Numerische Mechanik I

UNIKASSEL
VERSITÄT

Universität Kassel
Institut für Baustatik und Baudynamik
FB14 - Bauingenieur- und Umweltingenieurwesen

Inhaltsverzeichnis

UNIKASSEL
VERSITÄT

Anwendung Finite-Volumen-Verfahren auf die zweidimensionaler Flachwassergleichungen

Oleg Popov

1 Einleitung

In diese Arbeit wird das gesamte Wissen meiner Numerik Vorlesungen und Nebenfachvorlesungen dazu verwendet um zweidimensionale Flachwassergleichungen mit Finite-Volumen-Verfahren zu lösen. Bei Verfassung der Arbeit wurde versucht, so einfach wie möglich das gelehrte Wissen zu erklären und anschließend bei der Programmierung anzuwenden. Daher wurden hierzu aufgrund der Kapazität des Rahmen in dieser Arbeit nicht alle Einzelheiten geklärt. Im ersten Kapitel werden die zweidimensionale Flachwassergleichungen definiert. Anschließend werden die konservativen Größen in der Gleichung erklärt. In den weiteren Abschnitten wird vermittelt, wie man das Finite-Volumen-Verfahren auf diese Gleichungen anwenden kann. Für diese Arbeit wurde auch ein Matlab Programm geschrieben um numerische Berechnungen an einfachen Gitterstrukturen durchzuführen, deren Ergebnisse im letzten Abschnitt präsentiert werden.

2 Die Flachwassergleichungen

In diesem Abschnitt werden die Flachwassergleichungen wie es in der Numerik [1] üblich ist definiert. Dabei bezieht sich die Numerik im wesentlichen auf Gleichungen der konservativen Form. Die Herleitung solcher Formen sind jedoch aufwendig und werden daher nicht in diese Arbeit behandelt. Anschließend werden noch die Einschränkungen der zweidimensionalen Flachwassergleichungen erklärt. Außerdem wird folgende Notation $\partial_x := \frac{\partial}{\partial x}$ verwendet.

Für die Definition der Flachwassergleichungen müssen wir zunächst einmal zwei Funktionen definieren, welche die Tiefe des Wassers beschreiben. Hierzu betrachten wir zur Veranschaulichung die Abbildung 1. Also sei

$$b : \mathbb{R}^2 \to \mathbb{R}, (x_1, x_2) \longmapsto x_3 = b(x_1, x_2)$$

die zeitabhängige Flussbettfunktion und

$$s : \mathbb{R}^2 \to \mathbb{R}_0^+, (x_1, x_2, t) \longmapsto x_3 = s(x_1, x_2, t)$$

die zeit- und ortsabhängige Wasseroberfläche (siehe Bild 1). Zusätzlich definieren wir die Wasserhöhe H mit

$$H = s - b.$$

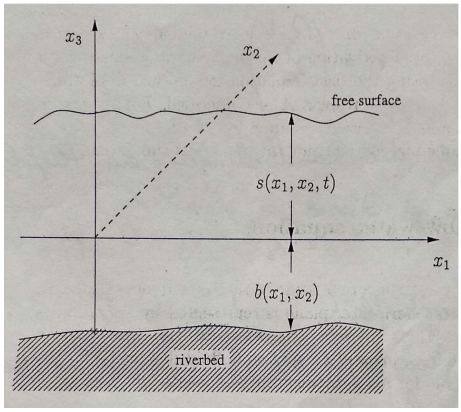

Abbildung 1: Flussbettfunktion und Wasseroberflächefunktion. Quelle: [1]

Aus [1] ist bekannt, dass die zweidimensionale Flachwassergleichungen in der konservativen Form

$$\partial_t \mathbf{u}(\mathbf{x}, t) + \sum_{j=1}^{2} \partial_{x_j} \mathbf{f}_j(\mathbf{u}(\mathbf{x}, t)) = \mathbf{g}(\mathbf{u}(\mathbf{x}, t)) \tag{1}$$

mit

$$\mathbf{u}(\mathbf{x}, t) = \begin{pmatrix} \phi \\ \phi v_1 \\ \phi v_2 \end{pmatrix}, \quad \mathbf{f}_j(\mathbf{u}(\mathbf{x}, t)) = \begin{pmatrix} \phi v_j \\ \phi v_1 v_j + \delta_{1j} \frac{1}{2} \phi^2 \\ \phi v_2 v_j + \delta_{2j} \frac{1}{2} \phi^2 \end{pmatrix}$$

und

$$\mathbf{g}_j(\mathbf{u}(\mathbf{x}, t)) = \begin{pmatrix} g \cdot q \\ g \cdot (q v_1 + \phi \partial_{x_1} b) \\ g \cdot (q v_2 + \phi \partial_{x_2} b) \end{pmatrix}, \quad j = 1, 2,$$

aufgeschrien werden können. Dabei ist $\phi = g \cdot H$ Geopotential, g ist eine Gravitationskonstante und $g = 9.81\frac{m}{s^2}$. v_1, v_2 Geschwindigkeit in x-Richtung und y-Richtung. Die Funktion $q = q(x_s(t), t)$ beschreibt die Wasserhöhenveränderung durch Regen und Verdampfung mit $x_s(t) = (x_1, x_2, s(x_1, x_2, t))$. Bei der Herleitung dieser Gleichungen wurden ein paar Einschränkungen gemacht die im nächsten Abschnitt erklärt werden.

2.1 Einschränkungen der Flachwassergleichungen

Bei der Herleitung der Flachwassergleichungen werden folgende Voraussetzungen gemacht:

1. Die Dichte ist räumlich und zeitlich konstant, d.h $\rho(\mathbf{x}, t) = \rho \neq 0$.
2. Die Körperkräfte sind durch die Gravitation festgelegt.
3. Die horizontalen Längenskalen sind wesentlich größer als die vertikalen. Somit wird die Beschleunigung in x_3-Richtung vernachlässigt.

Bei dem betrachteten Gebiet bzw. der Geometrie wird das Flussbett als undurchlässig vorausgesetzt.

3 Aufgabestellung

In diese Ausarbeitung wollen wir eine Simulation für Umströmung von Körpern in einem Becken ohne Regen und Wasserverdampfung machen. Somit vereinfachen sich die Gleichungen (1) zu

$$\partial_t \mathbf{u}(\mathbf{x}, t) + \sum_{j=1}^{2} \partial_{x_j} \mathbf{f}_j(\mathbf{u}(\mathbf{x}, t)) = 0 \tag{2}$$

Die Gleichung (2) wird mit dem Finite-Volumen-Verfahren gelöst. Anschließend wird hierauf ein Programm zur Simulation in Matlab geschrieben.

