

FRIEDRICH-ALEXANDER-UNIVERSITÄT ERLANGEN-NÜRNBERG
TECHNISCHE FAKULTÄT • DEPARTMENT INFORMATIK

Lehrstuhl für Informatik 10 (Systemsimulation)



Implementierung der Blasentrennung in waLBerla

Markus Siko

Studienarbeit

Implementierung der Blasentrennung in waLBerla

Markus Siko

Studienarbeit

Aufgabensteller: Prof. Dr. U. Rüde

Betreuer: Dr.-Ing. S. Donath, Dipl.Inf. Dipl.Math. S. Bogner

Bearbeitungszeitraum: 24.1.2011 – 27.1.2012

Erklärung:

Ich versichere, dass ich die Arbeit ohne fremde Hilfe und ohne Benutzung anderer als der angegebenen Quellen angefertigt habe und dass die Arbeit in gleicher oder ähnlicher Form noch keiner anderen Prüfungsbehörde vorgelegen hat und von dieser als Teil einer Prüfungsleistung angenommen wurde. Alle Ausführungen, die wörtlich oder sinngemäß übernommen wurden, sind als solche gekennzeichnet.

Erlangen, den 27. Januar 2012

.....

Kurzfassung

Das waLBERla Projekt, das vom Lehrstuhl für Systemsimulation der Friedrich-Alexander Universität Erlangen-Nürnberg entwickelt wird, soll einen All-in-One Löser für verschiedenste Flüssigkeitsszenarien darstellen. Im Rahmen dieser Arbeit soll die freie Oberflächen Methode erweitert werden, um ein korrektes Blasen aufspalten zu gewährleisten. Hierbei wird ein Algorithmus zur Erkennung von Volumentrennung und Vermessung der entstandenen Teilvolumina entwickelt. Als Ansatzpunkt für die Entwicklung des Algorithmus wurden Methoden aus der Bildsegmentierung verwendet und auf das waLBERla Framework zugeschnitten. Es werden dabei entstandene Probleme und Lösungen aufgezeigt und das Verfahren in Simulationen getestet.

Inhaltsverzeichnis

1	Einleitung	1
2	Grundlagen der freien Oberflächen Methode in waLBerla	3
3	Algorithmusentwicklung	6
3.1	Kriterium zur Blasentrennungserkennung	6
3.1.1	Blasenkrümmung	6
3.1.2	Umwandlung von Interface-Zellen in Liquid-Zellen und Überprüfung gegenüberliegender Normalen	8
3.2	Segmentierungsalgorithmen	11
3.3	Implementierter Algorithmus zur Blasentrennungserkennung und Volumenvermessung	13
3.3.1	Bounding Box	13
3.3.2	Iteratives Flutfüllungsverfahren zur Blasentrennungserkennung und Volumenvermessung	13
3.3.3	Performanz	18
4	Test des Algorithmus mittels Simulationen	20
4.1	Längliche Blase in Einströmung	20
4.2	Blasenbildung an einer überfluteten Düse	22
5	Zusammenfassung und Ausblick	24

1 Einleitung

Das widely applicable Lattice Boltzmann solver from Erlangen (waLBerla) Projekt, das am Lehrstuhl für Systemsimulation an der Friedrich-Alexander Universität entwickelt wird, soll unter anderem ein universal Lattice Boltzmann Löser für Fluid Simulationen sein. Aufgrund des häufigen Vorkommens in Natur, technischen Anwendungen und alltäglichen Leben stellen Zweiphasenströme einen wichtigen Aspekt in waLBerla dar. Gekennzeichnet sind diese Ströme durch Interaktionen zweier nicht mischbarer Fluide, die durch eine Grenzfläche (engl. Interface) voneinander getrennt sind. Diese Grenzfläche wird auch als freie Oberfläche bezeichnet und ist Namensgeber für die Methode die solche Zweiphasenströme behandelt. Betrachtet man nicht ein Flüssigkeitsvolumen, sondern ein Gasvolumen bzw. eine Gasblase in einer Flüssigkeit, so befindet man sich im Gebiet der Blasenströmungen [3].

Blasenströmungen haben einen hohen Stellenwert in industriellen und technischen Verfahren. Beispielsweise sind hier die Blasensäulen zu nennen. In diesen industriellen Apparaturen wird in eine Flüssigkeit Gas eingeperlt, um einen Stoffaustausch oder Reaktionen zwischen den beiden Phasen über die Grenzfläche zu ermöglichen. Angewendet werden solche Blasensäulen in chemischen, biologischen und physikalischen Reaktionen: zum Beispiel bei der Flotation oder auch Fermentation [3]. Ein weiteres Beispiel für Blasenströmungen ist die Entstehung von Schäumen. Schäume bestehen aus Gasblasen, die von flüssigen oder festen Wänden eingeschlossen sind. Neben Einsatzgebieten wie Leichtbau, Montage, Herstellung von Schaumstoffen und im speziellen auch in der Lebensmittelindustrie in Produkten wie Kuchen, Espresso oder Mousse. Simulationen werden in diesen Bereichen vor allem durchgeführt, um ein besseres Verständnis für den Entstehungsprozess von Blasen zu bekommen, bestehende Prozesse oder Apparaturen zu optimieren oder Neuentwicklungen voranzutreiben.

