Fehler definiert:

$$\begin{aligned} \text{FehlerI} &:= \|\mathbf{s}_r - \mathbf{s}_{r-1}\|_2, \\ \text{FehlerII} &:= \|\mathbf{s}_r - \mathbf{s}_{r-1}\|_2 + \|\nabla \cdot (\mathbf{s}_r - \mathbf{s}_{r-1})\|_2, \\ \text{FehlerIII} &:= \frac{\|\mathbf{s}_r - \mathbf{s}_{r-1}\|_2 + \|\nabla \cdot (\mathbf{s}_r - \mathbf{s}_{r-1})\|_2}{\|\mathbf{s}_{r-1} - \mathbf{s}_{r-2}\|_2 + \|\nabla \cdot (\mathbf{s}_{r-1} - \mathbf{s}_{r-2})\|_2}, \\ \text{FehlerIV} &:= \|LHS_r - RHS_r\|_2, \end{aligned}$$

wobei  $LHS$  und  $RHS$  entsprechend „left hand side“ und „right hand side“ bedeuten. Die Fehler sind so gewählt worden, dass sie die Konvergenz der Lösung zeigen.

### 5.1 Ergebnisse der ersten Lösungsmöglichkeit

#### 5.1.1 Das Gitter der Größe $10 \times 10$

Zuerst wird auf dem Rechteck ein grobes Gitter der Größe  $10 \times 10$  angelegt. Die Schrittweite ist definiert als

$$h = \frac{\text{Kantenbreite}}{\text{Anzahl der Schritte}} = \frac{1}{10} = 0,1.$$

Somit werden, angefangen bei 0, sowohl in  $x$ -, als auch in  $y$ -Richtung, immer 11 Punkte betrachtet. Insgesamt sind es damit 121 Gitterpunkte. Der Viskositätskoeffizient ist gesetzt auf  $\nu = 1$ . Die Anzahl der Iterationsschritte ist  $nsteps = 20$ . Die Zähigkeit  $\sigma$  des Fluides ist bei dem gesamten unsymmetrischen Fall auf 1 festgelegt worden.

Wie das Strömungsfeld  $s_r$  nach der Ausführung von dem zweiten Schritt aussieht, ist in der Abbildung 5.1 dargestellt. In diesem Schritt ist die Lösung für den Strömungsfluss berechnet worden, der auf einem konstanten Druck basiert.

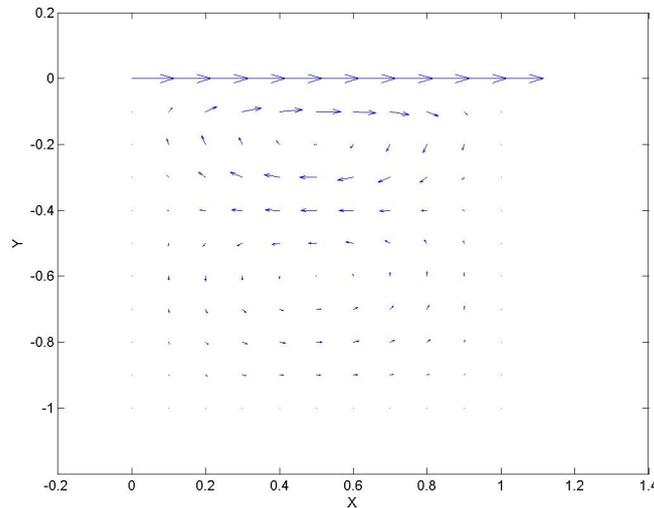


Abbildung 5.1: Das Strömungsfeld  $s_r$  nach dem Ausführen von Schritt 2 für  $h = 0,1$

Im dritten Schritt von dem implementierten Algorithmus wird die Druckfunktion  $p$  berechnet. Bereits nach sehr wenigen Schritten ist die Stabilisierung der Lösung deutlich sichtbar. Das stabile Endergebnis ist der Abbildung 5.2 zu entnehmen.

Unter Berücksichtigung der berechneten Druckfunktion wird letztlich im vierten Schritt der Strömungsverlauf  $s_r$  erneut berechnet. Auch der Strömungsfluss stabilisiert sich sehr rasch, so dass die stabile endgültige Lösung der Strömungsfunktion aus dem zweiten Schritt sehr ähnlich sieht, wie die Abbildung 5.3 zeigt.

In der Tabelle 5.1 werden die  $\ell_2$ -Fehler der iterativen Interpolation mit der ersten Lösungsmöglichkeit bei einer Auflösung von  $10 \times 10$  aufgelistet. Bei den Iterationsschritten handelt es sich um die Anzahl der Durchläufe der äußeren Schleife.

### 5.1.2 Das Gitter der Größe $20 \times 20$

Um die Ergebnisse nach der Ausführung des Programms noch besser darstellen zu können, wird das Gitter auf dem Rechteck von  $10 \times 10$  auf  $20 \times 20$  verfeinert. Die neue Schritt-

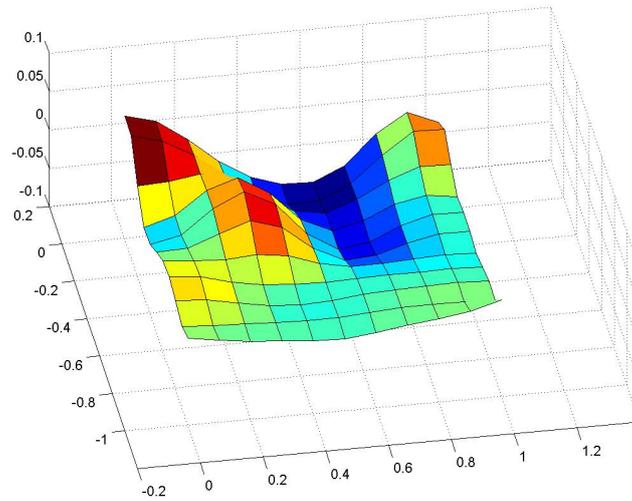


Abbildung 5.2: Das stabilisierte Druckfeld  $p$  nach dem Ausführen von Schritt 3 für  $h = 0,1$

