

Hochschule für Technik, Wirtschaft und Kultur Leipzig  
Fakultät Informatik, Mathematik und Naturwissenschaft  
Masterstudiengang Angewandte Mathematik

MASTERARBEIT  
zur Erlangung des akademischen Grades  
Master of Science (M.Sc.)

# Numerische Lösung der linearen Wellengleichung in Gebieten mit kompliziertem Rand

verfasst von Johannes Heun  
Leipzig, November 2014

Gutachter:  
Prof. Dr. Bernd Engelmann  
Dr. Oswald Knoth

# Inhaltsverzeichnis

<b>Abkürzungsverzeichnis</b>	<b>4</b>
<b>1 Einleitung</b>	<b>5</b>
1.1 Motivation . . . . .	5
1.2 Wellengleichung . . . . .	6
1.3 Ziel der Arbeit . . . . .	7
<b>2 Logische Rechteckgitter</b>	<b>9</b>
2.1 Einführung und Eigenschaften . . . . .	9
2.2 Lineares Differentialgleichungssystem . . . . .	11
2.3 Geometrie . . . . .	17
2.3.1 Flächeninhalt der Zellen . . . . .	17
2.3.2 Dualer Flächeninhalt der Kanten . . . . .	19
2.3.3 Mittelpunkt . . . . .	20
<b>3 Ortsdiskretisierung</b>	<b>22</b>
3.1 Einfluss der Gittergrößen . . . . .	22
3.1.1 Äquidistante Diskretisierung . . . . .	23
3.1.2 Variable Diskretisierung . . . . .	24
3.2 Gestörtes Gitternetz . . . . .	25
3.2.1 Gradient durch Divergenz . . . . .	26
3.2.2 Gradient durch MPFA . . . . .	27
3.2.3 Gradient über Mittelpunkt . . . . .	30

<b>4</b>	<b>Verfahren</b>	<b>32</b>
4.1	Implizite Mittelpunkregel . . . . .	32
4.2	Störmer-Verlet-Verfahren . . . . .	33
4.3	Component Splitting . . . . .	35
<b>5</b>	<b>Numerische Beispiele</b>	<b>37</b>
5.1	Logisches Rechteckgebiet . . . . .	37
5.1.1	Vergleichslösung . . . . .	38
5.1.2	Äquidistante Diskretisierung . . . . .	41
5.1.3	Variable Diskretisierung . . . . .	43
5.2	Gebiet mit kompliziertem Rand . . . . .	45
5.2.1	Piola-Transformation . . . . .	46
5.2.2	Fehleranalyse . . . . .	47
<b>6</b>	<b>Implementation in Matlab</b>	<b>50</b>
6.1	Eingabeparameter . . . . .	50
6.2	Programmaufbau . . . . .	51
6.2.1	Gebietsbestimmung . . . . .	51
6.2.2	Gradient und Divergenz . . . . .	52
6.2.3	Verfahren und Lösung . . . . .	53
<b>7</b>	<b>Schlusswort</b>	<b>55</b>
<b>A</b>	<b>Programme</b>	<b>56</b>
A.1	Divergenz . . . . .	56
A.2	Gradient-MPFA . . . . .	57
	<b>Literaturverzeichnis</b>	<b>67</b>

# Abkürzungsverzeichnis

$\Omega$	zu diskretisierendes Gebiet $\Omega \subset R^2$
$x_{i,j}$	Punkt in $\Omega$ mit den Koordinaten $(x_{i,j}, z_{i,j})$
$\Delta x_i$	Breite der Zelle $i$ in $x$ -Richtung
$\Omega_{i,j}$	Gitterzelle mit dem Mittelpunkt $x_{i,j}$
$p_{i,j}$	Funktionswert im Punkt $x_{i,j}$
$u_{i,j+\frac{1}{2}}$	Fluss durch eine senkrechten Kante im Punkt $x_{i,j+\frac{1}{2}}$
$w_{i+\frac{1}{2},j}$	Fluss durch eine waagerechten Kante im Punkt $x_{i+\frac{1}{2},j}$
$\psi$	Funktion durch Gebiet $\Omega$
$f^{p,u,w}$	Impulsfunktion; gibt Impuls auf den Druck $p$ der Zelle bzw. die Flüsse $u$ und $w$ der Kanten ab
$G$	diskrete Gradient-Menge, $p \rightarrow u$
$D$	diskrete Divergenz-Menge, $u \rightarrow p$
$\gamma_o$	Gewicht bzw. Breite einer Zelle mit Orientierung $o$

# Kapitel 1

## Einleitung

### 1.1 Motivation

Die Ausbreitung des Luftdruckes ist ein komplexer Vorgang. Dieser wird nicht nur durch das Medium bestimmt, sondern wird auch durch Hindernisse bzw. die Bodenbeschaffenheit beeinflusst.

Die meisten Modelle, die sich mit der Modellierung dieses Vorgangs beschäftigen, beruhen auf numerischen Berechnungen. Dabei wird die Atmosphäre in Teilgebiete bzw. Zellen unterteilt. Durch die Interaktion der Zellen miteinander und der Bestimmung ihrer Veränderung in bestimmten Zeitschritten, erhält man ein Modell.

Im Folgenden wird ein Ausschnitt der Atmosphäre untersucht und mittels numerischer Verfahren die Veränderung dessen Druckfeldes über einen festgelegten Zeitraum berechnet. Dabei wird ein zweidimensionales Gebiet  $\Omega$  betrachtet. Dieses Gebiet ist abgeschlossen und so groß gewählt, dass die Erdkrümmung vernachlässigt werden kann.

Im zweiten Kapitel wird die gegebene Problemstellung exakt formuliert. Dabei wird ein Gitternetz mit rechteckigen Zellen betrachtet, wobei im besonderen auf die Beziehung zwischen den Zellen eingegangen wird. Anschließend werden im dritten Kapitel verschiedene Möglichkeiten der Ortsdiskretisierung diskutiert, sowie deren Einfluss auf das zu lösende Differentialgleichungssystem betrachtet. Das vierte Kapitel befasst sich

mit den dafür verwendeten Lösungsverfahren. Dabei wird auf deren Vorschrift und deren Eigenschaften eingegangen, sowie ihre Vor- und Nachteile aufgezeigt. Im darauffolgenden Kapitel wird mittels dieser Verfahren eine Lösung für das Problem an einem konkreten Beispiel bestimmt. Dabei wird mit Hilfe einer „manufactured solution“ eine exakte Lösung erzeugt. Mit dieser kann die Genauigkeit der Verfahren bestimmt und anschließend verglichen werden. Im letzten Kapitel dieser Arbeit wird ein Überblick über die Umsetzung des entstandenen Verfahrens in der Programmiersprache Matlab gegeben.

## 1.2 Wellengleichung

Die Ausbreitung des Druckes in einem zweidimensionalen Gebiet  $\Omega \subset R^2$  soll mit Hilfe der Wellengleichung beschrieben und gelöst werden. Dazu wird das Gebiet  $\Omega$  diskretisiert, wodurch ein logisches Rechteckgitter entsteht (Abbildung 1.1).

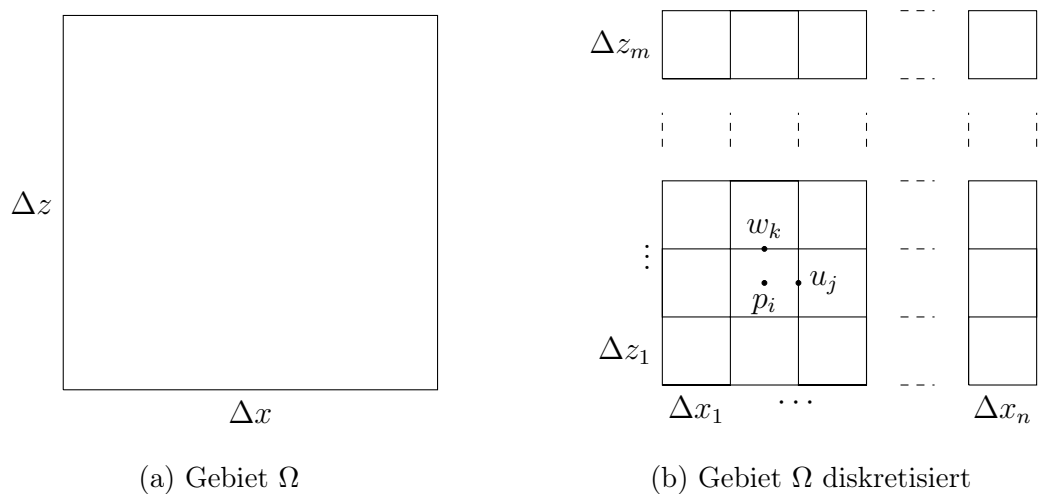


Abbildung 1.1: logisches Gitternetz

In diesem Gitter wird jeder Zelle ein Wert  $p$  zugewiesen, welcher den Druck in diesem Gebiet repräsentiert. Daneben werden den Kanten Flüsse zugewiesen, die den Druckaustausch benachbarter Zellen beschreibt. Dabei sind die Flüsse, die durch senkrechte Kanten gehen, mit  $u$  und die Flüsse, die durch waagerechte Kanten gehen, mit  $w$  gekennzeichnet.

Sowohl der Druck  $p$  einer Zelle als auch die Flüsse  $u$  und  $w$  der Kanten sind zeitlich veränderlich. Diese Veränderung bildet den Zusammenhang zwischen den Werten  $p$ ,  $u$  und  $w$  und wird in den nachfolgenden Gleichungen dargestellt:

$$\frac{\partial p}{\partial t} = -c \cdot \left[ \frac{\partial u}{\partial x} + \frac{\partial w}{\partial z} \right] \quad (1.1)$$

$$\begin{aligned} \frac{\partial u}{\partial t} &= -c \cdot \frac{\partial p}{\partial x} \\ \frac{\partial w}{\partial t} &= -c \cdot \frac{\partial p}{\partial z} \end{aligned} \quad (1.2)$$

Hierbei wird die zeitliche Änderung des Druckes  $p$ , einer Zelle, durch die Formel (1.1) dargestellt. Die Veränderung ist von der örtlichen Änderung der Flüsse, der angrenzenden Kanten  $u$  und  $w$  abhängig. Die Formel (1.2) hingegen beschreibt die zeitliche Änderung der Flüsse. Diese sind wiederum von der örtlichen Änderung des Drucks  $p$ , der benachbarten Zellen abhängig (genauer dazu siehe Kapitel 2.2).

Wird der Druck in Formel (1.1) nochmals partiell nach der Zeit abgeleitet und anschließend auf der rechten Seite die Formel (1.2) eingesetzt, dann ergibt sich die lineare, zweidimensionale Wellengleichung:

$$\frac{\partial^2 p}{\partial t^2} = c^2 \cdot \left[ \frac{\partial^2 p}{\partial x^2} + \frac{\partial^2 p}{\partial z^2} \right] \quad (1.3)$$

## 1.3 Ziel der Arbeit

In dieser Arbeit soll die lineare Wellengleichung für ein diskretisiertes Gebiet, wie es in Abbildung 1.1 zu sehen ist, mit Hilfe von numerischen Verfahren gelöst werden. Dabei werden unterschiedliche Varianten der örtlichen Diskretisierung betrachtet, welche die verwendeten numerischen Lösungsverfahren unterschiedlich beeinflussen.

Darüber hinaus soll der Fall eines gestörten Gebietes untersucht werden. Hierbei wird ein Teil des Gebietes  $\Omega$  heraus geschnitten, wodurch ein komplizierter Rand, beschrieben durch die Kurve  $\psi$ , entsteht. An diesem Rand bilden sich durch die Kurve defekte

Zellen und Kanten. Dadurch wird die Problemstellung derart verändert, dass für die Lösung des Problems mit einem einzelnen numerischen Verfahren, ein enormer Mehraufwand nötig ist. Um das Problem effizient zu lösen wird ein gekoppeltes Verfahren vorgestellt, welches die Nachteile der einzelnen Verfahren ausgleicht.

# Kapitel 2

## Logische Rechteckgitter

### 2.1 Einführung und Eigenschaften

Um die zeitliche und örtliche Veränderung des Luftdruckes in einem Gebiet  $\Omega$  zu beschreiben, wird die Atmosphäre in einzelne Teilgebiete  $\Omega_{i,j}$  unterteilt. Zwischen diesen Teilgebieten, die im Folgenden auch Gitterzelle genannt werden, erfolgt bei Einbeziehung der Kanten ein Druckausgleich, der mittels der Wellengleichung beschrieben wird.

Es gibt verschiedene Möglichkeiten die Ortsdiskretisierung des Gebietes  $\Omega$  durchzuführen (siehe Abbildung 2.1, vgl. [3]). Dabei kann das Gebiet in regelmäßigen Drei-, Vier- oder Sechsecken zerlegt werden. Darüber hinaus gibt es noch weitere Varianten, welche in dieser Arbeit aber keine Rolle spielen werden. Stattdessen wird ausschließlich die Diskretisierung in logische Gitterzellen betrachtet (siehe Abbildung 2.1 Mitte).

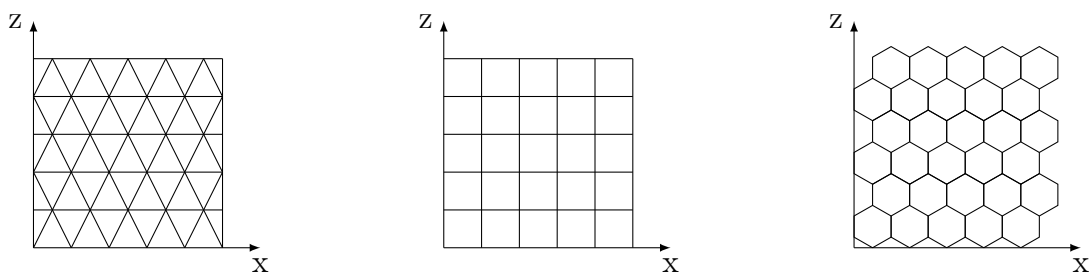


Abbildung 2.1: diskretisierungs Möglichkeiten

## 2.1. Einführung und Eigenschaften

Das Gebiet  $\Omega$  beschreibt einen zweidimensionalen Ausschnitt aus der Erdatmosphäre. Dieser wird so gewählt, dass er senkrecht auf der Erde steht. Das bedeutet, es wird ein Ausschnitt aus der x-z-Ebene betrachtet, wobei die x-Ebene die horizontale Richtung und die z-Ebene die vertikale Richtung bezüglich der Erdoberfläche darstellt. Weiterhin wird vorausgesetzt, dass der betrachtete Ausschnitt so groß gewählt wird, dass die Erdkrümmung vernachlässigt werden kann.

Durch die Diskretisierung des Gebietes  $\Omega$  entsteht ein logisches Gitternetz mit rechteckigen Gitterzellen  $\Omega_{i,j}$ . Der Druck in der Zelle  $\Omega_{i,j}$  wird durch den Druck eines Punktes innerhalb der entsprechenden Zelle repräsentiert. Dieser Punkt wird mit  $x_{i,j}$  bezeichnet und ist im Allgemeinen der Mittelpunkt der Zelle. Äquivalent dazu werden die Flüsse der Kanten, durch den Fluss eines Punktes auf der jeweiligen Kante repräsentiert:  $x_{i,j+\frac{1}{2}}, x_{i+\frac{1}{2},j}$ . Die Flüsse durch die senkrechten Kanten werden mit  $u$  und die Flüsse durch die waagerechten Kanten werden mit  $w$  bezeichnet (siehe Abbildung 2.2).

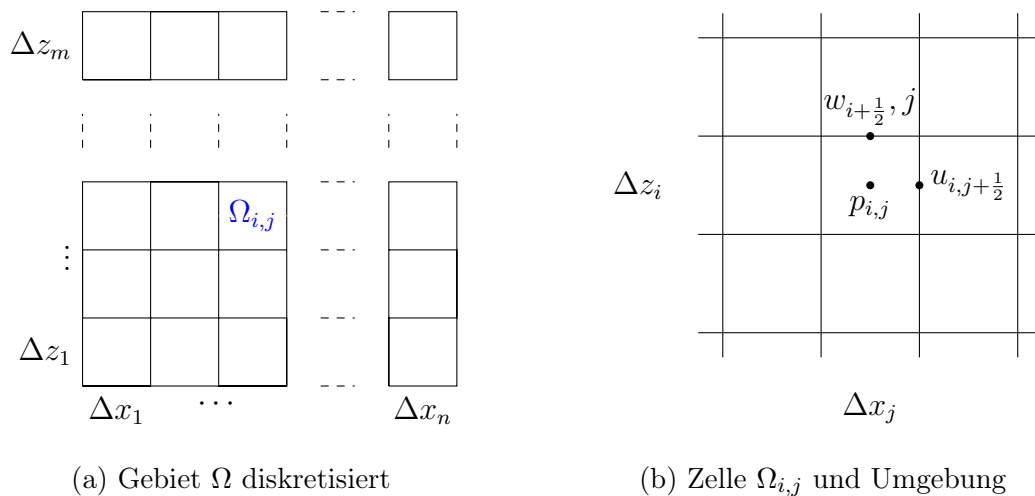


Abbildung 2.2: Diskretisierung

Die entstandenen Drücke der Zellen und Flüsse der Kanten können anschließend über die zeitliche Veränderung miteinander in Beziehung gesetzt werden. Dadurch ist es möglich die Änderung des Drucks einer Zelle über die Flüsse der angrenzenden Kanten zu bestimmen und analog dazu, die Änderung der jeweiligen Flüsse aus den benachbarten Zellen abzuleiten. Auf diesen Zusammenhang wird im nachfolgenden Abschnitt näher eingegangen.

## 2.2 Lineares Differentialgleichungssystem

In diesem Abschnitt wird die zweidimensionale Wellengleichung auf dem logischen Rechteckgitter untersucht. Durch Umformung ergibt sich dabei die Problemstellung in Form eines linearen Differentialgleichungssystems.

