

Bachelorarbeit

Simulation von Wasserströmungen
mit freier Oberfläche
berechnet mit Grafikprozessoren

Philipp Schierz

Fakultät für Physik und Geowissenschaften
Universität Leipzig
Juli 2011

Betreuer:

Prof. Dr. Eberhard Renner , Dr. Siegfried Fritsche

Inhaltsverzeichnis

1	Einleitung	4
2	Thermodynamische Herleitungen für Wasser	6
2.1	Zustandsgleichung für Wasser	6
2.2	Entropiegleichung	9
2.3	Bilanzgleichung der Entropie	11
3	Differentialgleichungen und Numerik der Wassersimulation in den inneren Zellen	14
3.1	Grundlagen	14
3.2	Differentialgleichungen	16
3.3	Runge-Kutta-Verfahren	17
3.4	Upwindverfahren	19
3.5	Quellterme und Zusammenfassung	22
3.6	Rand- und Anfangsbedingungen	24
3.7	Das CFL-Kriterium	25
3.8	Anmerkung zur Diskretisierung der Geschwindigkeiten	25
4	Modellierung der freien Oberfläche	27
4.1	Herangehensweise	27
4.2	Divergenzterm	27
4.3	Quellterme	29
5	Rechnen auf Grafikprozessoren	32
5.1	Warum Grafikprozessoren in der Computersimulation praktisch sind	32
5.2	Paralleles Rechnen auf der Grafikkarte	33
5.3	Beispiel: Matrixaddition auf der Grafikkarte	34
5.4	Nachteile	37
6	Funktionsweise und Aufbau des Programmes	38
6.1	Allgemeines	38
6.2	Programmablaufplan	38
7	Auswertung von Simulationen	41
7.1	Qualitativer Vergleich mit der Simulation von Bryan-Fritsch [5]	41
7.2	Die Wasserparabel	42
7.3	Rotierender Zylinderspaltschnitt inklusive Wärmefluss an den Randflächen	43

Inhaltsverzeichnis

8	Kritik und Ausblick	49
8.1	Kritik	49
8.2	Ausblick	50

1 Einleitung

Das Lösen von allgemeinen Differenzialgleichungen stellt in vielen Fällen eine große Herausforderung dar, insbesondere das Lösen von partiellen Differentialgleichungen. Der Großteil, der durch diese Gleichungen beschriebenen physikalischen Probleme, ist analytisch für die interessierenden Rand- und Anfangsbedingungen nicht lösbar. Bis vor wenigen Jahrzehnten stellte dieses Hindernis eine große Hürde in der Physik dar. Die einzige Alternative, die Numerik, war damals sehr aufwendig. Dies hat sich im Computerzeitalter geändert, da Computer aufwändige numerische Probleme berechnen können. Somit werden komplexe Sachverhalte mit einem vertretbaren Zeitaufwand veranschaulicht.

Solch ein Fall liegt bei der Berechnung von Wasserströmungen vor. Die Differentialgleichungen welche diese Strömungen beschreiben, sind die Navier-Stokes Gleichungen, sowie die Bilanzgleichungen für Dichte und Entropie.

Speziell ist diese Bachelorarbeit eine Vorbereitung zur numerischen Behandlung des Problems des rotierenden Zylinderspaltes.

Dieses Experiment sei hier zunächst kurz erläutert.

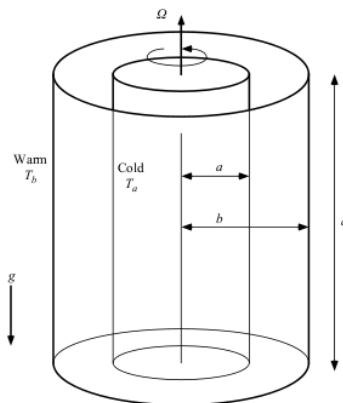


Abbildung 1.1: Der Rotierende Zylinder [1]

Ein kleiner (ca. 7 cm Radius) und ein großer (ca. 12 cm Radius) Zylinder werden ineinander gestellt. In den Zwischenraum wird Wasser gegeben und das gesamte System wird zum Rotieren gebracht. Der Innenzylinder wird nun gekühlt, der Außenzylinder beheizt (bei einem Temperaturunterschied von wenigen Kelvin). An der Oberfläche entstehen nach einer gewissen Zeit stabile Strömungen. Abbildung 1.2 stellt eine der möglichen stabilen Strömungen dar.

Man erkennt drei Wellenmoden im Wärmebild. Die Anzahl und die Form der Moden

1 Einleitung

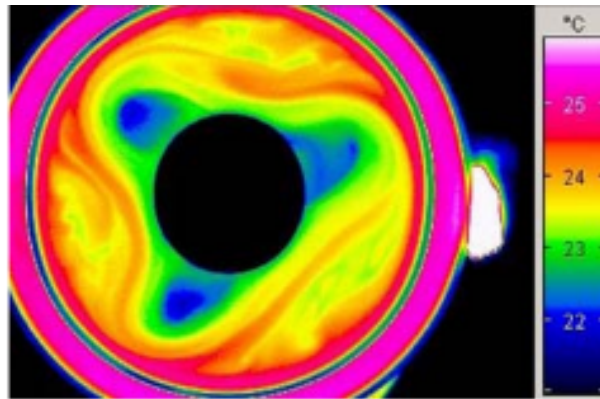


Abbildung 1.2: Wärmebild der Oberfläche [2]

variieren mit der Rotationsgeschwindigkeit und dem Temperaturunterschied zwischen Außen- und Innenzylinder.

Es wurden bereits mit dem Modell ASAM Versuche unternommen den rotierenden Zylinder zu simulieren, jedoch gab es bisher Schwierigkeiten die Moden zu erhalten. Dabei wurde das Wasser mit der Boussinesq-Approximation als inkompressibel behandelt und die Oberfläche wurde durch eine starre Wand simuliert. Im Institut für Troposphärenforschung, an dem diese Arbeit erstellt wurde, besteht nun die These, dass dieses Modell ergänzt werden muss. Diese Ergänzungen waren zum einen, das Wasser als nicht inkompressibel zu behandeln und zum anderen eine offene Oberfläche zu haben, die sich ausdehnen und zusammenziehen kann. Die Vorarbeit hierfür soll diese Arbeit leisten. Dabei wurde der Versuch unternommen kompressible Wasserströmungen mit offener Oberfläche in zwei Dimensionen zu simulieren.

2 Thermodynamische Herleitungen für Wasser

2.1 Zustandsgleichung für Wasser

Um die Strömungen von Wasser zu simulieren, ist es notwendig eine Zustandsgleichung $f(p, \rho, T) = 0$ für Wasser zu kennen. Diese sollte das System möglichst gut in den für das Zylinderspaltexperiment relevanten Bereichen beschreiben.

Diese sind:

- ein Temperaturbereich von 290K bis 300 K
- ein Druckbereich um 101325 Pa (Luftdruck)
- ein Dichtebereich um $998,2 \frac{\text{kg}}{\text{m}^3}$ (Dichte bei Normalbedingungen).

Die Herleitung, der in dieser Arbeit verwendeten Zustandsgleichung, geht von der Näherung eines konstanten Kompressibilitätsmoduls, sowie eines konstanten Ausdehnungskoeffizienten aus. Daraus ergibt sich in den nachfolgenden Schritten die Zustandsgleichung.

Man geht von den Gleichungen für Kompressibilität und thermaler Ausdehnung aus.

$$\begin{aligned}\kappa &= -\frac{1}{V} \left(\frac{\partial V}{\partial p} \right)_T \\ \alpha &= -\frac{1}{\rho} \left(\frac{\partial \rho}{\partial T} \right)_p\end{aligned}\tag{2.1}$$

Dabei erhält man durch Substitution ($V = \frac{m}{\rho}$) in der ersten Gleichung eine Formel die sich auf ρ bezieht:

$$\kappa = \frac{1}{\rho} \left(\frac{\partial \rho}{\partial p} \right)_T.\tag{2.2}$$

Nun bildet man das vollständige Differential von $\rho(T, p)$:

$$d\rho = \left(\frac{\partial \rho}{\partial T} \right)_p dT + \left(\frac{\partial \rho}{\partial p} \right)_T dp$$

und identifiziert mit den Gleichungen (2.1), (2.2):

$$\begin{aligned}\left(\frac{\partial \rho}{\partial T} \right)_p &= -\alpha \rho \\ \left(\frac{\partial \rho}{\partial p} \right)_T &= \kappa \rho.\end{aligned}$$

2 Thermodynamische Herleitungen für Wasser

Damit erhält man ein vollständiges Differential für ρ :

$$d\rho = -\alpha\rho dT + \kappa\rho dp$$

$$\frac{d\rho}{\rho} = -\alpha dT + \kappa dp,$$

womit durch Integration ein Ausdruck für ρ erhalten wird:

$$\ln(\rho) = \kappa p - \alpha T + D.$$

Umstellen nach ρ und setzen der Randbedingungen liefert die neue Zustandsgleichung ($C = e^D$):

$$\rho = Ce^{(\kappa p - \alpha T)}.$$

Dabei ergibt sich C folgendermaßen aus den Anfangswerten:

$$C = \frac{\rho_0}{e^{(\kappa p_0 - \alpha T_0)}}.$$

Ausformuliert lautet damit die Zustandsgleichung:

$$\rho = \rho_0 e^{(\kappa(p-p_0) - \alpha(T-T_0))}.$$

Um nun zu bestätigen, dass die benutzte Zustandsgleichung eine gute Näherung für Wasser im geforderten Bereich ist, wird sie mit den Ergebnissen der Arbeit von W. Wagner und A. Pruß [3] verglichen. Die Daten aus dieser Arbeit basieren auf mehreren 10000 Messdaten die benutzt wurden, um ein Programm zu schreiben, das die Zustände des Wassers möglichst genau angeben kann. Diese Daten wurden in dieser Arbeit verwendet, um die Koeffizienten κ , α zu fitten und zeigten eine gute Übereinstimmung im geforderten Bereich. In Abbildung 2.1 sieht man in wie weit die Zustandsgleichung mit realen Wasserdaten übereinstimmt.

Die Parameter κ , α wurden dabei als:

$$\kappa = (4.486 \cdot 10^{-10} \pm 2 \cdot 10^{-13}) \frac{1}{Pa}$$

$$\alpha = (2.4 \cdot 10^{-4} \pm 7 \cdot 10^{-7}) \frac{1}{K}$$

bestimmt.

Am Graphen und an den sehr kleinen Werten für κ und α ist zu erkennen, dass man die Zustandsgleichung ebenfalls durch eine einfache Ebenengleichung annähern könnte. In dieser Arbeit wurde dies jedoch nicht getan, da sich der Umgang mit der Exponentialfunktion selbst nicht als schwieriger erwies.

2 Thermodynamische Herleitungen für Wasser

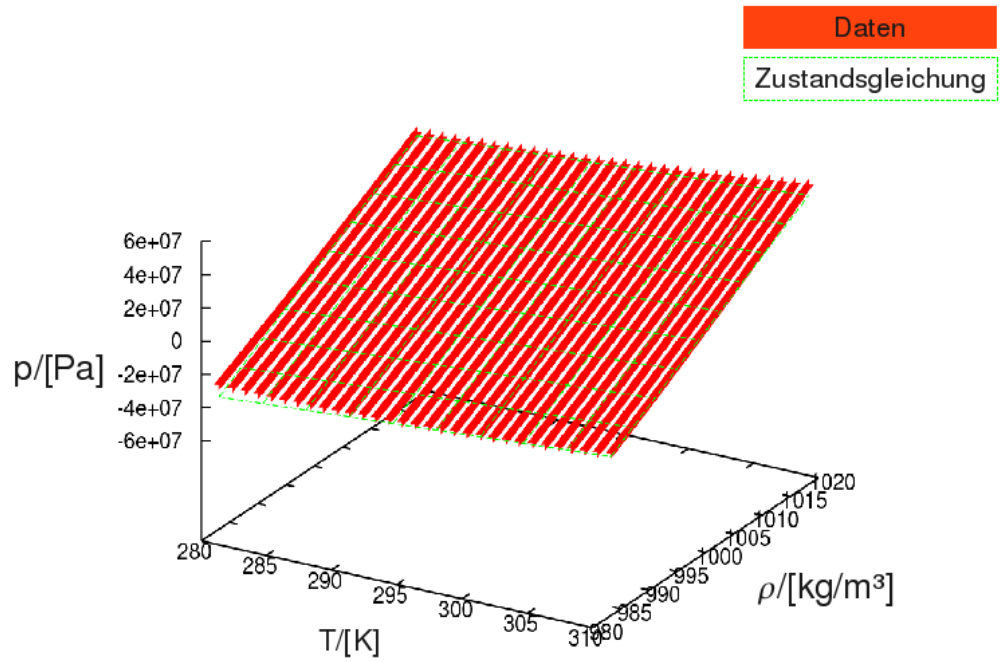


Abbildung 2.1: Vergleich der Zustandsgleichung mit Daten aus [3]

2.2 Entropiegleichung

Wie in der Einleitung erwähnt, werden zur Beschreibung von Wasserströmungen Differentialgleichungen verwendet. In der Differentialgleichung, die sich auf den Impuls bezieht (Gleichungen 3.2-3.4), wird der Druck benötigt, wobei die Dichte und die Entropie als bekannt vorausgesetzt werden können. Es wird also eine Gleichung benötigt, die aus den anderen bekannten Größen, Dichte und Entropie, den Druck berechenbar macht.

Das vollständige Differential der Entropie in Abhängigkeit von V und T stellt sich folgendermaßen dar:

$$dS = \left(\frac{\partial S}{\partial V}\right)_T dV + \left(\frac{\partial S}{\partial T}\right)_V dT.$$

Dabei ist die erste partielle Ableitung $\left(\frac{\partial S}{\partial V}\right)_T$ bereits über die Relation (Definition der isochoren Wärmekapazität):

$$\left(\frac{\partial S}{\partial T}\right)_V = \frac{C_V}{T}$$

bekannt.

Um einen Ausdruck für die zweite partielle Ableitung zu finden benutzen wir den 1. Hauptsatz:

$$dU = TdS - pdV,$$

und gehen über zur freien Energie mit $F = U - TS$:

$$dF = -SdT - pdV.$$

Dieser Ausdruck stellt ein vollständiges Differential dar, man erhält durch den Satz von Schwarz:

$$\left(\frac{\partial S}{\partial V}\right)_T = \left(\frac{\partial p}{\partial T}\right)_V$$

(Dies ist eine der sogenannten Maxwellrelationen).

$\left(\frac{\partial p}{\partial T}\right)_V$ erhält man aus der bereits hergeleiteten Zustandsgleichung.

$$p = \frac{1}{\kappa} \left(\ln\left(\frac{\rho}{C}\right) + \alpha T \right)$$

$$\left(\frac{\partial p}{\partial T}\right)_V = \frac{\alpha}{\kappa}$$

Somit gelangen wir zu einem vollständigem Differential für S :

$$dS = \frac{\alpha}{\kappa} dV + \frac{C_V}{T} dT,$$

2 Thermodynamische Herleitungen für Wasser

womit sich ein Ausdruck für die Entropie ergibt.

$$S = \frac{\alpha}{\kappa} V + C_V \ln(T)$$

Teilt man dies nun durch die Masse erhalten wir die spezifische Entropie.

$$s = \frac{\alpha}{\kappa} \frac{1}{\rho} + c_V \ln(T) \quad (2.3)$$

Damit haben wir das Ziel erst einmal erreicht, denn wenn wir jetzt die Temperatur durch die Zustandsgleichung ersetzen, erhalten wir die gesuchte Relation. Allerdings hat es sich bewährt eine sogenannte potentielle Temperatur einzuführen, die man an Stelle der doch recht unanschaulichen Werte der Entropie angibt. Die potentielle Temperatur ist dabei die Temperatur, die gegeben wäre, wenn sich das System bei einer bestimmten Entropie unter der Normalbedingung $\rho = \rho_0$ befinden würde. Die Meteorologie benutzt bei Luft die potentielle Temperatur im Allgemeinen als die Temperatur, die bei Normaldruck herrschen würde. Jedoch hat es sich in dieser Arbeit, aufgrund der Gleichungsstruktur bei Wasser, angeboten eine potentielle Temperatur bei Normaldichte einzuführen.