4 Anwendung der Finite-Volumen-Verfahren

Wie schon in Abschnitt 3 erklärt, gilt zunächst die Gleichung (2) als gegeben. Als nächstes definieren wir

$$\mathbf{F}(\mathbf{u}(\mathbf{x}, t)) := (\mathbf{f}_1(\mathbf{u}(\mathbf{x}, t)), \mathbf{f}_2(\mathbf{u}(\mathbf{x}, t)))^T$$

und

$$\operatorname{div}(\mathbf{F}(\mathbf{u}(\mathbf{x}, t))) := (\partial_{x_1}, \partial_{x_2}) \cdot (\mathbf{f}_1(\mathbf{u}(\mathbf{x}, t)), \mathbf{f}_2(\mathbf{u}(\mathbf{x}, t)))^T = \partial_{x_1} \mathbf{f}_1(\mathbf{u}(\mathbf{x}, t)) + \partial_{x_2} \mathbf{f}_2(\mathbf{u}(\mathbf{x}, t)). \tag{3}$$

Jetzt können wir mit (3) die Gleichungen (2) kompakter in der Form

$$\partial_t \mathbf{u}(\mathbf{x},t) + \operatorname{div}\left(\mathbf{F}(\mathbf{u}(\mathbf{x},t))\right) = 0 \tag{4}$$

aufschreiben. Für die Umformungen der Gleichungen (4) brauchen wir ein Gebiet und seine Diskretisierung. Also sei Ω unser zu diskretisierendes Gebiet. Hierfür bietet sich alternativ eine Auswahl an Diskretisierungselementen, wie z.B.: die Zerlegung in Dreiecke, Vierecke, krummlinige Elemente [2]. In dieser Arbeit beschränken wir uns jedoch ausschließlich auf die Diskretisierung durch nicht-überlappende Dreiecke, auch Triangulierung genannt. Der wesentliche Vorteil hierbei ist, dass durch die Triangulierung komplexere Geometrien diskretisiert werden können. Zudem gibt es hierfür ein frei zur Verfügung stehendes Programm [4]. Dieses Programm kann für alle weiteren Berechnungen eine Triangulierung in Matlab erzeugen. Wir schreiben also eine kurze Mathematische Definition zur Triangulierung unseres Gebiets Ω auf.

Sei eine derartige Zerlegung $\mathcal{T}(\Omega)$ von Ω gegeben, d.h.

$$\mathcal{T}(\Omega) := \{\Omega_i \subset \Omega, i = 1, \ldots, N \mid \Omega = \biguplus_{i=1}^{N} \Omega_i \text{ und } \Omega_i \cap \Omega_j = \emptyset\}.$$

Der nächste Schritt ist die Integration über eine Zelle bzw. Dreieck. Also sei $\Omega_i \subset \Omega$ und wir integrieren die Gleichungen (4) über Zelle Ω_i. Wir wollen das die Gleichung (4) für das ganze Gebiet Ω gilt. Daher setzen wir voraus, dass die Gleichung für jede Zelle Gültigkeit hat. Wenn wir alle Zellen aufsummieren bekommen wir damit ein Gebiet Ω welches samt die Gleichung erfüllt. Aus diesem Grund Integrieren wir die Gleichungen (4) über Zelle Ω_i und erhalten

$$\int_{\Omega_i} \partial_t \mathbf{u}(\mathbf{x},t) d\Omega + \int_{\Omega_i} \operatorname{div}\left(\mathbf{F}(\mathbf{u}(\mathbf{x},t))\right) d\Omega = 0. \tag{5}$$

Der nächste Schritt ist die Anwendung des sogenannten Gaußschen Integralsatzes. Für ein besseres Verständnis des nächsten Schrittes, wird dieser Satz von Gauß zunächst aufgeschrieben. (Quelle: [5]).

Satz 4.1. Es sei $V \subset \mathbb{R}^n$ eine kompakte Menge mit abschnittsweise glattem Rand $S = \partial V$, der Rand sei orientiert durch ein äußeres Normaleneinheitsvektorfeld $\mathbf{n}$. Ferner sei das Vektorfeld $\mathbf{F}$ stetig differenzierbar auf einer offenen Menge U mit $V \subseteq U$. Dann gilt

$$\int_V \operatorname{div}(\mathbf{F})\ d^{(n)}V = \oint_S \mathbf{F} \cdot \mathbf{n}\ d^{(n-1)}S \tag{6}$$

wobei $\mathbf{F} \cdot \mathbf{n}$ das Standardskalarprodukt der beiden Vektoren bezeichnet.

Wir wollen jetzt Satz (4.1) auf das zweite Integral der Gleichungen (5) anwenden. Mit $n = 2$, $\mathbf{F} = \mathbf{F}(\mathbf{u}(\mathbf{x},t))$, $V = \Omega_i \subset \mathbb{R}^2$, $S = \partial V = \partial\Omega_i$, $\mathbf{n} = (n_1, n_2)^T$ erhalten wir

$$\int_{\Omega_i} \operatorname{div}\left(\mathbf{F}(\mathbf{u}(\mathbf{x},t))\right) d\Omega \overset{\text{Satz 4.1}}{=} \oint_S \mathbf{F}(\mathbf{u}(\mathbf{x},t)) \cdot \mathbf{n}\ dS \tag{7}$$

Jetzt setzen wir (7) in Gleichungen (5) und erhalten

$$\begin{aligned} &\int_{\Omega_i} \partial_t \mathbf{u}(\mathbf{x},t) d\Omega + \oint_S \mathbf{F}(\mathbf{u}(\mathbf{x},t)) \cdot \mathbf{n}\ dS = 0 \\ \Leftrightarrow &\int_{\Omega_i} \partial_t \mathbf{u}(\mathbf{x},t) d\Omega = -\oint_S \mathbf{F}(\mathbf{u}(\mathbf{x},t)) \cdot \mathbf{n}\ dS \end{aligned} \tag{8}$$

Für die weiteren Umformungen von (8) brauchen wir die Definition des Zellmittelwertes.

Definition 4.2. Sei Ω_i eine Zelle. Wir Definieren als Zellmiteilwert $\tilde{\mathbf{u}}$ eine Funktion $\mathbf{u}(\mathbf{x},t)$ als

$$\tilde{\mathbf{u}} := \frac{1}{|\Omega_i|} \int_{\Omega_i} \mathbf{u}(\mathbf{x},t) d\Omega,$$

wobei $|\Omega_i|$ für die Fläche der Zelle Ω_i steht.

Jetzt können wir bei Gleichungen (8) das Integral mit der partiellen Ableitung vertauschen (hier ohne Beweis) und die Gleichungen mit $\frac{1}{|\Omega_i|}$ multiplizieren. Insgesamt bekommen wir

$$\frac{\mathrm{d}}{\mathrm{d}t} \int_{\Omega_i} \mathbf{u}(\mathbf{x},t) d\Omega = -\oint_S \mathbf{F}(\mathbf{u}(\mathbf{x},t)) \cdot \mathbf{n}\ dS \quad | \cdot \frac{1}{|\Omega_i|} \tag{9}$$

$$\Leftrightarrow \frac{\mathrm{d}}{\mathrm{d}t} \underbrace{\frac{1}{|\Omega_i|} \int_{\Omega_i} \mathbf{u}(\mathbf{x},t) d\Omega}_{\overset{\text{Def.4.2}}{=} \tilde{\mathbf{u}}} = -\frac{1}{|\Omega_i|} \oint_S \mathbf{F}(\mathbf{u}(\mathbf{x},t)) \cdot \mathbf{n}\ dS$$

$$\Leftrightarrow \frac{\mathrm{d}}{\mathrm{d}t} \tilde{\mathbf{u}} = -\frac{1}{|\Omega_i|} \oint_S \mathbf{F}(\mathbf{u}(\mathbf{x},t)) \cdot \mathbf{n}\ dS \tag{10}$$

Zum Schluss brauchen wir noch die Definition der Rotationsinvarianz.