Die Simulation von Blasenströmungen erfordert, dass alle Interaktionen zwischen den Gasblasen und der Flüssigkeit bewältigt werden können. In der freien Oberflächen Methode von waLBerla können Interaktion von Flüssigkeit mit den Gasblasen und das Vereinigen von Blasen simuliert werden. Die Vereinigung von Blasen ist auf triviale Weise erreichbar, indem immer dann, wenn sich das Interface verschiedener Blasen berührt, die Volumina der beiden Blasen addiert werden und die Blasen im weiteren Verlauf als eine Blase angesehen werden. Die Blasentrennung ist, aufgrund der Repräsentation der Gasblasen im verwendeten Modell, jedoch schwieriger zu erreichen. Eine genaue Erklärung dieser Problematik ist in Kapitel 2 zu finden. Diese Arbeit befasst sich mit der Erweiterung der freien Oberflächen Methode um eben jene Blasentrennung. Ein entsprechender Algorithmus wird entwickelt und dessen Implementierung vorgestellt. Das Verfahren muss dabei erkennen ob sich eine Blase getrennt hat und bei einer Trennung die entstandenen Volumina vermessen.

Im folgenden Kapitel werden zunächst die grundlegenden Strukturen der freien Oberflächen Methode genauer erläutert, um ein Fundament für die Entwicklung des Algorithmus zu schaffen. In Kapitel 3 werden die einzelnen Entwicklungsschritte des Algorithmus erläutert und

die Einbettung in das waLBerla-Framework beschrieben. Daraufhin folgt das Testen des Algorithmus anhand zweier Simulationen. Abschließend wird das Erarbeitete zusammengefasst und ein Ausblick auf weiterführende Arbeiten gegeben.

2 Grundlagen der freien Oberflächen Methode in waLBerla

Die Aufgabenstellung erfordert grundlegende Kenntnis über die in waLBerla verwendeten Strukturen und Zusammenhänge der freien Oberflächen Methode. Im speziellen wird auf das Modell eingegangen, das die Repräsentation der Gasblasen beschreibt, um die Problematik der Blasentrennung aufzuzeigen [5].

Das Modell der freien Oberflächen Methode lässt sich in zwei Teile aufteilen: das Modell zur Beschreibung der Fluidphase und das zur Beschreibung der Gasphase (Abbildung 2.1).

Das Modell der Gasphase beschreibt eine Gasblase mittels Initialvolumen und aktuellem Volumen. Das Initialvolumen gibt das Volumen der Blase zu Beginn der Simulation an und wird im weiteren Verlauf auch nicht mehr verändert. Im Gegensatz dazu steht das aktuelle Volumen, das sich innerhalb jedes Zeitschrittes ändern kann. Der Druck p der Blase berechnet sich aus dem Quotient von aktuellen und initialen Volumen. Die Blasen werden in einem geordneten Datenfeld gespeichert, wobei jede Blase eine eindeutige Identifikationsnummer erhält, die dem Platz der Blase im Datenfeld entspricht.

Das Modell der Fluidphase ist weitaus komplizierter. Der Phasenraum wird durch ein dreidimensionales Gitter diskretisiert. Es liegen zwei verschiedene Versionen dieses Gitters vor: das Quellgitter, das zu Beginn eines Zeitschritts vorhanden ist, und das Zielgitter, das nach dem Stream-Collide Schritt geschrieben wird. Die Gitter unterscheiden sich im Grunde dadurch, dass zwischen beiden ein Zeitschritt vergangen ist und sich somit Zellinhalte geändert haben können. Die drei Dimensionen werden entsprechend eines Koordinatensystems mit x , y und z bezeichnet. Die x -Richtung ist hierbei die Breite, die y -Richtung die Tiefe und die z -Richtung die Höhe des dreidimensionalen Raumes. Eine Zelle in dem Gitter entspricht einem Würfel und hat somit in allen Dimensionen gleich lange Kanten.