Die potentielle Temperatur ist damit über:

$$s = \frac{\alpha}{\kappa} \frac{1}{\rho_0} + c_V \ln(\theta) \quad (2.4)$$

definiert. θ in Abhängigkeit von ρ und T ergibt sich dann als:

$$\begin{aligned} \frac{\alpha}{\kappa} \frac{1}{\rho_0} + c_V \ln(\theta) &= \frac{\alpha}{\kappa} \frac{1}{\rho} + c_V \ln(T) \\ \ln(\theta) &= \left(\frac{\alpha}{c_V \kappa} \frac{1}{\rho} + \ln(T) - \frac{\alpha}{\kappa c_V} \frac{1}{\rho_0} \right) \\ \theta &= T e^{\left(\frac{\alpha}{c_V \kappa} \left(\frac{1}{\rho} - \frac{1}{\rho_0} \right) \right)}. \end{aligned} \quad (2.5)$$

Es sei nochmals darauf hingewiesen, dass die potentielle Temperatur nur ein Maß für die Entropie ist. Die Entropie und die potentielle Temperatur sind über Gleichung (2.4) bijektiv miteinander verknüpft. Es wäre möglich die gesamte Simulation direkt mit der Entropie durchzuführen, nur wären diese Werte unanschaulich.

Die Gleichung (2.3) wurde nun durch Gleichung (2.5) ersetzt und wird uns zur Druckbestimmung dienen. Wir ersetzen nun die Temperatur durch die Zustandsgleichung und erhalten unseren Druck in Abhängigkeit von θ und ρ :

$$p = \frac{\alpha}{\kappa} \left(\frac{\theta}{e^{\left(\frac{\alpha}{c_V \kappa} \left(\frac{1}{\rho} - \frac{1}{\rho_0} \right) \right)}} + \frac{1}{\alpha} \ln\left(\frac{\rho}{C}\right) \right)$$

2.3 Bilanzgleichung der Entropie

Die Entropie kann, wie die Dichte und der Impuls, als eine Größe betrachtet werden, die man entsprechend einer Bilanzgleichung transportieren kann. Dies ist möglich, da die Entropie eine extensive Zustandsvariable darstellt. Das Ziel dieses Abschnittes ist es die Bilanzgleichung für die Entropie und später für $\rho\theta$ zu formulieren.

Wir gehen von der Bilanzgleichung in einer allgemeinen Form aus (siehe dazu [8]):

$$\int \left[\frac{\partial \rho\psi(\vec{x}, t)}{\partial t} + \text{div}(\vec{v}\rho\psi(\vec{x}, t)) \right] dV = \int \left[-\nabla \vec{F}_\Psi + Q_\Psi \right] dV. \quad (2.6)$$

Dabei ist der Zusammenhang zwischen $\rho\psi$ und der betrachteten extensiven Größe Ψ gegeben durch:

$$\Psi = \int \rho\psi dV. \quad (2.7)$$

Die Größe $-\nabla \vec{F}_\Psi$ beschreibt die Änderungen durch äußere Wechselwirkungen. Dabei beschreibt "Außen" das Raumgebiet um das Integrationsvolumen.

$$\frac{d_e \Psi}{dt} = \int -\nabla \vec{F}_\Psi dV \quad (2.8)$$

(e für engl. exterior)

Dagegen ist Q_Ψ die Änderung der Extensiven Größe je Volumen und Zeit im Inneren des Integrationsvolumens.

$$\frac{d_i \Psi}{dt} = \int Q_\Psi dV \quad (2.9)$$

(i für engl. interior)

Im Fall der Entropie wird nur der zweite Term eine Rolle spielen. Dies liegt darin begründet, dass in der rechten Seite von Gleichung (2.6) die sogenannten Quellterme eingefügt werden müssen, welche unterschiedliche Darstellungsformen haben können. Diese hängen von der betrachteten extensiven Größe ab. Zum Beispiel hat der Impuls einen Quellterm der Form $-\nabla p$ ($= -\nabla \cdot F$; analog zu Gleichung 2.8), während die Dichte keinerlei Quellen besitzt. Im Falle der Entropie lassen sich die Quellen in der Form von Gleichung (2.9) darstellen.

Um Gleichung (2.6) auf die Entropie anzuwenden benötigt man für diese Ausdrücke entsprechend den Gleichungen (2.7) und (2.9), die nun bestimmt werden sollen.

Die Entropie kann man als Produkt aus Masse und spezifischer Entropie darstellen:

$$S(t) = m(t) \cdot s(t) = \int \rho(\vec{x}, t) dV \cdot s(t) = \int \rho(\vec{x}, t) \cdot s(t) dV.$$

2 Thermodynamische Herleitungen für Wasser

Dabei ist $s(t)$ die zeitabhängige spezifische Entropie der Masse im Integrationsvolumen. Diese wurde nur in das Integral gezogen, damit die Analogie zu Gleichung (2.7) deutlich wird ($\Psi = S$; $\psi = s$).

Damit erhalten wir die Zeitableitung.

$$\frac{d_i S(t)}{dt} = \frac{d_i}{dt} \left(\int \rho(\vec{x}, t) dV \right) \cdot s(t) + \int \rho(\vec{x}, t) dV \cdot \frac{d_i s(t)}{dt}$$

$\int \rho(\vec{x}, t) dV$ entspricht einer Masse. Masse kann weder erzeugt noch vernichtet werden, ist also quellenfrei.

$$\frac{d_i}{dt} \int \rho(\vec{x}, t) dV = 0$$

Wir erhalten für die strömungsunabhängige Änderung der Entropie den Ausdruck:

$$\frac{d_i S(t)}{dt} = \int \rho(\vec{x}, t) dV \cdot \frac{d_i s(t)}{dt} = \int \rho(\vec{x}, t) \cdot \frac{d_i s(t)}{dt} dV.$$

Und können nun in Gleichung (2.9) $Q_\Psi = \rho \cdot \frac{d_i s}{dt}$ identifizieren. Damit erhalten wir die Integraldarstellung der Bilanzgleichung.

$$\int \frac{\partial \rho s}{\partial t} dV + \int \text{div}(\vec{v} \rho s) dV = \int \left[\rho \cdot \frac{d_i s(t)}{dt} \right] dV \quad (2.10)$$

Um die übliche Differentialdarstellung einer Bilanzgleichung zu erhalten, beseitigen wir die Integrale. Die Gleichung (2.6) gilt für beliebige Volumina. Allerdings ist zu beachten, dass s die spezifische Entropie desjenigen Raumgebietes darstellt, über das integriert wird. Betrachten wir nun also infinitesimal kleine, beliebige Volumina bezieht sich s immer noch auf dieses. Somit hängt s nun zusätzlich vom Ort des infinitesimalen Volumens ab. Die Ableitung $\frac{d_i}{dt}$ soll jedoch wie zuvor eine reine Ableitung nach der Zeit bleiben, wofür sie in eine partielle Zeitableitung umgeschrieben werden muss. Es ergibt sich die differentielle Darstellung der Bilanzgleichung der Entropie.

$$\frac{\partial \rho(\vec{x}, t) \cdot s(\vec{x}, t)}{\partial t} + \text{div} \left[v(\vec{x}, t) \cdot \rho(\vec{x}, t) \cdot s(\vec{x}, t) \right] = \rho(\vec{x}, t) \cdot \frac{\partial_i s(t, \vec{x})}{\partial t} \quad (2.11)$$

Von der bekannten Formel für die Entropie $\partial S = \frac{\partial Q}{T}$ erhalten wir $\frac{\partial_i s(t, \vec{x})}{\partial t}$.

$$\frac{\partial S}{\partial Q} = \frac{1}{T} = \frac{\partial S}{\partial t} \frac{\partial t}{\partial Q}.$$

$$\dot{S} = \frac{\dot{Q}}{T}$$

Man geht nun zur spezifischen Entropie über, was durch $\frac{\partial m}{\partial t} = 0$ einfach ist.

$$\dot{s} = \frac{\dot{q}}{T}$$

2 Thermodynamische Herleitungen für Wasser

(Hier wurde die Ableitung $\frac{d_i}{dt}$ durch die Punktschreibweise ersetzt. Der Leser sollte im Hinterkopf behalten, dass die strömungsunabhängige, innere zeitliche Änderung gemeint ist.)

Der gesuchte Quellterm stellt sich nun durch

$$\rho \cdot \dot{s} = \rho \cdot \frac{\dot{q}}{T}$$

dar.

Wir erhalten die Bilanzgleichung.

$$\frac{\partial \rho s}{\partial t} + \text{div}(\vec{v} \rho s) = \rho \frac{\dot{q}}{T} \quad (2.12)$$

Nun wollen wir, wie im vorigen Abschnitt angekündigt, statt der Entropie die Größe $\rho\theta$ zum Rechnen verwenden. Dafür beseitigen wir zunächst die Dichte aus der Bilanzgleichung. Nach Produktregel folgt:

$$\frac{\partial \rho s}{\partial t} + \text{div}(\vec{v} \rho s) = \frac{\partial \rho}{\partial t} s + \text{div}(\vec{v} \rho) s + \rho \frac{\partial s}{\partial t} + \vec{v} \rho \nabla s \quad (2.13)$$

Damit erhält man durch Einsetzen in Gleichung (2.12) und mit der Relation $\frac{\partial \rho}{\partial t} + \text{div}(\vec{v} \rho) = 0$:

$$\frac{\partial s}{\partial t} + \vec{v} \nabla s = \frac{\dot{q}}{T} \quad (2.14)$$

Nun benutzt man die Definition der potentiellen Temperatur (Gleichung 2.4):

$$s = \frac{\alpha}{\kappa} \frac{1}{\rho_0} + c_V \ln(\theta) \quad (2.15)$$

und transformiert damit die Differentialgleichung 2.14, indem die Kettenregel angewendet wird.

$$\frac{ds}{d(\cdot)} = \frac{ds}{d(\theta)} \frac{d(\theta)}{d(\cdot)} = \frac{c_V}{\theta} \frac{d\rho\theta}{d(\cdot)} \quad (2.16)$$

Einsetzen in die Gleichung 2.14 liefert:

$$\begin{aligned} \frac{c_V}{\theta} \left(\frac{\partial \theta}{\partial t} + \vec{v} \nabla \theta \right) &= \frac{\dot{q}}{T} \\ \frac{\partial \theta}{\partial t} + \vec{v} \nabla \theta &= \frac{\theta}{c_V} \frac{\dot{q}}{T} \end{aligned}$$

Um nun wieder die typische Form einer Bilanzgleichung zu erhalten, geht man den Schritt der für Gleichung 2.14 gemacht wurde noch einmal rückwärts und erhält:

$$\frac{\partial \rho \theta}{\partial t} + \text{div}(\vec{v} \rho \theta) = \frac{\rho \theta}{c_V} \frac{\dot{q}}{T}.$$

Dies ist die Transportgleichung für $\rho\theta$, die in der Simulation benutzt wird.

3 Differentialgleichungen und Numerik der Wassersimulation in den inneren Zellen

3.1 Grundlagen

(In einigen der folgenden Abschnitte wird, um den allgemeineren Fall abzuhandeln, für drei Dimensionen argumentiert. In der Simulation werden jedoch nur zwei Dimensionen berücksichtigt.)

Eine Flüssigkeit wird, wie bereits erwähnt, durch drei wichtige Eigenschaften beschrieben: **Druck**, **Temperatur** und **Dichte**. Diese Eigenschaften können von Ort zu Ort variieren. Will man eine sich bewegende Flüssigkeit simulieren, braucht man als zusätzliche Eigenschaft die lokalen **Geschwindigkeiten** u, v, w (Geschwindigkeit in x, y, z Richtung), wie sie bereits in der Transportgleichung der Entropie auftraten. So werden jedem Raumpunkt (x, y, z) Temperatur, Druck, Dichte und die drei Geschwindigkeitskomponenten zugeordnet. Oder anders herum aufgefasst: Druck, Temperatur, Dichte und Geschwindigkeit sind Funktionen vom Ort und, da kein statisches System beschrieben werden soll, auch von der Zeit. Die Zeitentwicklung wird durch die in Abschnitt 3.2 gegebenen Differentialgleichungen, mit Hilfe der Numerik, beschrieben. Wie dies geschieht, wird in diesem Kapitel erläutert.

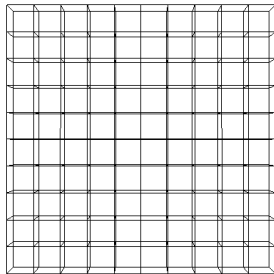


Abbildung 3.1: 10x10 2D Gitter

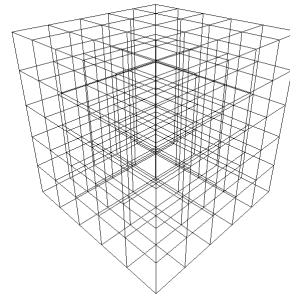


Abbildung 3.2: 5x5x5 3D Gitter

Die Wasserströmungen werden in dieser Arbeit mit der Finite-Volumen-Methode berechnet. Die Finite-Volumen-Methode beschreibt die räumliche Diskretisierung. Diese beinhaltet, dass man beispielsweise die Funktion $\rho(x, y, z, t)$ nicht kontinuierlich an jedem Raumpunkt kennt, sondern nur in vielen kleinen Volumina. Diese Volumina sind in dieser Arbeit Quader und bilden ein Gitter. In jeder Gitterzelle wird ein Wert der

3 Differentialgleichungen und Numerik der Wassersimulation in den inneren Zellen

Dichte gespeichert. Dieser soll der Mittelwert der realen, kontinuierlichen Dichteverteilung sein also:

$$\rho_{Zelle} = \frac{1}{V_z} \int_{V_z} \rho(x, y, z) dx^3.$$

Diese Arbeit wird auf zwei Dimensionen beschränkt sein, also auf zweidimensionale Gitter. In den Abbildungen 3.1 und 3.2 sind Bilder von zwei- und dreidimensionalen Gittern zu sehen.

Allerdings werden nicht alle Größen auf der Zellmitte diskretisiert. Der Grund dafür liegt in der Komplexität der später behandelten Differentialgleichungen und wird in Abschnitt 3.8 erläutert. Druck, Dichte und $\rho\theta$ sind Größen der Zellmitte, während man die Geschwindigkeiten auf dem Rand der Zelle diskretisiert.

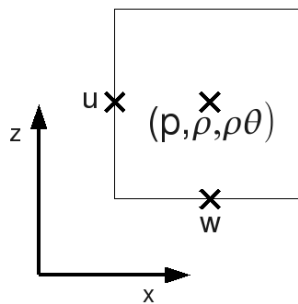


Abbildung 3.3: Einordnung der Größen in oder auf dem Rand der Zelle

Man benutzt also anstatt einer kontinuierlichen Verteilung von *Druck*, *Dichte*, $\rho\theta$ und \vec{v} bezüglich des Ortes eine diskrete. Daraus lässt sich schon erahnen, dass man hierbei die Realität nicht genau beschreibt (also einen numerischen Fehler macht), aber das je feiner das Gitter ist, die Realität besser angenähert wird.