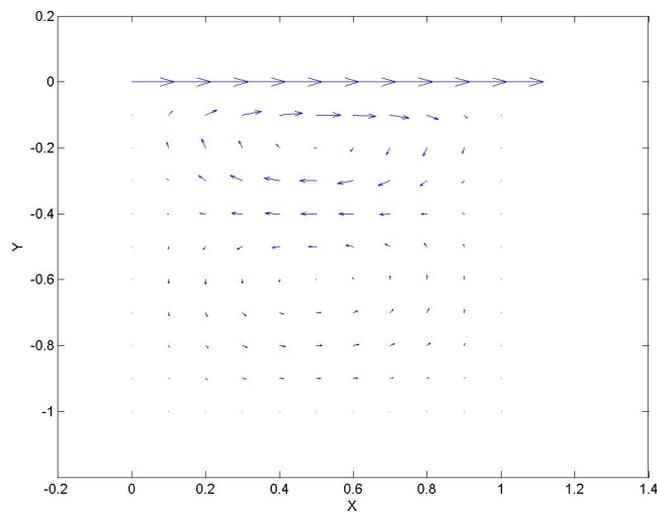


Abbildung 5.3: Das stabilisierte Strömungsfeld  $s_r$  nach dem Ausführen von Schritt 4 für  $h = 0,1$

Iteration	Fehler I	Fehler II	Fehler III	Fehler IV
1	9.1745e-015	1.1543e-013	1.1162	1.7878e-012
2	1.0444e-014	1.1622e-013	0.8787	1.7295e-012
3	9.1644e-015	1.0980e-013	1.0050	1.9198e-012
4	8.6536e-015	9.1599e-014	0.7240	2.7754e-012

Tabelle 5.1:  $\ell_2$ -Fehler der iterativen Interpolation mit der ersten Lösungsmöglichkeit bei einer Auflösung von  $10 \times 10$

weite ist dann entsprechend

$$h = \frac{\text{Kantenbreite}}{\text{Anzahl der Schritte}} = \frac{1}{20} = 0,05.$$

In  $x$ - und  $y$ -Richtung sind es folglich jeweils 21 Punkte, an denen die Einträge ausgerechnet werden. Die Gesamtanzahl aller Gitterpunkte liegt bei 441. Der Viskositätskoeffizient ist, wie im Fall des größeren Gitters, auf  $\nu = 1$  festgelegt.

Im zweiten Schritt von dem Algorithmus wird eine Lösung für das Strömungsfeld  $s_r$  berechnet, die noch auf einem konstanten Druck basiert. Wie dieses Feld bei dem verfeinerten Fall für  $20 \times 20$  Gitterpunkte aussieht zeigt die Abbildung 5.4.

Im dritten Schritt wird die Druckfunktion  $p$  bestimmt. Wie diese aussieht, ist in der Abbildung 5.5 dargestellt.

Im vierten Schritt wird das Strömungsfeld  $s_r$  unter dem Einbeziehen der im dritten Schritt ermittelten Druckfunktion berechnet. Die stabile endgültige Lösung ist auch bei dieser Gitterfeinheit gleich der Lösung aus Schritt 2, wie dies auch schon bei der groben Auflösung gezeigt wurde. Die Abbildung 5.6 zeigt die stabile endgültige Lösung nach dem Ausführen von Schritt 4.

In der Tabelle 5.2 sind die  $\ell_2$ -Fehler der iterativen Interpolation mit der ersten Lösungsmöglichkeit bei einer Auflösung von  $20 \times 20$  dargestellt.

### 5.1.3 Das Gitter der Größe $30 \times 30$

Anschließend soll die Schrittweite des Gitters nochmals verkleinert werden, und zwar auf

$$h = \frac{\text{Kantenbreite}}{\text{Anzahl der Schritte}} = \frac{1}{30} = 0,0333.$$

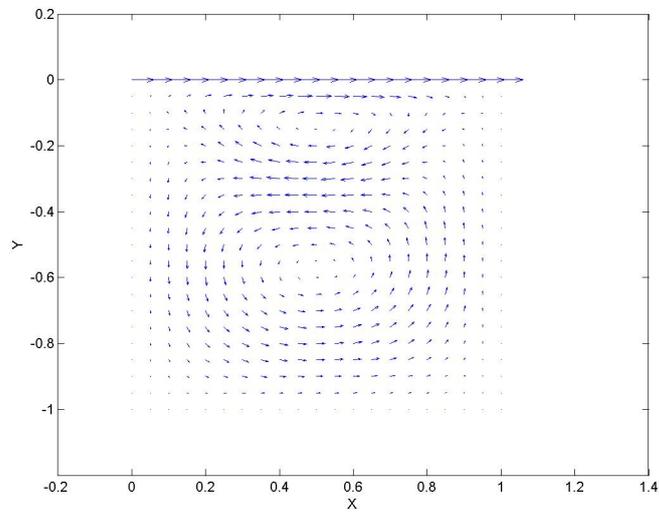


Abbildung 5.4: Das Strömungsfeld  $s_r$  nach dem Ausführen von Schritt 2 für  $h = 0,05$