Wie bereits mit den Formeln (1.1) bis (1.3) erklärt wurde, besteht ein direkter Zusammenhang zwischen dem Druck  $p_i$  einer Zelle und den Flüssen  $u_j$  und  $w_k$  auf den zugehörigen Kanten. Dieser Zusammenhang wird durch die partielle Ableitung von  $p_i$ ,  $u_j$  und  $w_k$  nach der Zeit  $t$  dargestellt, wie im Folgenden nochmals zu sehen ist:

$$\frac{\partial p}{\partial t} = -c \cdot \left[ \frac{\partial u}{\partial x} + \frac{\partial w}{\partial z} \right] + f^p \quad (2.1)$$

$$\begin{aligned} \frac{\partial u}{\partial t} &= -c \cdot \frac{\partial p}{\partial x} + f^u \\ \frac{\partial w}{\partial t} &= -c \cdot \frac{\partial p}{\partial z} + f^w \end{aligned} \quad (2.2)$$

Beim Vergleich der Formeln (2.1) und (2.2) mit den Formeln (1.1) und (1.2) auf Seite 7, fällt auf, dass bei den Ersten jeweils ein zusätzlicher Term  $f$  auftaucht. Diese Terme werden als Quellterme bezeichnet. Dabei unterscheidet man Quellterme  $f^p$  für die Zellen, sowie Quellterme  $f^u$  und  $f^w$  für die Kanten. Jeder dieser Terme gibt pro Zeitschritt einen Impuls ab, welcher entweder konstant, oder aber zeitabhängig unterschiedlich stark ist. Damit ist es möglich Gebiete mit Impulsgebern zu modellieren. Des Weiteren erleichtern die Terme die Untersuchung von Fehlern bei der numerischen Lösung (siehe Kapitel 5).

Jedem Punkt  $x_{i,j}$  in  $\Omega$  wird durch die Funktion  $p$  ein Druck  $p(x_{i,j})$  zugewiesen. Dadurch bildet  $p$  ein Skalarfeld  $P$  auf  $\Omega$ . Die Funktionen  $u$  und  $w$  hingegen weisen jedem Punkt  $x_{i,j}$  in  $\Omega$  einen Vektor zu. Dieser Vektor gibt Richtung und Stärke des Flusses in diesem Punkt an. Zusammengenommen bilden  $u$  und  $w$  somit ein Vektorfeld  $\vec{U}$ . Diese beiden Felder (Skalar- und Vektorfeld) lassen sich mit Hilfe der Formeln (2.1) und (2.2) ineinander umrechnen. Dabei berechnet Formel (2.1) den Druck (Skalar) einer Zelle aus den angrenzenden Flussvektoren. Durch Anwendung der Formel (2.1) auf alle

Zellen, erhält man ein Skalarfeld  $P$  welches aus dem Vektorfeld  $\vec{U}$  berechnet wurde. Diesen Vorgang bezeichnet man auch als Divergenz. Umgekehrt wird durch Formel (2.2) ein Flussvektor aus den Drücken der benachbarten Zellen bestimmt. Angewendet auf alle Kanten des Gebietes ergibt sich ein Vektorfeld  $\vec{U}$  welches aus dem Skalarfeld  $P$  berechnet wurde. Dieser Vorgang wird als Gradient bezeichnet.

Durch die Umformung der beiden Gleichungen (2.1) bzw. (2.2) lässt sich ein lineares Differentialgleichungssystem aufstellen, das sämtliche Zusammenhänge in dem logischen Gitternetz beschreibt.

## Divergenz bestimmen

Zur Bestimmung der Divergenz wird die Formel (2.1) über eine Gitterzelle  $\Omega_{i,j}$  integriert:

$$\iint_{\square} \left( \frac{\partial p}{\partial t} - f^p \right) dx dz = -c \iint_{\square} \left[ \frac{\partial u}{\partial x} + \frac{\partial w}{\partial z} \right] dx dz \quad (2.3)$$

Da im Folgenden nur die Zelle  $\Omega_{i,j}$  betrachtet wird, wird zur besseren Übersichtlichkeit auf die Indizierung  $i, j$  verzichtet. Die benötigten Flüsse  $u$  und  $w$ , die mit dem Druck  $p$  der Zelle  $\Omega_{i,j}$  in Verbindung stehen, werden dabei wie folgt bezeichnet:

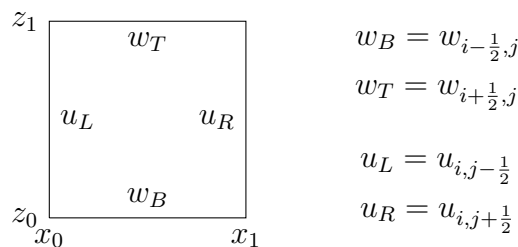


Abbildung 2.3: Betrachtung einer Zelle  $\Omega_{i,j}$

Um die Gleichung (2.3) zu lösen, werden beide Seiten getrennt betrachtet. Die rechte Seite der Gleichung wird dabei direkt gelöst, indem das Integral über das Rechteck aufgespalten und auf die Ortsableitung der Flüsse  $u$  und  $w$  einzeln angewendet wird. Im

letzten Schritt kommt dabei der Mittelwertsatz zum Einsatz:

$$\begin{aligned}
 \dots &= -c \iint_{\square} \frac{\partial u}{\partial x} dx dz - c \iint_{\square} \frac{\partial w}{\partial z} dx dz \\
 \dots &= -c \int_{z_0}^{z_1} \left( \int_{x_0}^{x_1} \frac{\partial u}{\partial x} dx \right) dz - c \int_{x_0}^{x_1} \left( \int_{z_0}^{z_1} \frac{\partial w}{\partial z} dz \right) dx \\
 \dots &= -c \int_{z_0}^{z_1} (u(x_1, z) - u(x_0, z)) dz - c \int_{x_0}^{x_1} (w(x, z_1) - w(x, z_0)) dx \\
 \dots &= -c [\gamma_R \cdot u_R - \gamma_L \cdot u_L] - c [\gamma_T \cdot w_T - \gamma_B \cdot w_B]
 \end{aligned}$$

Im Gegensatz zur rechten Seite, wird die linke Seite der Gleichung indirekt gelöst:

$$\begin{aligned}
 p(x_{i,j}) &:= \frac{1}{A_{\square}} \iint_{\square} p dx dz \\
 \frac{\partial}{\partial t} A_{\square} \cdot p(x_{i,j}) &= \frac{\partial}{\partial t} \iint_{\square} p dx dz \\
 A_{\square} \cdot p_t(x_{i,j}) &= \iint_{\square} \frac{\partial p}{\partial t} dx dz \\
 A_{\square} \cdot p_t(x_{i,j}) &= \iint_{\square} \left( \frac{\partial p}{\partial t} - f^p + f^p \right) dx dz \\
 A_{\square} \cdot (p_t(x_{i,j}) - f^p) &= \iint_{\square} \left( \frac{\partial p}{\partial t} - f^p \right) dx dz
 \end{aligned}$$

Wird die rechte und linke Seite der Gleichung zusammengefasst, dann ergibt sich folgender diskrete Zusammenhang für eine Zelle  $\Omega_{i,j}$ :

$$\begin{aligned}
 A_{\square} \cdot (p_t(x_{i,j}) - f^p) &= -c [\gamma_R \cdot u_R - \gamma_L \cdot u_L] - c [\gamma_T \cdot w_T - \gamma_B \cdot w_B] \\
 & \tag{2.4} \\
 p_t(x_{i,j}) &= -c \left[ \frac{\gamma_R \cdot u_R - \gamma_L \cdot u_L}{A_{\square}} \right] - c \left[ \frac{\gamma_T \cdot w_T - \gamma_B \cdot w_B}{A_{\square}} \right] + f^p
 \end{aligned}$$

Die entstandene Formel (2.4) gibt die zeitliche Veränderung des Druckes  $p_t(x_{i,j})$  einer Zelle  $\Omega_{i,j}$  an, welche von den vier benachbarten Flüssen  $u_L$ ,  $u_R$ ,  $w_B$  und  $w_T$  abhängig

ist. Dabei werden die Flüsse durch die Gewichte  $\gamma$ , die der Länge der Kanten entsprechen, und dem Flächeninhalt der Zelle  $A_{\square}$  gewichtet. Wenn das betrachtete Gebiet  $\Omega$  ungestört ist, dann lässt sich die Gleichung (2.4) noch weiter vereinfachen. Denn in diesem Fall entsprechen die Kantengewichte der Zelle  $\Omega_{i,j}$ , den Gitterbreiten  $\Delta x_j$  und  $\Delta z_i$ . Außerdem gilt  $A_{\square} = \Delta x_j \cdot \Delta z_i$ , woraus folgt:

$$p_t(x_{i,j}) = -c \left[ \frac{u_R - u_L}{\Delta x_j} \right] - c \left[ \frac{w_T - w_B}{\Delta z_i} \right] + f^p \quad (2.5)$$

Um die Gleichung (2.4) für die Berechnung der Divergenz auf alle Zellen des logischen Gitternetzes anwenden zu können, werden die Drücke und Flüsse neu nummeriert. Die Nummerierung des Druckes  $p_{i,j}$  einer Zelle erfolgt dabei in Anlehnung an ein Koordinatensystem (zeilenweise beginnend am Koordinatenursprung). Das heißt, die unterste Zeile des Gitternetzes besteht (von links nach rechts) aus  $p_1$  bis  $p_n$ , gefolgt von der zweiten Zeile mit  $p_{n+1}$  bis  $p_{2n}$ , usw. . Die Drücke in der obersten Zeile des Gitters erhalten folglich die Nummerierung  $p_{(m-1) \cdot n + 1}$  bis  $p_{m \cdot n}$ . Die Nummerierung der Flüsse  $u$  und  $w$  erfolgt analog.

Wendet man nun Formel (2.4) auf alle Zellen  $\Omega_{i,j}$  des logischen Gitternetzes an, ergibt sich ein System von linearen Differentialgleichungen:

$$\dot{p} = -c \cdot A_p^{-1} \cdot \bar{D} \begin{bmatrix} u \\ w \end{bmatrix} + f^p \quad (2.6)$$

Bei dieser Zusammenführung werden die Flächen der Zellen zu einer Matrix  $A_p$  und die Gewichte für die Flüsse zur Matrix  $\bar{D}$  zusammengefasst. Das Produkt  $-c \cdot A_p^{-1} \cdot \bar{D}$  führt eine Transformation von Fluss zum Druck bzw. die Berechnung der Divergenz durch und wird im Folgenden nur noch mit  $D$  bezeichnet. Die Matrix  $D$  entspricht dabei der Divergenzmatrix des Systems.

$$\dot{p} = D \begin{bmatrix} u \\ w \end{bmatrix} + f^p \quad (2.7)$$

## Gradient bestimmen

Die Bestimmung des Gradienten aus Formel (2.2) erfolgt ähnlich, wie die Bestimmung der Divergenz. Im Folgenden soll dies anhand der zeitlichen Änderung des Flusses  $u_{i,j+\frac{1}{2}}$  erläutert werden, welcher auf der Kante zwischen den Zellen  $\Omega_{i,j}$  und  $\Omega_{i,j+1}$  liegt.

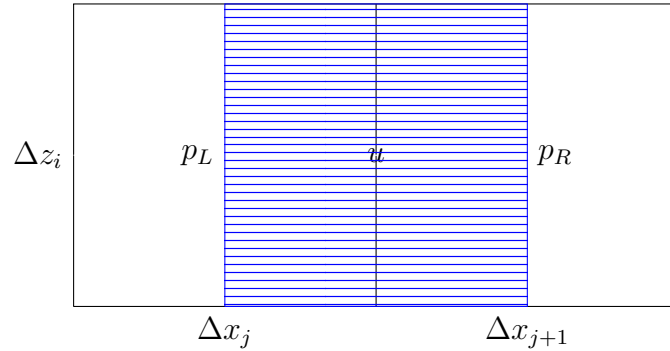


Abbildung 2.4: Fluss zwischen Zelle  $\Omega_{i,j}$  und  $\Omega_{i,j+1}$

Analog zu (2.3) wird die Formel (2.2) ebenfalls über ein Rechteckgebiet integriert:

$$\iint_{\square^D} \left( \frac{\partial u}{\partial t} - f^u \right) dx dz = -c \iint_{\square^D} \frac{\partial p}{\partial x} dx dz \quad (2.8)$$

Dieses Rechteckgebiet (im Folgenden auch als duale Fläche bezeichnet) setzt sich aus Teilflächen der angrenzenden Zellen zusammen. Dabei gilt bei ungestörten Zellen, dass sich die duale Fläche aus der Hälfte der linken und der Hälfte rechten Fläche zusammensetzt. Für die Berechnung von  $u_t(x_{i,j+\frac{1}{2}})$  wird der duale Flächeninhalt in Abbildung 2.4 mit dem schraffierten Bereich dargestellt.

Die Integration der Gleichung (2.8) erfolgt analog zur Integration der Gleichung (2.3) auf Seite 12. Führt man diese aus, dann ergibt sich:

$$A_{\square}^D \cdot \left( u_t(x_{i,j+\frac{1}{2}}) - f^u \right) = -c [\gamma_R \cdot p_R - \gamma_L \cdot p_L] \quad (2.9)$$

$$u_t(x_{i,j+\frac{1}{2}}) = -c \left[ \frac{\gamma_R \cdot p_R - \gamma_L \cdot p_L}{A_{\square}^D} \right] + f^u$$

Die entstandene Formel (2.9) liefert die zeitliche Veränderung des Flusses  $u_t(x_{i,j+\frac{1}{2}})$  zwischen den Zellen  $\Omega_{i,j}$  und  $\Omega_{i,j+1}$ , der von den benachbarten Drücken  $p_L$  und  $p_R$  abhängig ist. Analog zur Divergenzberechnung werden hierbei die Drücke mittels der Gewichte  $\gamma$  und des dualen Flächeninhaltes der Kante  $A_{\square}^D$  gewichtet. Ist das Gebiet  $\Omega$  ungestört, dann lassen sich die Gewichte explizit angeben. Dabei entsprechen die  $\gamma$ -s der Gitterhöhe  $\Delta z_i$  und für den dualen Flächeninhalt gilt  $A_{\square}^D = \frac{1}{2} (\Delta x_j + \Delta x_{j+1}) \cdot \Delta z_i$ . Somit ergibt sich:

$$u_t(x_{i,j+\frac{1}{2}}) = -2c \left[ \frac{p_R - p_L}{\Delta x_j + \Delta x_{j+1}} \right] + f^u \quad (2.10)$$

Die Bestimmung von  $w_t(x_{i+\frac{1}{2},j})$  erfolgt genau analog, wobei hier der Fluss zwischen zwei direkt übereinander liegenden Zellen bestimmt wird. Dabei setzt sich die duale Fläche aus den Flächenhälften dieser beiden Zellen zusammen. Für die allgemeine Bestimmung von  $w_t$  folgt somit die Gleichung (2.11) und für ein ungestörtes Gebiet  $\Omega$  die Gleichung (2.12):

$$w_t(x_{i+\frac{1}{2},j}) = -c \left[ \frac{\gamma_T \cdot p_T - \gamma_B \cdot p_B}{A_{\square}^D} \right] + f^w \quad (2.11)$$

$$w_t(x_{i+\frac{1}{2},j}) = -2c \left[ \frac{p_T - p_B}{\Delta z_i + \Delta z_{j+1}} \right] + f^w \quad (2.12)$$

Analog zur Divergenz werden im Folgenden die Formeln (2.9) und (2.11) auf sämtliche Kanten des Gitternetzes angewendet. Dabei werden alle Gradienten zusammen genommen, wodurch sich folgendes lineares Differentialgleichungssystem ergibt:

$$\begin{bmatrix} \dot{u} \\ \dot{w} \end{bmatrix} = -c \cdot A_u^{-1} \cdot \bar{G} \cdot p + \begin{bmatrix} f^u \\ f^w \end{bmatrix} \quad (2.13)$$

Auch hier werden die dualen Flächen zu einer Matrix  $A_u$  und die Gewichte  $\gamma$  der Drücke  $p$  zur Matrix  $\bar{G}$  zusammengefasst. Das Produkt  $-c \cdot Vol_u^{-1} \cdot \bar{G}$  führt eine Transformation

vom Druck zum Fluss bzw. die Berechnung des Gradienten durch und wird im Folgenden nur noch mit  $G$  bezeichnet. Die Matrix  $G$  entspricht dabei der Gradientenmatrix des Systems.

$$\begin{bmatrix} \dot{u} \\ \dot{w} \end{bmatrix} = G \cdot p + \begin{bmatrix} f^u \\ f^w \end{bmatrix} \quad (2.14)$$

## Zusammenfassung

Die Flüsse  $u$  und  $w$  unterscheiden sich nur geringfügig voneinander. Deshalb wird aus Gründen der Übersichtlichkeit im Folgenden nur noch der Fluss  $u$  angegeben. Dieser repräsentiert dabei beide Flüsse  $u$  und  $w$ . Analog hierzu steht  $f^u$  für die Impulse der  $u$ - und  $w$ -Kanten.

Im Folgenden werden die Gleichungen der Divergenz (2.7) und des Gradienten (2.14) zusammengefasst. Dadurch entsteht ein lineares Differentialgleichungssystem. Dieses System soll in den nachfolgenden Kapiteln für verschiedene Rechteckgitter und mit verschiedenen numerischen Verfahren gelöst werden.

$$\begin{bmatrix} \dot{u} \\ \dot{p} \end{bmatrix} = \begin{bmatrix} 0 & G \\ D & 0 \end{bmatrix} \cdot \begin{bmatrix} u \\ p \end{bmatrix} + \begin{bmatrix} f^u \\ f^p \end{bmatrix} \quad (2.15)$$

## 2.3 Geometrie

In diesem Kapitel wird auf verschiedene geometrische Probleme eingegangen, welche bei der Problemformulierung und den Lösungsstrategien auftreten. Unter anderem wird die Bestimmung der Flächen bzw. dualen Flächen bei defekten Zellen untersucht.