In Abbildung 3.4 findet man ein Beispiel für die Luftschichtung nach der barometrischen Höhenformel (jedem Gitterpunkt wurde eine Dichte zugeordnet).

Zusätzlich zur räumlichen Diskretisierung benötigt man eine zeitliche. Diese wird mit einem Zeitintegrationsverfahren realisiert. So kann man aus einer Gleichung $\frac{\partial \rho}{\partial t} = f(t, \rho)$ und einer bekannten aktuellen Dichte die Dichte nach einer diskreten Zeitspanne Δt näherungsweise bestimmen (numerisch integrieren). Somit erhält man ρ zu diskreten Zeitpunkten.

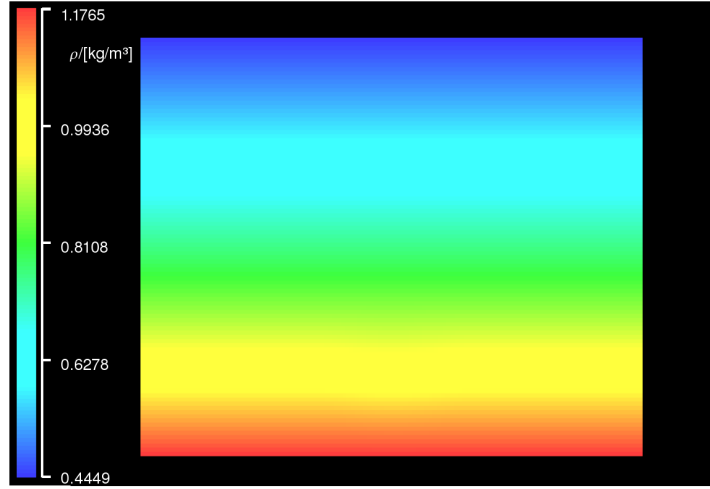


Abbildung 3.4: Luftschichtung

3.2 Differentialgleichungen

Die Differentialgleichungen, die der Strömungssimulation des Wassers zugrunde liegen, sind die Kontinuitätsgleichung für die Dichte, die Navier-Stokes Gleichung und die Bilanzgleichung für die Entropie. Ausgeschrieben sehen diese wie folgt aus:

Kontinuitätsgleichung der Dichte:

$$\frac{\partial \rho}{\partial t} = -\frac{\partial \rho u}{\partial x} - \frac{\partial \rho v}{\partial y} - \frac{\partial \rho w}{\partial z} \quad (3.1)$$

Navier-Stokes-Gleichungen (unter Vernachlässigung des Stresstensors und des Viskositätsterms):

$$\frac{\partial \rho u}{\partial t} = -\frac{\partial u(\rho u)}{\partial x} - \frac{\partial v(\rho u)}{\partial y} - \frac{\partial w(\rho u)}{\partial z} - \frac{\partial p}{\partial x} + \rho a_x \quad (3.2)$$

$$\frac{\partial \rho v}{\partial t} = -\frac{\partial u(\rho v)}{\partial x} - \frac{\partial v(\rho v)}{\partial y} - \frac{\partial w(\rho v)}{\partial z} - \frac{\partial p}{\partial y} + \rho a_y \quad (3.3)$$

$$\frac{\partial \rho w}{\partial t} = -\frac{\partial u(\rho w)}{\partial x} - \frac{\partial v(\rho w)}{\partial y} - \frac{\partial w(\rho w)}{\partial z} - \frac{\partial p}{\partial z} + \rho a_z \quad (3.4)$$

(\vec{a} bezeichnet die Beschleunigung aufgrund eines äußeren Potentials)

Bilanzgleichung für die potentielle Temperatur (Entropie):

$$\frac{\partial \rho \theta}{\partial t} = -\frac{\partial u \rho \theta}{\partial x} - \frac{\partial v \rho \theta}{\partial y} - \frac{\partial w \rho \theta}{\partial z} + \frac{\rho \theta}{c_V T} \dot{q} \quad (3.5)$$

Wir erinnern uns an die Eigenschaften von Wasser aus dem vorigen Abschnitt und finden zur Dichte und den drei Geschwindigkeitskomponenten jeweils eine eigene Differentialgleichung die, in der Zeit integriert, die Entwicklung dieser Größe angibt. Wie diese Zeitentwicklung mit Hilfe der Numerik gewonnen wird, ist in Abschnitt 3.3 erläutert. Aus der letzten Differentialgleichung folgt die Zeitentwicklung der potentiellen Temperatur (der Entropie).

Die benötigte Gleichung für den Druck lautet:

$$p = \frac{\alpha}{\kappa} \left(\frac{\theta}{e^{\left(\frac{\alpha}{c_V \kappa} \left(\frac{1}{\rho} - \frac{1}{\rho_0}\right)\right)}} + \frac{1}{\alpha} \ln\left(\frac{\rho}{C}\right) \right).$$

3.3 Runge-Kutta-Verfahren

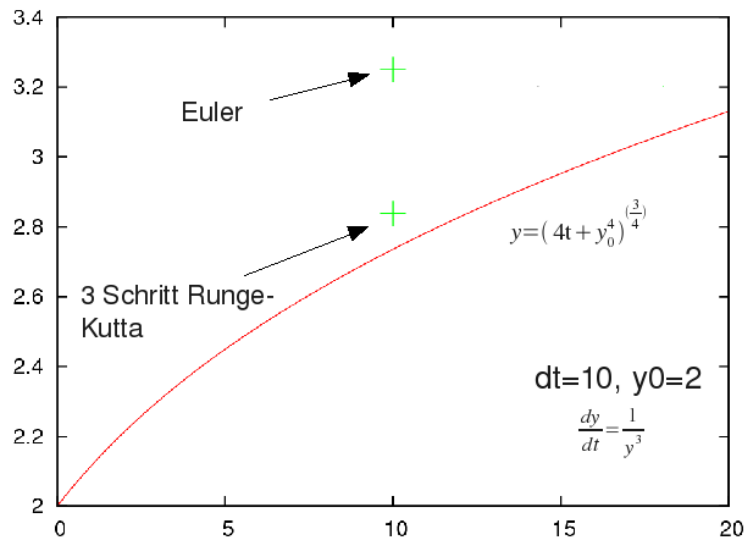


Abbildung 3.5: Runge-Kutta- im Vergleich mit dem Eulerverfahren

Das Zeitintegrationsverfahren stellt ein wichtiges Element der Simulation dar und nähert eine Differentialgleichung der Form: $\frac{dy}{dt} = f(y(t), t)$ bzw. in Integralschreibweise:

$$y(t) = \int f(y(t), t) dt.$$

3 Differentialgleichungen und Numerik der Wassersimulation in den inneren Zellen

Die Funktion $f(y(t), t)$ bezeichnet man dabei als Right-Hand Side, da sie die rechte Seite der zu lösenden Differentialgleichung darstellt.

Ein Zeitintegrationsverfahren kann immer nur aus dem zu einem Zeitpunkt bekannten Wert $y(t)$ den Wert nach einer endlichen Zeit $y(t + \Delta t)$ berechnen. Bei der Methode dieser Berechnung unterscheidet man zwischen expliziten und impliziten Zeitintegrationsverfahren. Bei den expliziten Zeitintegrationsverfahren geht $y(t + \Delta t)$ direkt aus $y(t)$ hervor, während bei den impliziten Verfahren eine Gleichung:

$$y(t + \Delta t) = h(t, y(t), y(t + \Delta t))$$

entsteht, die es zu lösen gilt. In dieser Arbeit wurde ein explizites Integrationsverfahren gewählt.

Das einfachste explizite Integrationsverfahren ist das Eulerverfahren $y = y_0 + f(y(t), t)\Delta t$, welches jedoch sehr ungenau ist. Das in dieser Arbeit verwendete Zeitintegrationsverfahren ist ein explizites dreistufiges Runge-Kutta-Verfahren. Dieses ist ein sogenanntes Mehrschrittverfahren, da zur Berechnung von $y(t + \Delta t)$ mehrere (in diesem Falle drei) Zwischenschritte vollzogen werden. Zusätzlich ist es wesentlich genauer als z.B. das Eulerverfahren.

Zur Berechnung eines Zeitschrittes werden an drei Stellen die Anstiege $f(y(t), t)$ (Right-Hand Sides) berechnet.

Der erste Anstieg wird an der Stelle y_0 zum Zeitpunkt t_0 berechnet:

$$k1 = f(y_0, t_0).$$

Danach wird ein Zwischenschritt $y_1 = y_0 + \frac{\Delta t}{2}k1$ vollzogen und an diesem Punkt der neue Anstieg berechnet:

$$k2 = f(y_1, t_0 + \frac{\Delta t}{2}).$$

Es erfolgt ein weiterer Zwischenschritt $y_2 = y_0 - \Delta t \cdot k1 + 2 \cdot \Delta t \cdot k2$, an welchem erneut ein Anstieg berechnet wird:

$$k3 = f(y_2, t_0 + \Delta t).$$

Diese drei Anstiege werden nun benutzt, um das neue y_n zu berechnen:

$$y_n = y_0 + \Delta t \left(\frac{1}{6}k1 + \frac{4}{6}k2 + \frac{1}{6}k3 \right).$$

Um genauer herauszufinden warum und wie genau das Runge-Kutta-Verfahren funktioniert sei hier auf die mathematische Literatur zB. [4] verwiesen. In Abbildung 3.5 ist ein Beispiel für den resultierenden Punkt des dreistufigen Runge-Kutta-Verfahrens für die Differentialgleichung $\frac{dy}{dt} = \frac{1}{y^3}$ gegeben, die durchgehende Linie ist die analytische Lösung $y = (4t + y_0^4)^{\frac{1}{4}}$.

Schauen wir uns nun die Differentialgleichungen (Gleichungen 3.1-3.5) an, sehen wir dass in unserem Fall die Funktion $f(y(t), t)$ zu einem großen Teil aus räumlichen Ableitungen besteht. Die Werte dieser örtlichen Gradienten werden mit Hilfe der

bekanntem, diskreten Werte des Gitters genähert (dies wird das Thema des Abschnitts 3.4 und 3.5 sein). Dazu ist es allerdings notwendig, dass für jeden Zwischenschritt y_1, y_2 alle Werte auf dem gesamten Gitter berechnet werden, damit die örtlichen Ableitungen bei jedem Zwischenschritt berechnet werden können. Das Runge-Kutta-Verfahren kann also nicht auf jeden Gitterpunkt einzeln, sondern nur auf das gesamte Gitter angewendet werden. Dabei stellt das Gitter selbst die räumliche Diskretisierung dar und das Runge-Kutta-Verfahren berechnet die zusätzliche zeitliche Diskretisierung.

3.4 Upwindverfahren

In Abschnitt 3.3 wurde beschrieben, wie die Zeitintegration in der Wassersimulation erfolgt. Dabei muss in jedem Schritt die Right-Hand Side berechnet werden. Im Falle der in dieser Arbeit benutzten Differentialgleichungen beinhaltet jede von ihnen die Divergenz. Eine Methode diese zu berechnen, die gute Ergebnisse bei der Anwendung auf das hier benutzte Differentialgleichungssystem gibt, ist das Upwindverfahren. Wie in diesem die Divergenz berechnet wird, soll hier für die Dichte erläutert werden und erfolgt für die anderen Größen analog. Beim Upwindverfahren werden die Geschwindigkeiten benötigt, die auf dem Rand der Zelle diskretisiert wurden, und die Dichtewerte der Nachbarzellen, um den Wert der Divergenz zu berechnen.

Wir starten mit der betreffenden Differentialgleichung der Form:

$$\frac{\partial \rho}{\partial t} = -\frac{\partial \rho u}{\partial x} - \frac{\partial \rho v}{\partial y} - \frac{\partial \rho w}{\partial z} = -\text{div}(\vec{v}\rho).$$

Diese Differentialgleichung beschreibt die Dichteadvektion, also den Transport von Dichte mit einer Geschwindigkeit \vec{v} .

Wir integrieren beide Seiten über das Zellvolumen und benutzen den Gaußschen Integralsatz:

$$\int_{V_z} \frac{\partial \rho}{\partial t} dx^3 = \int_{V_z} \text{div}(\vec{v}\rho) dx^3 = \int_{S_{V_z}} \vec{v}\rho \vec{ds}.$$

Entsprechend der Finite-Volumen-Methode ist die zeitliche Änderung einer Gitterzelle in dieser eine Konstante:

$$\int_{V_z} \frac{\partial \rho}{\partial t} dx^3 = V_z \frac{\partial \rho}{\partial t}.$$

S_{V_z} stellt die Außenflächen der Gitterzelle dar und \vec{ds} zeigt in die jeweilige Flächennormalenrichtung. Mit Abbildung 3.6 wird ersichtlich, dass sich das Integral aufsplitten lässt:

$$\int_{S_{V_z}} \vec{v}\rho \vec{ds} = \int_{A_{rechts}} \vec{v}\rho \vec{ds} + \int_{A_{links}} \vec{v}\rho \vec{ds} + \int_{A_{oben}} \vec{v}\rho \vec{ds} + \int_{A_{unten}} \vec{v}\rho \vec{ds} + \int_{A_{vorne}} \vec{v}\rho \vec{ds} + \int_{A_{hinten}} \vec{v}\rho \vec{ds}$$

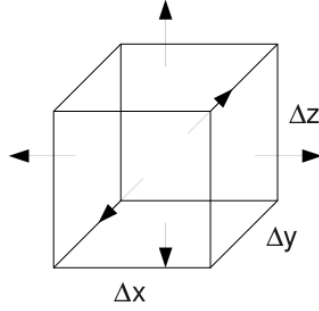


Abbildung 3.6: Gitterzelle mit Flächennormalen

Nun nehmen wir konstante Dichteflüsse auf den Flächen an, also $u\rho(x + \frac{\Delta x}{2})$ etc. ($u = v_x$; die Dichte auf der Fläche wird später noch zu interpolieren sein), und erhalten:

$$\begin{aligned} \int_{S_{V_z}} \vec{v}\rho d\vec{s} &= A_{rechts}u(x + \frac{\Delta x}{2})\rho(x + \frac{\Delta x}{2}) - A_{links}u(x - \frac{\Delta x}{2})\rho(x - \frac{\Delta x}{2}) \\ &+ A_{hinten}v(y + \frac{\Delta y}{2})\rho(y + \frac{\Delta y}{2}) - A_{vorne}v(y - \frac{\Delta y}{2})\rho(y - \frac{\Delta y}{2}) \\ &+ A_{oben}w(z + \frac{\Delta z}{2})\rho(z + \frac{\Delta z}{2}) - A_{unten}w(z - \frac{\Delta z}{2})\rho(z - \frac{\Delta z}{2}). \end{aligned}$$

An dieser Stelle wird einer der Gründe ersichtlich, warum es sinnvoll ist die Geschwindigkeit auf dem Rand der Zelle zu initialisieren. Dort wird sie bei der Divergenzberechnung gebraucht, um den Transport von Dichte, Entropie und Impuls zu beschreiben. Des Weiteren wird hier die Bedeutung der Advektionsgleichung (Kontinuitätsgleichung) deutlich. Die Veränderung einer Erhaltungsgröße in der Zelle entspricht dem Fluss derselben Größe über die Außenflächen. Bei einer Bilanzgleichung tritt hier zusätzlich ein Quellterm auf. Beispielsweise erscheint beim Impuls die Kraft als Quellterm. Fassen wir nun unsere Ergebnisse für die Zeitableitung und die Divergenz zusammen erhalten wir:

$$V_z \frac{\partial \rho}{\partial t} = A_{rechts}u_{rechts}\rho(x + \frac{\Delta x}{2}) - A_{links}u_{links}\rho(x - \frac{\Delta x}{2}) \quad (3.6)$$

$$+ A_{hinten}v_{hinten}\rho(y + \frac{\Delta y}{2}) - A_{vorne}v_{vorne}\rho(y - \frac{\Delta y}{2}) \quad (3.7)$$

$$+ A_{oben}w_{oben}\rho(z + \frac{\Delta z}{2}) - A_{unten}w_{unten}\rho(z - \frac{\Delta z}{2}). \quad (3.8)$$

$$(3.9)$$

3 Differentialgleichungen und Numerik der Wassersimulation in den inneren Zellen

Teilen wir beide Seiten durch das Volumen ergibt sich:

$$\begin{aligned} \frac{\partial \rho}{\partial t} &= \frac{u_{rechts}\rho(x + \frac{\Delta x}{2})}{\Delta x} - \frac{u_{links}\rho(x - \frac{\Delta x}{2})}{\Delta x} \\ &+ \frac{v_{hinten}\rho(y + \frac{\Delta y}{2})}{\Delta y} - \frac{v_{vorne}\rho(y - \frac{\Delta y}{2})}{\Delta y} \\ &+ \frac{w_{oben}\rho(z + \frac{\Delta z}{2})}{\Delta z} - \frac{w_{unten}\rho(z - \frac{\Delta z}{2})}{\Delta z}. \end{aligned}$$

Erst jetzt kommen wir zum Kern des eigentlichen Upwindverfahrens. Wie man erkennen kann, muss man für die obige Gleichung die Dichte auf dem Rand der Zelle berechnen. In dieser Arbeit wird dazu das Upwindverfahren dritter Ordnung benutzt, wobei folgendermaßen vorgegangen wird.