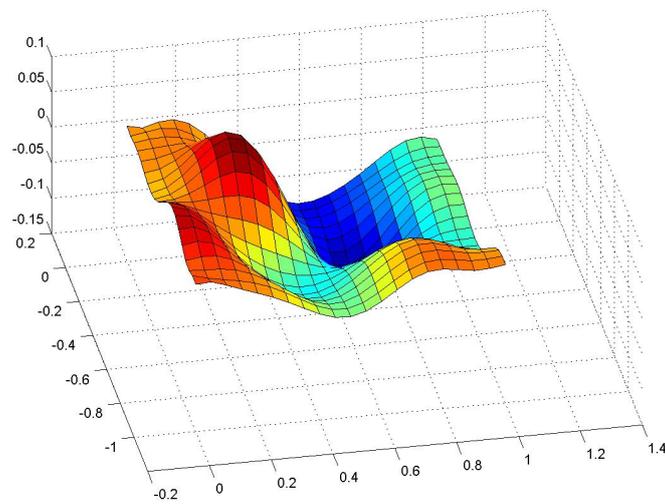


Abbildung 5.5: Das stabilisierte Druckfeld  $p$  nach dem Ausführen von Schritt 3 für  $h = 0,05$

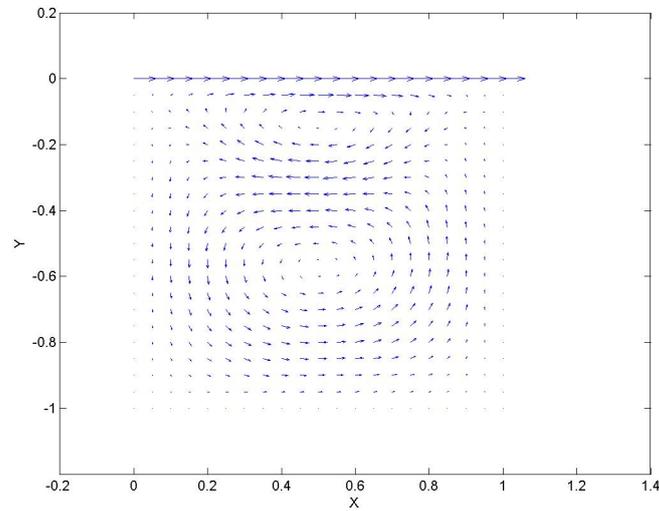


Abbildung 5.6: Das stabilisierte Strömungsfeld  $s_r$  nach dem Ausführen von Schritt 4 für  $h = 0,05$

Iteration	Fehler I	Fehler II	Fehler III	Fehler IV
1	1.2516e-012	2.0835e-011	0.6716	2.4088e-010
2	1.1314e-012	1.8407e-011	0.5829	2.4011e-010
3	7.5470e-013	1.2681e-011	0.9852	2.4697e-010
4	6.8819e-013	9.7237e-012	1.1483	2.4829e-010

Tabelle 5.2:  $\ell_2$ -Fehler der iterativen Interpolation mit der ersten Lösungsmöglichkeit bei einer Auflösung von  $20 \times 20$

Damit beträgt die Größe des Gitters  $30 \times 30$  Punkte, d.h. wieder entsprechend 31 Punkte auf der  $x$ -Achse und 31 Punkte auf der  $y$ -Achse. Insgesamt ergeben sich 961 Punkte. Der Viskositätskoeffizient  $\nu$  ist nach wie vor  $\nu = 1$ .

Das Ergebnis der Berechnung des Strömungsflusses  $s_r$  aus dem zweiten Schritt ist in der Abbildung 5.7 dargestellt. Beachte, dass bei dieser Lösung der Druck noch konstant ist.

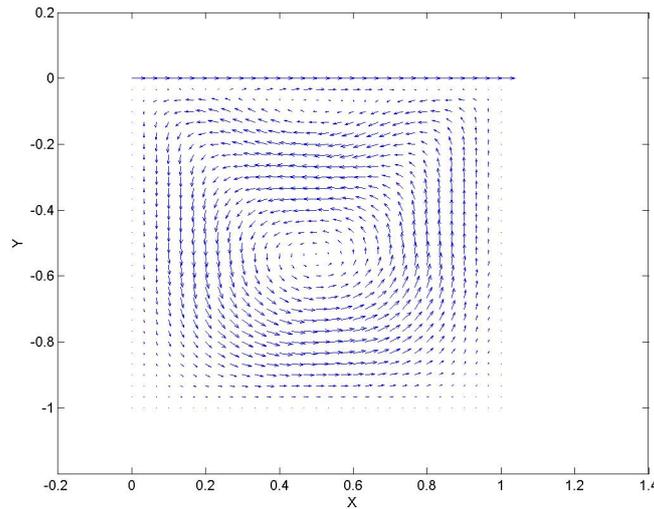


Abbildung 5.7: Das Strömungsfeld  $s_r$  nach dem Ausführen von Schritt 2 für  $h = 0,0333$

Im dritten Schritt wird wieder die Druckfunktion  $p$  ausgerechnet, die im vierten Schritt bei der erneuten Interpolation des Strömungsflusses benötigt wird. Den Druck mit der Auflösung von  $30 \times 30$  zeigt die Abbildung 5.8.

Schließlich zeigt die Abbildung 5.9, was sich für die Interpolation aus Schritt 4 ergibt. In der Tabelle 5.3 sind die  $\ell_2$ -Fehler der iterativen Interpolation mit der zweiten Lösungsmöglichkeit bei einer Auflösung von  $30 \times 30$  dargestellt.