Definition 4.3. Die Differentialgleichung (1) heißt Rotationsinvariant, falls zu jedem Vektor $\mathbf{n} \in \mathbb{R}^2$ mit $||\mathbf{n}||_2 = 1$ eine reguläre Matrix $\mathbf{T}(\mathbf{n}) \in \mathbb{R}^{m \times m}$ existiert, derart dass

$$\mathbf{f}_1(\mathbf{u}(\mathbf{x},t)) n_1 + \mathbf{f}_2(\mathbf{u}(\mathbf{x},t)) n_2 = \mathbf{T}(\mathbf{n})^{-1} \cdot \mathbf{f}_1(\mathbf{T}(\mathbf{n}) \cdot \mathbf{u}(\mathbf{x},t))$$

gilt.

Aus [1] ist bekannt das die zweidimensionalen Flachwassergleichungen rotationsinvariant sind mit der Matrix

$$\mathbf{T}(\mathbf{n}) = \begin{pmatrix} 1 & 0 & 0 \\ 0 & n_1 & n_2 \\ 0 & -n_2 & n_1 \end{pmatrix}. \tag{11}$$

Dabei bezeichnet der auftretende Vektor $\mathbf{n} = (n_1, n_2)^T$ den nach außen zeigenden von der betrachtete Zelle Ω_i Normaleneinheitsvektor. Außerdem für die Flachwassergleichungen gilt

$$\mathbf{f}_1(\mathbf{T}(\mathbf{n}) \cdot \mathbf{u}(\mathbf{x},t)) = \begin{pmatrix} \phi(\mathbf{v} \cdot \mathbf{n}) \\ \phi(\mathbf{v} \cdot \mathbf{n})^2 + \frac{1}{2}\phi^2 \\ \phi(\mathbf{v} \cdot \mathbf{n})(\mathbf{v} \cdot (-n_2, n_1)^T) \end{pmatrix} \tag{12}$$

Auf den Beweis der Aussage (12) und Rotationsinvarianz der Flachwassergleichungen, wird bei dieser Ausarbeitung verzichtet. Bei der Anwendung der Rotationsinvarianz auf die Gleichungen (10) erhalten wir

$$\begin{aligned} \frac{\mathrm{d}}{\mathrm{d}t}\tilde{\mathbf{u}} &= -\frac{1}{|\Omega_i|}\oint_S \mathbf{f}_1(\mathbf{u}(\mathbf{x},t))\cdot n_1 + \mathbf{f}_2(\mathbf{u}(\mathbf{x},t))\cdot n_2 \; dS \\ \overset{\text{Def.4.3}}{\Leftrightarrow} \frac{\mathrm{d}}{\mathrm{d}t}\tilde{\mathbf{u}} &= -\frac{1}{|\Omega_i|}\oint_S \mathbf{T}(\mathbf{n})^{-1}\cdot \mathbf{f}_1(\mathbf{T}(\mathbf{n})\cdot\mathbf{u}(\mathbf{x},t)) \; dS \end{aligned} \tag{13}$$

unsere räumliche Diskretisierung. Hier kann man unmittelbar den Vorteil von Rotationsinvarianz in dieser Gleichung sehen, denn statt der zwei Funktionsauswertungen $\mathbf{f}_1(\mathbf{u}(\mathbf{x},t))$ und $\mathbf{f}_2(\mathbf{u}(\mathbf{x},t))$ brauchen wir jetzt nun nur eine Funktionsauswertung. Allerdings kann das Integral

$$\oint_S \mathbf{T}(\mathbf{n})^{-1}\cdot \mathbf{f}_1(\mathbf{T}(\mathbf{n})\cdot\mathbf{u}(\mathbf{x},t)) \; dS$$

am Rand nicht ausgewertet werden. Das liegt daran, dass wir die Werte am Rand bei $\mathbf{u}(\mathbf{x},t)$ nicht kennen bzw. zwei unterschiedliche Werte am Rand haben könnten. Dies bedeutet, dass zwei benachbarte Zellen unterschiedliche Mittelwerte $\tilde{\mathbf{u}}$ besitzen, was man auch in Abbildung 2 für sämtliche Dreieckselemente mit der x-Koordinate gleich Null sehen kann.

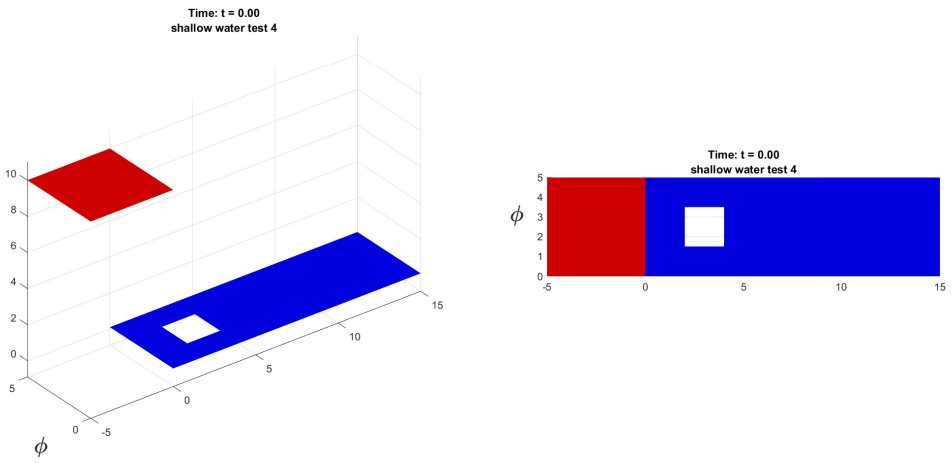

Abbildung 2: Unstetige Anfangsbedingung. Links Seitenansicht, rechts Draufsicht.

Um diesem Integral Auszuwerten, verwenden wir die Idee von Finite-Volumen-Verfahren. Wir approximieren dieses Integral mit einer numerische Flussfunktion. Somit bekommen wir eine Approximation am Rand für benachbarte Zellen durch

$$\oint_S \mathbf{T}(\mathbf{n})^{-1}\cdot \mathbf{f}_1(\mathbf{T}(\mathbf{n})\cdot\mathbf{u}(\mathbf{x},t)) \; dS \approx \oint_S \mathbf{T}(\mathbf{n})^{-1}\cdot \mathbf{H}(\mathbf{T}(\mathbf{n})\cdot\tilde{\mathbf{u}}^+, \mathbf{T}(\mathbf{n})\cdot\tilde{\mathbf{u}}^-) \; dS, \tag{14}$$

mit $\tilde{\mathbf{u}}^+$ als Zellenmittelwert für aktuelle Zelle und $\tilde{\mathbf{u}}^-$ als Zellenmittelwert für entsprechend benachbarte Zelle.