### 2.3.1 Flächeninhalt der Zellen

Zur Berechnung des Flächeninhaltes einer Zelle  $\Omega_{i,j}$  gilt es zwei Zellenarten zu unterscheiden; ungestörte Zellen und defekte Zellen (siehe Abbildung 2.5). Die ungestörten

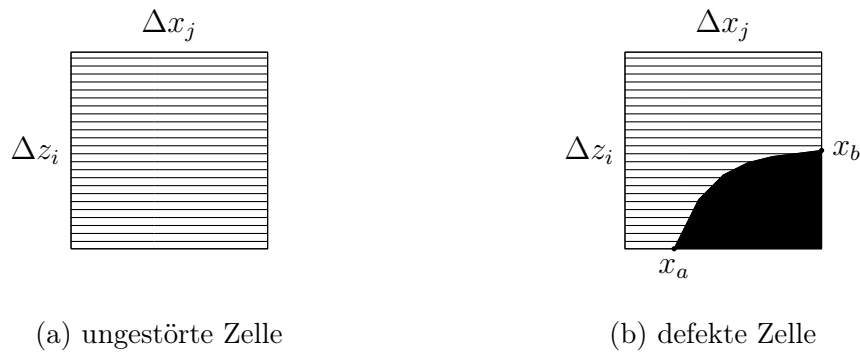


Abbildung 2.5: Flächeninhalt einer Zelle  $\Omega_{i,j}$

Zellen sind Rechteckzellen, deren Flächeninhalt über das Kreuzprodukt zweier benachbarter Kanten gebildet wird. Dadurch ergibt sich für die Fläche:

$$|A_{i,j}| = |\Delta x_j \times \Delta z_i| \quad (2.16)$$

Die zweite Zellenart sind die defekten Zellen. Diese Zellen sind ebenfalls Rechteckzellen mit dem Unterschied, dass am Rand des Rechteckes ein Teil weggeschnitten wurde. An dieser Stelle entsteht ein defekter bzw. komplizierter Rand. Im Folgenden soll dieser Rand durch eine Funktion  $\psi$  beschrieben werden und es gilt, dass der Flächeninhalt der Zelle durch den Bereich über dieser Funktion definiert wird.

Sei nun  $\Omega_{i,j}$  eine solche defekte Zelle mit defektem Rand. Dann ergibt sich der Flächeninhalt der Zelle aus dem Kreuzprodukt der Seiten abzüglich des Flächenintegrals der Funktion  $\psi$  über dem Gebiet der Zelle.

$$|A_{i,j}| = |\Delta x_j \times \Delta z_i| - \int_{x_a}^{x_b} (\psi - z_a) dx \quad (2.17)$$

Mit den Formeln (2.16) und (2.17) lässt sich der Inhalt sämtlicher Flächen des diskretisierten Gebietes  $\Omega$  bestimmen. Dabei werden die Flächen zu einer Matrix  $A_p$  zusammengefasst (siehe Formel (2.6)). Da die Zellen und ihre zugehörigen Drücke  $p_i$  zeilenweise zu einem Vektor zusammengefasst werden, folgt, dass die Matrix  $A_p$  die Form einer Diagonalmatrix annimmt. Dabei gehört der Flächeninhalt  $A_{i,i}$  zur Zellen mit dem Druck  $p_i$ .

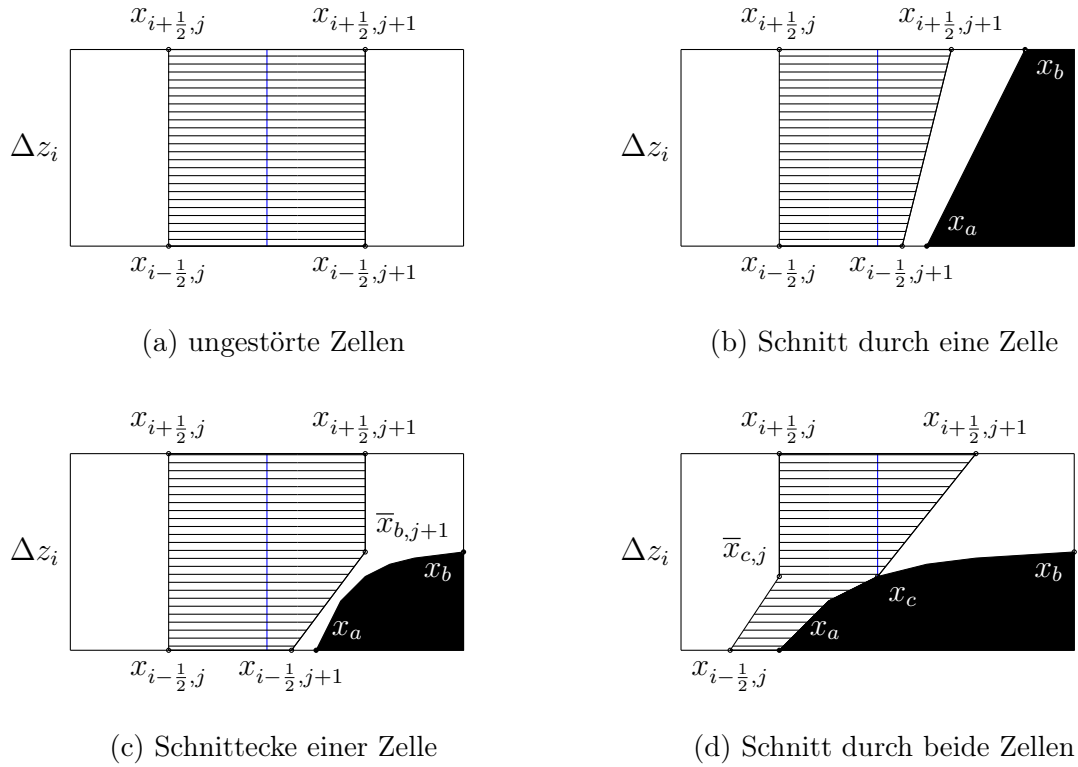


Abbildung 2.6: dualer Flächeninhalt

### 2.3.2 Dualer Flächeninhalt der Kanten

Ähnlich wie bei den Zellen, wird auch den Kanten  $u$  und  $w$  eine Fläche und deren Flächeninhalt zugeordnet (im Nachfolgenden als duale Fläche bzw. dualer Flächeninhalt bezeichnet). Dies erfolgt in Anlehnung an die in [1] beschriebene Flächenbestimmung. Die Fläche setzt sich dabei aus den Teilflächen der zwei angrenzenden Zellen zusammen (siehe Abb. (2.6)). Im Folgenden wird die Berechnung des dualen Flächeninhaltes anhand der Kanten  $u$  erläutert. Die Berechnung der dualen Flächen zu  $w$  erfolgt daraufhin analog.

Betrachtet werden zwei benachbarte Zellen  $\Omega_{i,j}$  und  $\Omega_{i,j+1}$ . Innerhalb dieser wird der duale Flächeninhalt derjenigen Kante gesucht, die diese beiden Zellen gemeinsam haben (in Abb. 2.6 blau dargestellt). Um nun diese Fläche zu bestimmen, werden die vier Kantenmittelpunkte der Kanten benötigt, die einen gemeinsamen Endpunkt mit der gesuchten Kante haben:  $x_{i-\frac{1}{2},j}$ ,  $x_{i-\frac{1}{2},j+1}$ ,  $x_{i+\frac{1}{2},j+1}$ ,  $x_{i+\frac{1}{2},j}$ .

In dem Fall, dass beide Zellen ungestört sind (d.h. die Funktion  $\psi$  verläuft unterhalb der Zellen) wird der Flächeninhalt durch die Kantenmittelpunkten aufgespannt (siehe Abb. 2.6a).

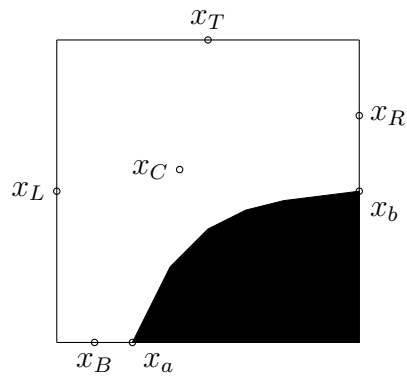
Im zweiten, betrachteten Fall verläuft die Kurve  $\psi$  durch die Zellen (Abb. 2.6b bis 2.6d). Dabei wird angenommen, dass es nur einen Ein- bzw. Austrittspunkt je Zellenpaar gibt. Dies kann durch zusätzliche Glättung der Kurve bzw. eine feinere Diskretisierung erreicht werden. Der Ein- bzw. Austrittspunkt des Zellenpaars wird nachfolgend mit  $x_a$  bzw.  $x_b$  bezeichnet. Darüber hinaus sei der Schnittpunkt von  $\psi$  mit der gemeinsamen Kante der Zellen durch  $x_c$  gekennzeichnet.

Liegt einer der Schnittpunkte  $x_a$  oder  $x_b$  am linken oder rechten Rand des Zellenpaars, bzw. existiert der Schnittpunkt  $x_c$ , dann entsteht ein neuer Hilfspunkt  $\bar{x}$  (siehe Abb. 2.6c und 2.6d). Die Hilfspunkte  $\bar{x}_{b,j+1}$  und  $\bar{x}_{c,j}$  haben dabei die gleiche  $z$ -Koordinate, wie  $x_b$  bzw.  $x_c$  und die gleiche  $x$ -Koordinate, wie der Kantenmittelpunkt im ungestörten Zellenabschnitt. Mit Hilfe dieses zusätzlichen Punktes wird dann der jeweilige duale Flächeninhalt aufgespannt.

Die in den Abbildungen 2.6 nicht schraffierten Flächen gehören dabei jeweils zur dualen Fläche der linken bzw. rechten Kante. Existiert diese Kante nicht, das heißt sie liegt unterhalb der Funktion  $\psi$ , dann wird die entsprechende unschraffierte Fläche der benachbarten dualen Flächen hinzugefügt.

### 2.3.3 Mittelpunkt

Es gibt verschiedene Punkte, die wichtig für das Rechnen im Gitternetz sind. Zu diesen Punkten gehören auch die Kanten- und Zellenmittelpunkte. Im Folgenden wird eine Zelle  $\Omega_{i,j}$  betrachtet, durch die eine Kurve  $\psi$  verläuft, deren Schnittpunkte mit der Zelle durch  $x_a$  und  $x_b$  gegeben sind. Darüber hinaus sind die Kanten und Mittelpunkte wie folgt bestimmt:

Abbildung 2.7: Mittelpunkte einer Zelle  $\Omega_{i,j}$ 

Die Kantenmittelpunkte lassen sich aus dem Mittelwert der zugehörigen Eckpunkte bzw. der Begrenzung  $x_a$  und  $x_b$  bestimmen. Im Gegensatz dazu wird der Schwerpunkt der Fläche  $x_C$  über den Mittelwert aller Kantenpunkte und aller Punkte des Kurvenstückes bestimmt.

# Kapitel 3

## Ortsdiskretisierung

Dieses Kapitel befasst sich mit dem Einfluss der Gittergrößen auf das Differentialgleichungssystem. Dabei werden drei verschiedene Fälle untersucht, wobei in den ersten beiden Fällen ein ungestörtes Gitternetz und im dritten Fall ein gestörtes Gitternetz betrachtet wird. Neben der Störung unterscheidet sich ebenfalls die Art der Diskretisierung. Dabei wird im ersten und dritten Fall eine Äquidistante Diskretisierung und im zweiten Fall eine Diskretisierung mit unterschiedlichen Gittergrößen untersucht.

### 3.1 Einfluss der Gittergrößen

Wie bereits gesagt wurde, gibt es verschiedene Möglichkeiten das Gebiet  $\Omega$  zu diskretisieren. Im Folgenden soll deshalb auf die Zerlegung von  $\Omega$  in ein logisches Gitternetz genauer eingegangen werden.

In einem solchen Gitternetz gibt es mit der Breite der Zellen  $\Omega_{i,j}$  eine veränderliche Größe. Damit können zwei Varianten der Diskretisierung betrachtet werden. Die erste Variante entspricht einer äquidistanten Diskretisierung von  $\Omega$ , d.h. alle Zellen haben die gleiche Größe.

Die zweite Variante entspricht einer variablen Diskretisierung, d.h. aus einer äquidistanten Diskretisierung werden einige wenige Zellen auf ein Bruchteil ihrer Ursprungsgröße

reduziert, wodurch Zellen unterschiedlicher Größe entstehen. Dadurch ist es Möglich eine Änderung der Ausbreitungsgeschwindigkeit  $c$  des Luftdruckes in  $\Omega$  zu modellieren. Wie bereits in Kapitel 2.1 auf Seite 9 erläutert wurde, findet zwischen den Zellen ein Druckausgleich statt, dessen Geschwindigkeit mit der Konstanten  $c$  gesteuert wird. Befindet sich nun beispielsweise ein Gas in dem Gebiet, welches die Ausbreitungsgeschwindigkeit vergrößert, dann kann dies durch Verkleinerung der Gitterzellen modelliert werden. Dabei gilt, dass die dadurch entstehenden kleineren Gitterzellen ein größeres Gebiet beschreiben (z.B. wird ein  $4 \text{ m}^2$  großes Gebiet auf  $1 \text{ m}^2$  reduziert, repräsentiert aber trotzdem ein reales Gebiet von  $4 \text{ m}^2$ ). Mit dieser Vorgehensweise ist es möglich eine Änderung der Ausbreitungsgeschwindigkeit, in eine Änderung der Ortsdiskretisierung zu transformieren, was die Problemlösung vereinfacht.

#### 3.1.1 Äquidistante Diskretisierung

Im Folgenden wird ein äquidistantes Gitternetz betrachtet. Das heißt, es wird vorausgesetzt, dass sämtliche Gitterzellen die gleiche Größe haben ( $\Delta z_i = \Delta x_j$  für alle  $i, j$ ). Es wird dann der Einfluss dieser Diskretisierung auf die Berechnung der Divergenz (2.6) und des Gradienten (2.13) untersucht.

Die Divergenz- und Gradientenberechnung wird mittels der Konstanten  $c$ , den Matrizen  $A_p$  und  $A_u$  für die Flächeninhalte der Zellen bzw. der dualen Flächeninhalte der Kanten, und den Gewichtsmatrizen  $\bar{D}$  und  $\bar{G}$  realisiert.

Die Matrix  $A_p$  beinhaltet die berechneten Flächeninhalte der Zellen, welche aufgrund der äquidistanten Diskretisierung für alle Zellen  $\Omega_{i,j}$  gleich groß sind. Aufgrund der linearen Nummerierung hat  $A_p$  die Form einer Diagonalmatrix, wobei der Eintrag  $A_p(i, i)$  dem Flächeninhalt der Zelle zum Druck  $p_i$  entspricht. Analog hierzu beinhaltet die Matrix  $A_u$  die dualen Flächeninhalte, die ebenfalls alle gleich groß sind. Diese Matrix besteht aus zwei übereinanderliegenden Diagonalmatrizen, eine für die  $u$ -Kanten und eine für die  $w$ -Kanten.

Die Gewichtsmatrix  $\bar{D}$  beinhaltet die Kantengewichte der  $u$  und  $w$  Kanten. Da  $\Delta z_i =$

$\Delta x_j$  für alle  $i, j$  gilt, sind auch hier sämtliche Kantengewichte gleich. Analog dazu beinhaltet die Matrix  $\bar{G}$ , die Gewichte der Zellen für die Gradientenberechnung. Für alle vier Matrizen gilt, dass diese schwach besetzt sind, was für die programmiertechnische Umsetzung von Vorteil ist.

Durch Zusammenfassung der entsprechenden Matrizen, ergeben sich die Divergenzmatrix  $D$ , sowie die Gradientenmatrix  $G$ . Da aufgrund der äquidistanten Diskretisierung alle  $\Delta x_j$  und  $\Delta z_i$  gleich groß sind, ergeben sich in den Formeln (2.5), (2.10) und (2.12) die gleichen Gewichte für die Drücke  $p$  und Flüsse  $u$  und  $w$ , wodurch für die Gradienten- und Divergenzmatrix gilt:  $G = -D^T$ .

#### 3.1.2 Variable Diskretisierung

Neben der äquidistanten Diskretisierung, lässt sich durch Änderung der Größe einzelner  $\Delta x_j$  bzw.  $\Delta z_i$ , eine variable Diskretisierung erreichen. Im Extremfall kann durch große Unterschiede in den Gitterbreiten, ein großer Unterschied in den Flächeninhalten der Zellen entstehen. Dieser Fall soll im Folgenden betrachtet werden, d.h. es wird ein äquidistantes Gitternetz so verändert, dass einige wenige Zeilen und Spalten auf einen Bruchteil der ursprünglichen Größe reduziert werden.

Dies hat einen bedeutenden Einfluss auf die Matrizen in den Gleichungen (2.6) und (2.13). Im Speziellen bewirkt dies, dass die Matrizen  $A_p$  und  $A_u$  einige kleine Einträge enthalten und in den Matrizen  $\bar{D}$  und  $\bar{G}$  einige Kantengewichte deutlich kleiner sind. Werden nun die Matrizen  $D$  und  $G$  bestimmt, dann ist die Gleichung  $G = -D^T$  nicht mehr in allen Komponenten erfüllt.

Als Beispiel wird eine äquidistante Diskretisierung vorausgesetzt, bei der eine Zeile und eine Spalte um einen Faktor  $k$  verkleinert wird. Zuerst wird die Divergenzberechnung nach Formel (2.6) betrachtet. Hier werden für jede Zelle, die Kantengewichte durch den Flächeninhalt der Zelle geteilt. Wird nun eine Zelle durch den Faktor  $k$  beeinflusst, dann wird ihr Flächeninhalt, sowie zwei ihrer vier Kantengewichte um den Faktor  $k$  verringert. Anschließend werden die Gewichte durch den Flächeninhalt geteilt. Dadurch

treffen normale Kanten auf den verringerten Flächeninhalt, wodurch sich an dieser Stelle der Wert in der Divergenzmatrix um den Faktor  $k$  erhöht.

Bei der Gradientenberechnung hingegen entsteht eine wesentlich kleinere Erhöhung. Es werden hier zwei benachbarte Zellen betrachtet, bei der eine um den Faktor  $k$  verändert wurde. Dabei gilt, dass die Veränderung der Zellenbreite nicht die Schnittkante der Zellen betrifft. Berechnet man nun den zugehörigen Wert in der Gradientenmatrix, wird die nicht veränderte Kantenlänge durch den dualen Flächeninhalt geteilt, das durch die Diskretisierung um den Faktor  $\frac{k+1}{2k}$  vergrößert wurde. Somit entsteht, bei genügend großem  $k$  eine Verdoppelung des Gradientenwertes und nicht eine Vervielfachung um den Wert  $k$ .

## 3.2 Gestörtes Gitternetz

Bisher wurde ein ungestörtes, logisches Gitternetz betrachtet. In diesem Abschnitt wird der Fall untersucht, in dem aus dem diskretisierten Gitternetz ein Teil heraus geschnitten wird. Hierbei entstehen, wie im Abschnitt 3.1.2, Zellen unterschiedlicher Größe.