Zur Interpolation der Randdichte werden drei Zellen benutzt. Um die Formel zu erhalten, mit der die Randdichte aus drei Zellen interpoliert werden kann, benutzt man ein Polynom zweiten Grades und bildet den Mittelwert für die Zellen. Dieser entspricht den für diese Zellen gesetzten Dichtewerten.

$$\rho_{Zelle1} = \frac{1}{\Delta x} \int_0^{\Delta x} \alpha x^2 + \beta x + \gamma dx \quad (3.10)$$

$$\rho_{Zelle2} = \frac{1}{\Delta x} \int_{\Delta x}^{2\Delta x} \alpha x^2 + \beta x + \gamma dx \quad (3.11)$$

$$\rho_{Zelle3} = \frac{1}{\Delta x} \int_{2\Delta x}^{3\Delta x} \alpha x^2 + \beta x + \gamma dx \quad (3.12)$$

Damit erhält man drei Gleichungen für die drei Variablen α, β, γ und kann somit die drei Koeffizienten des Polynoms berechnen. Damit ergibt sich die Dichte auf der Fläche zwischen den beiden Zellen. Zusätzlich wird beim Upwindverfahren die Wahl der drei Zellen zur Interpolation von der Richtung der Geschwindigkeit (=Windrichtung bei Luft) abhängig gemacht. Es werden zwei Zellen der Richtung genommen aus der der Wind kommt und eine der anderen Richtung (siehe Abbildung 3.7). Daher der Name 'Upwind'.

Als Formel zur Interpolation nach dem Upwindverfahren ergibt sich nach der beschriebenen Methode (mit der Nummerierung wie in Abbildung 3.7):

$$\rho(Rand) = \frac{\rho_{Zelle3}}{3} + \frac{5\rho_{Zelle2}}{6} - \frac{\rho_{Zelle1}}{6}$$

Somit erhalten wir den Right-Hand Side Anteil, der durch die Divergenz entsteht.

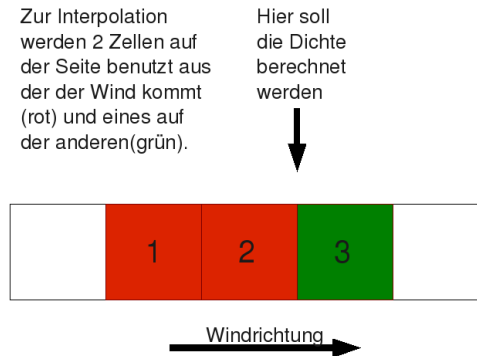


Abbildung 3.7: Zur Veranschaulichung des Upwindverfahrens

3.5 Quellterme und Zusammenfassung

Mit der Berechnung der Divergenz, entsprechend dem vorigen Abschnitt, ist die Right-Hand Side für die quellenfreie Differentialgleichung 3.1 der Dichte vollständig abgehandelt. Die Gleichungen für potentielle Temperatur und Impuls besitzen jedoch Quellterme welche zur Right-Hand Side hinzu kommen.

Der einfachere dieser beiden Quellterme ist der der potentiellen Temperatur:

$$\frac{\rho\theta}{c_V} \frac{\dot{q}}{T}$$

Dieser beschreibt die Erhöhung der potentiellen Temperatur (Entropie) durch einen Wärmefluss. In der Simulation wird dieser nur am Rand berechnet, falls die Flüssigkeit erwärmt/gekühlt werden soll. Dabei setzt man \dot{q} als:

$$\dot{q} = \frac{\vec{F}}{m} \lambda \text{grad}(T) \quad (\vec{F} = \text{Flächennormalenvektor}).$$

Wenn man nun eine Temperatur der Wand vorgibt und die Temperatur in der zu heizenden Zelle kennt, kann man bspw. den x-Gradienten zur rechten Wand approximieren:

$$\frac{\partial T}{\partial x} = \frac{(T_{\text{Wand}} - T_{\text{Zelle}})}{\Delta x}.$$

So erhält man den Wert für den Quellterm der Entropiedichte.

Etwas aufwändiger gestaltet sich die endgültige Berechnung der Right-Hand Side der Impulsdichte. Der Divergenzterm der Impulsdichte wird, wie in Abschnitt 3.4

3 Differentialgleichungen und Numerik der Wassersimulation in den inneren Zellen

beschrieben, im Inneren der Zelle berechnet. Die Geschwindigkeit ist aber auf dem Rand der Zelle diskretisiert. Dieses Problem wird in der Simulation folgendermaßen gelöst:

Zuerst wird in jedem Zeitschritt die Impulsdichte in der Zelle berechnet. Dafür wird die Dichte auf den Rand interpoliert, damit dort mit den Geschwindigkeiten die Impulsdichte berechnet werden kann. Mit Hilfe der nun bekannten Impulsdichten am Rand der Zellen wird die Impulsdichte in die Zellmitte gemittelt (siehe Abbildung 3.8).

Impuls in der Mitte der Zelle bestimmen:

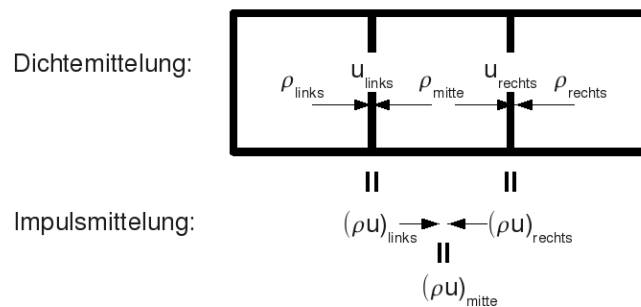


Abbildung 3.8: Impulsdichtenmittlung

Für die nun in der Zelle befindlichen Werte kann das Upwindverfahren benutzt werden, um den Divergenzterm berechnen zu können.

Dieser Divergenzterm stellt nun jedoch die Änderung in der Zellmitte dar, gesucht ist aber die Änderung auf dem Zellrand. Dafür werden nun die Ergebnisse der Divergenzberechnung wieder auf den Rand interpoliert.

Nachdem nun der Divergenzterm auf der Zellfläche bekannt ist, müssen noch der Druckgradient und die Beschleunigung durch ein äußeres Potenzial addiert werden. Dafür wird aus der Dichte in der Zellmitte und $\rho\theta$ der Druck des aktuellen Zustandes ermittelt. Dadurch ergibt sich recht einfach mit der vorher auf den Rand interpolierten Dichte der Quellterm für die Impulsdichten.

Quellterm für ρu auf linker Fläche:

$$\text{gemittelter Divergenzterm für } \rho u \text{ auf linker Fläche} a_x - \frac{(p_{links} - p_{rechts})}{\Delta x}$$

3 Differentialgleichungen und Numerik der Wassersimulation in den inneren Zellen

Quellterm für ρv auf vorderer Fläche:

$$\text{gemittelter Divergenzterm für } \rho v + \rho_{\text{auf vorderer Fläche}} a_y - \frac{(p_{\text{hinten}} - p_{\text{vorne}})}{\Delta y}$$

Quellterm für ρw auf unterer Fläche:

$$\text{gemittelter Divergenzterm für } \rho w + \rho_{\text{auf unterer Fläche}} a_z - \frac{(p_{\text{oben}} - p_{\text{unten}})}{\Delta z}$$

Zusammenfassung

Damit ist der Ablauf eines Zeitschrittes beschrieben. Für das gesamte Gitter wird in zwei Schritten einmal die Divergenz, mit Hilfe des Upwindverfahrens, berechnet und danach der Quellterm hinzuaddiert. Damit ist die Right-Hand Side in jeder Gitterzelle berechnet und es kann mit Hilfe des Runge-Kutta-Verfahrens ein neuer Zwischenzustand, oder nach drei Zwischenschritten ein neuer Zustand $y(t + \Delta t)$, berechnet werden. Dadurch erhält man eine Approximation der Funktionen:

$$\rho(x, y, z, t), \rho\theta(x, y, z, t), u(x, y, z, t), v(x, y, z, t) \text{ und } w(x, y, z, t).$$

3.6 Rand- und Anfangsbedingungen

Randbedingungen

Bei der Berechnung von räumlichen Ableitungen mit Hilfe der Finite-Volumen-Methode ist es wesentlich was am Rand des Gebietes getan wird. Die Randbedingungen können hierbei recht gut mit physikalischen Überlegungen bestimmt werden. Bei der Wasserströmung, die in dieser Arbeit berechnet wird kann dies z.B. eine feste Wand sein. Eine feste Wand hat die Eigenschaft, dass die Geschwindigkeitskomponenten orthogonal zu der Wand Null sein müssen. Dementsprechend findet dort kein Transport von z.B. Dichte statt und es muss an der Wand keine Right-Hand-Side für den Impuls berechnet werden. Die Berechnung des Wärmeflusses am Rand wurde bereits im vorigen Abschnitt erläutert.

Die schwierigeren Randbedingungen sind die an der sich deformierenden Oberfläche. Diese werden im Kapitel 4 behandelt.

Anfangsbedingungen

Zunächst ist es wichtig das Programm mit einer stabilen Schichtung zu initialisieren. Das heißt, dass ein Dichteprofil erstellt werden muss, in dem $\frac{dp}{dz} = -\rho g$ ist. Beachtet man dies nicht, versucht sich dieses Gleichgewicht einzustellen und produziert dabei von oben nach unten laufende Wellen, die erst nach sehr langer Zeit wieder verschwinden. Damit wird der tatsächlich interessierende Sachverhalt (wie zB. Strömungen aufgrund von Temperaturunterschieden) überdeckt. θ wird dabei normalerweise überall mit dem gleichen Wert θ_0 initialisiert.

3 Differentialgleichungen und Numerik der Wassersimulation in den inneren Zellen

In dieser Arbeit wurde dieses Dichteprofil iterativ erstellt. Als Anfangswert wurde dafür Normaldruck in der oberen Reihe der Zellen angenommen. Der Druck und die Dichte in der nächsten Schicht wurden nun berechnet, indem der Druck nach dem Banachschen Fixpunktsatz so lange variiert wurde bis die Gleichung

$$\frac{p_{oben} - p_{unten}}{\Delta z} = - \frac{\rho_{oben}(p_{oben}) + \rho_{unten}(p_{unten})}{2} g$$

zu einer hohen Genauigkeit erfüllt ist. Dieses iterative Verfahren wird bis zur untersten Stelle durchgeführt und die jeweils resultierende Dichte gespeichert. Auf diese Weise wurde eine stabile Dichteschichtung erhalten, die so gut wie keine Geschwindigkeiten in z-Richtung produziert, da der Quellterm des Impulses Null ist. Es ist möglich auch die Geschwindigkeiten mit einem Anfangswert zu initialisieren, jedoch werden diese meist auf Null gesetzt.

Ebenso wird die Oberflächenzellhöhe in den meisten Fällen mit dz initialisiert, damit am Anfang alle Zellen die gleiche Größe besitzen.

3.7 Das CFL-Kriterium

Eine Grenze für die Wahl von Orts- und Zeitdiskretisierung der Strömungssimulation ist die sogenannte CFL-Zahl (Courant-Friedrichs-Lewy-Zahl).

$$c = \frac{u \Delta t}{\Delta x} < 1$$

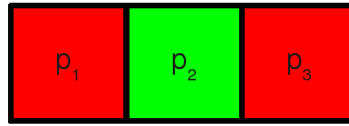
Dabei ist u die Schallgeschwindigkeit im strömenden Medium (im Wasser $\approx 1500 \frac{m}{s}$). Wie man zu dieser Relation kommt findet sich in Referenz [6].

Diese stellt eine Bedingung, damit eine numerische Lösung der Differentialgleichungen erhalten werden kann. Eine Folge des CFL Kriteriums ist, dass mit einem feineren Gitter, also kleineren Gitterzellen, auch ein kleinerer Zeitschritt gewählt werden muss. Zum Beispiel bedeutet dies, dass man bei einem Wassermmodell mit Gitterzellengrößen von 100 m einen Zeitschritt von 0.06 Sekunden benutzen kann. Demgegenüber konnte in dieser Arbeit bei einer Gitterzellengröße von 0.001 m nur ein Zeitschritt von 10^{-7} Sekunden verwendet werden. Demensprechend viel Rechenzeit braucht es um eine Sekunde Simulationszeit zu berechnen.

Räumliche und zeitliche Diskretisierung können nicht unabhängig voneinander feiner oder gröber werden.

3.8 Anmerkung zur Diskretisierung der Geschwindigkeiten

Im Prozess der Erstellung dieser Arbeit wurde auch mit der Möglichkeit experimentiert, die Geschwindigkeiten ebenfalls in der Zellmitte zu initialisieren. Dies schlug aus folgendem Grund fehl:



$$p_3, p_1 > p_2$$

$$p_3 = p_1$$

Abbildung 3.9: Fehler durch Diskretisierung

Zur Right-Hand Side der Impulsdichte gehört der Druckgradient $grad(p)$. Würde dieser Druckgradient in der Mitte der Zelle bestimmt, so würde dieser aus einer symmetrischen Anordnung von umliegenden Zellen berechnet werden, z.B. mit der Methode der Zentralen Differenzen:

$$\frac{p_{links} - p_{rechts}}{2 \cdot \Delta z}$$

Dabei kann die in Abbildung 3.9 dargestellte Situation zustande kommen. In dieser ist der Druckgradient gleich Null. Das heißt, dass ein solcher Druck nicht zur Änderung der Geschwindigkeiten führt. Eine solche Geschwindigkeit wäre jedoch notwendig um den Druck ausgleichen zu können. Folglich entsteht in der Simulation ein Schachbrettmuster, das nicht wieder verschwindet und der Realität widerspricht.