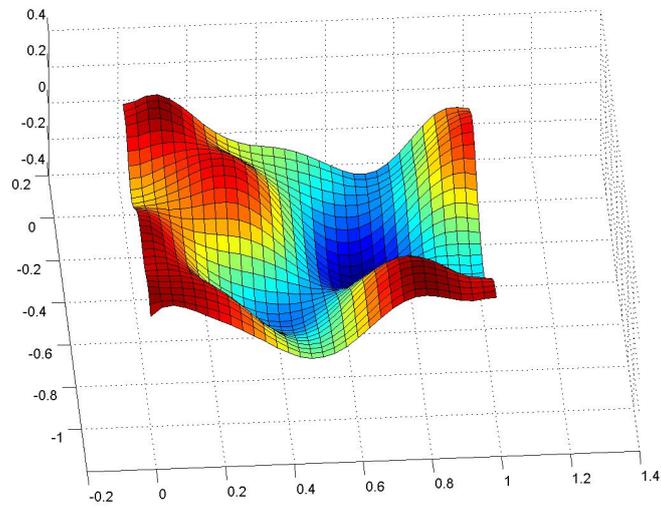


Abbildung 5.8: Das stabilisierte Druckfeld  $p$  nach dem Ausführen von Schritt 3 für  $h = 0,0333$

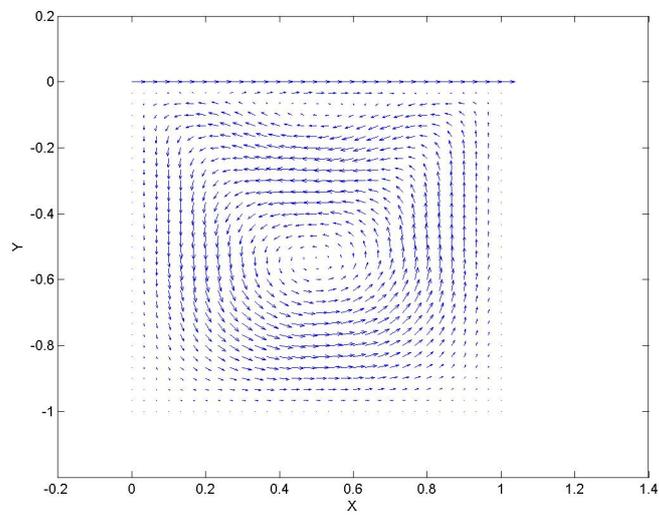


Abbildung 5.9: Das stabilisierte Strömungsfeld  $s$ , nach dem Ausführen von Schritt 4 für  $h = 0,0333$

Iteration	Fehler I	Fehler II	Fehler III	Fehler IV
1	6.8759e-011	9.6769e-010	0.9040	6.4852e-009
2	1.3755e-010	2.1079e-009	2.0230	6.4051e-009
3	1.7488e-010	2.4914e-009	0.5041	6.4049e-009
4	1.9778e-010	2.3280e-009	0.6582	6.5063e-009

Tabelle 5.3:  $\ell_2$ -Fehler der iterativen Interpolation mit der ersten Lösungsmöglichkeit bei einer Auflösung von  $30 \times 30$

## 5.2 Ergebnisse der zweiten Lösungsmöglichkeit

### 5.2.1 Das Gitter der Größe $10 \times 10$

Die Ergebnisse der symmetrischen Lösungsmöglichkeit werden auch zuerst auf dem groben Gitter der Größe  $10 \times 10$  getestet. Die Schrittweite ist definiert als

$$h = \frac{\text{Kantenbreite}}{\text{Anzahl der Schritte}} = \frac{1}{10} = 0,1.$$

Es werden in  $x$ - und  $y$ -Richtung jeweils 11 Punkte betrachtet. Insgesamt also 121 Gitterpunkte. Der Viskositätskoeffizient ist gesetzt auf  $\nu = 1$ . Die Anzahl der Iterationsschritte ist auch hier  $nsteps = 20$ . Die Zähigkeit ist 0.0949.

Wie das Strömungsfeld  $s_r$  nach der Ausführung von dem zweiten Schritt aussieht, ist in der Abbildung 5.10 dargestellt. In diesem Schritt ist die Lösung für den Strömungsfluss berechnet worden, der auf einem konstanten Druck basiert.

Im dritten Schritt von dem implementierten Algorithmus II wird die Druckfunktion  $p$  berechnet. Bereits nach sehr wenigen Schritten ist die Stabilisierung der Lösung deutlich sichtbar. Das stabile Endergebnis des symmetrischen Falles ist der Abbildung 5.11 zu entnehmen.

Unter Berücksichtigung der berechneten Druckfunktion wird letztlich im vierten Schritt der Strömungsverlauf  $s_r$  erneut berechnet. Auch der Strömungsfluss stabilisiert sich sehr rasch, so dass die stabile endgültige Lösung der Strömungsfunktion aus dem zweiten Schritt sehr ähnlich sieht, wie die Abbildung 5.12 zeigt.

In der Tabelle 5.4 werden die  $\ell_2$ -Fehler der iterativen Interpolation mit der zweiten Lösungsmöglichkeit bei einer Auflösung von  $10 \times 10$  aufgelistet. Bei den Iterationsschritten handelt es sich um die Anzahl der Durchläufe der äußeren Schleife.