Betrachtet wird ein diskretisiertes Gebiet  $\Omega$  mit äquidistanter Zellengröße. In diesem ist eine Funktion  $\psi$  definiert, die den herausgeschnittenen Bereich beschreibt, d.h. ein Punkt  $x_{i,j}$  aus  $\Omega$  wird weggeschnitten, genau dann wenn gilt:  $z_{i,j} - \psi(x_{i,j}) < 0$ .

Durch die Kurve  $\psi$  entstehen somit am Rand des Gitters defekte Zellen, die nicht mehr durch ihre vier Eckpunkte beschrieben werden können. Das führt dazu, dass sich der Mittelpunkt der Zelle, sowie die Mittelpunkte der Kanten am defekten Rand des Gitters verschieben. Das hat wiederum zur Folge, dass die Strecke zwischen Kanten- und Zellenmittelpunkt nicht mehr senkrecht auf der entsprechenden Kante steht.

Betrachtet man den Fluss auf einer Kante, dann gilt dass dieser im Kantenmittelpunkt definiert ist. Zudem fließt er senkrecht aus der Kante heraus und muss den Zellenmittelpunkt erreichen. Liegt ein ungestörtes Gitternetz vor, dann trifft der Fluss genau den Zellenmittelpunkt. Betrachtet man dagegen eine defekte Zelle in einem gestörten

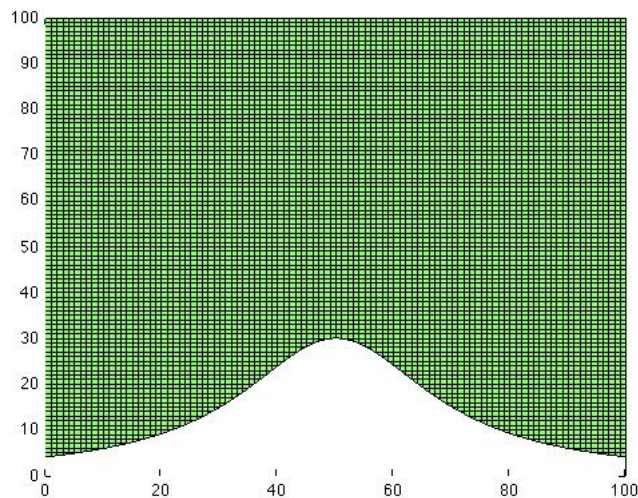


Abbildung 3.1: gestörtes Gitter

Gitternetz, dann wird der Zellenmittelpunkt im Allgemeinen nicht getroffen.

Somit lässt sich durch diese Verschiebung der Mittelpunkte, der Gradient nicht mehr so einfach berechnen. Es gibt nun verschiedene Möglichkeiten eine Gradientenmatrix aufzustellen, wobei im Folgenden drei verschiedene Varianten vorgestellt werden. Die Divergenzmatrix ist von dieser Änderung nicht betroffen und wird, wie im vorherigen Abschnitt erläutert, berechnet.

### 3.2.1 Gradient durch Divergenz

Die erste Variante zur Aufstellung der Gradientenmatrix, orientiert sich an einem nicht defekten Gitter mit äquidistanter Diskretisierung. In diesem Fall lässt sich der Gradient mit Hilfe die Divergenzmatrix ausdrücken und es gilt  $G = -D^T$ .

Aus diesem Grund soll in Kapitel 5.2 untersucht werden, ob sich auch im defekten Gitternetz dieser Zusammenhang als sinnvoll erweist.

### 3.2.2 Gradient durch MPFA

In der zweiten Variante wird der Gradient mit Hilfe der MPFA Methode berechnet, wie sie in [3] beschrieben wird.

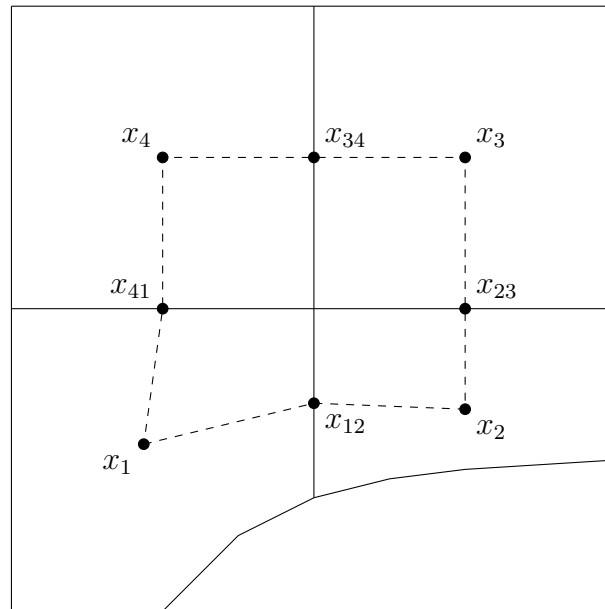


Abbildung 3.2: duales Gitter

Bei der MPFA-Methode (Multi Point Flux Approximation) werden die Flüsse durch die Kanten halbseitig bestimmt. Dafür werden von je vier benachbarten Zellen, deren Mittelpunkte, sowie die Mittelpunkte der Kanten, die diese Zellen gemeinsam haben betrachtet (siehe Abbildung 3.2).

Dadurch erhält man acht Punkte, mit denen sich ein duales Gitter aufspannen lässt. In diesem Gitter sind die Funktionswerte  $p(x_k)$  in den vier Mittelpunkten mit  $k = 1, 2, 3, 4$  der Zellen bekannt, während die Funktionswerte in den Mittelpunkten der Kanten unbekannt sind. Unter der Annahme, dass die Funktionswerte auf den Kanten (bezeichnet mit  $\bar{p}$ ) ebenfalls definiert, aber unbekannt sind, lässt sich innerhalb jeder Zelle mittels der drei Punkten und durch Interpolation, eine lineare Funktion  $p^k(x)$  bestimmen. Um nun die Funktionswerte in den Kantenmittelpunkten bestimmen zu können, muss gelten, dass die Normalenkomponenten des Gradienten der Funktionswerte  $p^k(x)$  in den Kantenmittelpunkten gleich ist.

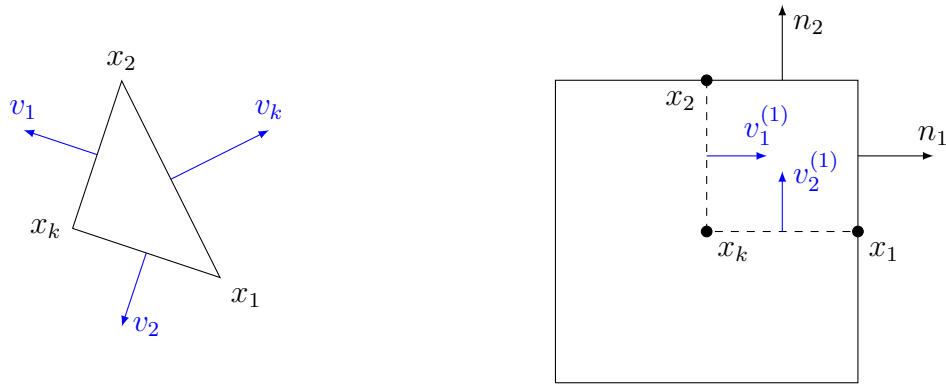


Abbildung 3.3: Zusammenhang der Mittelpunkte in einer Zelle

Im Folgenden wird eine Zelle mit dem Dreieck  $x_k x_1 x_2$  betrachtet (siehe Abbildung 3.3), das die Punkte der Zelle aufspannen. Die lineare Funktion  $p^k$  sei dabei definiert durch:

$$p^k(x) = \sum_{i=1}^3 p_i^k \gamma_i(x) \quad (3.1)$$

$\gamma_i$  ist hier die Gewichtsfunktion, welche aus der Interpolation hervor geht und  $p_i^k$  ist der jeweilige Funktionswert in den drei Punkten. Um nun die Flüsse zu bestimmen, wird der Gradient von  $p^k(x)$  bestimmt. Dieser lässt sich durch die Gradienten der Gewichte ausdrücken, für die gilt:

$$\text{grad } \gamma_i(x) = -\frac{1}{2|A_D|} v_i \quad (3.2)$$

Der Wert  $A_D$  beschreibt den Flächeninhalt des Dreiecks. Des Weiteren ist der Vektor  $v_i$ , die Seitennormale, die dem Punkt  $x_i$  gegenüber liegt. Das heißt  $v_i$  steht senkrecht auf der Seite und hat die gleiche Länge wie diese. Aus diesem Grund ist die Summe der drei Vektoren  $v_1$ ,  $v_2$  und  $v_k$  gleich null. Somit kann einer der Vektoren, durch die anderen beiden ausgedrückt werden. Wird nun der Gradient von  $p^k$  gebildet, ergibt sich:

$$\text{grad } p^{(k)} = \frac{1}{2|A_D^{(k)}|} \left[ v_1^{(k)}(p_1 - p_k) + v_2^{(k)}(p_2 - p_k) \right] \quad (3.3)$$

Daraus ergeben sich die Flüsse für die Halbseiten wie folgt:

$$\begin{aligned} \begin{bmatrix} f_1^{(k)} \\ f_2^{(k)} \end{bmatrix} &= \begin{bmatrix} \gamma_1 \cdot e_1^T \\ \gamma_2 \cdot e_2^T \end{bmatrix} \text{grad } c^{(k)} \\ &= \frac{1}{2|A_D^{(k)}|} \begin{bmatrix} \gamma_1 \cdot e_1^T \\ \gamma_2 \cdot e_2^T \end{bmatrix} \begin{bmatrix} v_1^{(k)} & v_2^{(k)} \end{bmatrix} \begin{bmatrix} \bar{p}_1 - p_k \\ \bar{p}_2 - p_k \end{bmatrix} \end{aligned} \quad (3.4)$$

Die  $\gamma$ -s entsprechen den Kantenlängen zwischen zwei benachbarten Zellen und die Vektoren  $e_i$  sind die Einheitsvektoren, welche senkrecht auf der entsprechenden Seite stehen. Durch Zusammenfassen einiger Terme zu einer Matrix  $G_k$ :

$$G_k = \frac{1}{2|A_D^{(k)}|} \begin{bmatrix} \gamma_1 \cdot e_1^T \\ \gamma_2 \cdot e_2^T \end{bmatrix} \begin{bmatrix} v_1^{(k)} & v_2^{(k)} \end{bmatrix} = \frac{1}{2|A_D^{(k)}|} \begin{bmatrix} \gamma_1 \cdot e_1^T v_1^{(k)} & \gamma_1 \cdot e_1^T v_2^{(k)} \\ \gamma_2 \cdot e_2^T v_1^{(k)} & \gamma_2 \cdot e_2^T v_2^{(k)} \end{bmatrix} \quad (3.5)$$

ist es möglich die Bestimmung der Flüsse verkürzt aufzuschreiben:

$$\begin{bmatrix} f_1^{(k)} \\ f_2^{(k)} \end{bmatrix} = G_k \begin{bmatrix} \bar{p}_1 - p_k \\ \bar{p}_2 - p_k \end{bmatrix} \quad (3.6)$$

Im Folgenden wird nun das Gitternetz aus Abbildung 3.4 betrachtet. In diesem lassen sich für jede der Zellen eine Fussgleichung bestimmen, d.h. es gilt:

$$\begin{aligned} \begin{bmatrix} f_1^{(1)} \\ f_4^{(1)} \end{bmatrix} &= G_1 \begin{bmatrix} \bar{p}_1 - p_1 \\ \bar{p}_4 - p_1 \end{bmatrix}, & \begin{bmatrix} f_1^{(2)} \\ f_2^{(2)} \end{bmatrix} &= G_2 \begin{bmatrix} p_2 - \bar{p}_1 \\ \bar{p}_2 - p_2 \end{bmatrix} \\ \begin{bmatrix} f_3^{(3)} \\ f_2^{(3)} \end{bmatrix} &= G_3 \begin{bmatrix} p_3 - \bar{p}_3 \\ p_3 - \bar{p}_2 \end{bmatrix}, & \begin{bmatrix} f_3^{(4)} \\ f_4^{(4)} \end{bmatrix} &= G_4 \begin{bmatrix} \bar{p}_3 - p_4 \\ p_4 - \bar{p}_4 \end{bmatrix} \end{aligned} \quad (3.7)$$

Durch diese Gleichungen werden für jede Kante zwei Flüsse bestimmt. Aufgrund der Stetigkeitsforderung müssen diese Flüsse gleich sein. Durch Gleichsetzen folgt daraus, dass vier Gleichungen für die Flüsse entstehen. Die Flüsse lassen sich zu einem Vektor  $f$  zusammenfassen, genauso wie die Zellenmittelpunkte den Vektor  $p$  bzw. die Kantenmittelpunkte den Vektor  $\bar{p}$  bilden. Dadurch ergibt sich ein lineares Differentialgleichungssystem:

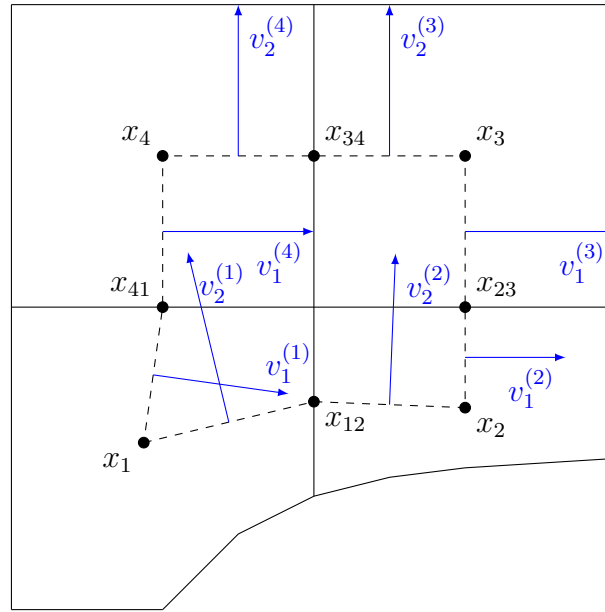


Abbildung 3.4: Globale Indizierung bei MPFA

$$f = A_1 \bar{p} + B_1 p = A_2 \bar{p} + B_2 p \quad (3.8)$$

Durch lösen des Systems können die unbekannt Funktionswerte  $\bar{p}$  eliminiert werden. Dadurch ergibt sich die Gleichung (3.9), welche die Flüsse durch die Kantenhalbseiten bestimmt:

$$f = \left[ A_1 (A_1 - A_2)^{-1} (B_2 + B_1) + B_1 \right] p \quad (3.9)$$

Wendet man dieses Verfahren auf alle Zellen des Gitternetzes an, erhält man für alle Kanten zwei Teilflüsse  $f_1$  und  $f_2$ . Die Summe dieser Teilflüsse, geteilt durch die Kantenlänge, ergibt den gesuchten Fluss  $u$  bzw.  $w$ . Führt man dies für alle Kanten durch, erhält man die gesuchte Gradientenmatrix  $G$ .

### 3.2.3 Gradient über Mittelpunkt

Die dritte Variante die Gradientenmatrix  $G$  zu bestimmen erfolgt über eine Verschiebung der Mittelpunkte. Wie in Abbildung (3.5) zu sehen ist, werden die Kanten- und

Zellenmittelpunkte in den defekten Zellen verschoben. Damit stehen die Verbindungen zwischen den Mittelpunkten, wieder senkrecht auf den Kanten und die Gradienten lassen sich wieder mit den Gleichungen (2.9) und (2.11) bestimmen.

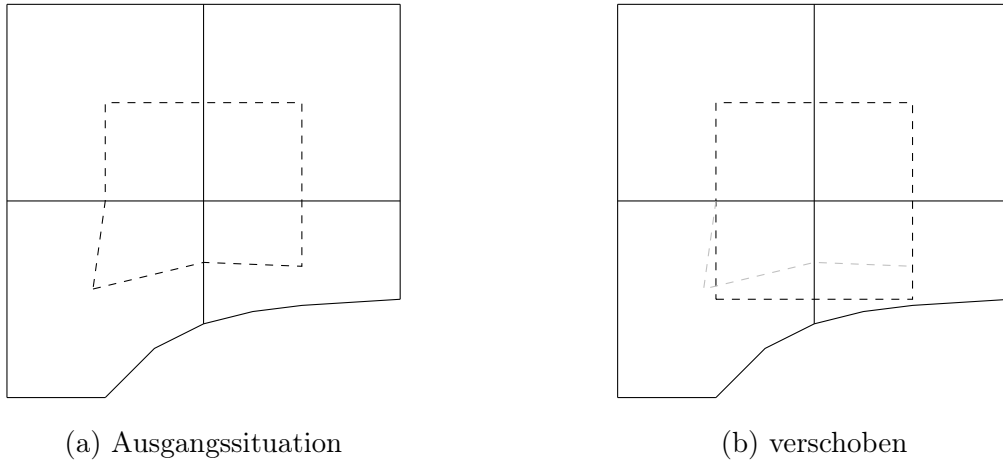


Abbildung 3.5: Korrektur der Mittelpunkte

# Kapitel 4

## Verfahren

In diesem Kapitel werden die numerischen Verfahren vorgestellt, welche zur Lösung des linearen Differentialgleichungsproblems (2.15) verwendet werden. Es gibt viele verschiedene Verfahren, welche eine Lösung für das Problem liefern können. In der vorliegenden Arbeit werden im Wesentlichen nur zwei dieser Verfahren betrachtet.

Dabei handelt es sich um das Störmer-Verlet-Verfahren und die implizite Mittelpunkregel. Diese Verfahren werden sowohl einzeln auf die Problemstellung angewendet, als auch in einer kombinierten Version. Diese Kombination ist das dritte und letzte Verfahren welches zur Lösungsfindung untersucht wird.

### 4.1 Implizite Mittelpunkregel

Das erste Verfahren, welches zur Lösung des Differentialgleichungssystems eingesetzt wird, ist die implizite Mittelpunkregel. Dies ist ein Einschrittverfahren vom Runge-Kutta-Typ und hat folgende Vorschrift:

$$\frac{y_{n+1} - y_n}{\Delta t} = M \cdot \frac{1}{2} (y_{n+1} + y_n) + \frac{1}{2} (f_{n+1}^y + f_n^y), \quad y = \begin{pmatrix} v \\ p \end{pmatrix} \quad (4.1)$$

Wie der Name bereits sagt, gehört Formel (4.1) zu den impliziten Verfahren. Das

bedeutet, dass bei jedem Ausführen ein lineares Gleichungssystem gelöst werden muss. Die Matrix  $M$  ist hierbei die Matrix aus Formel (2.15) in welcher die Matrizen  $G$  und  $D$  in der Diagonale stehen. Obwohl die drei Matrizen  $D$ ,  $G$  und  $M$  des Problems schwach besetzt sind, ist das Lösen eines derart großen Gleichungssystems zeitaufwendig.