4 Modellierung der freien Oberfläche

4.1 Herangehensweise

Die Oberfläche wurde in dieser Arbeit modelliert, indem die oberen Zellen die Möglichkeit besitzen sich auszudehnen oder zusammenzuziehen um sich dem äußeren Luftdruck anzupassen (siehe Abbildung 4.1). Das heißt das sich das Gitter deformiert. Dadurch muss die Numerik, die bei den inneren Zellen angewendet wird, angeglichen werden, da hier veränderliche Volumina in diesen Zellen vorliegen. Die veränderlichen Volumina führen außerdem zu einem zusätzlichen Quellterm $\frac{d\rho}{dt}$ in den Differentialgleichungen. In diesem Kapitel werden diese abgeänderten Vorgehensweisen beschrieben.

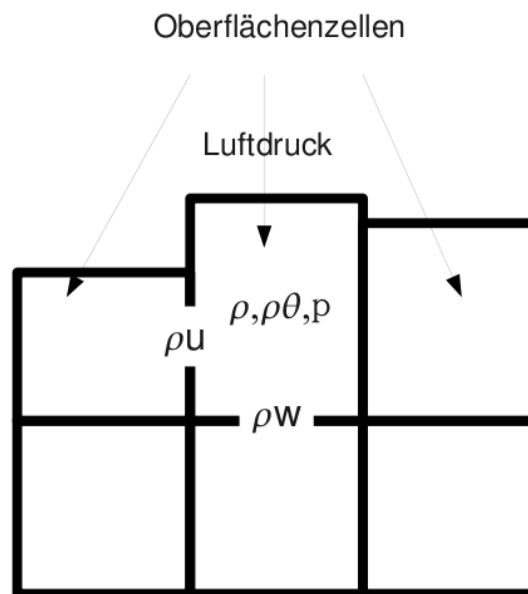


Abbildung 4.1: Visualisierung der Oberflächenmodellierung

4.2 Divergenzterm

Der Divergenzterm wird, wie im Abschnitt 3.4 beschrieben, mit Hilfe des Upwindverfahrens berechnet. Dazu wird die betreffende Größe auf die Mitte der Fläche interpoliert, durch die der Transport erfolgt. Anschließend wird der Fluss bestimmt.

4 Modellierung der freien Oberfläche

Bei den Oberflächenzellen wird nun die Transportfläche immer als die Fläche gewählt, die zu der Zelle gehört, von der die Geschwindigkeitsrichtung ausgeht.

Dabei kann " $\frac{\text{Fläche}}{\text{Volumen}}$ " nicht mehr zu $\Delta x, \Delta z, \dots$ gekürzt werden und die Gleichungsstruktur 3.9 bleibt erhalten. Die Interpolation der Dichte auf den Zellrand gestaltet sich in diesem Fall etwas anders. Bei der Berechnung der Dichte am unterem Rand der Zelle muss berücksichtigt werden, dass jetzt eines der Volumina größer oder kleiner als bisher sein kann.

Der Ansatz für eine Interpolationsformel ist damit beispielsweise für die obere Zellschicht:

$$\rho_{Zelle1} = \frac{1}{\Delta z} \int_0^{\Delta z} (\alpha z^2 + \beta z + \gamma) dz \quad (4.1)$$

$$\rho_{Zelle2} = \frac{1}{\Delta z} \int_{\Delta z}^{2\Delta z} (\alpha z^2 + \beta z + \gamma) dz \quad (4.2)$$

$$\rho_{Zelle3} = \frac{1}{h} \int_{2\Delta z}^{2\Delta z+h} (\alpha z^2 + \beta z + \gamma) dz. \quad (4.3)$$

Wobei h die Zellhöhe der Oberflächenzelle meint.

Hinsichtlich der Interpolation der Dichte für den Divergenzterm in x-Richtung sieht es etwas komplizierter aus. Man kann nun kein eindimensionales Polynom mehr für die Dichte ansetzen, sondern muss eine zweite Dimension in das Polynom einbringen. Dies erfolgt in dieser Arbeit linear, d.h. dass bei der Interpolation eine lineare Abhängigkeit in z-Richtung angenommen wird. Dabei wird zu den drei Zellen, die für die Interpolation bei Innenzellen angenommen wurden, noch eine hinzugenommen. Diese befindet sich unter der Zelle deren Randdichtewert interpoliert werden soll (siehe Abbildung 4.2).

Der Ansatz für die Interpolation stellt sich dann hier als:

$$\rho_1 = \frac{1}{\Delta x h_1} \int_0^{h_1} \int_0^{\Delta x} (\alpha + \beta x + \gamma x^2 + \delta z) dx dz$$

$$\rho_2 = \frac{1}{\Delta x h_2} \int_0^{h_2} \int_{dx}^{2\Delta x} (\alpha + \beta x + \gamma x^2 + \delta z) dx dz$$

$$\rho_3 = \frac{1}{\Delta x h_3} \int_0^{h_3} \int_{2dx}^{3\Delta x} (\alpha + \beta x + \gamma x^2 + \delta z) dx dz$$

für ρ_4 entweder:

$$\rho_4 = \frac{1}{\Delta x \Delta z} \int_{-\Delta z}^0 \int_0^{\Delta x} (\alpha + \beta x + \gamma x^2 + \delta z) dx dz$$

4 Modellierung der freien Oberfläche

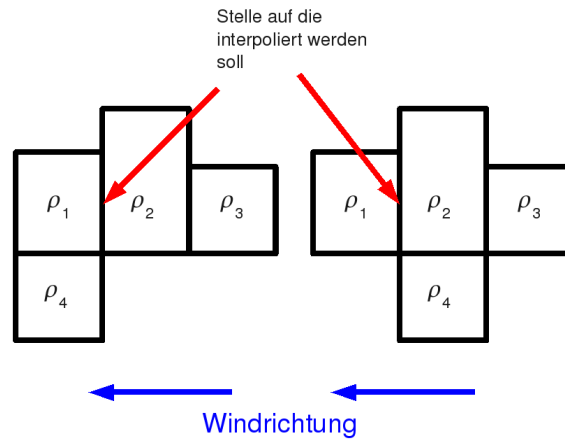


Abbildung 4.2: zur Interpolation an der Oberfläche am Beispiel der Dichte (links interpolation für den rechten Rand von Zelle 1, rechts interpolation für den linken Rand von Zelle 2)

oder

$$\rho_4 = \frac{1}{\Delta x \Delta z} \int_{-\Delta z}^0 \int_{\Delta x}^{2\Delta x} (\alpha + \beta x + \gamma x^2 + \delta z) dx dz$$

dar.

Wobei h_i die Zellhöhe der i -ten Oberflächenzelle meint.

Ermittelt man die Koeffizienten des Interpolationspolynoms kann man diesen an der gewünschten Stelle auswerten.

Nachdem eine Formel zur interpolation der Dichte auf den Zellrand gegeben ist kann der Fluss wie im Abschnitt 3.4 bestimmt werden.

4.3 Quellterme

Bei einer offenen Oberfläche gibt es es eine zusätzliche Geschwindigkeit am oberem Rand der Oberflächenzelle. Im geschlossenem System war diese entsprechend den Randbedingungen gleich Null gesetzt worden. Diese Geschwindigkeit bzw. die ihr zugehörige Impulsdichte wird analog zu den Impulsdichten im Innenraum zeitintegriert. Der über der Zelle befindliche Druck wird dabei als der stets konstante Luftdruck behandelt (siehe Abbildung 4.3).

Die Differentialgleichung der Oberflächenimpulsdichte lautet damit:

$$\frac{\partial \rho w}{\partial t} = -\frac{dp}{dz} + \rho g - \text{div}(\vec{v} \rho w),$$

wie für die Innenzellen auch. Diese zusätzliche Geschwindigkeit ist nun dafür verantwortlich, dass es noch eine weitere Differentialgleichung für die Höhe gibt, die ebenfalls

4 Modellierung der freien Oberfläche

berechnet wird und die die Oberflächendehformierung beschreibt. Bei der Höhe handelt es sich um eine einfache Länge, die sich mit der Geschwindigkeit ändert mit der sich das Wasser nach oben bewegt. Für solch einen Fall lautet die einfache Differentialgleichung:

$$dh = w \cdot dt.$$

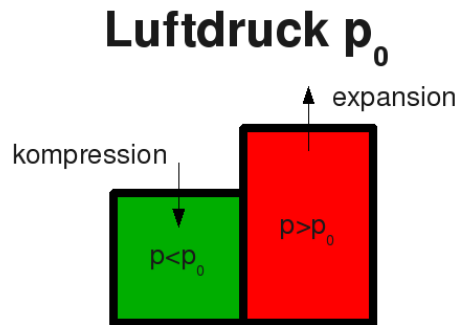


Abbildung 4.3: der Luftdruck deformiert die Oberfläche

Im Fall des sich ausdehnenden Wassers muss die Kontinuitätsgleichung für die Dichte einen Quellterm $\frac{d\rho}{dt} \neq 0$ bekommen, damit sich die Masse in der Zelle trotz vergrößertem oder verkleinertem Volumen nicht ändert. Die Volumenausdehnung einer Zelle in z-Richtung ist leicht beschreibbar, sobald man die Geschwindigkeit w am oberem Rand der Zelle kennt.

$$dV = Fdz; dz = wdt$$

(dabei ist F die Fläche die bewegt wird)

Aus der Definition der Dichte $\rho = \frac{m}{V}$ folgt damit:

$$\frac{d\rho}{dt} = -\frac{m}{V^2} \frac{dV}{dt} = -\frac{\rho}{V} Fw.$$

Damit folgen zusätzliche Quellterme für Impulsdichte und $\rho\theta$:

$$\frac{d\rho}{dt} \vec{v} = -\frac{\rho}{V} Fw\vec{v},$$

$$\frac{d\rho}{dt} \theta = -\frac{\rho}{V} Fw\theta.$$

4 Modellierung der freien Oberfläche

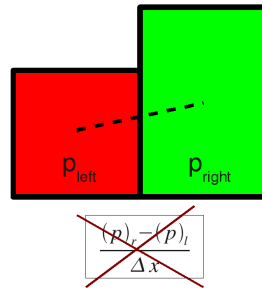


Abbildung 4.4: zur Vermeidung eines schiefen Druckgradienten

Dies sind die Quellterme, die in der Oberflächenmodellierung hinzukommen. Die anderen Quellterme Kraft und Wärmeflüsse werden analog zu den Innenzellen, unter Berücksichtigung des veränderten Volumens, berechnet .

Eine weitere Anmerkung sei hinsichtlich der Berechnung des Druckgradienten gegeben. Sobald sich die Zellen deformieren, wird es etwas schwieriger eine Ableitung allein in x-Richtung zu erhalten (siehe 4.4). Dies wurde hier gelöst, indem eine dritte Zelle hinzugenommen und eine Ebengleichung:

$$f(x, z) = a + bx + cz$$

aufgestellt wurde, die durch diese drei Punkte verläuft. Damit ist der Parameter b der Anstieg in x-Richtung.

5 Rechnen auf Grafikprozessoren

5.1 Warum Grafikprozessoren in der Computersimulation praktisch sind

Es gibt viele Beispiele der physikalischen Computersimulation, bei denen das Rechnen auf der Grafikkarte praktisch sein kann. Das ist immer dann der Fall, wenn viele gleiche oder ähnliche Objekte miteinander verrechnet werden. Dabei darf das Ergebnis für ein Objekt nicht von dem Ergebnis eines anderen Objektes abhängen.

Ein Beispiel dafür ist die Gittersimulation dieser Arbeit. Hier wird für jeden Gitterpunkt ein neuer Wert berechnet, indem man den kompletten Satz der alten Werte zu Hilfe nimmt, z.B. ist für jeden Gitterpunkt eine Dichte und eine Randgeschwindigkeit gespeichert. Aus diesen Dichte- und Geschwindigkeitsdaten wird numerisch der Anstieg von $\rho\vec{v}$ an dieser Stelle berechnet, welcher dann verwendet wird, um (entsprechend der Kontinuitätsgleichung) einen Zeitintegrationsschritt für diese Gitterzelle zu vollziehen.

Ein weiteres Beispiel wäre die Molekulardynamik. Hier soll die Kraft auf jedes einzelne Molekül berechnet werden, die aus den Anziehungen oder Abstoßungen der anderen Moleküle resultiert. Es muss also jedes Molekül mit jedem verrechnet werden.

Bei diesen beiden Beispielen wird deutlich, dass wenn ein Prozessor nacheinander die Berechnungen für jedes Molekül oder jede Gitterzelle einzeln vornimmt, er recht lange brauchen wird, wenn es viele Moleküle oder viele Gitterzellen gibt. Um den Zeitaufwand zu minimieren, bietet es sich an mehrere Prozessoren arbeiten zu lassen, die sich in die Molekül- oder Gitterzellenanzahl "reinteilen". Dies wird bereits seit längerem auch mit Computern getan, die mehrere Rechenkerne besitzen.

An dieser Stelle ist es allerdings praktischer eine Grafikkarte zu benutzen, da diese für solche parallelen Rechnungen ausgelegt ist. Dies wird in den folgenden Abschnitten beschrieben.

Es soll jedoch bereits hier Einschränkung des parallelen Rechnens benannt werden. Wie oben erwähnt darf dafür ein Ergebnis nicht von einem anderen abhängen. Die Bedeutung dessen wird vor allem bei der Molekulardynamik deutlich. Hier gilt "Aktio=Reaktio". Das bedeutet, dass das Molekül 1 auf Molekül 2 die gleiche Kraft ausübt, wie das 2. auf das 1.. Demzufolge müsste diese nur einmal berechnet werden und dann in das Kraftergebnis beider Moleküle hineinaddiert werden. Das funktioniert beim parallelen Rechnen jedoch nicht! Aus einem einfachen Grund:

Wenn sich ein Prozessor das Molekül 2 vornimmt und wissen möchte, wie groß die berechnete Kraft von Molekül 2 auf Molekül 1 war, muss er davon ausgehen, dass diese auch bereits fertig berechnet ist. Da die beiden jedoch parallel rechnen sollen, ist das nicht möglich und der Prozessor der Molekül 2 berechnet muss die Kraft von

Molekül 1 auf Molekül 2 neu berechnen. Das ist nicht der Fall, wenn ein Prozessor nacheinander alle Moleküle durchgeht.

Dieses Kriterium des parallelen Rechnens muss bei der Programmierung immer im Hinterkopf behalten werden.

In der Strömungssimulation gibt es ebenfalls einen solchen Effekt. Hier müssen die Flüsse an den Zellrändern berechnet werden, jeweils links und rechts. Jedoch entspricht der rechte Fluss einer Zelle dem linken Fluss derjenigen Zelle die sich rechts daneben befindet. Dies muss auf der Grafikkarte ebenfalls doppelt berechnet werden, obwohl es die selben Werte sind.

5.2 Paralleles Rechnen auf der Grafikkarte

Interessant und hilfreich zum Verständnis des parallelen Rechnens auf der Grafikkarte ist sicherlich die Frage warum eine Grafikkarte dafür so gut geeignet ist.

Grafikkarten werden dafür benutzt aufwendige virtuelle Umgebungen zu berechnen. Stellen wir uns beispielsweise eine Kiste vor:



Abbildung 5.1: Kiste in virtueller Umgebung

Diese besteht im wesentlichen nur aus 6 Flächen auf die eine Textur (ein Bild) gelegt wird, die wie Holz aussieht. Um diese Texturen soll es jetzt im Wesentlichen gehen. Möchten wir eine solche Kiste z.B. anders beleuchten oder einen Schattenwurf auf die Kiste berechnen, muss jedes Pixel der Textur neu berechnet werden. Und genau dafür ist der sogenannte Fragmentshader (manchmal auch Pixelshader) der Grafikkarte zuständig. Er kann auf eine grafikartenabhängige Anzahl von Texturen zugreifen und aus ihnen eine neue Textur berechnen. Damit ist es z.B. auch möglich, Spiegelungen im Wasser zu berechnen. Auf der Grafikkarte gibt es mehrere (bei großen Grafikkarten mehrere Hundert) Shadereinheiten, die parallel die neuen Pixelwerte errechnen. Die Texturgrößen sind dabei begrenzt, jedoch liegt die Grenze recht hoch und 8192×8192 ist eine Pixelanzahl die viele Grafikkarten verarbeiten können.