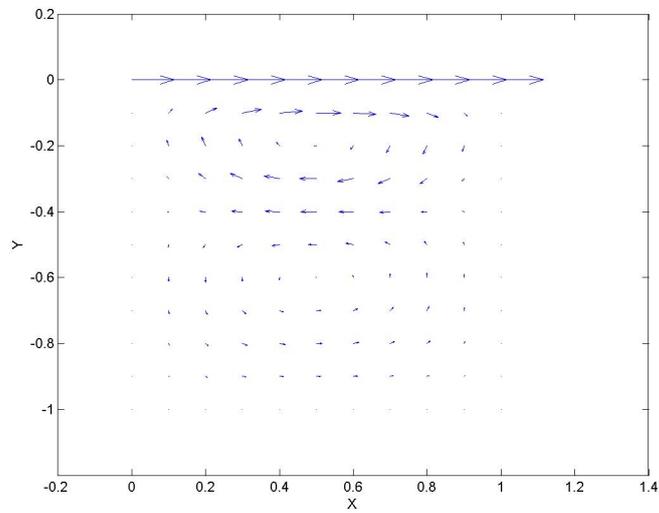


Abbildung 5.10: Das Strömungsfeld  $s_r$  nach dem Ausführen von Schritt 2 für  $h = 0,1$

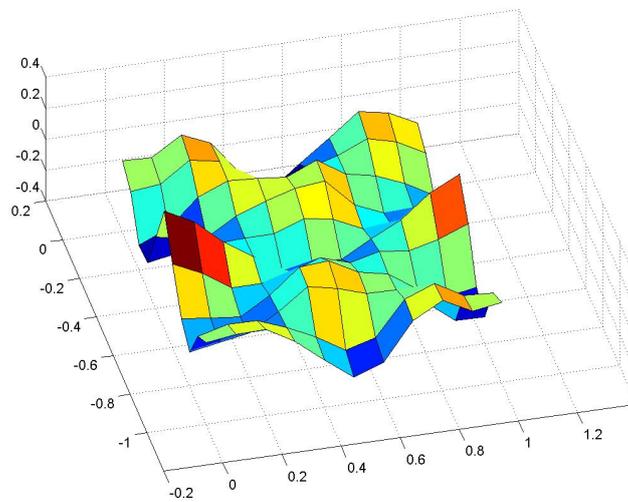


Abbildung 5.11: Das stabilisierte Druckfeld  $p$  nach dem Ausführen von Schritt 3 für  $h = 0,1$

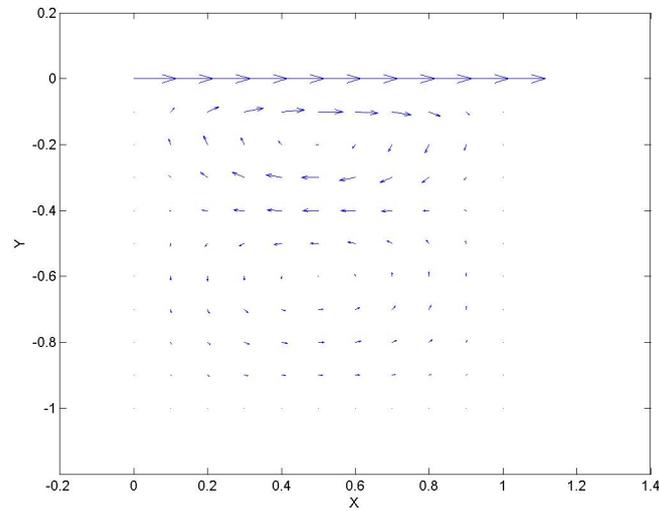


Abbildung 5.12: Das stabilisierte Strömungsfeld  $s_r$  nach dem Ausführen von Schritt 4 für  $h = 0,1$

Iteration	Fehler I	Fehler II	Fehler III	Fehler IV
1	8.9261e-015	9.5629e-014	1.0825	1.9737e-012
2	8.8893e-015	1.2685e-013	0.8749	1.9190e-012
3	6.6776e-015	8.2869e-014	1.0603	2.1122e-012
4	9.5306e-015	1.1598e-013	0.9754	2.2204e-012

Tabelle 5.4:  $\ell_2$ -Fehler der iterativen Interpolation mit der zweiten Lösungsmöglichkeit bei einer Auflösung von  $10 \times 10$

### 5.2.2 Das Gitter der Größe $20 \times 20$

Zur Verfeinerung der Ergebnisse wird das Gitter auf dem Rechteck von  $10 \times 10$  auf  $20 \times 20$  verfeinert. Die neue Schrittweite ist dann

$$h = \frac{\text{Kantenbreite}}{\text{Anzahl der Schritte}} = \frac{1}{20} = 0,05.$$

In  $x$ - und  $y$ -Richtung sind es damit jeweils 21 Punkte, an denen die Einträge ausgerechnet werden. Die Gesamtanzahl aller Gitterpunkte liegt bei 441. Der Viskositätskoeffizient ist, wie im Fall des gröberen Gitters, auf  $\nu = 1$  festgelegt.  $\sigma$  ist 0,0894.

Im zweiten Schritt von dem Algorithmus wird eine Lösung für das Strömungsfeld  $s_r$  berechnet, die noch auf einem konstanten Druck basiert. Wie dieses Feld bei dem verfeinerten Fall für  $20 \times 20$  Gitterpunkte aussieht zeigt die Abbildung 5.13.

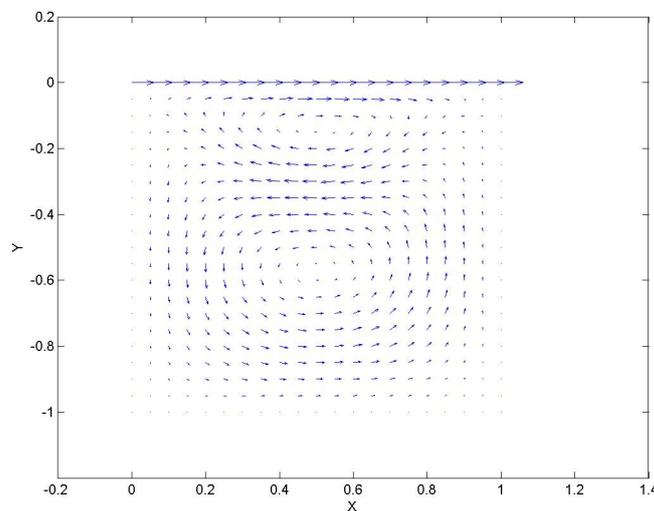


Abbildung 5.13: Das Strömungsfeld  $s_r$  nach dem Ausführen von Schritt 2 für  $h = 0,05$

Im dritten Schritt wird die Druckfunktion  $p$  bestimmt. Wie diese aussieht, ist in der Abbildung 5.14 dargestellt.