Es gibt sehr viele numerische Flussfunktionen, aus diesem Grund wird in diese Arbeit auf eine Definition und genauere Erklärung verzichtet. Die Definition und Eigenschaften der numerische Flussfunktion können in [3] auf der Seite 47 nachgeschlagen werden. Für die Simulation in Matlab wurde die Harten-Lax-Van Leer-Methode (Flussfunktion) verwendet (abgekürzt HLL-Methode). Diese Flussfunktion sieht wie folgt aus,

$$\mathbf{H}^{HLL}(\tilde{\mathbf{u}}^+, \tilde{\mathbf{u}}^-) = \begin{cases} \tilde{\mathbf{u}}^+ & S_l \geq 0 \\ \frac{S_r \mathbf{f}(\tilde{\mathbf{u}}^+) - S_l \mathbf{f}(\tilde{\mathbf{u}}^-) + S_l S_r (\tilde{\mathbf{u}}^- - \tilde{\mathbf{u}}^+)}{S_r - S_l} & S_l \leq 0 \leq S_r \\ \tilde{\mathbf{u}}^- & S_r \leq 0 \end{cases} \tag{15}$$

mit

$$S_l = \min\{\mathbf{v}_j \cdot \mathbf{n} - \sqrt{\phi_j}, \mathbf{v}_i \cdot \mathbf{n} - \sqrt{\phi_i}\}, \quad S_r = \max\{\mathbf{v}_j \cdot \mathbf{n} - \sqrt{\phi_j}, \mathbf{v}_i \cdot \mathbf{n} - \sqrt{\phi_i}\}, \tag{16}$$

für zweidimensionale Flachwassergleichungen. Dabei gilt $\mathbf{v} = (v_1, v_2)^T, \mathbf{n} = (n_1, n_2)^T$ und i steht für aktuelle Zelle und j für die Nachbarzelle.

Alles zusammengefasst bekommen wir für benachbarte Zellen nachfolgend, dass

$$\frac{\mathrm{d}}{\mathrm{d}t}\tilde{\mathbf{u}} = -\frac{1}{|\Omega_i|} \oint_S \mathbf{T}(\mathbf{n})^{-1} \cdot \mathbf{H}^{HLL}(\mathbf{T}(\mathbf{n}) \cdot \tilde{\mathbf{u}}^+, \mathbf{T}(\mathbf{n}) \cdot \tilde{\mathbf{u}}^-) \, dS \tag{17}$$

Finite-Volumen-Verfahren für zweidimensionale Flachwassergleichungen. Die Gleichungen (17) gelten für beliebige Zellen die nicht am Rand des Gebiets liegen. In diese Arbeit verwenden wir die Dreieckelemente, d.h. die Gleichungen (17) müssen für jedes Dreieck an drei Kanten ausgewertet werden. Die Auswertung ist relativ einfach, weil wir bei dem Finite-Volumen-Verfahren nur einen Mittelwert der Funktion über eine Kante zu integrieren haben. Also gilt somit für alle Dreiecke im inneren des Gebiets (Randbedingungen werden im nächsten Abschnitt behandelt):

$$\frac{\mathrm{d}}{\mathrm{d}t}\tilde{\mathbf{u}} = -\frac{1}{|\Omega_i|} \sum_{j \in N(i)} |\Gamma_{ij}| \cdot \mathbf{T}(\mathbf{n})^{-1} \cdot \mathbf{H}^{HLL}(\mathbf{T}(\mathbf{n}) \cdot \tilde{\mathbf{u}}^+, \mathbf{T}(\mathbf{n}) \cdot \tilde{\mathbf{u}}^-) \tag{18}$$

mit $N(i) = \{j \in \mathbb{N} \mid \partial\Omega_i \cap \partial\Omega_j \neq \emptyset\}$. Dabei bezeichnet Ω_i für $i = 1, \ldots, N$ die Dreiecke einer Triangulierung des Gebiets Ω und Γ_{ij} die Kante zwischen den Dreiecken Ω_i und Ω_j.

4.1 Randbedingungen

In diese Arbeit wird nur eine Art an Randbedingung behandelt. Also wir wollen davon ausgehen, dass es sich bei den Rändern um feste Wände handelt, dementsprechend gilt dort $\mathbf{v} \cdot \mathbf{n} = 0$. Dies können wir direkt in $\mathbf{f}_1(\mathbf{T}(\mathbf{n}) \cdot \mathbf{u}(\mathbf{x}, t))$ einsetzen und erhalten nach

(12) an den Randkanten

$$\mathbf{f}_1(\mathbf{T}(\mathbf{n}) \cdot \mathbf{u}(\mathbf{x}, t)) = \begin{pmatrix} 0 \\ \frac{1}{2}\phi^2 \\ 0 \end{pmatrix}. \tag{19}$$

Jetzt setzen wir (19) in (13) ein und erhalten

$$\frac{\mathrm{d}}{\mathrm{d}t}\tilde{\mathbf{u}} = -\frac{1}{|\Omega_i|} \oint_S \mathbf{T}(\mathbf{n})^{-1} \cdot \begin{pmatrix} 0 \\ \frac{1}{2}\phi^2 \\ 0 \end{pmatrix} \, dS. \tag{20}$$

Nun fehlt uns noch die Zeitintegration, die im nächsten Abschnitt erklärt wird.

5 Zeitintegration

Für die Zeitintegration wurde bei dieser Arbeit nicht so viel Zeit investiert. Dies liegt daran, dass bei sehr feinen Ortsdiskretisierungen, der hierzu verwendete Rechner sehr schnell an seine Grenzen kommt, so dass mit explizitem Euler-Verfahren das Matlab Programm mit der Fehlermeldung „Out of Memory“ beendet wird. Der zweite Grund ist, dass das Matlab Programm bei explizitem Euler-Verfahren schon sehr lange braucht um Ergebnisse bei feinerer Ortsdiskretisierung zu erzeugen. Das kann natürlich auch an der Implementierung liegen. Den bei der Implementierung wurde nur beachtet, dass die in Matlab eingebaute App zur Übersetzung des Codes in C++, implementierte Codes auch übersetzen kann. Aus diesen Gründen wurden alle Bespiele nur mit explizitem Euler-Verfahren berechnet. Alles zusammengefasst kann ein Zeitschritt eines Finite-Volumen-Verfahrens mit explizitem Euler-Verfahren zur Lösung der Differentialgleichung (2) in der Form

$$u_i^{n+1} = u_i^n - \Delta t \cdot \frac{1}{|\Omega_i|} \sum_{j \in N(i)} |\Gamma_{ij}| \cdot \mathbf{T}(\mathbf{n})^{-1} \cdot \mathbf{H}^{HLL}(\mathbf{T}(\mathbf{n}) \cdot \tilde{\mathbf{u}}^+, \mathbf{T}(\mathbf{n}) \cdot \tilde{\mathbf{u}}^-) \tag{21}$$

mit $N(i) = \{j \in \mathbb{N} \mid \partial\Omega_i \cap \partial\Omega_j \neq \emptyset\}$ für Dreiecke im inneren des Gebiets geschrieben werden. Für die Dreiecke mit kanten am Randgebiet müssen wir das explizite Euler-Verfahren auf (20) anwenden und erhalten

$$u_i^{n+1} = u_i^n - \Delta t \cdot \frac{|\Gamma_{ij}|}{|\Omega_i|} \mathbf{T}(\mathbf{n})^{-1} \cdot \begin{pmatrix} 0 \\ \frac{1}{2}\phi^2 \\ 0 \end{pmatrix}. \tag{22}$$

Außerdem muss beim expliziten Verfahren im mehrdimensionalen Fall eine verallgemeinerte CFL-Bedingung berücksichtigt werden. Wir wollen die Zeitschrittweite so wählen, dass für jedes Dreieck die Ungleichung

$$\Delta t \leq CFL \cdot \frac{\min_{l=1,2,3} k_l}{\max_{j=1,2,3} |\lambda_j|} \tag{23}$$

mit CFL $\in (0, 1)$ erfüllt ist. Dabei sind die charakteristischen Ausbreitungsgeschwindigkeiten der Flachwassergleichungen durch $\lambda_1 = \mathbf{v} \cdot \mathbf{n} - \sqrt{\phi}$, $\lambda_2 = \mathbf{v} \cdot \mathbf{n}$ und $\lambda_3 = \mathbf{v} \cdot \mathbf{n} + \sqrt{\phi}$ gegeben und für $l = 1, \ldots, 3$ bezeichnet k_l den Abstand des Schwerpunktes zu einer der drei Mittelpunkten der Dreieckskanten.