Die weiteren Schritte sind nun recht einfach. Es ist möglich Texturen zu verrechnen, ohne sie auf dem Bildschirm anzeigen zu lassen. In einen Pixel passen typischerweise 4 32-Bit Floatzahlen hinein (für Computerspiele sind das der Rot-, Grün-, Blauwert und der Alphakanal(die Transparenz)). Man kann nun diese Texturen nutzen und damit z.B. für Moleküle den Ort und die Geschwindigkeit speichern und verrechnen, oder bei den Gittersimulation die Dichte, den Impuls und andere Größen. Die Grafikkarte kann nun aus einer Textur in der z.B. der Ort steht und einer in der die Geschwindigkeit steht eine Textur mit den neuen Orten nach einem Zeitschritt berechnen.

Mehrere hundert Shadereinheiten können auf diesem Weg sehr schnell eine große Anzahl von Molekülen oder viele Gitterzellenwerte berechnen. Im Gegensatz zum parallelen Rechnen auf mehreren Computerkernen wird dabei nicht auf eine mit unter recht aufwändige weise das Modellgebiet in mehrere kleinere Objektanzahlen zerteilt, sondern jedes Objekt wird einzeln berechnet und die Zuteilung der Pixel auf die Shadereinheiten nimmt die Grafikkarte selbst vor. Man muss sich also weniger Gedanken über die Parallelisierung machen, als beim Rechnen mit mehreren Computerkernen.

5.3 Beispiel: Matrixaddition auf der Grafikkarte

Ein einfaches Beispiel für das Rechnen auf der Grafikkarte ist eine Matrixaddition. Damit klar wird, wie dies programmiert werden kann wird, hier ein möglichst einfaches Codebeispiel gegeben. Dabei werden zwei Klassen verwendet eine Shader und eine Texturklasse. Diese sind aus dem Modell von Stefan Horn entnommen, Information dazu findet man in [7]. In unterem Beispiel sollen 4 Matrizen zusammenaddiert werden.

Um die Verhältnisse der Rechenzeiten zu erhalten wurde zusätzlich ein Programm geschrieben das 4000 mal drei 500x500 Matrizen zu einer anderen 500x500 Matrix addiert. Diese Rechnung wurde 20 mal durchgeführt. Der Rechner mit dem der Test ausgeführt wurde hat einen CPU mit der Bezeichnung: Intel(R) Core(TM)2 Duo CPU T7300 @ 2.00GHz und eine nVidia Grafikkarte vom Typ: GeForce 8600M GT. Im Mittel brauchte der CPU in diesem Test 22800 Millisekunden während die Grafikkarte die gleiche Rechnung im Schnitt in 2800 Millisekunden bewerkstelligte. Damit war die Grafikkarte ca. **8 mal schneller**. Interessant war auch, dass wenn man die Matrixengröße auf 10x10 reduzierte der CPU im Schnitt nur 5.9 Millisekunden benötigte, während die Grafikkarte mit 66.4 Millisekunden langsamer war. Das zeigt, dass das Rechnen auf der Grafikkarte nur bei großen Datensätzen, für die gleiche Rechenoperationen ausgeführt werden sollen, Sinn macht.

5 Rechnen auf Grafikprozessoren

Zuerst müssen einige OpenGL Befehle abgehandelt werden um mit Texturen arbeiten zu können.

```
int nx=100;
int nz=100;

SDL_Init(SDL_INIT_VIDEO); // init SDL video
SDL_SetVideoMode(nx, nz, 24, SDL_OPENGL); // open SDL window
glwInit(); // init GL extension wrapper
glEnable(GL_TEXTURE_2D);
```

Abbildung 5.2: OpenGL Befehle

Danach werden Shader und Texturen definiert und initialisiert. Dabei bezeichnet 'add' eine Datei die im selben Verzeichnis liegt und in der der Shadercode steht. Der Shadercode wird bei der Initialisierung compiliert. In diesem steht bereits welche Operationen der Shader später auszuführen hat.

```
shader_1 *s_add; //define shader

s_add=new shader_1("header", "add"); //construct shader with code from a file

texture *ma1; //define Textures
texture *ma2;
texture *ma3;
texture *ma4;
texture *magpu;

ma1=new texture(nx,nz); //construct Textures with resolution nx * nz
ma2=new texture(nx,nz);
ma3=new texture(nx,nz);
ma4=new texture(nx,nz);
magpu=new texture(nx,nz);
```

Abbildung 5.3: Shader und Texturinitialisierung

```
uniform sampler2D t0;
uniform sampler2D t1;
uniform sampler2D t2;
uniform sampler2D t3;

void main ()
{
    vec2 pos=gl_TexCoord[0].xy;

    gl_FragData[0].r=texture2D(t0, pos).r+texture2D(t1, pos).r+texture2D(t2, pos).r+texture2D(t3, pos).r;
}
```

Abbildung 5.4: Shadercode

Nun erfolgen ein paar weiter OpenGL Befehle.

```
glMatrixMode(GL_PROJECTION); //additional OpenGL commands
glLoadIdentity();
gluOrtho2D(0,1,0,1);
glMatrixMode(GL_MODELVIEW);
glLoadIdentity();
glDisable(GL_BLEND);
```

Abbildung 5.5: OpenGL Befehle

5 Rechnen auf Grafikprozessoren

Nun füllen wir die Texturen mit Daten und laden diese in den Grafikkartenspeicher.

```
for(int i=0; i<nx; i++)
{
    for(int j=0; j<nz; j++)
    {
        ma1->data[4*(i+nx*j)+0]=1.;
        ma2->data[4*(i+nx*j)+0]=5.;
        ma3->data[4*(i+nx*j)+0]=9.;
        ma4->data[4*(i+nx*j)+0]=13.;
    }
}
ma1->ram2gpu();
ma2->ram2gpu();
ma3->ram2gpu();
ma4->ram2gpu();
```

Abbildung 5.6: Texturwerte setzen

Nun sind die Werte gesetzt und die Addition kann ausgeführt werden.

```
s_add->t[0]=ma1;
s_add->t[1]=ma2;
s_add->t[2]=ma3;
s_add->t[3]=ma4;
s_add->tout[0]=magpu;
s_add->bind();
s_add->step();
```

Abbildung 5.7: Shaderaufruf

Folgendermaßen kann man nun die Ergebnisse ausgeben.

```
magpu->gpu2ram();
for(int i=0; i<nx; i++)
{
    for(int j=0; j<nz; j++)
    {
        cout << magpu->data[4*(i+(nx*j))+0] << endl;
    }
}
```

Abbildung 5.8: Datenabfrage aus dem Grafikkartenspeicher

5.4 Nachteile

Ein Nachteil des parallelen Rechnens wurde bereits erwähnt. Es ist der Fakt, dass eine Recheneinheit nicht auf das Ergebnis einer Anderen zugreifen darf und dass dadurch viele Dinge mehrfach berechnet werden müssen. Dieser Nachteil wird jedoch in den meisten Fällen durch die Zeitersparnis des parallelen Rechnens wettgemacht.

Ein anderer Nachteil, der sich in der Programmierung der Wasserströmungssimulation gezeigt hat, ist der Fakt, dass bisher standartmäßig nur 32-Bit Floatdaten gespeichert werden können. Die Dichte des Wassers liegt um $1000 \frac{kg}{m^3}$, die für den Druck relevanten Dichteschwankungen jedoch liegen teilweise im $10^{-5} \frac{kg}{m^3}$ Bereich. Diese Schwankung kann durch die hohe Normaldichte des Wassers nicht mehr von den Floatzahlen gespeichert werden, da diese nur ca. 7 Stellen nach der führenden Kommastelle speichern kann (in der wissenschaftlichen Schreibweise). Deshalb war es an vielen Stellen nötig, Größen mit einem Profil zu versehen. Das heißt, dass nicht die Dichte selbst, sondern $\rho - \rho_0$ gespeichert wurde, um die relevanten Dichteschwankungen auflösen zu können. Auch an der Stelle im Programm, wo aus ρ und $\rho\theta$ der Druck berechnet wurde, gab es hier Schwierigkeiten die mit Hilfe von Taylorentwicklungen gelöst wurden, wodurch die Größen: $\rho - \rho_0$ bzw $(\rho\theta) - (\rho\theta)_0$ direkt verwendet werden konnten.

6 Funktionsweise und Aufbau des Programmes

6.1 Allgemeines

Das Grundgerüst der Computersimulation wurde in C++ geschrieben, die Unterprogramme, die auf der Grafikkarte ausgeführt werden, in der Programmiersprache GLSL (OpenGL Shading Language). Zum Einbinden der Shader in den C++ Code wurden die OpenGL und SDL Bibliotheken verwendet. Die Ausgabe der Daten als Bilder erfolgt mit Hilfe eines separaten, selbst geschriebenen Bildbetrachters.

Im wesentlichen besteht das Programm nur aus einer Hauptschleife in der die Shader in der richtigen Reihenfolge aufgerufen und in der nach der gewünschten Schrittanzahl Daten ausgegeben werden. Dies gestaltet sich durch die Einführung einer Shaderklasse und einer Texturklasse recht einfach, da hiermit die Aufrufe der Grafikprogramme und das Anbinden der Texturen sehr übersichtlich wird. In Abbildung 6.1 ist der Aufruf für den Advektionsshader als Beispiel gegeben. Die ersten drei Zeilen definieren die Eingabetexturen die 4. die Ausgabertextur.

```
//advektionsupdate für rho, theta und die zentralgeschwindigkeiten wird berechnet
s_advection->t[0]=zc;
s_advection->t[1]=zf;
s_advection->t[2]=h;
s_advection->tout=zck1;
s_advection->bind();
glUniform1fARB(glGetUniformLocationARB(s_advection->Handle,"dt"),dt);
glUniform1fARB(glGetUniformLocationARB(s_advection->Handle,"nx"),nx);
glUniform1fARB(glGetUniformLocationARB(s_advection->Handle,"nz"),nz);
glUniform1fARB(glGetUniformLocationARB(s_advection->Handle,"dx"),dx);
glUniform1fARB(glGetUniformLocationARB(s_advection->Handle,"dy"),dy);
glUniform1fARB(glGetUniformLocationARB(s_advection->Handle,"dz"),dz);
glUniform1fARB(glGetUniformLocationARB(s_advection->Handle,"rhoprof"),rhoprof);
glUniform1fARB(glGetUniformLocationARB(s_advection->Handle,"rhothetaprof"),rhothetaprof);
s_advection->step();
```

Abbildung 6.1: Advektionsshader Aufruf

6.2 Programmablaufplan

In Abbildung 6.2 ist der Programmablauf in einer Übersicht dargestellt und wird im Folgenden noch einmal kurz erläutert.

Initialisierungsbedingungen

In einem ersten Schritt werden die Anfangsbedingungen berechnet und gesetzt. Wie dies geschieht, ist in Abschnitt 3.6 beschrieben.

6 Funktionsweise und Aufbau des Programmes

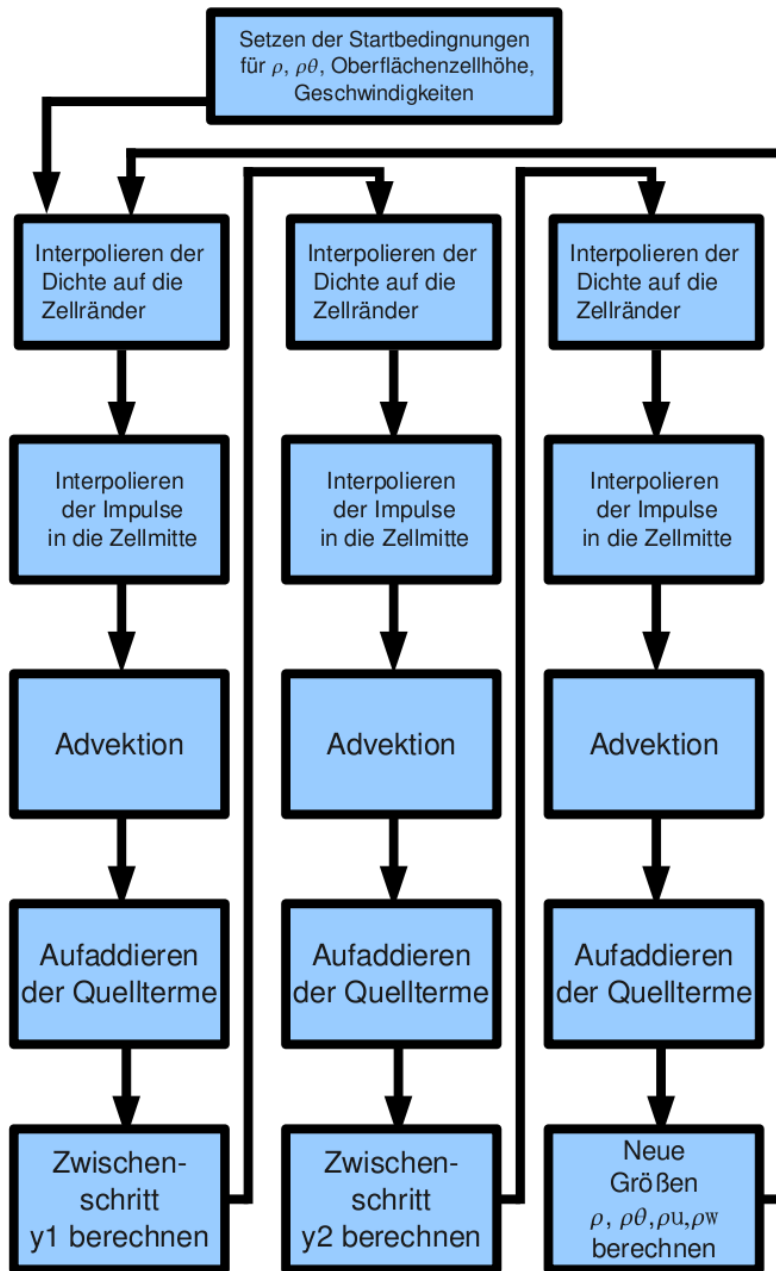


Abbildung 6.2: Programmablaufplan

Dichteinterpolation

Die Zelldichten auf der linken oder unteren Fläche einer Zelle werden berechnet, indem der Mittelwert der beiden an die Fläche angrenzenden Zellen gebildet wird.

Impulsinterpolation

Nachdem die Dichte auf den Flächen bekannt ist, kann mit den Geschwindigkeiten auf dem Zellrand der Impuls gebildet und in die Zelle als Mittelwert interpoliert werden.

Advektion

In der Advektionssubroutine wird der Divergenzterm berechnet. Dazu wird die Dichte mit Hilfe des Upwindverfahrens auf den Zellrand interpoliert und der Fluss durch die Flächen bestimmt. Das Ergebnis wird als erster Anteil der Right-Hand Side in eine Textur gespeichert.

Aufaddieren der Quellterme

Die Quellterme: Kraft, Wärmefluss durch das beheizen von Zellen und die Quellterme durch die Expansion der Oberflächenzellen werden auf die Right-Hand-Side-Textur aufaddiert. Für den Impuls wird hierbei der vorher berechnete Divergenzterm auf die Fläche gemittelt und die wirkende Kraft auf der Fläche hinzuaddiert.