Im vierten Schritt wird das Strömungsfeld  $s_r$  unter dem Einbeziehen der im dritten Schritt ermittelten Druckfunktion berechnet. Die stabile endgültige Lösung ist auch bei dieser Gitterfeinheit gleich der Lösung aus Schritt 2, wie dies auch schon bei der groben Auflösung gezeigt wurde. Die Abbildung 5.15 zeigt die stabile endgültige Lösung nach dem Ausführen von Schritt 4.

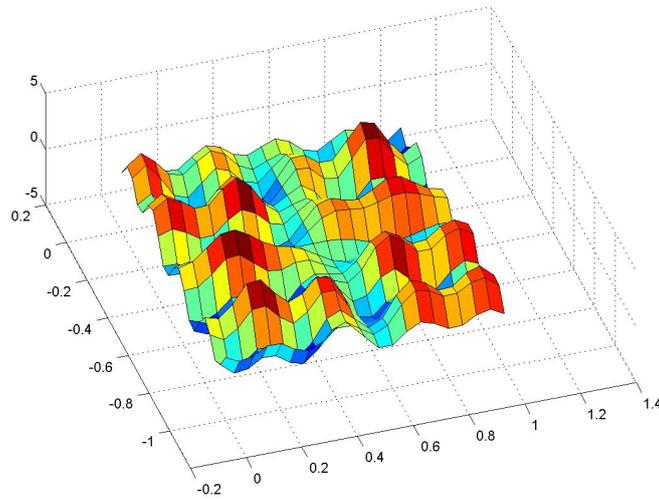


Abbildung 5.14: Das stabilisierte Druckfeld  $p$  nach dem Ausführen von Schritt 3 für  $h = 0,05$

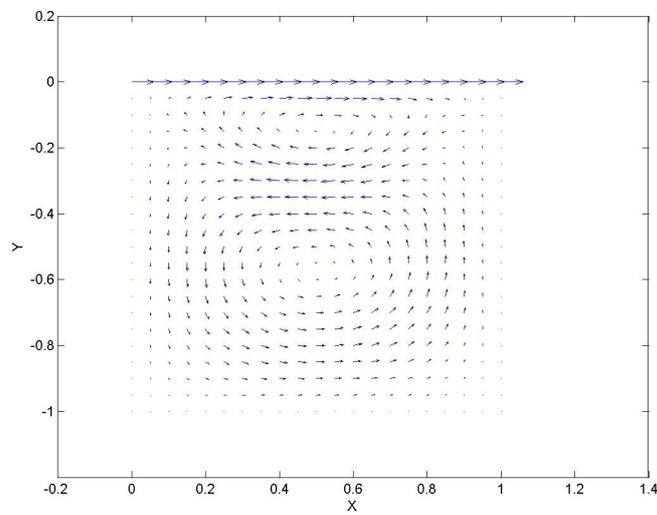


Abbildung 5.15: Das stabilisierte Strömungsfeld  $s_r$  nach dem Ausführen von Schritt 4 für  $h = 0,05$

In der Tabelle 5.5 sind die  $\ell_2$ -Fehler der iterativen Interpolation mit der zweiten Lösungsmöglichkeit bei einer Auflösung von  $20 \times 20$  dargestellt.

Iteration	Fehler I	Fehler II	Fehler III	Fehler IV
1	1.3262e-012	2.0491e-011	1.1412	2.4041e-010
2	1.2787e-012	2.0662e-011	0.8287	2.5517e-010
3	7.9096e-013	1.4321e-011	1.1099	2.4523e-010
4	1.8423e-012	3.0985e-011	1.1898	2.3372e-010

Tabelle 5.5:  $\ell_2$ -Fehler der iterativen Interpolation mit der zweiten Lösungsmöglichkeit bei einer Auflösung von  $20 \times 20$

### 5.2.3 Das Gitter der Größe $30 \times 30$

Auch beim symmetrischen Lösungsansatz soll die Schrittweite des Gitters anschließend nochmal verkleinert werden, und zwar auf

$$h = \frac{\text{Kantenbreite}}{\text{Anzahl der Schritte}} = \frac{1}{30} = 0,0333.$$

Damit beträgt die Größe des Gitters  $30 \times 30$  Punkte, d.h. wieder entsprechend 31 Punkte auf der  $x$ -Achse und 31 Punkte auf der  $y$ -Achse. Insgesamt ergeben sich 961 Punkte. Der Viskositätskoeffizient  $\nu$  ist nach wie vor  $\nu = 1$ .  $\sigma$  ist 0,0730.

Das Ergebnis der Berechnung des Strömungsflusses  $s_r$  aus dem zweiten Schritt ist in der Abbildung 5.16 dargestellt. Beachte, dass bei dieser Lösung der Druck noch konstant ist.

Im dritten Schritt wird wieder die Druckfunktion  $p$  ausgerechnet, die im vierten Schritt bei der erneuten Interpolation des Strömungsflusses benötigt wird. Den Druck mit der Auflösung von  $30 \times 30$  zeigt die Abbildung 5.17.