6 Numerische Ergebnisse

Für die Erzeugung numerischer Ergebnisse wurde ein Matlab Programm geschrieben. Dieses Programm besteht aus zwei Teilen bzw. zwei unabhängigen Programmen. Der erste Teil erzeugt mit Hilfe von DistMesh aus [4] ein Gitter mit Dreiecken. Der zweite Teil ist unabhängig vom ersten Teil und verwendet nur erzeugte Gitter aus ersten Teils. Dieser Teil verwendet teilweise eingebaute Funktionen in Matlab wie „triangulation“ für einfachen Zugriff auf anliegender Dreiecke einer Kante und zum ploten wird die eingebaute Funktion „trisurf“ verwendet.

Bei der Berechnung werden folgende Größen geplotet. Im ersten und zweiten Quadrant, die konservative Größe ϕ. Diese ist nicht gleich der Wasseroberflächenhöhe, denn wie schon in Abschnitt 2 erklärt, es gilt $\phi = g \cdot H$. Mit anderen Worten wird die Wasseroberflächenhöhe mal Gravitationskonstante geplotet. Im dritten und vierten Quadranten werden Beträge der Geschwindigkeit $\mathbf{v} = (v_1, v_2)^T$ in der $||\ .\ ||_2$ für alle Dreiecke geplotet. Das Programm ist so geschrieben, dass es mit beliebigem Gitter rechnen kann. Das liegt daran, dass alle Wände als „feste Wand“ betrachtet werden. In diese Ausarbeitung werden zwei berechnete Beispiele präsentiert. Beide Beispiele wurden mit CFL= 0.5 berechnet. Das erste Beispiel heißt „Rechteck im rechteck“. Die Anfangsbedingung in feiner Auflösung befindet in Abbildung 3. Die Ausbreitung der Welle von links nach rechts zum Zeitpunkt t = 1 sehen wir in den Abbildungen 4 und 5. Dabei kann man sehen, dass wie erwartet die Strömung symmetrisch verläuft. Allerdings gibt es hierbei bei der Ecke der Abbildung 5 im vierten Quadranten mit x-Koordinate gleich Null, eine kleine Abweichung in der Symmetrie. Das kann an kleine Konvergenzordnung des Expliziten Euler-Verfahren liegen oder an die gewählte numerische Flussfunktion. Bei den Abbildungen 6 und 8 für gröbere Gitter und Abbildungen 7 und 9, kann man den weiteren Verlauf der Welle beobachten.