Zwischenschritt y_1, y_2 , bzw Zeitschritt

Entsprechend dem Runge-Kutta-Verfahren werden hier aus den berechneten Right-Hand-Sides komplette Zwischenzustände berechnet und gespeichert. Diese enthalten wieder die gleichen Zustandsgrößen (nur zu einem neuem Zwischenzeitpunkt), die auch initialisiert wurden: $\rho, \rho\theta, \vec{v}, \text{Oberflächenzellhöhe}, \text{Oberflächenzellgeschwindigkeit}$. Mit diesen Zwischenzuständen kann jeweils ein neuer Zwischenzustand berechnet werden. Am Ende, wenn alle Right-Hand Sides bekannt sind, wird ein Zeitschritt ausgeführt.

Wie man in Abbildung 6.2 sieht, werden damit für einen Zeitschritt drei mal die gleichen Righthandsideberechnungen für unterschiedliche Zwischenzustände ausgeführt.

7 Auswertung von Simulationen

7.1 Qualitativer Vergleich mit der Simulation von Bryan-Fritsch [5]

Der erste Test, den die Simulation unterlief, sollte zunächst zeigen, dass die Advektionsroutinen richtig arbeiten. Dieser Test wurde noch mit der Zustandsgleichung für Luft durchgeführt. Dazu wurde ein Druckprofil initialisiert, das einer stabil geschichteten Atmosphäre entspricht. Danach wurde als Anfangswert ein kreisförmiges Gebiet mit einer potentiellen Temperatur nach der Formel:

$$\theta = 2 \cdot \cos\left(\frac{\pi L}{2}\right)^2$$

(L=Abstand zum Mittelpunkt des kreisförmigen Gebietes) gesetzt. Der Rest des Gebietes erhielt die Normaltemperatur $\theta = 293K$. Aus dieser gesetzten potentiellen Temperatur und dem Druck wurde dann ein Dichteprofil berechnet. Das Anfangsprofil ist in Abbildung 7.1 zu sehen.

Die oft als "Theta Bubbel" bezeichnete Warmluftblase verhielt sich qualitativ richtig. Dies ist an den Ergebnissen 7.2 und 7.3 zu sehen.

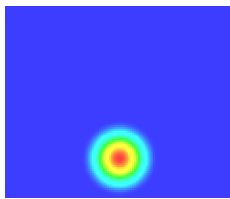


Abbildung 7.1: Initialisiertes Thetaprofil

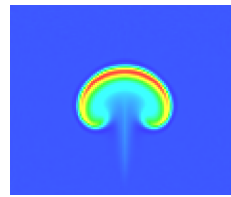


Abbildung 7.2: Nach 700 Sekunden

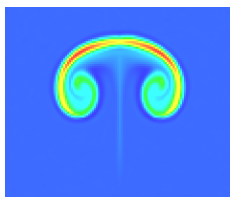


Abbildung 7.3: Nach 1000 Sekunden

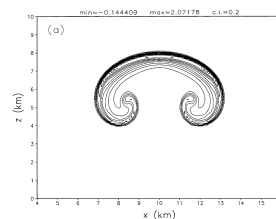


Abbildung 7.4: zum Vergleich mit dem Paper von Bryan und Fritsch [5]

7.2 Die Wasserparabel

Wird Wasser in Rotation versetzt, stellt sich eine parabelförmige Oberfläche ein. Diese Form kann analytisch berechnet werden, wenn man beachtet, dass sich die Wasseroberfläche im Gleichgewicht so einstellt, dass die Kraft die Normale zur Fläche ist. Damit erhält man eine analytische Lösung, für die Form der Wasseroberfläche:

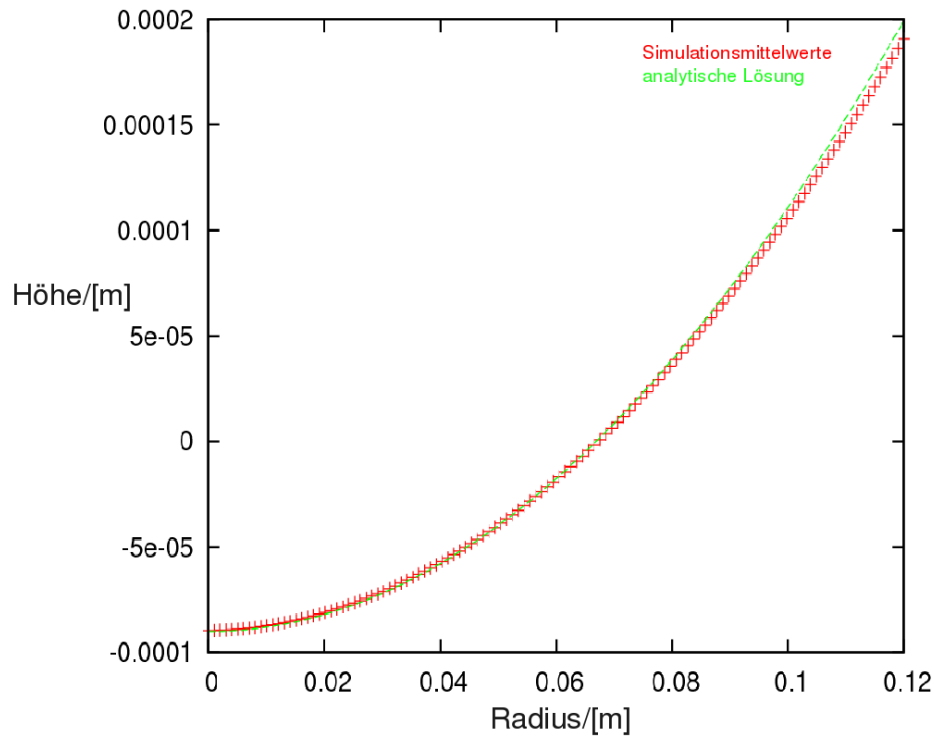


Abbildung 7.5: Wasseroberflächenparabel (Rotationsgeschwindigkeit $\omega = 2\pi \cdot 0.1 \frac{1}{s}$)

$$h(r) = \frac{\omega^2}{2g} r^2 + const.$$

Dieses einfache Experiment wurde nun in einer Simulation durchgeführt, um die Funktionalität der Algorithmen, welche die Oberflächendeformation beschreiben zu testen. In der Simulation laufen dabei jedoch horizontale Wellen durch die Oberfläche, die durch die Trägheit des Wassers und das plötzliche Anschalten der Rotation zustande kommen. Deswegen wurde eine zeitliche Mittelung durchgeführt, um die Wellen aus dem Oberflächenprofil herauszurechnen. In Abbildung 7.5 sieht man die analytische Lösung im Vergleich zum Mittelwert des Oberflächenprofils.

7.3 Rotierender Zylinderspaltschnitt inklusive Wärmefluss an den Randflächen

In diesem Simulationslauf sollte getestet werden welche zweidimensionalen Strömungen beim rotierenden Zylinderspaltschnitt entstehen. In Abbildung 7.6 ist zu sehen, welche zweidimensionale Fläche des rotierenden Zylinders hier gemeint ist.

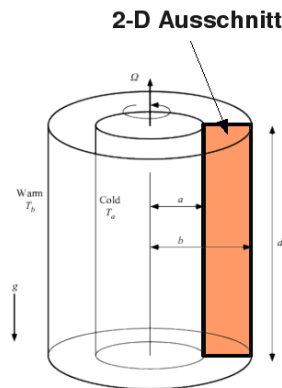


Abbildung 7.6: zweidimensionaler Zylinderspaltschnitt [1]

Die Rotationsgeschwindigkeit wurde im Simulationslauf auf $\omega = 2\pi \cdot 0.1 \frac{1}{s}$ gesetzt. Der Schnitt hat eine Breite von 12 cm und eine Höhe von 10 cm. Die rechte Wand besitzt eine konstante Temperatur von $T_r = 295K$, die linke von $T_l = 291K$. Da die Anfangstemperatur des Wassers $T_{anfang} = 293K$ beträgt, erwärmt sich das Wasser an der rechten Wand und kühlt sich links ab.

Durch diese Anfangsbedingungen kommen zwei Prozesse in Gang. Zum einen drückt die Zentrifugalkraft das Wasser nach außen und verursacht anfängliche Wellen. Damit ein stabiles Stömungsmuster entsteht in dem es keine weitere Impulsänderung in den Zellen gibt muss diese Kraft, zusammen mit der Gravitation, durch ein passendes Druck- und Strömungsprofil ausgeglichen werden.

Zum anderen bewirkt der Temperaturunterschied zwischen den Wänden das eine Konvektionszelle ausgebildet wird. Das erwärmte Wasser an der rechten Wand steigt auf, das Kalte an der linken Wand sinkt herunter (siehe Abbildung 7.7).

Die Oberflächenwellen beruhigen sich erst nach ca. 23 s. Zu dieser Zeit bilden sich Geschwindigkeitswirbel unter der Oberfläche (siehe Abbildung 7.8), die jedoch nach 45 s Simulationszeit wieder verschwinden.

Nach einer Simulationszeit von ca. 60 s entsteht ein stabiler Zustand. In diesem gibt es eine Warmwasserkonzentration an der Oberfläche und der rechten Wand und eine Kaltwasserkonzentration am Boden und der linken Wand. Auffällig hierbei ist der Fakt, dass es an der Oberfläche einen kleinen Bereich gibt, in dem sich das warme Wasser

7 Auswertung von Simulationen

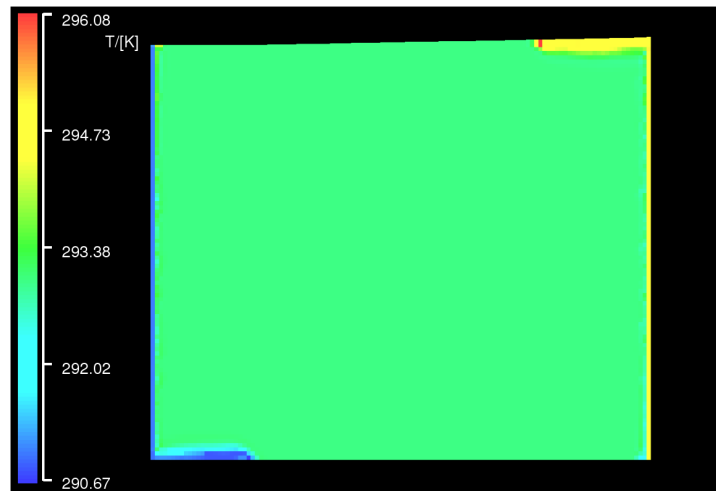


Abbildung 7.7: Temperatur nach 0,6 s. Das warme Wasser sammelt sich oben das kalte unten.

sammelt und in dem sogar die Temperatur erhöht gegenüber der Wandtemperatur ist (siehe Abbildung 7.9). Zur Veranschaulichung wurde in Abbildung 7.10 die Temperatur der Oberflächenzellen in Abhängigkeit des Radiuses geplottet. Vergleicht man die Lage des Warmbereiches an der Oberfläche mit dem Betrag der Geschwindigkeit in diesem Bereich erkennt man, dass dieser rasch abnimmt (siehe Abbildung 7.11). Dies ist ein Indiz wie es zu einem "Wärmestau" kommen kann.

Dieser Warmwasserbereich könnte ein Hinweis darauf sein, dass mit Hilfe dieser Simulation (nach dem sie auf drei Dimensionen erweitert wurde) die scharfen Oberflächentemperaturunterschiede des Zylinderspaltexperimentes erhalten werden könnten.

Zum Vergleich wurde ebenfalls eine Simulation durchgeführt, in der die Oberfläche nicht offen ist. Hier hat sich dementsprechend an der Oberfläche ein unterschiedliches Ergebnis gezeigt (siehe Abbildung 7.12).

Nach intensiver Beschäftigung mit den Simulationsdaten ist jedoch ein systematischer Fehler aufgefallen, dessen Auswirkung auf den Endzustand schwer abzuschätzen ist. Im Abschnitt 8.1 wird erläutert, warum es in dieser Simulation numerische Fehler in der Geschwindigkeitsberechnung gab, die prinzipiell mit weiterer Programmierarbeit vermeidbar oder zumindest stark reduzierbar wären. In Abbildung 7.13 ist zu sehen, dass dadurch die betragskleinen Geschwindigkeiten scharfe Unstetigkeitskanten aufweisen, die aus der Verrechnung mit der Dichte resultieren. Obwohl sich diese Fehler vor allem in den betragskleinen Geschwindigkeiten bemerkbar machen, könnten sie das Ergebnis stark verfälscht haben.