Schließlich zeigt die Abbildung 5.18, was sich für die Interpolation aus Schritt 4 ergibt. In der Tabelle 5.3 sind die  $\ell_2$ -Fehler der iterativen Interpolation mit der zweiten Lösungsmöglichkeit bei einer Auflösung von  $30 \times 30$  dargestellt.

In der Tabelle 5.6 sind die  $\ell_2$ -Fehler der iterativen Interpolation mit der zweiten Lösungsmöglichkeit bei einer Auflösung von  $30 \times 30$  dargestellt.

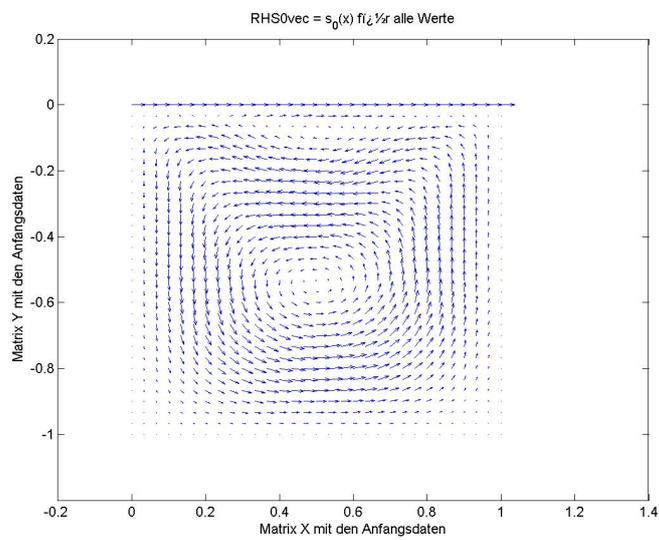


Abbildung 5.16: Das Strömungsfeld  $s_r$ , nach dem Ausführen von Schritt 2 für  $h = 0,0333$

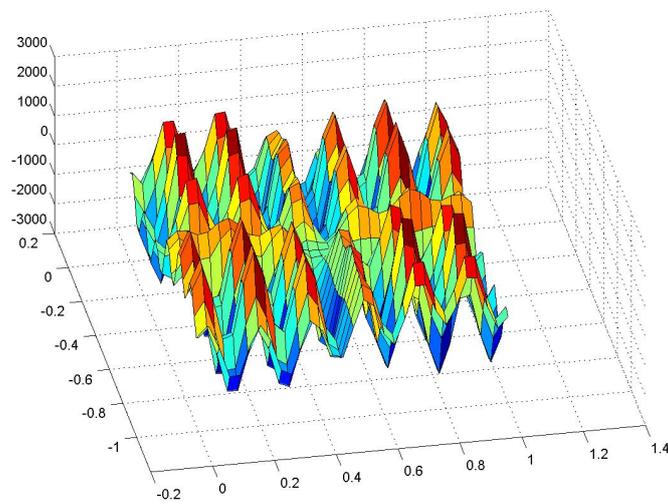


Abbildung 5.17: Das stabilisierte Druckfeld  $p$  nach dem Ausführen von Schritt 3 für  $h = 0,0333$

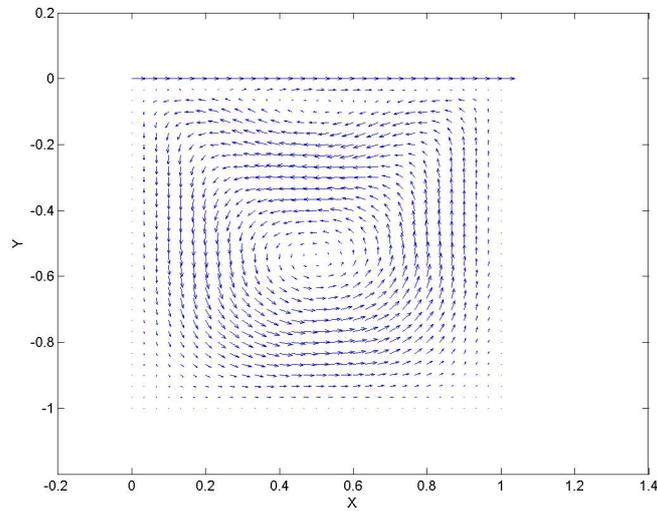


Abbildung 5.18: Das stabilisierte Strömungsfeld  $s_r$  nach dem Ausführen von Schritt 4 für  $h = 0,0333$

Iteration	Fehler I	Fehler II	Fehler III	Fehler IV
1	1.4460e-010	2.3618e-009	0.4082	6.5594e-009
2	2.6914e-010	4.3459e-009	0.4397	6.5236e-009
3	2.9868e-011	5.5075e-010	4.6803	6.3611e-009
4	3.4597e-010	5.5936e-009	0.9589	6.5471e-009

Tabelle 5.6:  $\ell_2$ -Fehler der iterativen Interpolation mit der zweiten Lösungsmöglichkeit bei einer Auflösung von  $30 \times 30$

## 6 Zusammenfassung und Ausblick

In der vorliegenden Arbeit wurde ein Verfahren zur numerischen Behandlung von Strömungen mit divergenzfreien Ansatzräumen entwickelt. Diesem Verfahren liegt die ursprüngliche Methode von Frau Svenja Lowitzsch [1] zugrunde. Dabei wird zum ersten Mal auf den Fall des nichtkonstanten Druckes eingegangen. Die mathematische Vorgehensweise des Verfahrens wurde theoretisch dargestellt. Anschließend wurde es in dem interaktiven Programmpaket für numerische Berechnungen, MATLAB / SIMULINK implementiert.