Auf der anderen Seite hat das Verfahren, aufgrund seiner impliziten Struktur den Vorteil, dass es sehr robust ist und unabhängig von der Schrittweite  $\Delta t$ , stets eine Lösung liefert. Trotz allem sollte man die Schrittweite möglichst klein wählen, da mit steigendem  $\Delta t$  die Genauigkeit der Lösung abnimmt.

Eine weitere Eigenschaft der impliziten Mittelpunkregel ist, dass sie ein Verfahren zweiter Ordnung ist. Das bedeutet, dass die Abweichung bzw. der Fehler zwischen der numerischen und der exakten Lösung, von  $\Delta t^2$  abhängig ist.

## 4.2 Störmer-Verlet-Verfahren

Das zweite Verfahren, mit welchem eine Lösung für das Problem berechnet wird, ist das Störmer-Verlet-Verfahren. Dieses Verfahren ist im Gegensatz zur impliziten Mittelpunkregel explizit und lässt sich mit Hilfe der beiden Eulerverfahren in Formel (4.2) herleiten.

Als erstes wird dazu das explizite und implizite Verfahren kombiniert. Dabei wird die Berechnung des nächsten Zeitschrittes in zwei Teilschritte zerlegt, wobei der explizite Charakter beibehalten wird. Je nachdem, wie man die Eulerverfahren miteinander kombiniert, können die beiden Teilschritte auch vertauscht werden. Aufgrund weiterer Umformungen werden hier zuerst die Randwerte  $u$  neu berechnet und anschließend die Zellenwerte  $p$ . Dieses Verfahren bezeichnet man auch als symplektisches Eulerverfahren, welches angewendet auf das Problem (2.15) unter Formel (4.3) zu sehen ist.

Um das Störmer-Verlet-Verfahren zu erhalten, wird der Anfangsschritt geteilt. Das heißt,  $u_{n+1}$  wird jeweils in zwei Schritten berechnet, wobei je ein halber Schritt vor und nach der Berechnung von  $p_{n+1}$  erfolgt. Dadurch entsteht ein dritter Teilschritt, in dem der Zwischenwert  $u_{n+\frac{1}{2}}$  berechnet wird. Berücksichtigt man außerdem, dass ein

explizites Verfahren entstehen soll, erhält man das Störmer-Verlet-Verfahren (4.4).

Eulerverfahren:

$$\begin{aligned} \textit{explizit} : \quad y_{n+1} &= y_n + \Delta t (M \cdot y_n + f_n^y) \\ \textit{implizit} : \quad y_{n+1} &= y_n + \Delta t (M \cdot y_{n+1} + f_{n+1}^y) \end{aligned} \tag{4.2}$$

symplektisches Eulerverfahren:

$$\begin{aligned} u_{n+1} &= u_n + \Delta t (G \cdot (p_{n+1} + p_n) + f_{n+1}^u + f_n^u) \\ p_{n+1} &= p_n + \Delta t (D \cdot u_{n+1} + f_{n+1}^p) \end{aligned} \tag{4.3}$$

Störmer-Verlet-Verfahren:

$$\begin{aligned} u_{n+\frac{1}{2}} &= u_n + \Delta t \cdot \frac{1}{2} (G \cdot p_n + f_n^u) \\ p_{n+1} &= p_n + \Delta t (D \cdot u_{n+\frac{1}{2}} + f_{n+\frac{1}{2}}^p) \\ u_{n+1} &= u_{n+\frac{1}{2}} + \Delta t \cdot \frac{1}{2} (G \cdot p_{n+1} + f_{n+1}^u) \end{aligned} \tag{4.4}$$

Dieses Verfahren ist, ebenso wie die Mittelpunkregel, ein Verfahren zweiter Ordnung. Dies macht es im Folgenden leichter, die zwei Verfahren bzgl. ihrer Genauigkeit zu vergleichen.

Im Unterschied zur Mittelpunkregel, ist das Störmer-Verlet-Verfahren ein explizites Verfahren. Das heißt, es kann ohne größeren Aufwand ein neuer Zeitschritt berechnet werden. Der Nachteil dabei ist, dass dies eine Einschränkung der Zeitschrittweite  $\Delta t$  zur Folge hat.

Bezogen auf das Gitter, ist die Schrittweite abhängig von der Größe der Gitterzellen, sowie der Konstanten  $c$ . Wie in Kapitel 2.1 beschrieben, sind die Druckänderungen einer Gitterzelle  $\Omega_{i,j}$  von den vier direkt benachbarten Zellen abhängig. Betrachtet man jetzt in  $\Omega_{i,j}$  den Druck zum Zeitpunkt  $t$ , dann wird mit dem nächsten Zeitschritt bestimmt, wie viel davon zu den benachbarten Zellen fließt. Dabei muss allerdings gelten, dass nicht mehr abfließen kann, als vorhanden ist. Wählt man nun die Schrittweite  $\Delta t$  zu

groß, dann kann ein negativer Druckwert entstehen. Das ist nicht erlaubt und kann zum Zusammenbruch des Verfahrens führen.

Aus diesem Grund ist die Schrittweite beschränkt und es muss für alle  $\Delta x_i$ ,  $\Delta z_j$  und die Konstante  $c$  gelten:

$$\Delta t < \frac{\Delta x_i \cdot \Delta z_j}{c \cdot \sqrt{\Delta x_i^2 + \Delta z_j^2}} \quad (4.5)$$

## 4.3 Component Splitting

In diesem Abschnitt werden die beiden oben vorgestellten Verfahren (Mittelpunktregel und Störmer-Verlet-Verfahren) zu einem gemeinsamen gekoppelt (in Anlehnung an [4]). Das Ziel der Kopplung ist, dass ein Teil der Zellen explizit mittels des Störmer-Verlet-Verfahrens und der andere Teil implizit mit der Mittelpunktregel berechnet werden kann.

Um beide Verfahren miteinander zu koppeln, müssen diese zuerst auf eine ähnliche Struktur gebracht werden. Das Störmer-Verlet-Verfahren (4.4) führt einen Zeitschritt in drei Teilschritten aus, während die Mittelpunktregel (4.1) nur aus einem Schritt besteht.

Für die Kopplung wird nun die Mittelpunktregel ebenfalls in drei Schritte aufgeteilt. Als erstes wird, in Anlehnung an das symplektische Eulerverfahren (4.3), eine separate Berechnung der  $u$ - und  $p$ -Werte vorgenommen.

$$\begin{aligned} u_{n+1} &= u_n + \Delta t \left( G \cdot \frac{1}{2} (p_{n+1} + p_n) + f_n^u \right) \\ p_{n+1} &= p_n + \Delta t \left( D \cdot \frac{1}{2} (u_{n+1} + u_n) + f_n^p \right) \end{aligned} \quad (4.6)$$

Anschließend teilt man (analog zum Störmer-Verlet-Verfahren) die Berechnung von  $u_{n+1}$  in zwei Teilschritte auf. Diese Aufteilung wird so vorgenommen, dass die erste und die dritte Gleichung, mit denen des Störmer-Verlet-Verfahrens übereinstimmen.

Dadurch entsteht die folgende Dreiteilung:

$$\begin{aligned}
 u_{n+\frac{1}{2}} &= u_n + \Delta t \cdot \frac{1}{2} (G \cdot p_n + f_n^u) \\
 p_{n+1} &= u_n + \Delta t \left( D \cdot \frac{1}{2} (u_{n+1} + u_n) + f_{n+\frac{1}{2}}^p \right) \\
 u_{n+1} &= u_{n+\frac{1}{2}} + \Delta t \cdot \frac{1}{2} (G \cdot p_{n+1} + f_{n+1}^u)
 \end{aligned} \tag{4.7}$$

Wie bereits erwähnt wurde unterscheidet sich die entstandene, geteilte Mittelpunkregel vom Störmer-Verlet-Verfahren nur in der Berechnung von  $p_{n+1}$ . Das heißt, die Kopplung beider Verfahren, lässt sich durch eine Kopplung der beiden mittleren Teilschritte erreichen und führt zu folgendem System:

$$\begin{aligned}
 u_{n+\frac{1}{2}} &= u_n + \Delta t \cdot \frac{1}{2} (G \cdot p_n + f_n^u) \\
 p_{n+1} &= p_n + \Delta t \left( D_0 \cdot u_{n+\frac{1}{2}} + D_1 \cdot \frac{1}{2} (u_{n+1} + u_n) + f_{n+\frac{1}{2}}^p \right), \quad D = D_0 + D_1 \\
 u_{n+1} &= u_{n+\frac{1}{2}} + \Delta t \cdot \frac{1}{2} (G \cdot p_{n+1} + f_{n+1}^u)
 \end{aligned} \tag{4.8}$$

Betrachtet man das entstandene Verfahren, fällt auf, dass die Matrix  $D$  in zwei Teile aufgeteilt wurde. Dabei gilt, dass diejenigen Zellen, die explizit durch das Störmer-Verlet-Verfahren berechnet werden sollen, in  $D_0$  enthalten sind und die Zellen, die implizit durch die Mittelpunkregel berechnet werden, sollen in  $D_1$ . Das heißt, wenn  $D_0 = D$  gilt (also  $D_1 = 0$ ), dann liegt das Störmer-Verlet-Verfahren vor und wenn  $D_1 = D$  gilt (also  $D_0 = 0$ ), dann entsteht die Mittelpunkregel.

Durch die Kopplung werden auch die jeweiligen Eigenschaften der beiden Verfahren übernommen. Das heißt, die Schrittweite ist wiederum von den Zellengrößen abhängig und ein Teil des Problems muss implizit berechnet werden. Dabei gilt allerdings, dass die Aufteilung von  $D$  in  $D_0$  und  $D_1$  für eine Verbesserung des Verfahrens gegenüber den einzelnen Verfahren sorgt. Die Schrittweite  $\Delta t$  ist nur noch von den Zellen in  $D_0$  abhängig und es muss nur noch der der Teil in  $D_1$  implizit berechnet werden. Dies bringt eine Zeit- und Rechensparnis.

Für bestimmte Arten der Diskretisierung (siehe nächstes Kapitel) liefert dieses Verfahren eine bessere Lösung als wenn man eines der Verfahren alleine anwendet.

# Kapitel 5

## Numerische Beispiele

In diesem Kapitel wird ein quadratisches Gebiet  $\Omega$  betrachtet, welches eine Kantenlänge von 10 *km* hat. Des Weiteren wird  $\Omega$  diskretisiert, so dass ein logisches Rechteckgebiet mit je 100 Zeilen und 100 Spalten entsteht. Im Folgenden soll das Differentialgleichungssystem (2.15) auf diesem Gebiet gelöst werden, wobei die im vorhergehenden Kapitel beschriebenen, numerischen Verfahren zum Einsatz kommen.

Im ersten Abschnitt wird ein ungestörtes Gitternetz betrachtet. Dabei werden zwei verschiedene Ortsdiskretisierungen untersucht. Die Erste ist eine äquidistante Diskretisierung und die Zweite eine variable Diskretisierung.

Im zweiten Teil dieses Kapitels wird das Gebiet  $\Omega$  durch eine Kurve  $\psi$  gestört. Diese Kurve, welche durch das Gebiet verläuft, teilt es in zwei Abschnitte. Der Abschnitt unterhalb der Kurve wird dabei aus dem Gebiet heraus geschnitten. Dadurch entsteht ein komplizierter Rand, der sich negativ auf die Lösung der Differentialgleichung (2.15), sowie die numerischen Verfahren auswirkt.

### 5.1 Logisches Rechteckgebiet

Im Nachstehenden wird ein ungestörtes Gitternetz betrachtet. In diesem Gitternetz wird die Lösung der jeweiligen numerischen Verfahren für zwei verschiedene Diskreti-

sierungen untersucht. Zuvor wird eine exakte Lösung bestimmt, die für den späteren Vergleich und zur Bestimmung des Fehlers heran gezogen werden soll.

### 5.1.1 Vergleichslösung

Um die numerischen Verfahren miteinander vergleichen zu können, ist eine exakte Lösung des Problems notwendig. Eine solche Lösung soll in diesem Abschnitt, mit Hilfe einer „manufactured solution“, bestimmt werden.

Die Vergleichslösung muss die Bedingungen der Wellengleichung in Formel (2.1) und (2.2) auf Seite 11 erfüllen, sowie die Nullrandbedingungen. Realisiert wird dies über die Quellterme  $f^p$ ,  $f^u$  und  $f^w$ . Diese können stets so gewählt werden, dass die Bedingungen der Wellengleichung für beliebige stetige Funktionen  $p$ ,  $u$  und  $w$  erfüllt sind.

Für das Gitternetz wird folgende Funktion für die  $p$ -Werte vorgegeben:

$$p(x, z, t) = \cos\left(\pi \frac{x}{h_x}\right) \cdot \cos\left(\pi \frac{z}{h_z}\right) \cdot \cos(t) \quad (5.1)$$

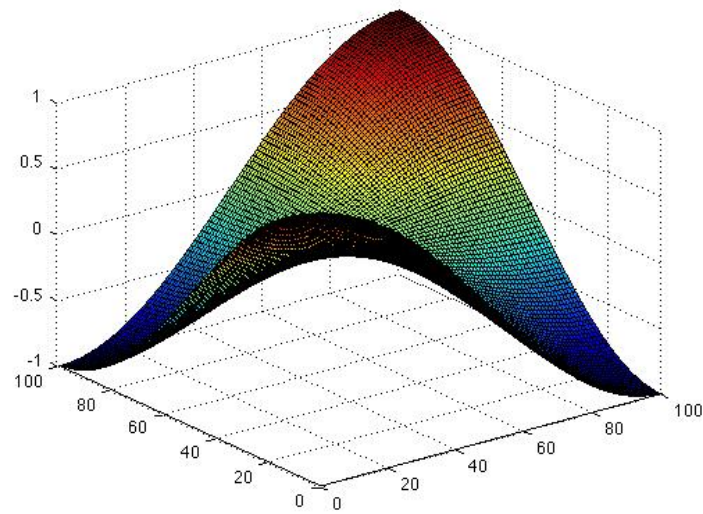
In jedem Punkt des Gebietes  $\Omega$  ist ein Fluss, mit Richtung und Stärke, definiert. Dieser Fluss wird in zwei Komponenten  $u$  und  $w$  aufgeteilt. Die  $u$  Komponente fließt dabei in  $x$  Richtung und die  $w$  Komponente fließt in  $z$  Richtung. Unter Berücksichtigung der Nullrandbedingungen in  $\Omega$ , werden die Flüsse  $u$  und  $w$  wie folgt definiert:

$$\begin{aligned} u(x, t) &= \sin\left(2\pi \frac{x}{h_x}\right) \cdot \cos(t) \\ w(z, t) &= \sin\left(2\pi \frac{z}{h_z}\right) \cdot \cos(t) \end{aligned} \quad (5.2)$$

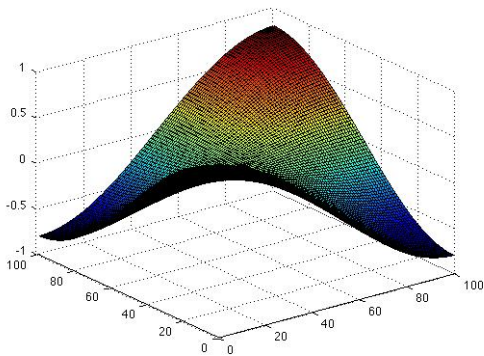
Dabei wird mit  $h_x$  und  $h_z$  die Breite bzw. Höhe des Gebietes  $\Omega$  angegeben.