7 Auswertung von Simulationen

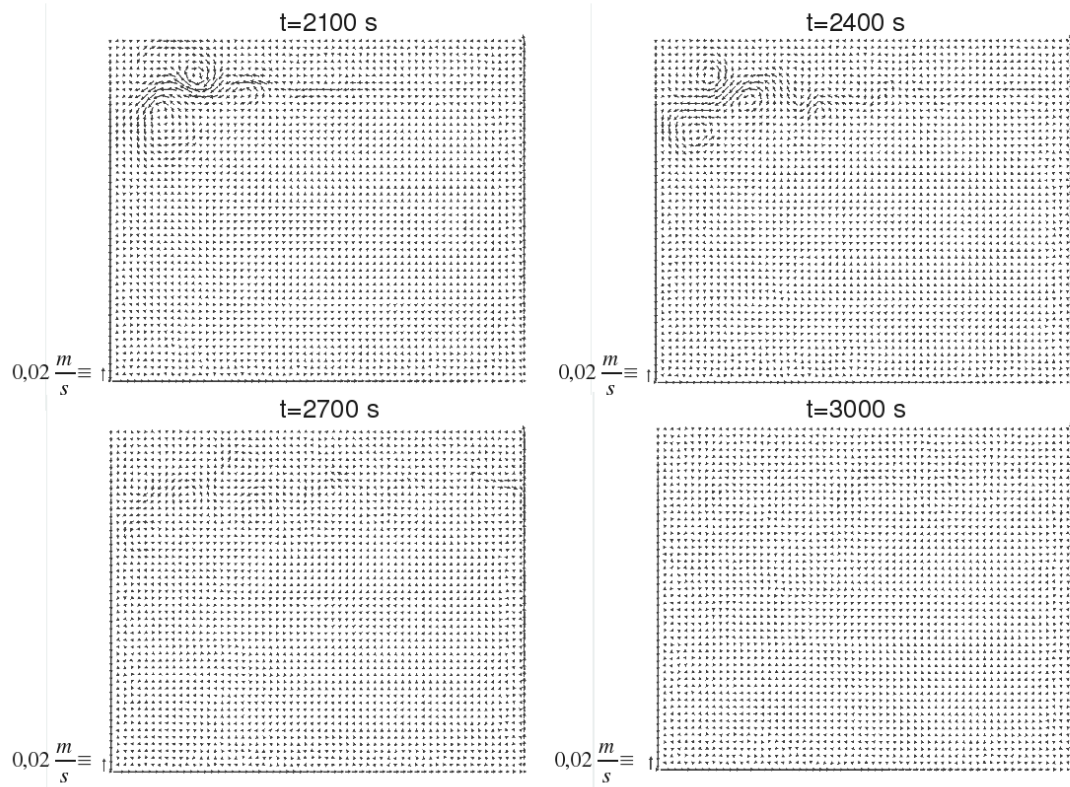


Abbildung 7.8: Das Geschwindigkeitsfeld

7 Auswertung von Simulationen

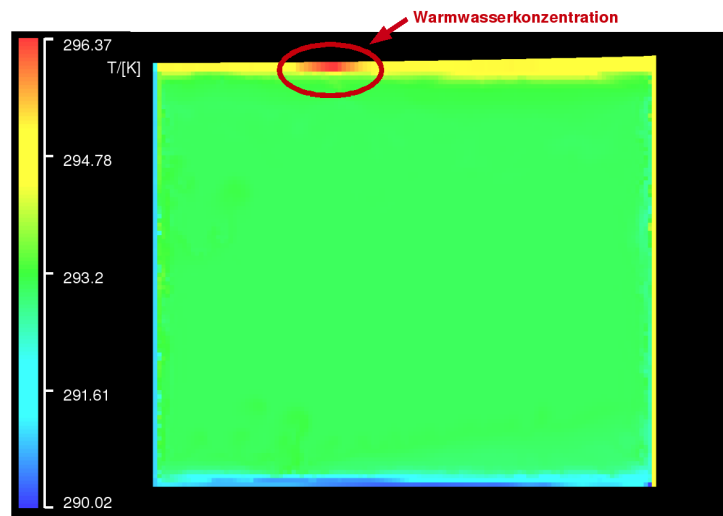


Abbildung 7.9: Temperatur nach 60 s

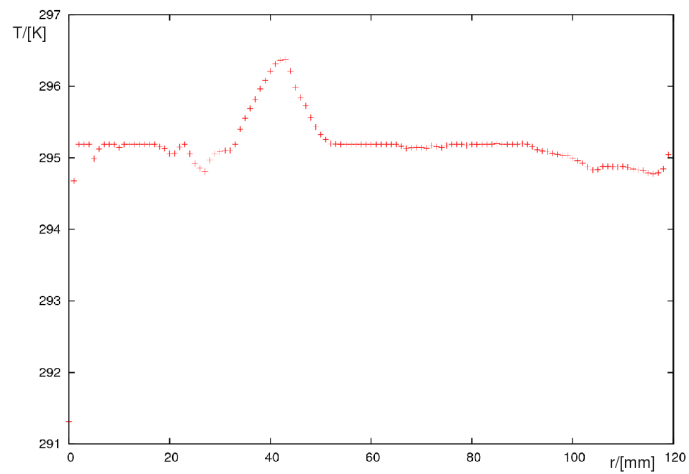


Abbildung 7.10: Temperatur in den Oberflächenzellen nach 90 s

7 Auswertung von Simulationen

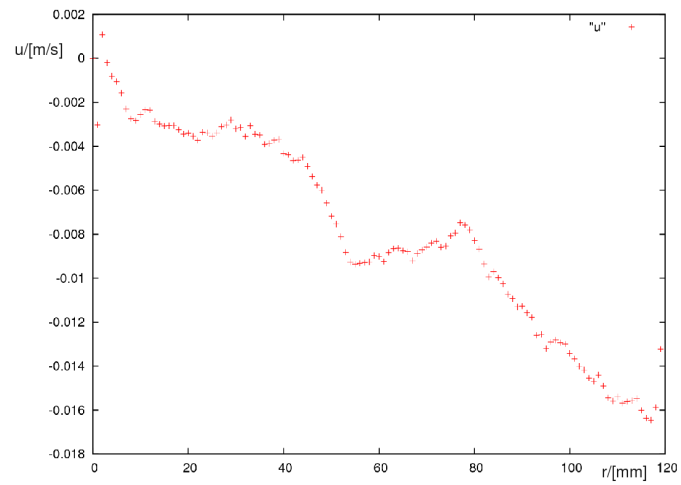


Abbildung 7.11: Geschwindigkeitskomponente in x-Richtung in den Oberflächenzellen nach 90 s

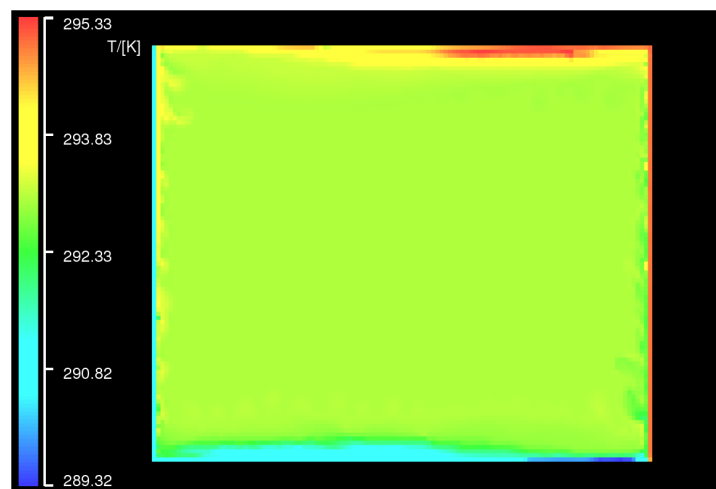


Abbildung 7.12: Temperatur im Simulationslauf ohne offene Oberfläche nach 70 s

7 Auswertung von Simulationen

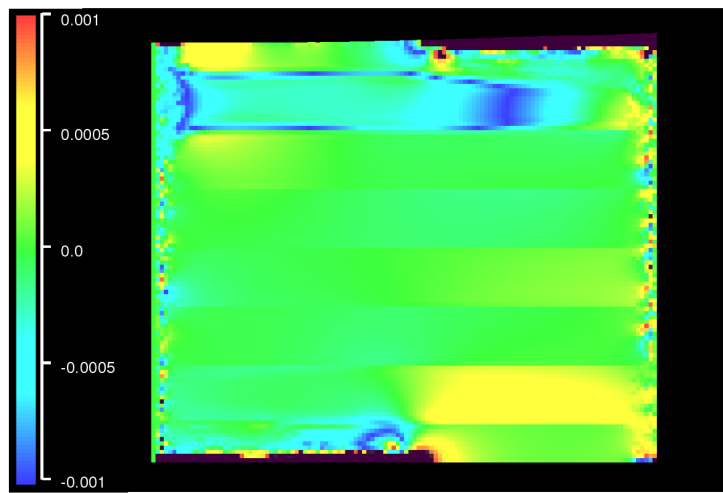


Abbildung 7.13: systematische Fehler in der Geschwindigkeitsberechnung der u Komponente (der schwarze Bereich ist aushalb der Skala)

8 Kritik und Ausblick

8.1 Kritik

Die Priorität dieser Arbeit lag bei der Implementierung des Wassers, sowie der Modellierung einer freien Oberfläche für Wasser. Ob dies qualitativ gelingt oder nicht kann man bereits unter Vernachlässigung einiger Quellterme erkennen. Diese Vernachlässigung wurde nicht beseitigt, da die Implementierung den Rahmen einer Bachelorarbeit zu sehr überdehnt hätte. Diese Quellterme sind zunächst einmal der Spannungstensor in der Differentialgleichung des Impulses:

$$\sum_{j=1}^3 \partial_{x_j} S_{ij} \tag{8.1}$$
$$S_{ij} = \mu \left[(\partial_{x_j} v_i + \partial_{x_i} v_j) - \frac{2}{3} \delta_{ij} \sum_{k=1}^3 \partial_{x_k} v_k \right]$$

Dabei ist 8.1 der Term, der auf die Differentialgleichung für die i-te Impulskomponente zusätzlich aufaddiert werden müsste.

Zusätzlich könnte man die Wärmeleitung zwischen den einzelnen Zellen implementieren:

$$\dot{q} = \frac{\vec{F}}{m} \lambda \text{grad}(T) \quad (\vec{F} = \text{Flächennormalenvektor}),$$

da sonst die gesamte Strömung als nahezu isentrop berechnet wird. Möchte man diese Simulation mit konkreten Messwerten vergleichen, könnte es wichtig werden diese beiden Quellterme zu berücksichtigen.

Ebenfalls wichtig ist eine Anmerkung zu der in dieser Arbeit verwendeten Zustandsgleichung. Sie beschreibt Wasser in dem für diese Arbeit benötigtem Bereich sehr gut. Sie dürfte jedoch versagen, sobald man in einen Bereich kommt, in dem der Wärmeausdehnungskoeffizient oder die Kompressibilität nicht mehr als konstant angesehen werden können. Demzufolge ist diese Simulation auf einen bestimmten Zustandsbereich beschränkt.

Zusätzlich sollte allgemein angemerkt werden, dass in dieser Art von Simulationen Gleichgewichtsformeln der Thermodynamik benutzt werden, um die Zwischenzustände schnell ablaufender Prozesse zu berechnen. Dies ist eine zusätzliche Fehlerquelle der Berechnungen, da dies die thermodynamischen Gleichungen in der Regel nicht exakt liefern.

8 Kritik und Ausblick

Des Weiteren wäre es in programmiertechnischer Hinsicht besser nicht die Geschwindigkeiten, sondern die Impulse in den Texturen zu speichern. Dadurch könnte man vermeiden, dass sich Fehler in den Floatingpointoperationen aufaddieren, wie man dies in Abbildung 8.1 sehen kann. Diese resultieren daraus, dass im Programm die Geschwindigkeiten mit Hilfe der Dichte aus den Impulsen ermittelt werden und später aus diesen Geschwindigkeiten wieder der Impuls berechnet wird. Dies ist an sich unnötig. Dieser Fehler sollte vor einer weiteren Benutzung des Programms ausgeglichen werden.

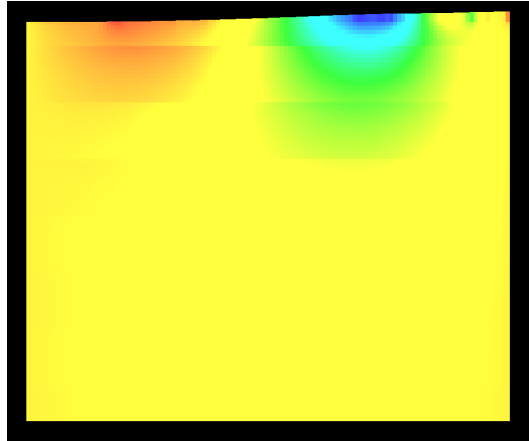


Abbildung 8.1: Fehler durch Floatingpointoperationen

8.2 Ausblick

Diese Arbeit soll eine Vorbereitung für die Simulation des rotierenden Zylinders sein, daher soll im Ausblick nun anklängen, was die nächsten Schritte sind, um den rotierenden Zylinder zu simulieren.

Zunächst einmal getalltet es sich einfach von zwei auf drei Dimensionen überzugehen, indem der Impuls in y -Richtung mitgerechnet wird und die Divergenzterme in y -Richtung ebenfalls einbezogen werden. Dafür wäre, insofern das bestehende Programm weiterentwickelt werden soll, eine Neugestaltung der Texturen nötig, um die zusätzlichen Werte speichern zu können.

Außerdem wäre es empfehlenswert auf Zylinderkoordinaten überzugehen, um die Verwendung von "teilweise gefüllten", sogenannten Cutcells, zu vermeiden. Diese Cutcells entstehen an den Rändern, wenn man versucht die Flüssigkeit, die sich in einem Zylinder befindet in einem kartesischem Gitter unterzubringen (siehe Abbildung 8.2).

Um in einem auf Zylinderkoordinaten basierendem Gitter rechnen zu können, müssten die Differentialgleichungen von x, y, z auf r, ϕ, z transformiert werden.

8 Kritik und Ausblick

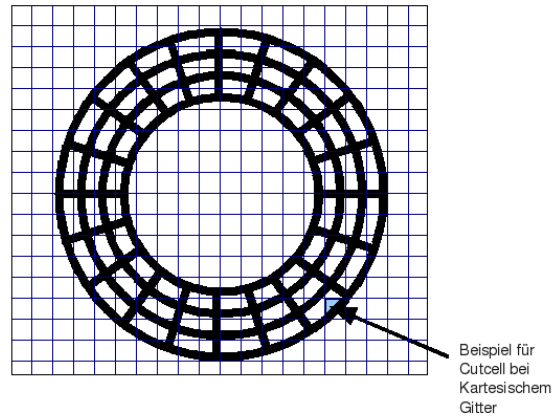


Abbildung 8.2: Vergleich von Kartesischem- mit Zylinderkoordinatengitter

Die zweidimensionalen Simulationen haben gezeigt, dass im Gegensatz zum geschlossenen System, eine offene Oberfläche bewirkt, dass hier der Impuls nicht oder auf andere Art reflektiert wird. Dadurch entstehen an der Oberfläche Strömungen, die bei geschlossener Simulation nicht entstehen würden und für die in der Einleitung erwähnten Moden verantwortlich sein könnten.

Literaturverzeichnis

- [1] ANTHONY RANDRIAMAMPINANINA, WOLF-GERRIT FRÜH, PETER L. READ and PIERRE MAUBERT (2006). Direct numerical simulations of bifurcations in an air-filled rotating baroclinic annulus. *Journal of Fluid Mechanics*, 561, pp 359-389 doi:10.1017/S0022112006000711
- [2] Harlander, U.; von Larcher, Th.; Wang, Y.; Egbers, Ch.; 2009 PIV- and LDV-measurements of baroclinic wave interactions in a thermally driven rotating annulus *Experiments in Fluids*, DOI: 10.1007/s00348-009-0792-5, Springer-Verlag, 2009
- [3] W. Wagner and A. Pruß. The IAPWS Formulation 1995 for the Thermodynamic Properties of Ordinary Water Substance for General and Scientific Use. *J. Phys. Chem. Ref. Data* 31, 387 (2002); doi:10.1063/1.1461829 (149 pages)
- [3] Von Heinz Schade, Ewald Kunz, Frank Kameier. *Strömungslehre*. Verlag: Gruyter 2007; 3., neu bearb. Aufl.
- [4] Ernst Hairer, Gerhard Wanner, Syvert Paul Norsett. *Solving Ordinary Differential Equation I*. Springer-Verlag Berlin Heidelberg 1987
- [5] Bryan, G. H., and Fritsch, J. M. (2002). A Benchmark Simulation for Moist Non-hydrostatic Numerical Models. *Monthly Weather Review*, 130(12), 2917-2928.
- [6] Dale Durran. *Numerical Methods for Fluid Dynamics*. Springer-Verlag GmbH, 11/2010
- [7] <http://asamwiki.tropos.de/index.php/ASAMgpu>
- [8] Wilford Zdunkowski, Andreas Bott (2003). *DYNAMICS OF THE ATMOSPHERE: A COURSE IN THEORETICAL METEOROLOGY*. Cambridge University Press 2003

Danksagung

Die erste Danksagung geht an Stefan Horn der im Institut für Troposphärenforschung, der mich direkt betreut hat. Er hat mir in etlichen Diskussionen und bei vielen Schwierigkeiten mit viel Geduld zur Seite gestanden. Durch ihn konnte ich einiges an Wissen und Verständnis erlangen, das zu Erstellung dieser Bachelorarbeit nötig war. Stefan hat mich motiviert auf der Grafikkarte zu rechnen und mir die nötigen Programmierkenntnisse beigebracht.

Außerdem danke ich Dr. Oswald Knoth und dem Institut für Troposphärenforschung dafür, meine Bachelorarbeit dort schreiben und ebenfalls Fragen stellen zu können. Hinzu kommt der Dank an Dr. Siegfried Fritzsche, der der Zweitbetreuer dieser Arbeit

Literaturverzeichnis

ist.

Ein weiterer Dank geht an meine Schwester Elisa Schierz die mir bei Formulierungen und Orthografie half.

Selbstständigkeitserklärung

Hiermit erkläre ich, die vorliegende wissenschaftliche Arbeit selbstständig und nur mit den angegebenen Hilfsmitteln angefertigt zu haben und dass alle Stellen, die dem Sinn nach anderer Literatur entnommen sind, durch Angabe der Quellen als Entlehnung kenntlich gemacht worden sind.

Leipzig den 08.07.2011