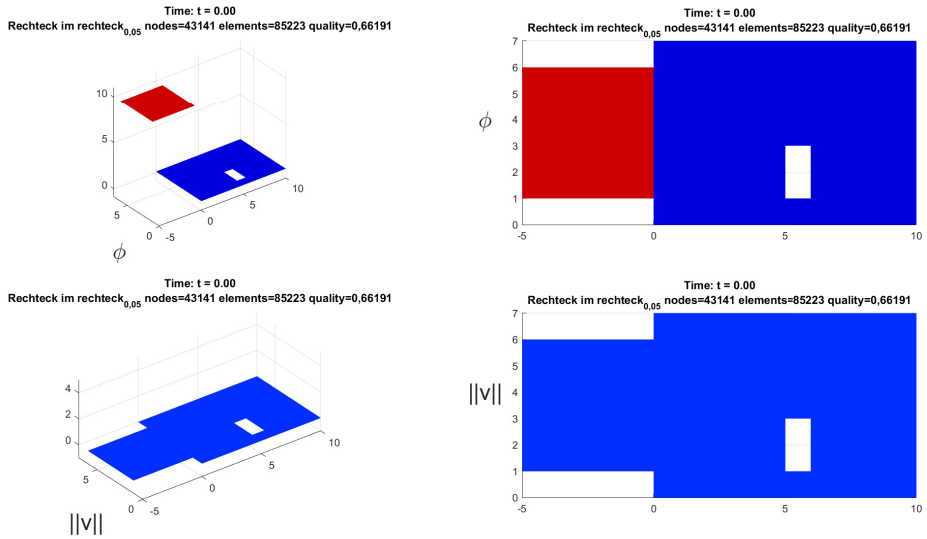

Abbildung 3: Bsp. 1: Rechteck im rechteck mit 85223 Elementen zu Zeit t =0

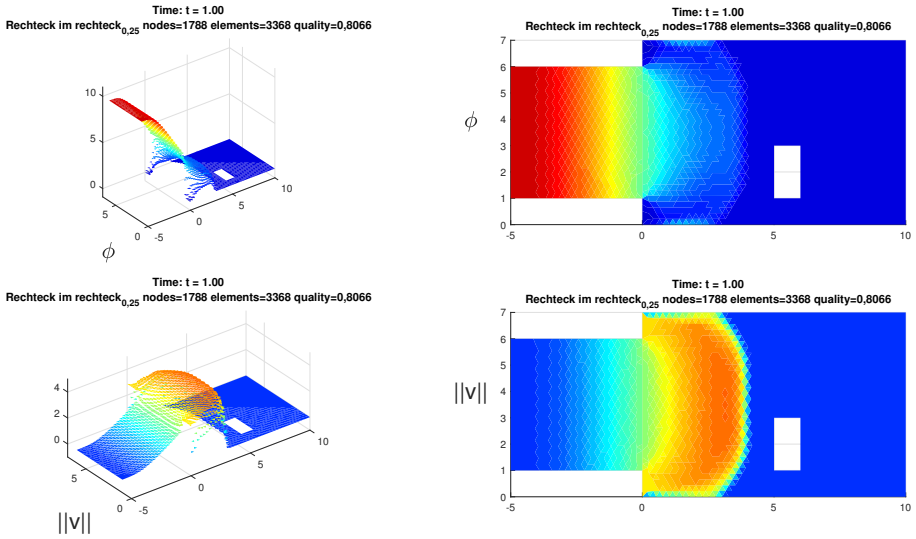

Abbildung 4: Bsp. 1: Rechteck im rechteck mit 3368 Elementen zu Zeit t =1

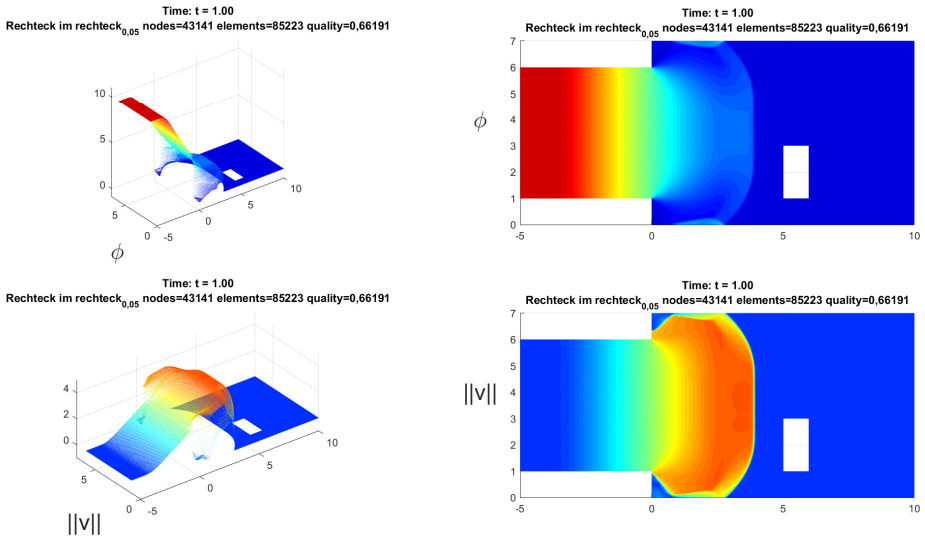

Abbildung 5: Bsp. 1: Rechteck im rechteck mit 85223 Elementen zu Zeit t =1

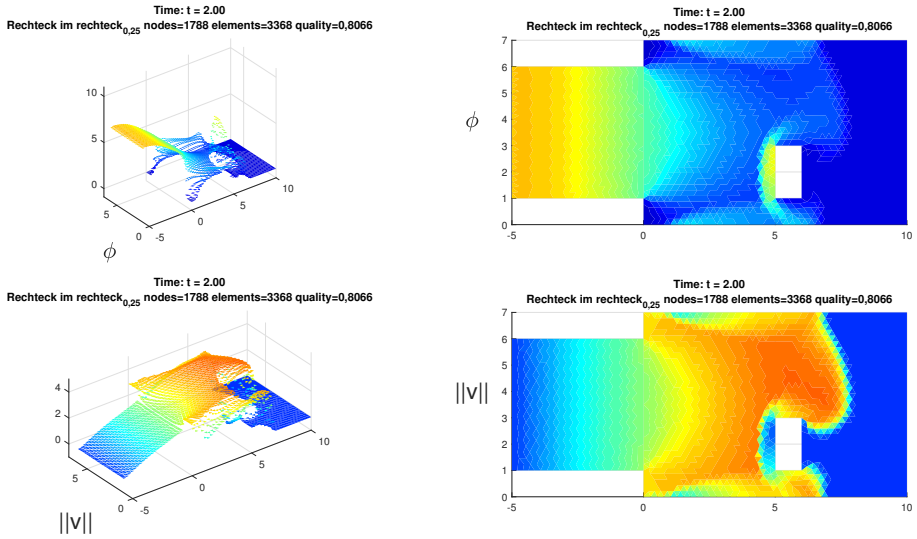

Abbildung 6: Bsp. 1: Rechteck im rechteck mit 3368 Elementen zu Zeit t =2

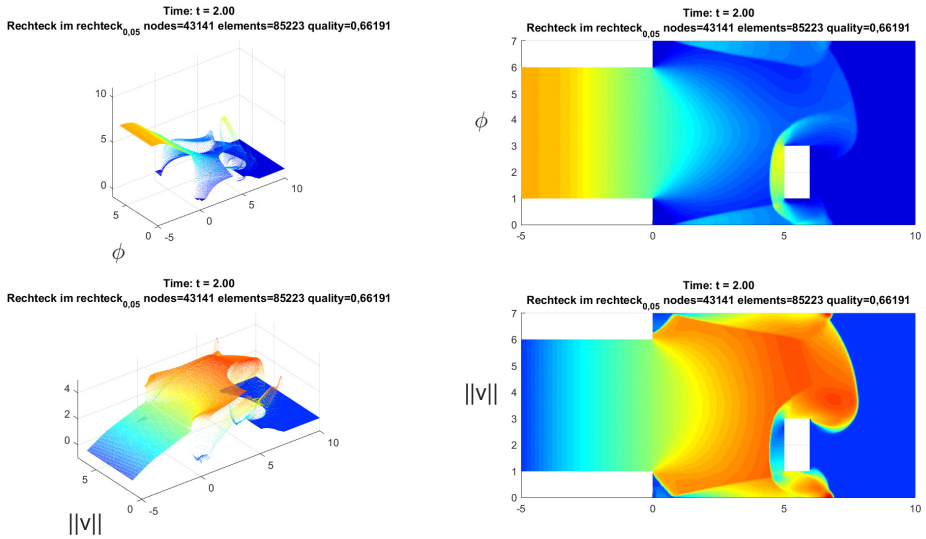

Abbildung 7: Bsp. 1: Rechteck im rechteck mit 85223 Elementen zu Zeit t =2

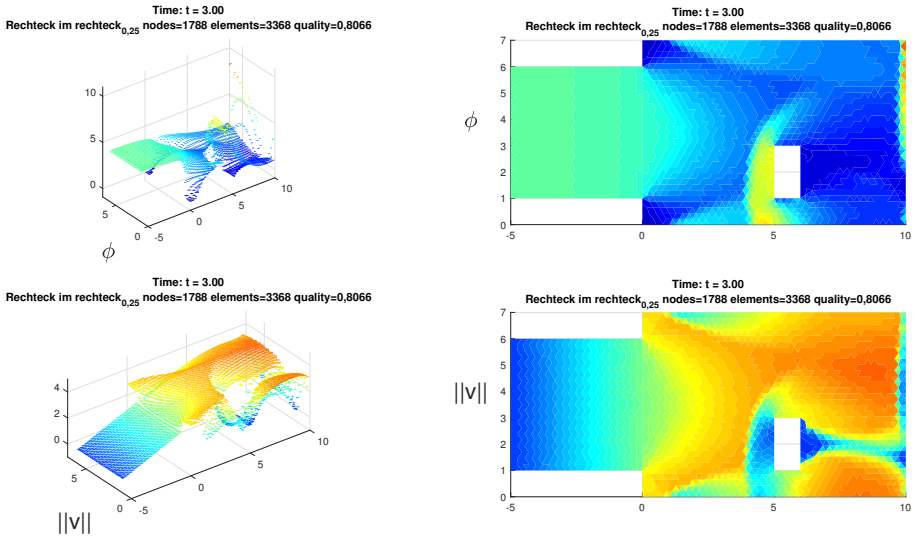

Abbildung 8: Bsp. 1: Rechteck im rechteck mit 3368 Elementen zu Zeit t =3

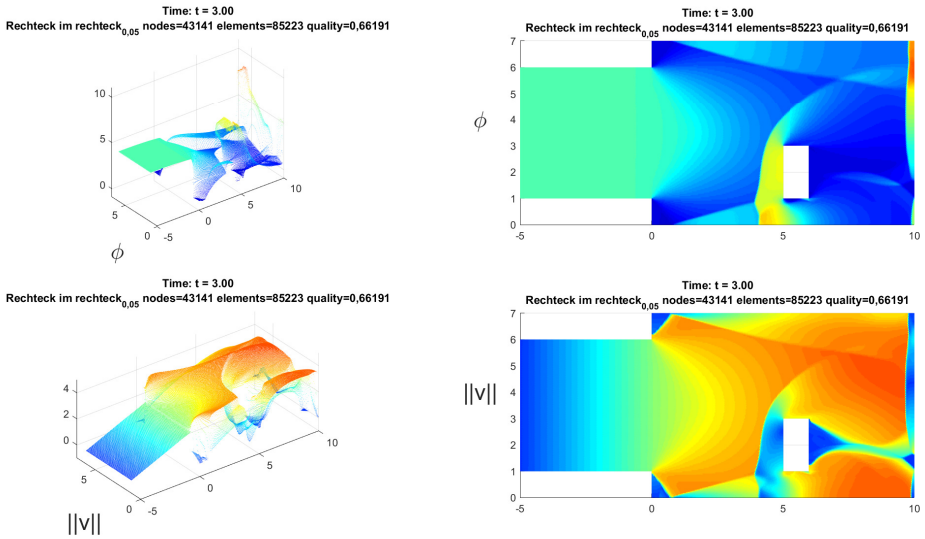

Abbildung 9: Bsp. 1: Rechteck im rechteck mit 85223 Elementen zu Zeit t =3

Die Anfangsbedingung für das zweite Beispiel „mesh_test“ ist im Abbildung 10 dargestellt. Zudem sind Bilder zu unterschiedlichen Zeiten für ein gröberes Gitter in den Abbildungen 11,13,15 und für ein feineres Gitter in den Abbildungen 12,14,16 dargestellt. Beim zweiten Beispiel wurde untersucht, wie sich die Wellen binnen 100 Sekunden entwickeln. Dabei fällt auf, dass die Wellen immer kleiner werden und das Wasser sich schon fast gar nicht mehr bewegt (siehe Abbildung 15). Dieser Zusammenhang geht nicht aus den Flachwassergleichungen hervor, die eine reibungsfreie Wasserbewegung beschreiben. Aufgrund der Ergebnisse die möglicher Weise auf ein Reibungsverhältnis deuten, muss der Grund woanders liegen. Der Grund hierfür kann unterschiedlich sein, jedoch hätte vermutlich die numerische Diffusion den größten Einfluss auf die Wellenausbreitung. Bei der Verfeinerung des Gitters kann man beobachten, dass die numerische Diffusion wie erwartet nachlässt (siehe Abbildung 16, sowie andere Abbildungen die kleine Unterschiede aufzeigen). Dies bedeutet auch, dass die Berechnung (bis die Wellen sich nicht mehr bewegen) länger andauert. Somit lässt die numerische Diffusion bei feinerem Gitter nach, denn für $\frac{\Delta t}{\Delta x} \to 0$ mit Beachtung der CFL-Zahl soll die numerische Lösung gegen die exakte Lösung konvergieren.