## Kurvenverlauf

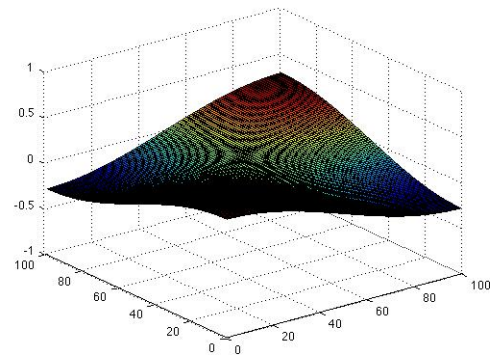
Die unter (5.1) und (5.2) angegebenen Funktionen bilden die gesuchte Vergleichslösung, mit der die Genauigkeit der numerischen Verfahren untersucht wird. Die Funktion



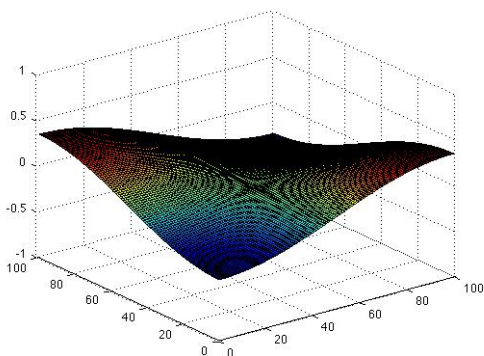
(a) Anfangsbedingung:  $t = 0$  s



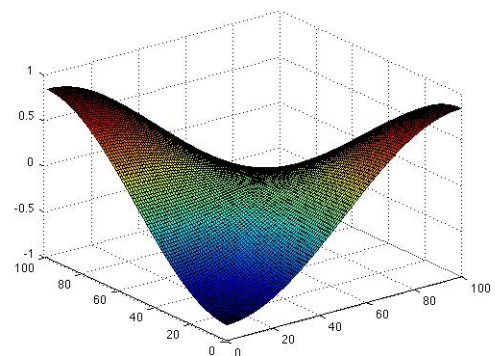
(b)  $t = 256$  s



(c)  $t = 512$  s



(d)  $t = 768$  s

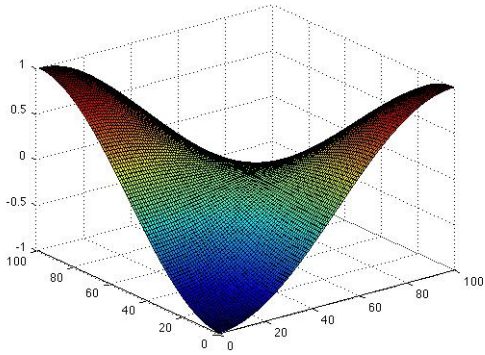


(e)  $t = 1024$  s

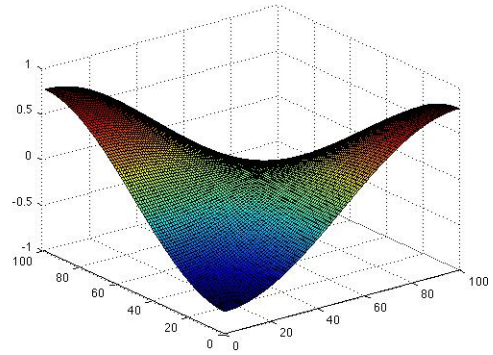
Abbildung 5.1: exakter Kurvenverlauf

## 5.1. Logisches Rechteckgebiet

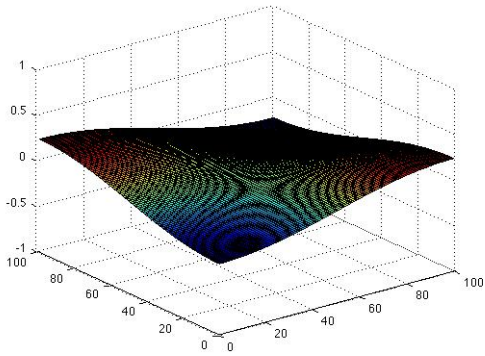
---



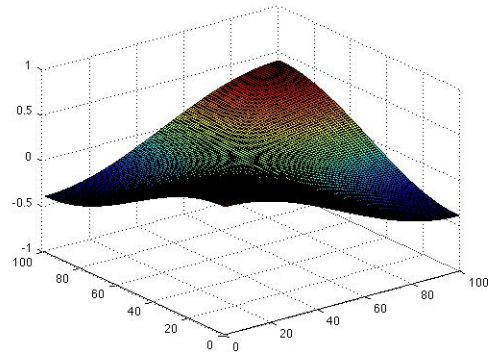
(f)  $t = 1280$  s



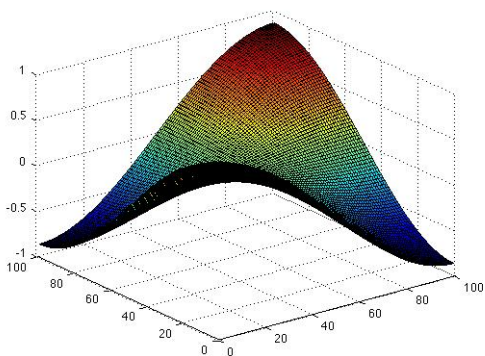
(g)  $t = 1536$  s



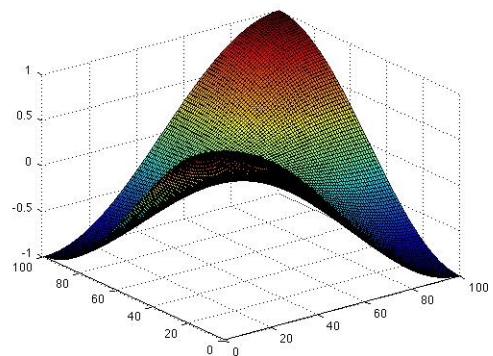
(h)  $t = 1792$  s



(i)  $t = 2048$  s



(j)  $t = 2304$  s



(k)  $t = 2513$  s

Abbildung 5.1: exakter Kurvenverlauf

$p$  ist periodisch und hat eine Periodenlänge von  $800 \cdot \pi (\approx 2513,3)$  Sekunden. Um eine Vorstellung von ihrem Kurvenverlauf zu bekommen, wird dieser in Abbildung 5.1 dargestellt.

### 5.1.2 Äquidistante Diskretisierung

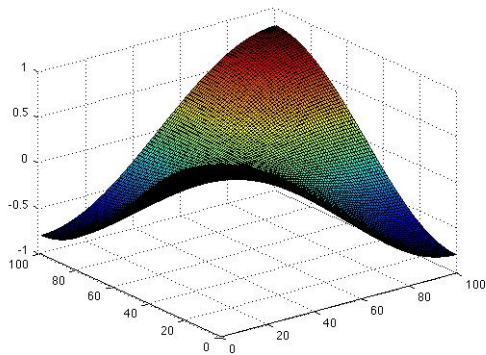
Betrachtet wird ein logisches Rechteckgebiet mit äquidistanter Diskretisierung. Folglich gilt, dass jede Zelle  $\Omega_{i,j}$  quadratisch ist und eine Breite von 100 m hat. Auf diesem Gitter lässt sich nun die Problemstellung (2.15) mit den beiden vorgestellten Verfahren (Mittelpunktregel (4.1) und Störmer-Verlet-Verfahren (4.4)) lösen.

Dazu werden die beiden Verfahren auf die Anfangsbedingungen in Abbildung 5.1a angewendet. Das Störmer-Verlet-Verfahren verwendet dabei eine maximal mögliche Schrittweite von  $\Delta t = 0.2$  s (Schrittweitebeschränkung durch Formel (4.5) auf Seite 35). Bei Anwendung der Mittelpunktregel wird hingegen eine gröbere Schrittweite von  $\Delta t = 0.8$  s verwendet.

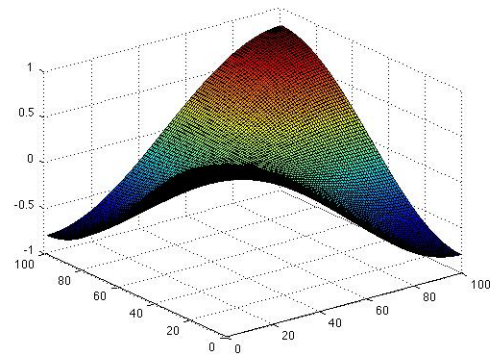
Bevor der Fehler der beiden Verfahren berechnet wird, wird ein Ausschnitt der Entwicklung grafisch (Abb. 5.2) betrachtet. Wie man in der Abbildung erkennen kann, verhalten sich beide Verfahren gleich und es sind kaum Unterschiede in der Entwicklung zu erkennen. Auch ein Vergleich mit der exakten Lösung (Abbildungen 5.1b und 5.1c) lässt optisch keine Unterschiede erkennen.

Die Frage ist nun, wie weit die numerischen Lösungen jeweils von der exakten Lösung abweichen. Zu diesem Zweck wird die Lösung der Verfahren über 10 Perioden lang berechnet. Innerhalb dieses Zeitraums wird alle 4 Sekunden der Fehler zwischen der numerischer und der exakten Lösung bestimmt. Der Fehler wird dabei punktgenau berechnet. Dabei wird die betragsmäßige Abweichung in jedem Diskretisierungspunkt bestimmt und mittels des Flächeninhaltes der jeweiligen Zelle gewichtet. Anschließend werden diese Werte für alle Zellen summiert und durch den Flächeninhalt des Gebietes  $\Omega$  geteilt. Der dabei entstandene Fehler wird im folgenden Punktfehler genannt.

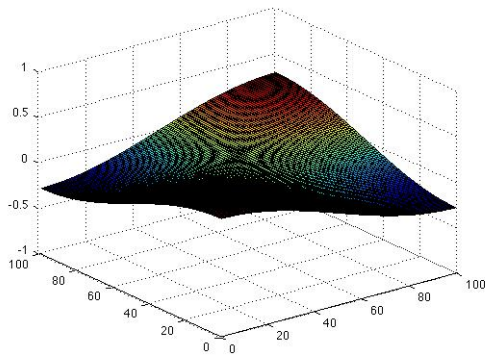
Bei der Berechnung ergibt sich, dass das Störmer-Verlet-Verfahren einen durchschnitt-



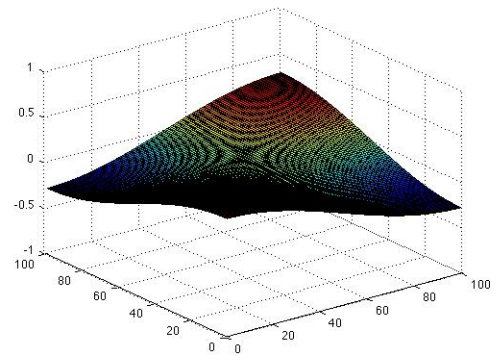
(a) MP:  $t = 256$  s



(b) Verlet:  $t = 256$  s



(c) MP:  $t = 512$  s



(d) Verlet:  $t = 512$  s

Abbildung 5.2: Vergleich Mittelpunkt mit Verlet

lichen Fehler von  $0.560182e^{-4}$  und die Mittelpunkregel einen Fehler von  $0.560104e^{-4}$  hat. Wie sich dabei erkennen lässt, unterscheidet sich der Fehler erst ab der 5-ten Stelle nach dem Komma, weshalb sich sagen lässt, dass beide Verfahren etwa gleich gut (bzw. gleich schlecht) arbeiten.

Um dies zu bestätigen, wurde in jedem Zeitschritt (aller 4 Sekunden über 10 Perioden) die Differenz der Fehler von Mittelpunkregel und Störmer-Verlet-Verfahren gebildet. Anschließend wurde der Mittelwert aller Differenzen (in jedem Zeitschritt) berechnet. Dabei lässt sich erkennen, dass dieser mit zunehmender Laufzeit gegen 0 konvergiert (siehe Abb. 5.3).

Nach 10 Perioden ergab sich für die durchschnittliche Differenz der Fehler ein Wert

von  $0.78228^{-10}$ . Im Vergleich zu den einzelnen Fehlern der Verfahren (Mittelpunktregel  $0.560104e^{-4}$ ; Störmer-Verlet-Verfahren  $0.560182e^{-4}$ ) ist dieser um sechs Exponenten Stellen kleiner, weshalb von einer Konvergenz gegen 0 auszugehen ist.

Abschließend gilt es noch einzuschätzen, wie weit der Fehler der Funktion  $p$  auf  $\Omega$  ins Gewicht fällt. Dazu wird ein Punktwert in  $\Omega$  bestimmt. Dieser berechnet sich analog zum Punktfehler mit dem Unterschied, dass in jedem Diskretisierungspunkt der Wert der exakten Lösung mit dem Flächeninhalt der Zelle gewichtet wird.

Vergleicht man nun den Fehler von Mittelpunktregel und Störmer-Verlet-Verfahren, dann ist grafisch bereits erkennbar, dass dieser keine große Abweichung zur exakten Lösung hat. Dies lässt sich auch rechnerisch bestätigen, indem der Punktfehler der Verfahren ( $0.5601e^{-4}$ ) mit dem Punktwert der exakte Lösung (0.258) verglichen wird. Hierbei ergibt sich, dass der Fehler lediglich  $\frac{1}{4600}$  des Punktwertes ausmacht.

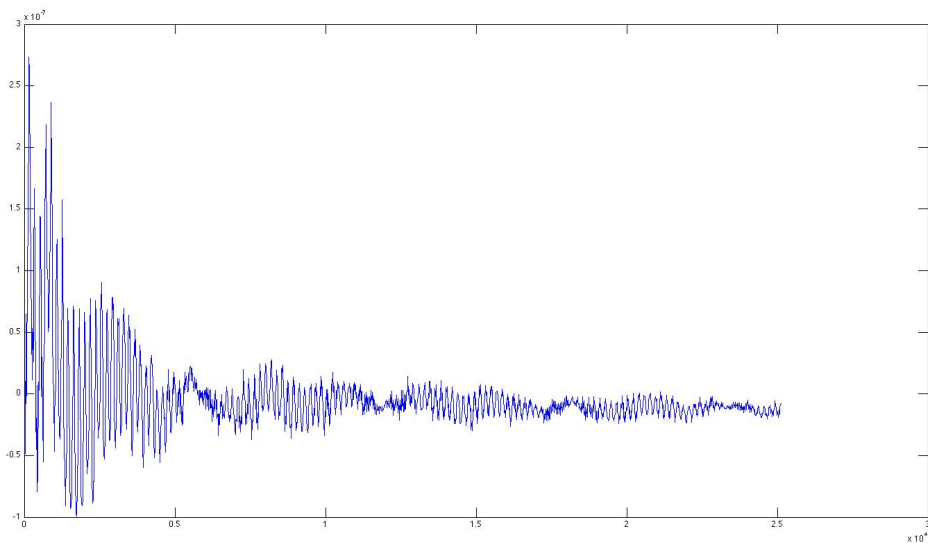


Abbildung 5.3: Fehlerdifferenz MP/Verlet

### 5.1.3 Variable Diskretisierung

Im vorhergehenden Abschnitt wurde ein logisches Gitternetz mit gleich bleibender Diskretisierung betrachtet. Diese Diskretisierung wird in diesem Abschnitt dahingehend

verändert, dass vereinzelte Zeilen, sowie Spalten auf einen Bruchteil der ursprünglichen Größe verkleinert werden. Die Verkleinerung der  $\Delta x_i$  und  $\Delta z_i$  führt zu einer Verkleinerung der entsprechenden Zellen. Diese Änderung der Gittergröße hat einen enormen Einfluss auf die bisher verwendeten Lösungsverfahren.

Bei Anwendung der impliziten Mittelpunkregel muss in jedem Schritt ein Gleichungssystem gelöst werden. Durch die großen Unterschiede in den Gittergrößen wird der Aufwand dafür stark vergrößert, wodurch die Berechnung eines Zeitschrittes exponentiell länger dauert.

Beim Störmer-Verlet-Verfahren ist die Schrittweite durch Formel (4.5) beschränkt. Hierbei ist die Beschränkung von den Zellgrößen abhängig. Das bedeutet, dass durch einige wenige kleine Zellen, auch sämtliche Größeren mit einem deutlich verringerten  $\Delta t$  berechnet werden müssen. Wird beispielsweise das Gitter mit einer Zellengrößen von 100 m x 100 m so verändert, dass einige, wenige Zellen sehr klein (1 m x 1 m groß) sind, dann führt das dazu, dass die Schrittweite um den Faktor 100 geringer ausfällt, als ohne die kleinen Zellen. Somit sind auch 100-mal mehr Berechnungen notwendig, wodurch wiederum der Arbeitsaufwand steigt.

### **Lösung mittels gekoppeltem Verfahren**

Um dieses Problem zu umgehen, wird stattdessen die Kopplung der beiden Verfahren (Mittelpunkregel und Störmer-Verlet-Verfahren) verwendet. Dabei werden die kleinen Zellen implizit, d.h. mit der Mittelpunkregel, und die größeren Zellen explizit, mit Hilfe des Störmer-Verlet-Verfahren bestimmt. Für eine nähere Erläuterung siehe Kapitel 4.3.

Durch diese Teilung in der Berechnung wird erreicht, dass die Beschränkung der Schrittweite nicht mehr von den kleineren Zellen beeinflusst wird, sondern nur noch von den großen Zellen abhängig ist.

Um ein optimales Ergebnis zu erreichen, ist ein Auswahlverfahren notwendig, welches jeder Zelle ein Verfahren zuordnet. Die Auswahlstrategie, die hier verwendet wird, sortiert die Zellen dabei nicht nur nach der Größe aus, sondern auch nach der Lage.

Dabei werden für die implizite Rechnung zuerst die kleinen Zellen herausgesucht. Zu diesen kommen Anschließend noch all diejenigen Zellen hinzu, die mindestens eine Kante mit einer kleinen Zelle gemeinsam haben. Das ist notwendig, da der Einfluss der kleinen Zellen sich auch auf die Kanten auswirkt und somit die benachbarten Zellen indirekt beeinflusst.

Mit diesem neuen Verfahren wird wieder der Fehler zur exakten Lösung untersucht. Zum Vergleich wird das gleiche Problem zusätzlich mit der Mittelpunkregel gelöst. Dabei ergibt sich für die Mittelpunkregel, bei einer Schrittweite von 0.8 s, ein Fehler von  $0.5629e^{-4}$ . Im Gegenzug dazu, liegt der Fehler beim kombinierten Verfahren bei  $1.4749e^{-4}$ .

Wie sich erkennen lässt ist der Fehler des kombinierten Verfahrens etwa 2,5-mal so groß, wie bei der Mittelpunkregel, obwohl die Schrittweite 4-mal genauer ist. Im Vergleich zur exakten Lösung, mit einem durchschnittlichen Punktwert von 0.258, ist der Fehler trotzdem noch verschwindend gering. Betrachtet man darüber hinaus die Rechenzeit beider Verfahren, dann braucht die Mittelpunkregel 6-mal mehr Zeit, um die Lösung zu bestimmen, als das kombinierte Verfahren. Zusammenfassend ist das kombinierte Verfahren besser zu bewerten, da es einen erheblichen Rechenvorteil gegenüber der Mittelpunkregel hat und die Erhöhung des Fehlers nicht wesentlich ist.

## 5.2 Gebiet mit kompliziertem Rand

In diesem Abschnitt wird wiederum ein Gebiet mit äquidistanter Ortsdiskretisierung betrachtet. Aus diesem Gebiet wird mittels der Kurve  $\psi$  ein Teil herausgeschnitten und es entsteht ein Gebiet mit einem komplizierten Rand. Dadurch entstehen Randzellen mit unterschiedlichen Größen und Formen, was die Berechnung erschwert.

### 5.2.1 Piola-Transformation

Zur Definition einer exakten Lösung müssen die Nullrandbedingungen in  $\Omega$  erfüllt sein. Durch den defekten Rand ist dies aber nicht ohne weiteres möglich. Aus diesem Grund wird die exakte Lösung nun auf dem Einheitsquadrat definiert und anschließend nach  $\Omega$  transformiert. Dazu wird die Piola-Transformation verwendet. Im Folgenden wird die Durchführung dieser Transformation an einem Beispiel verdeutlicht, das gleichzeitig die exakte Lösung für den späteren Vergleich mit den numerischen Verfahren darstellt.

#### Transformation in $\Omega_0$

Im ersten Schritt wird eine Transformation von  $\Omega$  nach  $\Omega_0$  durchgeführt:

$$\phi : \begin{cases} \Omega \rightarrow \Omega_0 \\ (x, z) \mapsto (\xi, \zeta) \end{cases} \quad (5.3)$$

$\Omega_0$  ist dabei das Einheitsquadrat. Durch die Funktion  $\phi$  wird eine Transformation beschrieben, die jedem Punkt in  $\Omega$  ein Punkt in  $\Omega_0$  zuweist. In  $\Omega_0$  werden dann die Funktionen  $p_0$ ,  $u_0$  und  $w_0$  wie im Folgenden definiert.