Zu Beginn der Arbeit wurde auf die theoretischen Grundlagen näher eingegangen. Es wurde erklärt wie die Navier-Stokes-Gleichungen definiert sind und worum es sich beim zweidimensionalen „driven-cavity-problem“ handelt. Weiterhin wurden die Anwendungen der divergenzfreien radialen Basisfunktionen basierend auf dem zweidimensionalen „driven-cavity-problem“ untersucht.

Danach wurden die numerischen Grundlagen des implementierten Verfahrens dargestellt. Es wurde in die Theorie der Interpolation mit radialen Basisfunktionen eingeführt. Anschließend wurde die eigentliche numerische Methode erläutert. Es wurden die Fälle mit konstantem und normalem Druck durchgearbeitet. Bei der Beschreibung der Implementierung des Verfahrens ist auf den Programmaufbau eingegangen worden. Weiterhin ist die Implementierung der einzelnen vier Schritte des Algorithmus und die beim Verfahren verwendeten Funktionen genau erläutert worden.

Abschließend sind die Ergebnisse, die das implementierte Verfahren geliefert hat, bildlich und tabellarisch dargestellt worden. Zur Präsentation der Ergebnisse wurden drei Gitter mit unterschiedlichen Feinheiten ausgesucht. Im Anhang der Arbeit befindet sich eine CD mit der elektronischen Fassung dieser Arbeit und den implementierten Programmen.

Das Verfahren hat insgesamt widersprüchliche Ergebnisse geliefert. In diversen Diskussionen zeigte sich, dass an dem theoretischen Ausgangspunkt der Arbeit von Frau Svenja Lowitzsch [1] noch weitere Untersuchungen nötig gewesen wären. Die Einbeziehung von Ansatzräumen divergenzfreier Vektorfelder in eine Numerik der Navier-Stokes-Gleichungen bedarf einer gründlichen Überarbeitung, wie sich aus Diskussionen mit den Referenten ergab. Auch hat man weder bei der Arbeit von Frau Lowitzsch, noch bei der vorliegenden Arbeit, auch bei hoher Auflösung der Gitter keine Wirbelentwicklung in den Ecken des Raumes sehen können.

In Richtung der Numerik kann das Verfahren noch weiter ausgebaut werden, indem zur Berechnung und Darstellung der Ergebnisse noch feinere Gitter verwendet werden. Weiterhin sollte die Tatsache, dass es sich bei den definierten Matrizen um sparse-Matrizen handelt, programmiertechnisch besser ausgenutzt werden. Desweiteren können höhere Reynoldszahlen eingesetzt werden, um zu überprüfen, wie weit man dabei gehen kann. Man könnte auch ganz aus MATLAB aussteigen und das Verfahren in einer anderen höheren Programmiersprache implementieren.

In der vorliegenden Arbeit ist das Verfahren auf das zweidimensionale „driven-cavity-problem“ angewandt worden. Man sollte versuchen, das Verfahren für dreidimensionale Felder entsprechend umzuschreiben, um es auch auf dreidimensionalen Problemen testen zu können. Auch ist bei der Entwicklung der Arbeit nur der stationäre Fall betrachtet worden. Der nichtstationäre Fall wäre interessant für weitere Untersuchungen.

## 7 Literaturverzeichnis

- [1] Svenja Lowitzsch, *Approximation and interpolation employing divergence-free radial basis functions with applications*, Dissertation, May 2002.
- [2] Svenja Lowitzsch, *Matrix-valued radial basis functions: stability estimates and applications*, *Advanced in Computational Mathematics* (2005)23: 299-315, Springer 2005.
- [3] Holger Wendland, *Piecewise polynomial, positive definite and compactly supported radial functions of minimal degree*, *Advanced in Computational Mathematics* 4, (1995) 389-396.
- [4] Thorsten Kattelans, *Spektrale Least-Squares Verfahren für inkompressible Navier-Stokes-Gleichungen*, Diplomarbeit, März 2007, Universität Duisburg-Essen.
- [5] I. N. Bronstein und K. A. Semendjajew, *Taschenbuch der Mathematik*, 2000.
- [6] Holger Wendland, *Scattered Data Approximation*, Cambridge University Press 2005.
- [7] Holger Wendland, *Ein Beitrag zur Interpolation mit radialen Basisfunktionen*, Diplomarbeit, 1994.
- [8] M.D.Buhmann, *Radial basis functions*, *Acta Numerica*, Volume 9 (2000), 1-38.
- [9] Holger Wendland, *Konstruktion und Untersuchung radialer Basisfunktionen mit kompakten Träger*, Dissertation, Universität Göttingen, 1996.
- [10] Rodrigo B. Platte, Haroldo F. de Campos Velho, Julio C.Ruiz, *Algorithms for pressure correction in the driven cavity problem*, CiteSeer (Scientific Literature Digital Library), 1998.
- [11] Heike Haschke, *Splitting-Techniken zur spektralen Approximation der Navier-Stokes- und Boussinesq-Gleichungen*, Dissertation, Universität Essen, 2001.
- [12] Jörg Frochte, *Ein Splitting-Algorithmus höherer Ordnung für die Navier-Stokes-Gleichung auf der Basis der Finite-Element-Methode*, Dissertation, Universität Duisburg-Essen, 2005.

- [13] H. Sohr, *Zur Regularitätstheorie der instationären Gleichungen von Navier-Stokes*, Math.Z. 184 (1983), 359-375.
- [14] J. Serrin, *The initial value problem for the Navier-Stokes equations*, Nonlinear Probl. 83 (1963), 69-98.
- [15] Z. Wu, *Hermite-Birkhoff Interpolation of scattered data by radial basis functions*, Approximation Theory and its Applications, Vol. 8/2, 1992.