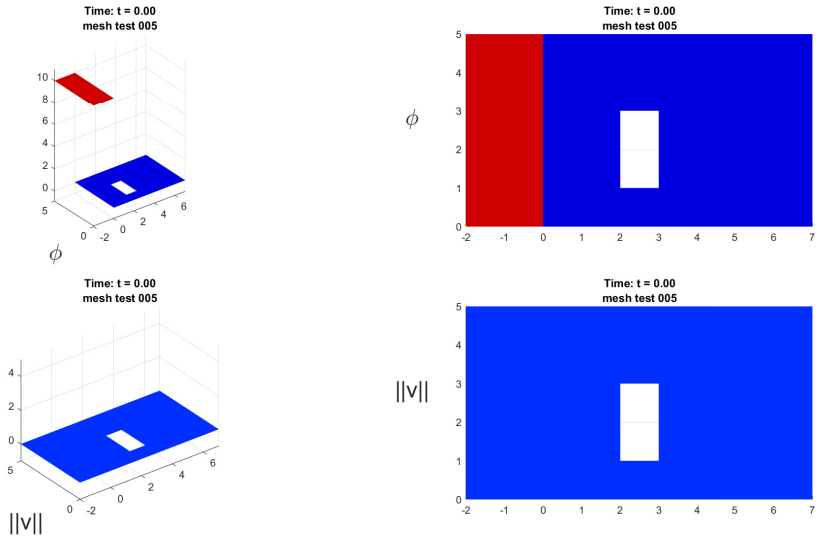

Abbildung 10: Bsp. 2: mesh_test mit 39455 Elementen zu Zeit t =0

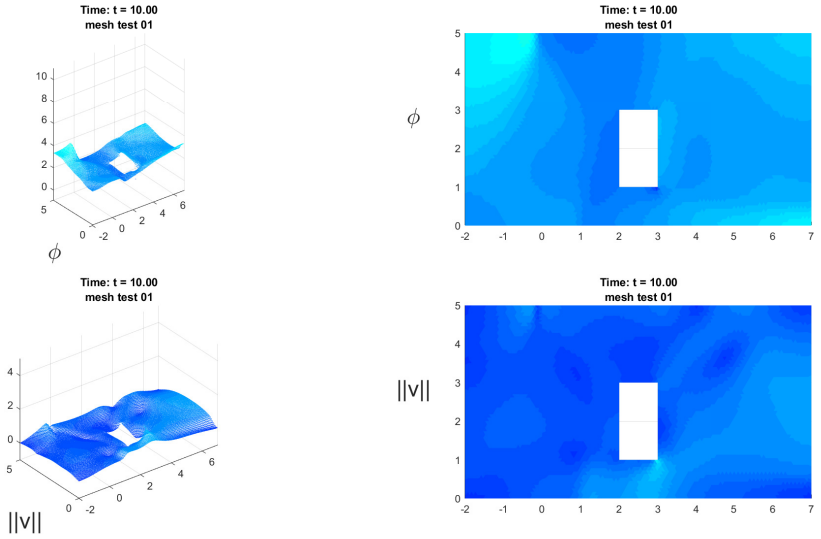

Abbildung 11: Bsp. 2: mesh_test mit 9752 Elementen zu Zeit t =10

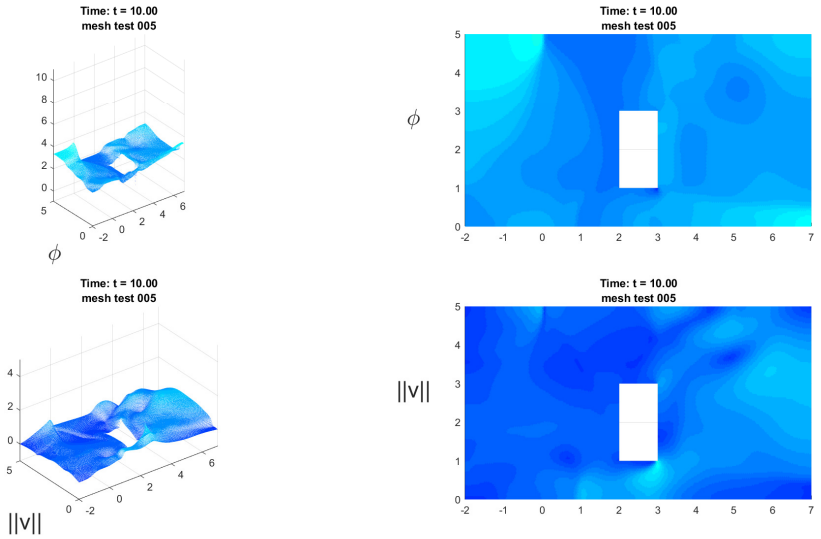

Abbildung 12: Bsp. 2: mesh_test mit 39455 Elementen zu Zeit t =10

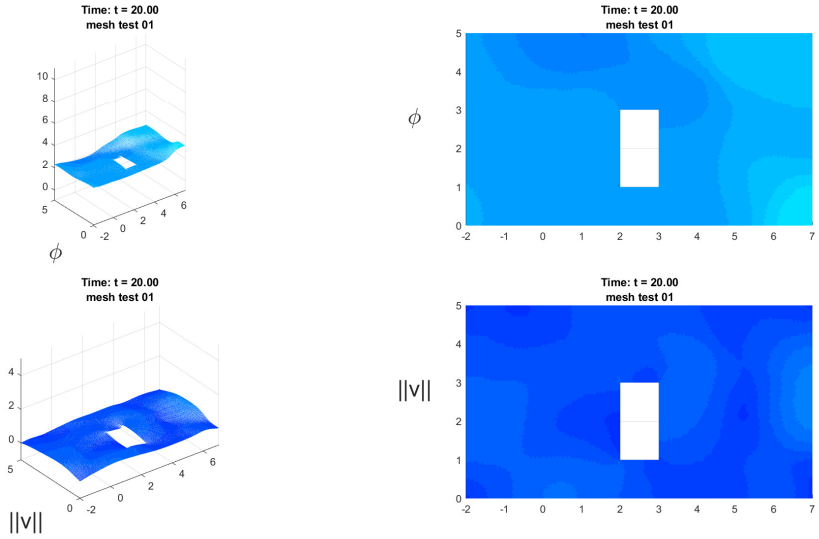

Abbildung 13: Bsp. 2: mesh_test mit 9752 Elementen zu Zeit t =20

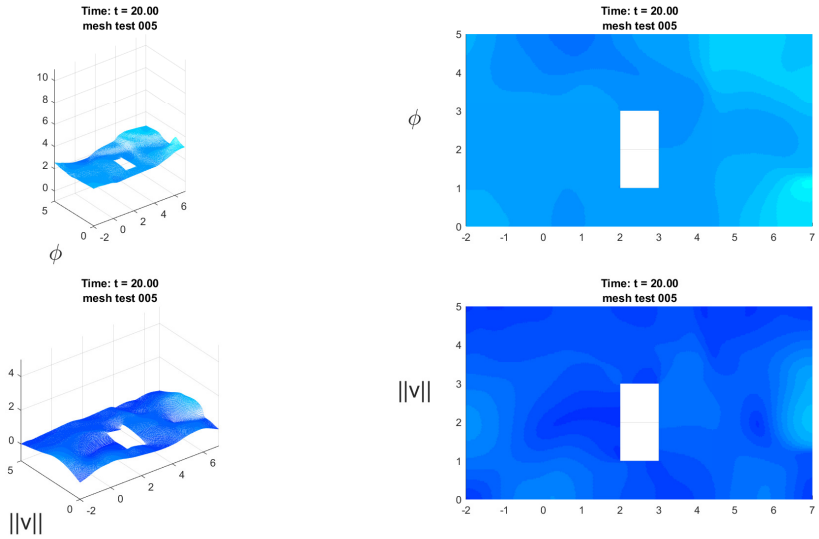

Abbildung 14: Bsp. 2: mesh_test mit 39455 Elementen zu Zeit t =20

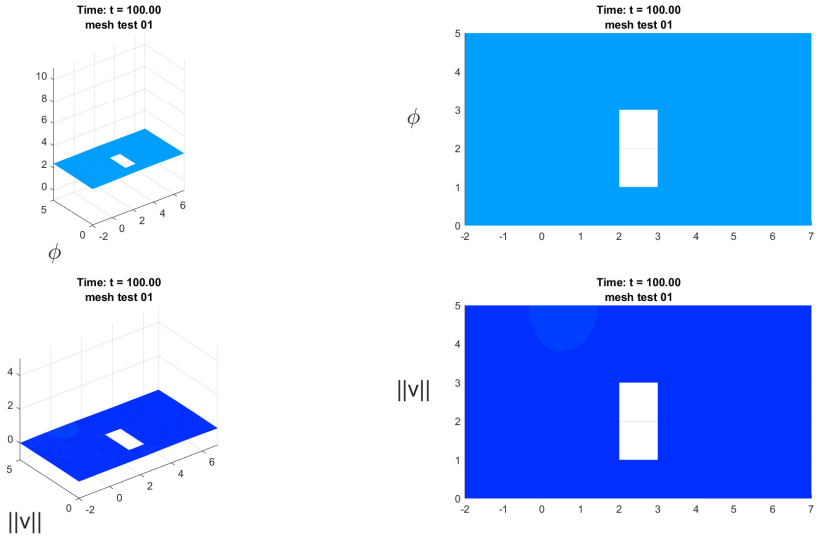

Abbildung 15: Bsp. 2: mesh_test mit 9752 Elementen zu Zeit t =100

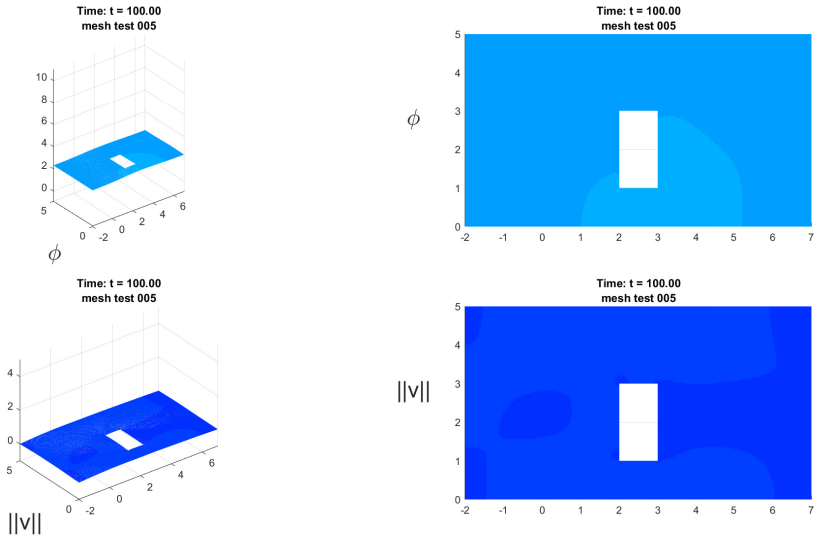

Abbildung 16: Bsp. 2: mesh_test mit 39455 Elementen zu Zeit t =100

Literatur

[1] A. Meister. *Numerik gewöhnliche Differentialgleichungen.* Vorlesungsskript, Universität Kassel, Wintersemester 2015/16.

[2] O. Wünsch. *Numerische Berechnung von Strömungen.* Vorlesungsskript, Universität Kassel, Wintersemester 2018/2019.

[3] S. Ortlebt. *Ein diskontinuierliches Galerkin-Verfahren hoher Ordnung auf Dreiecksgittern mit modaler Filterung zur Lösung hyperbolischer Erhaltungsgleichungen.* 2012, kassel university press GmbH, Kassel. ISBN online: 978-3-86219-219-9

[4] P.-O. Persson, G. Strang. *DistMesh - A Simple Mesh Generator in MATLAB.* http://persson.berkeley.edu/distmesh

[5] *Gaußscher Integralsatz.* https://de.wikipedia.org/wiki/Gaußscher_Integralsatz

[6] D. Kuhl. *Numerische Mechanik I.* Vorlesungsskript, Universität Kassel, Wintersemester 2016/17.