Die Funktion  $p_0$  gibt zu jedem Zeitpunkt  $t$  und in jedem Punkt  $\xi_{i,j}$  des Gebietes  $\Omega_0$  einen Funktionswert an:

$$p_0(\xi, \zeta, t) = \cos(\pi\xi) \cdot \cos(\pi\zeta) \cdot \cos(t) \quad (5.4)$$

In jedem Punkt  $\xi_{i,j}$  des Gebietes  $\Omega_0$  ist ein Fluss, mit Richtung und Stärke, definiert. Dieser Fluss wird in zwei Komponenten  $u_0$  und  $w_0$  aufgeteilt. Die  $u_0$  Komponente fließt dabei in  $\xi$  Richtung und die  $w_0$  Komponente fließt in  $\zeta$  Richtung. Unter Berücksichtigung der Nullrandbedingungen in  $\Omega_0$ , werden die Flüsse  $u_0$  und  $w_0$  wie folgt definiert:

$$\begin{aligned} u_0(\xi, t) &= \sin(2\pi\xi) \cdot \cos(t) \\ w_0(\zeta, t) &= \sin(2\pi\zeta) \cdot \cos(t) \end{aligned} \quad (5.5)$$

## Rücktransformation in $\Omega$

Nun wurden die Funktionen  $p_0$ ,  $u_0$  und  $w_0$  in  $\Omega_0$  definiert und erfüllt dort die Nullrandbedingung. Für die Lösung des Differentialgleichungssystems werden allerdings die Funktionswerte  $p(x, z, t)$ ,  $u(x, t)$  und  $w(z, t)$  benötigt. Das bedeutet, dass die Funktionswerte wieder zurück transformiert werden müssen. Die Funktionswerte von  $p_0(\xi, \zeta, t)$  bleiben dabei unverändert, so dass gilt:

$$p(x, z, t) := p_0(\xi, \zeta, t), \quad \text{mit} \quad \begin{array}{l} \xi = \phi(x) \\ \zeta = \phi(z) \end{array} \quad (5.6)$$

Die Flüsse in den Punkten  $\xi_{i,j}$  lassen sich hingegen nicht einfach übernehmen, da sonst die Nullrandbedingungen nicht mehr erfüllt sind. Aus diesem Grund wird die Rücktransformation mit Hilfe der Piola-Transformation durchgeführt. Dadurch bleiben die Nullrandbedingung in normalen Richtung auch nach der Transformation erhalten.

$$\begin{bmatrix} u(x, t) \\ w(z, t) \end{bmatrix} := \frac{1}{\det(J)} J(\xi, \zeta) \cdot \begin{bmatrix} u_0(\xi, t) \\ w_0(\zeta, t) \end{bmatrix}, \quad \text{mit} \quad \begin{array}{l} \xi = \phi(x) \\ \zeta = \phi(z) \end{array} \quad (5.7)$$

Die Matrix  $J$  in Gleichung (5.7) ist dabei die Jacobi-Matrix der Umkehrabbildung von  $\phi$ .

### 5.2.2 Fehleranalyse

Wie im Kapitel 3.2 bereits erwähnt wurde, gibt es für ein Gitter mit defektem Rand verschiedene Varianten den Gradienten  $G$  zu bestimmen. Dabei wurden drei Möglichkeiten vorgestellt, für die im Folgenden der Fehler bzgl. der exakten Lösung bestimmt wird.

Als Grundlage dafür dient wieder das quadratische Gebiet  $\Omega$  aus 5 auf Seite 37 mit einer Kantenlänge von 10 km und einer äquidistanten Diskretisierung in 100 Zeilen x 100 Spalten. Darüber hinaus wird eine Kurve  $\psi$  durch das Gebiet gelegt, die definiert

## 5.2. Gebiet mit kompliziertem Rand

---

ist durch:

$$\psi(x) = \frac{h}{1 + \frac{x-x_0}{a}} \quad (5.8)$$

Dabei gilt für die Konstanten:  $h = 3005$  m,  $x_0 = 5000$  m und  $a = 2000$  m. Betrachtet man die grafische Darstellung des Gebiets, dann sieht die Diskretisierung mit Anfangsbedingung wie folgt aus:

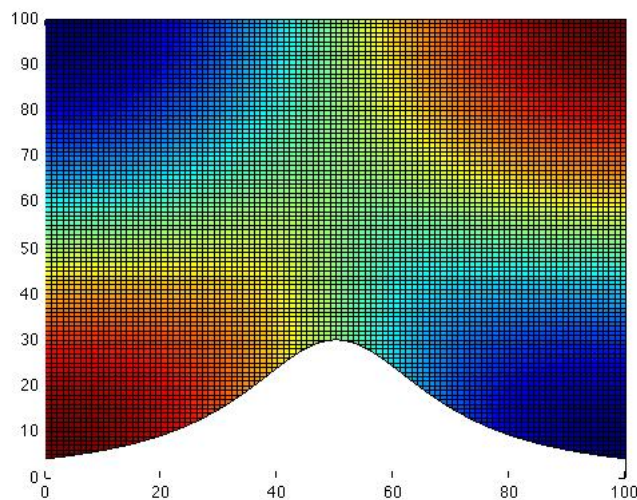


Abbildung 5.4: gestörtes Gitter

Dieses Gebiet entspricht dem aus Abbildung 5.1a auf Seite 39. Der Unterschied hierbei ist, dass die Funktionswerte durch die Kurve  $\psi$  in das Gebiet „hineingeschoben“ wurden. Somit hat dieses Beispiel ebenfalls eine Periodenlänge von 2513 Sekunden und weist einen Punktwert von 0.258 auf.

Löst man nun das Differentialgleichungssystem für dieses Gebiet über eine Periodenlänge, dann ergibt sich, dass das MPFA-Verfahren im Vergleich am besten abschneidet. Dieses hat einen Punktfehler von  $4.6723e^{-04}$ , der etwa 10-mal so groß ist, wie im ungestörten, äquidistanten Gitternetz. Im Vergleich zum durchschnittlichen Punktwert von 0.258 ist dies aber immer noch recht klein.

Bestimmt man die Lösung mittels des Gradienten, welcher sich aus der Divergenz herleitet, fällt der Fehler etwa doppelt so hoch aus wie beim MPFA-Verfahren. Dabei muss

allerdings berücksichtigt werden, dass für die Bestimmung des Gradienten kein Rechenaufwand notwendig ist. Der Gradient mit der MPFA-Methode muss hingegen für jedes Gebiet separat bestimmt werden. Im verwendeten Beispiel war die Rechendauer zur Bestimmung des Gradienten, im Vergleich zur Bestimmung der numerischen Lösung, zeitlich vernachlässigbar. Zusammenfassend bedeutet dies, dass der Gradient, welcher sich aus der Divergenz herleitet, zwar einen geringen, zeitlichen Vorteil hat, dies aber nicht die Halbierung der Genauigkeit rechtfertigt.

Die letzte Variante zur Bestimmung des Gradienten liefert, im Vergleich zum MPFA-Verfahren, eine schlechtere Genauigkeit, wobei der entstandene Fehler 7-mal größer ausgefallen ist.

# Kapitel 6

## Implementation in Matlab

Dieses Kapitel befasst sich mit der Umsetzung der Problemlösung in der Programmiersprache Matlab. Dabei wird auf alle notwendigen Eingabeparameter, sowie die wichtigsten Funktionen und Programmabläufe eingegangen. Es sei darauf hingewiesen, dass aufgrund des Umfangs, die Implementation nicht vollständig aufgeführt bzw. erläutert werden kann. Stattdessen wird im Nachstehenden nur auf die wichtigsten Methoden eingegangen, die für das Verständnis des Programmes notwendig sind.

### 6.1 Eingabeparameter

Damit mittels des Programms eine Lösung für das Differentialgleichungsproblem bestimmt werden kann, müssen verschiedene Parameter gegeben sein. Dazu gehören die Gebiets- und Diskretisierungsparameter, sowie die Anfangswerte für die Zellen und Kanten.

Die Gebietsparameter sind feste Größen, die durch die natürliche Umgebung bestimmt werden. Dazu gehört die Beschreibung des Gebietes  $\Omega$ , das durch ein rechteckiges Gebiet mit den Längen  $\Delta x$  und  $\Delta z$  festgelegt wird. Natürliche Hindernisse, wie z.B Berge, werden dabei durch die Funktion  $\psi$  beschrieben. Des Weiteren muss die Ausbreitungsgeschwindigkeit des Druckes  $c$  vorgegeben werden.

Neben den Gebietsparametern müssen die Werte für die Diskretisierung vorgegeben sein. Da im vorliegenden Fall das Gebiet in Rechteckzellen diskretisiert wird, muss die Anzahl  $n_x$ ,  $n_z$  der Spalten bzw. Zeilen, sowie die zugehörigen Breiten  $\Delta x_i$ ,  $\Delta z_j$  vorgegeben werden. Für die Diskretisierung in der Zeit ist daneben eine Schrittweite  $\Delta t$  anzugeben.

Zum Schluss müssen die Anfangswerte für den Druck in den Zellen vorgegeben werden. Die Flüsse zwischen diesen müssen nicht notwendigerweise bekannt sein, sondern können stattdessen mit dem Wert 0 initialisiert werden.

## 6.2 Programmaufbau

In diesem Abschnitt wird ein Überblick über die verschiedenen Phasen des Programmes gegeben.

### 6.2.1 Gebietsbestimmung

Nachdem die Eingabeparameter angelegt wurden, werden aus diesen die benötigten Gebietsvariablen bestimmt. Diese sind für das Gitter stets gleich, das heißt, bei gleichem Gebiet und gleichbleibender Ortsdiskretisierung liefert der nachfolgende Code die gleichen Ergebnisse. Somit fällt für ein bestimmtes Gebiet nur einmal diese Rechnung bzw. der damit verbundene Rechenaufwand an. Das Ergebnis kann anschließend abgespeichert und bei nochmaliger Rechnung auf diesem Gebiet, neu geladen werden.

```
12 %% Gebietsparameter
13
14 % 1. Gitter ueber Gebiet (OHNE levelfun)
15 [WeiFU,WeiFW] = randWeight;
16 [x,z,gx,gz] = Grid(WeiFW(1,:),WeiFU(:,1));
17
18 % 2. Schnittpunkte mit "levelfun" bestimmen
19 [SPx,SProw,SPcol] = SPLevelFun(gx,gz);
20
```

```

21 % 3. Einfluss "levelfun" auf Gewichte und Gitter
22 [WeifUc,WeifWc] = ...
23     cutWeifun(WeifU,WeifW,gx,gz,SPx,SProw,SPcol);
24 [Xq,Rq,Xf,Rf,PolyMidP,PolyMidPU] = ...
25     PolygonGrid(gx,gz,SPx,SProw,SPcol);
26
27 % 4. Volumina des Gitternetztes bestimmen
28 Vol = VolMatrix(gx,gz,SPx(:,1),SProw,SPcol);
29 VolU = VolUMatrix(gx,gz,SPx(:,1),SPx(:,2),...
30     SProw,SPcol,WeifUc,WeifWc);

```

Im ersten Programmabschnitt (siehe oben) wird über das Gebiet  $\Omega$  das Gitternetz gelegt. Dabei werden die Kantengewichte des Gitters (=Länge der jeweiligen Kanten), sowie die Koordinaten der Gitterpunkte bestimmt. Im zweiten Schritt wird die Funktion  $\psi$  (=levelfun im Programmcode) durch das Gitter gezogen. Dadurch entstehen Schnittpunkte mit dem Gitter, welche berechnet werden müssen. Mit Hilfe dieser Schnittpunkte wird im dritten Schritt der Einfluss auf die Kantengewichte bestimmt. Darüber hinaus wird ein Polygon-Gitter erstellt, welches zur Veranschaulichung die Kanten der Funktion  $\psi$  grafisch dargestellt wird. Im vierten und letzten Schritt der Gebietsbestimmung werden die Flächeninhalte der Zellen sowie die dualen Flächeninhalte zu den Kanten bestimmt.

## 6.2.2 Gradient und Divergenz

Nachdem das Gitter angelegt wurde, werden die Matrizen aus dem Differentialgleichungssystem (2.15) erstellt. Das heißt, es wird die Divergenzmatrix, sowie die Gradientenmatrix angelegt. Dabei stehen für die Berechnung der Gradientenmatrix drei Varianten zur Verfügung, aus denen eine ausgewählt werden muss.

```

32 %% Divergenz und Gradient
33
34 % 1. Divergenz
35 Dz = DivMatrix(WeifUc,WeifWc);
36 D = SolveGlsB(Vol,Dz); %D = -cS*inv(Vol) *Dz
37

```

```

38 % 2. Gradient
39 switch useG
40     case 1
41         % 2.1) Grad = -Div^T
42         G = SolveGlsA(VolU,Dz'); %G= -cS*inv(VolU)*(-1)*Dz'
43
44     case 2
45         % 2.2) MPFA (Multipoint Flux Approximation)
46         G = -cS*GradMPFA(gx,gz,WeiFUc,WeiFWc,PolyMidP);
47
48     case 3
49         % 2.3) p lebt in Zellenmittelpunkt
50         G = Grad3(WeiFU,WeiFW,VolU);
51
52     otherwise
53         disp('Gradient ex. nicht. ');
54 end

```

Die Berechnung des Gradienten nach der MPFA-Methode (=case 2) hat sich im vorhergehenden Kapitel als die Beste heraus gestellt, weshalb hierfür der Quellcode in Anhang A.2 auf Seite 57 angegeben wird. Darüber hinaus wird in Anhang A.1 auf Seite 56 der Quellcode zur Divergenzberechnung bereitgestellt.

### 6.2.3 Verfahren und Lösung

In den vorhergehenden Abschnitten wurden alle notwendigen Matrizen für das Differentialgleichungssystem (2.15) aufgestellt. Um dieses nun zu lösen wird im Folgenden eines der gegebenen Verfahren aus Kapitel 4 ausgewählt. Für das gewählte Verfahren werden dazu weitere Parameter bestimmt. Diese umfassen spezielle Matrizen, die sich aus den Parametern des Gitters zusammensetzen, sowie die Schrittweite  $\Delta t$ . Daneben werden die Anfangsbedingungen angelegt und in einem Vektor  $r$  gespeichert.

```

1 function calcProc_param(G,D,WeiFUc,WeiFWc,Vol)
2 % legt Parameter fuer die verwendeten Verfahren fest
3 global useMidP useStVer useCompS;
4

```

```
5
6 % 3.1) Mid Point rule
7 if useMidP
8     calcMidP_param(G,D);
9 end
10
11 % 3.2) Stoermer-Verlet
12 if useStVer
13     calcSymp1_param(WeiFUc,WeiFWc);
14 end
15
16 % 3.3) Component Splitting
17 if useCompS
18     calcCompS_param(G,D,Vol,WeiFUc,WeiFWc);
19 end
20
21 %% Anfangsbedingungen
22 r(:,1) = funrexakt(F1exakt,0);
```

Mit Hilfe dieser Parameter kann anschließend das Problem gelöst werden. Dazu wird die entsprechende Funktion des verwendeten Verfahrens aufgerufen. Diese wird dann so oft ausgeführt, bis die entsprechende Schrittweite erreicht wurde.

```
166 % Mid Point rule
167     if useMidP == true
168         rMidP = calcMidP(rMidP,F1,F2,(iter-1)*dt);
169
170
171 % Stoermer-Verlet
172     if useStVer == true
173         rSymp1 = calcSymp1(rSymp1,A,B,F1,F2,(iter-1)*dt);
174
175
176 % Component splitting
177     if useCompS == true
178         rCompS=calcCompS(rCompS,A,B,F1,F2,(iter-1)*dt);
```

# Kapitel 7

## Schlusswort

In der vorliegenden Arbeit wurde die numerische Lösung der zweidimensionalen Wellengleichung auf einem defekten Gebiet  $\Omega$  bestimmt.

Dieses Gebiet ist dabei durch einen komplizierten Rand charakterisiert. Dieser Rand beeinflusst die Diskretisierung des Gebietes soweit, dass Zellen unterschiedlicher Größe entstehen. Diese Größenunterschiede beeinflussen die Lösbarkeit und den Rechenaufwand der verwendeten numerischen Verfahren. Durch geschickte Kopplung, der impliziten Mittelpunkregel und des expliziten Störmer-Verlet-Verfahrens, lässt sich ein Verfahren erzeugen, welches die Nachteile der einzelnen Verfahren ausgleicht.

Das gekoppelte Verfahren führt eine Aufteilung der Zellen durch, wobei diese entweder dem expliziten oder dem impliziten Verfahren zugeordnet werden. Durch geschickte Verteilung kann damit die Schrittweitenbeschränkung des expliziten Verfahrens deutlich verbessert und der Rechenaufwand minimiert werden. Die Aufteilung der Zellen muss dabei nicht per Hand durchgeführt werden, sondern lässt sich anhand der Größe der einzelnen Gitterzellen berechnen.

Dadurch erhält man ein numerisches Verfahren, welches trotz geringer Einbuße in der Genauigkeit, eine numerische Lösung in deutlich besserer Zeit, als ein einzelnes Verfahren, und mit nur gering größerem Aufwand liefert.

# Anhang A

## Programme

### A.1 Divergenz

```
1 function Div=DivMatrix(WeiFU,WeiFW)
2 %Divergenzmatrix bestimmen
3 global nx;
4
5 nn=nx(1)*nx(2);
6 nU=nn;
7 nW=nn;
8 mm=nU+nW;
9 ia=zeros(4*nn,1);
10 ja=zeros(4*nn,1);
11 sa=zeros(4*nn,1);
12
13 i=1;
14 for ix=2:nx(1)
15     for iz=1:nx(2)
16         icol=(ix-1)+(iz-1)*nx(1);
17         irow=(ix-1)+(iz-1)*nx(1);
18
19         ia(i)=irow;
20         ja(i)=icol;
21         sa(i)=WeiFU(iz,ix);
22         i=i+1;
23
24         ia(i)=irow+1;
```

```

25     ja(i)=icol;
26     sa(i)=-WeiFU(iz,ix);
27     i=i+1;
28 end
29 end
30
31 for ix=1:nx(1)
32     for iz=2:nx(2)
33         icol=nU+ix+(iz-2)*nx(1);
34         irow=ix+(iz-2)*nx(1);
35
36         ia(i)=irow;
37         ja(i)=icol;
38         sa(i)=WeiFW(iz,ix);
39         i=i+1;
40
41         ia(i)=irow+nx(1);
42         ja(i)=icol;
43         sa(i)=-WeiFW(iz,ix);
44         i=i+1;
45     end
46 end
47
48 Div=sparse(ia(1:i-1),ja(1:i-1),sa(1:i-1),nn,mm);
49 end

```

## A.2 Gradient-MPFA

```

1 function G = GradMPFA(gx,gz,WeiFUc,WeiFWc,PolyMidP)
2 eps = 10^-4;
3
4 nx = length(gx)-1;
5 nz = length(gz)-1;
6 nn=nx*nz;
7
8 G = zeros(2*nx*nz,nx*nz);
9
10 %Spaltenweise Betrachtung der Punkte
11 for ix=0:nx
12     for iz=0:nz

```

```

13     % mittlere Zellen um [ix+0.5, iz+0.5]
14     ibl = ix+nx*(iz-1); % index: bottom left
15     ibr = ibl + 1;      % index: bottem right
16     itl = ibl + nx;    % index: top left
17     itr = ibr + nx;    % index: top right
18
19     x = zeros(4,2);     % Mittelpunkte
20     e = zeros(0,1);    % Kantenlaengen
21     k = zeros(0,2);    % Mittelpunkt Kante
22     n = zeros(0,2);    % Normalen auf Kante
23
24     ci = zeros(4);
25     cf = zeros(4);
26     ik = 1;
27
28     if (ix == 0 || ix == nx) ...
29         && (iz == 0 || iz ==nz)           % Ecken
30         % nichts zu berechnen
31
32     elseif ix == 0 && iz > 0 && iz < nz    % Rand-links
33         if PolyMidP(1,ibr)~=0 && (WeiFUc(iz,ix+1)~=0 ...
34             || WeiFWc(iz+1,ix+1)~=0)
35             x(ik,:) = PolyMidP(:,ibr)';
36             [e,k,n] = MPFA_ibr(x(ik,:),e,k,n,WeiFUc,...
37                 WeiFWc,gx,gz,ix,iz,eps);
38
39             ci(ik) = ibr;      cf(ik) = ibl;
40             ik = ik+1;
41         end
42
43     if PolyMidP(1,itr)~=0 && (WeiFWc(iz+1,ix+1)~=0 ...
44         || WeiFUc(iz+1,ix+1)~=0)
45         x(ik,:) = PolyMidP(:,itr)';
46         [e,k,n] = MPFA_itr(x(ik,:),e,k,n,WeiFUc,...
47             WeiFWc,gx,gz,ix,iz,eps);
48
49         ci(ik) = itr;      cf(ik) = nn+ibr;
50         ik = ik+1;
51     end
52
53     elseif ix == nx && iz > 0 && iz < nz    % Rand-rechts

```

```

54         if PolyMidP(1,itl)~=0 && (WeiFUc(iz+1,ix+1)~=0 ...
55                                 || WeiFWc(iz+1,ix)~=0)
56             x(ik,:) = PolyMidP(:,itl)';
57             [e,k,n] = MPFA_itl(x(ik,:),e,k,n,WeiFUc,...
58                             WeiFWc,gx,gz,ix,iz,eps);
59
60             ci(ik) = itl;         cf(ik) = itl;
61             ik = ik+1;
62         end
63
64         if PolyMidP(1,ibl)~=0 && (WeiFWc(iz+1,ix)~=0 ...
65                                 || WeiFUc(iz,ix+1)~=0)
66             x(ik,:) = PolyMidP(:,ibl)';
67             [e,k,n] = MPFA_ibl(x(ik,:),e,k,n,WeiFUc,...
68                             WeiFWc,gx,gz,ix,iz,eps);
69
70             ci(ik) = ibl;         cf(ik) = nn+ibl;
71             ik = ik+1;
72         end
73
74     elseif iz == 0 && ix > 0 && ix < nx      % Rand-unten
75         if PolyMidP(1,itr)~=0 && (WeiFWc(iz+1,ix+1)~=0 ...
76                                 || WeiFUc(iz+1,ix+1)~=0)
77             x(ik,:) = PolyMidP(:,itr)';
78             [e,k,n] = MPFA_itr(x(ik,:),e,k,n,WeiFUc,...
79                             WeiFWc,gx,gz,ix,iz,eps);
80
81             ci(ik) = itr;         cf(ik) = nn+ibr;
82             ik = ik+1;
83         end
84
85         if PolyMidP(1,itl)~=0 && (WeiFUc(iz+1,ix+1)~=0 ...
86                                 || WeiFWc(iz+1,ix)~=0)
87             x(ik,:) = PolyMidP(:,itl)';
88             [e,k,n] = MPFA_itl(x(ik,:),e,k,n,WeiFUc,...
89                             WeiFWc,gx,gz,ix,iz,eps);
90
91             ci(ik) = itl;         cf(ik) = itl;
92             ik = ik+1;
93         end
94

```

```

95     elseif iz == nz && ix > 0 && ix < nx      % Rand-oben
96         if PolyMidP(1,ibl)~=0 && (WeiFWc(iz+1,ix)~=0 ...
97             || WeiFUc(iz,ix+1)~=0)
98             x(ik,:) = PolyMidP(:,ibl)';
99             [e,k,n] = MPFA_ibl(x(ik,:),e,k,n,WeiFUc,...
100                 WeiFWc,gx,gz,ix,iz,eps);
101
102             ci(ik) = ibl;      cf(ik) = nn+ibl;
103             ik = ik+1;
104         end
105
106         if PolyMidP(1,ibr)~=0 && (WeiFUc(iz,ix+1)~=0 ...
107             || WeiFWc(iz+1,ix+1)~=0)
108             x(ik,:) = PolyMidP(:,ibr)';
109             [e,k,n] = MPFA_ibr(x(ik,:),e,k,n,WeiFUc,...
110                 WeiFWc,gx,gz,ix,iz,eps);
111
112             ci(ik) = ibr;      cf(ik) = ibl;
113             ik = ik+1;
114         end
115
116     else                                     % Zellen-mitte
117         if PolyMidP(1,ibr)~=0 && (WeiFUc(iz,ix+1)~=0 ...
118             || WeiFWc(iz+1,ix+1)~=0)
119             x(ik,:) = PolyMidP(:,ibr)';
120             [e,k,n] = MPFA_ibr(x(ik,:),e,k,n,WeiFUc,...
121                 WeiFWc,gx,gz,ix,iz,eps);
122
123             ci(ik) = ibr;      cf(ik) = ibl;
124             ik = ik+1;
125         end
126
127         if PolyMidP(1,itr)~=0 && (WeiFWc(iz+1,ix+1)~=0 ...
128             || WeiFUc(iz+1,ix+1)~=0)
129             x(ik,:) = PolyMidP(:,itr)';
130             [e,k,n] = MPFA_itr(x(ik,:),e,k,n,WeiFUc,...
131                 WeiFWc,gx,gz,ix,iz,eps);
132
133             ci(ik) = itr;      cf(ik) = nn+ibr;
134             ik = ik+1;
135         end

```

```

136
137     if PolyMidP(1,itl)~=0 && ...
138         (WeiFUc(iz+1,ix+1)~= 0 || WeiFWc(iz+1,ix)~=0)
139         x(ik,:) = PolyMidP(:,itl)';
140         [e,k,n] = MPFA_itl(x(ik,:),e,k,n,WeiFUc,...
141             WeiFWc,gx,gz,ix,iz,eps);
142
143         ci(ik) = itl;         cf(ik) = itl;
144         ik = ik+1;
145     end
146
147     if PolyMidP(1,ibl)~=0 && ...
148         (WeiFWc(iz+1,ix)~= 0 || WeiFUc(iz,ix+1)~=0)
149         x(ik,:) = PolyMidP(:,ibl)';
150         [e,k,n] = MPFA_ibl(x(ik,:),e,k,n,WeiFUc,...
151             WeiFWc,gx,gz,ix,iz,eps);
152
153         ci(ik) = ibl;         cf(ik) = nn+ibl;
154         ik = ik+1;
155     end
156 end
157
158 x = x(1:ik-1,:);
159 ci = ci(1:ik-1);
160 cf = cf(1:ik-1);
161
162 % Flussberechnung
163 if length(ci)>=2
164     f = MPFA(x,k,e,n);
165
166     if length(f(1,:)) == 4 && length(ci) == 4
167         % geschlossen
168         for jf=1:length(cf)
169             for jc = 1: length(ci)
170                 G(cf(jf),ci(jc)) = ...
171                     G(cf(jf),ci(jc)) + f(jf,jc)/(2*e(jf));
172             end
173         end
174     else
175         % offen
176         for jf=2:length(cf)

```

```
177         for jc = 1: length(ci)
178             G(cf(jf),ci(jc)) = ...
179                 G(cf(jf),ci(jc)) + f(jf,jc)/(2*e(jf));
180         end
181     end
182 end
183 end
184 end
185 end
186
187 [ia,ja,sa]=find(G);
188 i=length(sa);
189 G=sparse(ia(1:i),ja(1:i),sa(1:i),2*nx*nz,nx*nz);
190 end
191
192
193 %% Hilfsfunktion
194 function [e,k,n] = ...
195     MPFA_ibr(x,e,k,n,WeiFUc,WeiFWc,gx,gz,ix,iz,eps)
196
197 %Kante 1
198 if WeiFUc(iz,ix+1) ~=0 && isempty(k(:,1))
199     % erste Kante der ersten Zelle
200     e = [e; WeiFUc(iz,ix+1)];
201     kh(1,1) = gx(ix+1);
202     kh(1,2) = gz(iz+1) - 0.5*WeiFUc(iz,ix+1);
203
204     k = [k;kh];
205     n = [n;[1 0]];
206 elseif WeiFUc(iz,ix+1) ~=0
207     % erste Kante aber vorherige Zelle ex.
208     %     -> Kante bereits eingeplant
209 else
210     e = [e; 10];
211     kh(1,1) = levelfunSPz(gx(ix+1),x(1),x(2),eps);
212     kh(1,2) = x(2);
213
214     k = [k;kh];
215     n = [n;[1 0]];
216 end
217
```

```

218 %Kante 2
219 e = [e; WeiFWc(iz+1,ix+1)];
220 if WeiFUc(iz,ix+1)~=0
221     kh(1,1) = gx(ix+1) + 0.5*WeiFWc(iz+1,ix+1);
222 else
223     kh(1,1) = gx(ix+2) - 0.5*WeiFWc(iz+1,ix+1);
224 end
225
226 kh(1,2) = gz(iz+1);
227
228 k = [k;kh];
229 n = [n;[0 1]];
230 end
231
232
233 function [e,k,n] = ...
234     MPFA_itr(x,e,k,n,WeiFUc,WeiFWc,gx,gz,ix,iz,eps)
235
236 %Kante 1
237 if WeiFWc(iz+1,ix+1) ~=0 && isempty(k(:,1))
238     % erste Kante der ersten Zelle
239     e = [e; WeiFWc(iz+1,ix+1)];
240     if WeiFUc(iz+1,ix+1)== (gz(iz+2)-gz(iz+1))
241         kh(1,1) = gx(ix+1) + 0.5*WeiFWc(iz+1,ix+1);
242     else
243         kh(1,1) = gx(ix+2) - 0.5*WeiFWc(iz+1,ix+1);
244     end
245     kh(1,2) = gz(iz+1);
246
247     k = [k;kh];           n = [n;[0 1]];
248 elseif WeiFWc(iz+1,ix+1) ~=0
249     % erste Kante aber vorherige Zelle ex.
250     % -> Kante bereits eingeplant
251 else
252     e = [e; 10];
253     k = zeros(1,2);
254     k(1,1) = x(1);
255     k(1,2) = levelfun(x(1));
256
257     n = zeros(1,2);
258     n(1,:) = [0 1];

```

```
259 end
260
261 %Kante 2
262 if WeiFUc(iz+1,ix+1) ~=0
263     e = [e; WeiFUc(iz+1,ix+1)];
264     kh(1,1) = gx(ix+1);
265     kh(1,2) = gz(iz+2) - 0.5*WeiFUc(iz+1,ix+1);
266
267     k = [k;kh];
268     n = [n;[1 0]];
269 else
270     e = [e; 10];
271     kh(1,1) = levelfunSPz(gx(ix+1),x(1),x(2),eps);
272     kh(1,2) = x(2);
273
274     k = [k;kh];
275     n = [n;[1 0]];
276 end
277 end
278
279
280 function [e,k,n] = ...
281     MPFA_itl(x,e,k,n,WeiFUc,WeiFWc,gx,gz,ix,iz,eps)
282
283 %Kante 1
284 if WeiFUc(iz+1,ix+1) ~=0 && isempty(k(:,1))
285     % erste Kante der ersten Zelle
286     e = [e; WeiFUc(iz+1,ix+1)];
287     kh(1,1) = gx(ix+1);
288     kh(1,2) = gz(iz+2) - 0.5*WeiFUc(iz+1,ix+1);
289
290     k = [k;kh];
291     n = [n;[1 0]];
292
293 elseif WeiFUc(iz+1,ix+1) ~=0
294     % erste Kante aber vorherige Zelle ex.
295     % -> Kante bereits eingeplant
296
297 else
298     e = [e; 10];
299     k = zeros(1,2);
```

```
300     k(1,1) = levelfunSPz(x(1),gx(ix+1),x(2),eps);
301     k(1,2) = x(2);
302
303     n = zeros(1,2);
304     n(1,:) = [1 0];
305 end
306
307 %Kante 2
308 if WeiFWc(iz+1,ix) ~=0
309     e = [e; WeiFWc(iz+1,ix)];
310     if WeiFUc(iz+1,ix+1) == (gz(iz+2)-gz(iz+1))
311         kh(1,1) = gx(ix+1) - 0.5*WeiFWc(iz+1,ix);
312     else
313         kh(1,1) = gx(ix) + 0.5*WeiFWc(iz+1,ix);
314     end
315
316     kh(1,2) = gz(iz+1);
317
318     k = [k;kh];
319     n = [n;[0 1]];
320 else
321     e = [e; 10];
322     kh(1,1) = x(1);
323     kh(1,2) = levelfun(x(1));
324
325     k = [k;kh];
326     n = [n;[0 1]];
327 end
328 end
329
330
331 function [e,k,n] = ...
332     MPFA_ib1(x,e,k,n,WeiFUc,WeiFWc,gx,gz,ix,iz,eps)
333
334 %Kante 1
335 if WeiFWc(iz+1,ix) ~=0 && isempty(k(:,1))
336     % erste Kante der ersten Zelle
337     e = [e; WeiFWc(iz+1,ix)];
338     if WeiFUc(iz,ix+1)~=0
339         kh(1,1) = gx(ix+1) - 0.5*WeiFWc(iz+1,ix);
340     else
```

```
341         kh(1,1) = gx(ix) + 0.5*WeiFWc(iz+1,ix);
342     end
343     kh(1,2) = gz(iz+1);
344
345     k = [k;kh];
346     n = [n;[0 1]];
347
348 elseif WeiFWc(iz+1,ix) ~=0
349     % erste Kante aber vorherige Zelle ex.
350     %     -> Kante bereits eingeplant
351
352 else
353     e = [e; 10];
354     k = zeros(1,2);
355     k(1,1) = x(1);
356     k(1,2) = levelfun(x(1));
357
358     n = zeros(1,2);
359     n(1,:) = [0 1];
360 end
361
362 %Kante 2
363 if WeiFUc(iz,ix+1) ~=0 && length(k(:,1)) == 4
364     % bereits eingefuegt
365 elseif WeiFUc(iz,ix+1) ~=0
366     e = [e; WeiFUc(iz,ix+1)];
367     kh(1,1) = gx(ix+1);
368     kh(1,2) = gz(iz+1) - 0.5*WeiFUc(iz,ix+1);
369
370     k = [k;kh];
371     n = [n;[1 0]];
372 else
373     e = [e; 10];
374     kh(1,1) = levelfunSPz(x(1),gx(ix+1),x(2),eps);
375     kh(1,2) = x(2);
376
377     k = [k;kh];
378     n = [n;[1 0]];
379 end
380 end
```

# Literaturverzeichnis

- [1] Marc Theodoor Dröge. „Cartesian Grid Methods for Turbulent Flow Simulation in Complex Geometries“. Dissertation. University of Groningen, Jan. 2007. ISBN: 90-367-2904-1.
- [2] Luis Eça, M Hoekstra, A Hay und D Pelletier. „A manufactured solution for a two-dimensional steady wall-bounded incompressible turbulent flow“. In: *International Journal of Computational Fluid Dynamics* 21.3-4 (2007), Seiten 175–188.
- [3] Mathias Erfurth. „Diskretisierung des viskosen Terms der 2-Dimensionalen Navier-Stokes Gleichungen auf logischen Rechteckgittern“. Diplomarbeit. Leipzig: Hochschule für Technik, Wirtschaft und Kultur Leipzig, 17. März 2011.
- [4] J.G.Verwer. „Component splitting for semi-discrete Maxwell equations“. In: *BIT Numerical Mathematics* 51.2 (1. Juni 2011), Seiten 427–445.
- [5] Helmut Podhaisky Karl Strehmel Rüdiger Weiner. *Numerik gewöhnlicher Differentialgleichungen. 2.*, überarbeitete und erweiterte Auflage. Wiesbaden: Vieweg+Teubner Verlag, 2012. ISBN: 978-3-8348-1847-8.
- [6] Ken Mattsson Martin Almquist Ilkka Karasalo. „Atmospheric Sound Propagation Over Large-Scale Irregular Terrain“. In: *Journal of Scientific Computing* 61 (14. Feb. 2014), Seiten 369–397.
- [7] Joseph B. Klemp William A. Gallus. „Behavior of Flow over Step Orography“. In: *Monthly Weather Review* 128.4 (2000), Seite 1153.

## Eidesstattliche Erklärung

Hiermit erkläre ich, dass ich die vorliegende Masterarbeit „Numerische Lösung der linearen Wellengleichung in Gebieten mit kompliziertem Rand“ selbstständig und nur unter der Verwendung der angegebenen Hilfsmittel und Literatur angefertigt, sowie Zitate bzw. die sinngemäße Wiedergabe von Zitaten kenntlich gemacht habe.

Ich versichere, dass die vorliegende Masterarbeit nicht anderweitig als Prüfungsleistung verwendet wurde und noch nicht veröffentlicht worden ist.

Leipzig, der 24. November 2014

Johannes Heun