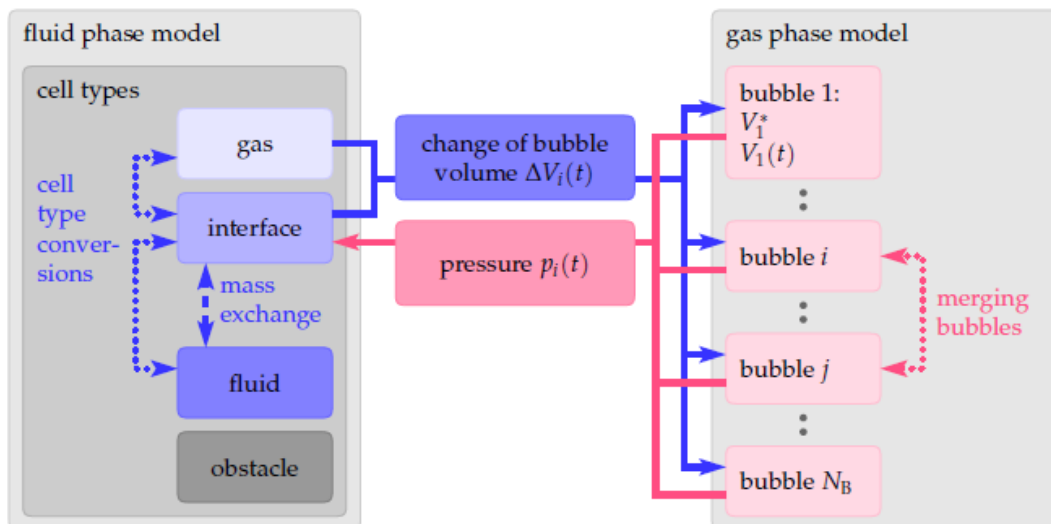


Abbildung 2.1: Überblick über das Modell der freien Oberflächen Strömung [5].

Jede Zelle ist dabei genau einem der folgenden vier Zelltypen zuzuordnen:

- Hindernis-
- Flüssigkeits-
- Gas-
- Interface-Zelle

Hindernis- und Flüssigkeitszellen sind bereits aus dem Standard Lattice Boltzmann Verfahren bekannt. Gas- und Interface-Zellen ergeben im geschlossenen Verbund zusammen eine Gasblase auf dem Gitter. Die Interface-Zellen sind als Grenzfläche zwischen den beiden Phasen definiert und enthalten somit sowohl Anteile von Gas als auch Flüssigkeit. Sie umgeben die Gaszellen und trennen diese von den Flüssigkeitszellen ab. In waLBerla können Interface-Zellen nur dann existieren, wenn als direkter Nachbar sowohl eine Gas- als auch eine Flüssigkeitszelle vorhanden ist (Abbildung 2.2). Das Volumen, das eine einzelne Gas- oder Flüssigkeitszelle enthält, wird zu Beginn einer Simulation über den Parameter dx festgelegt, der angibt wie die Länge der Kanten einer Zelle in der Realität wäre. Der jeweilige Zelleninhalt ergibt sich dann zu dx^3 . Da sich in Interface-Zellen nicht nur eine Phase befindet, wird eine zusätzliche Variable benötigt, die den Füllstand der Zelle mit Flüssigkeit beziffert. Jede Interface-Zelle besitzt außerdem noch eine Oberflächennormale. In waLBerla liegen diese Normalen immer in normierter Form vor und zeigen ins Innere der Blase hinein. Mit Hilfe dieser Normalen lässt sich sowohl die Blasenoberfläche modellieren als auch die Oberflächenkrümmung berechnen.

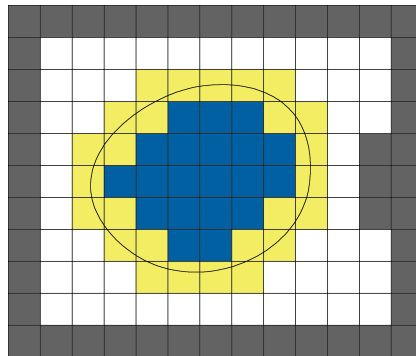


Abbildung 2.2: Ein Beispiel für ein zweidimensionales Gitter: Die Flüssigkeitszellen (weiss) sind umgeben von der Hinderniszellen (dunkelgrau). Im Inneren des Fluids befindet sich eine Blase bestehend aus den Gaszellen (blau) und den Interface-Zellen (gelb)[5].

Durch Bewegungen der Flüssigkeit und des Gases kann sich der Flüssigkeitsanteil in den Zellen ändern. Werden dabei bestimmte Grenzwerte für den Flüssigkeitsanteil erreicht oder wird eine Nachbarzelle einer Flüssigkeits- oder Gaszelle umgewandelt, so kommt es zur Konversion der Zelle. Interface-Zellen können dabei sowohl in Flüssigkeits- als auch Gaszellen umgewandelt werden und umgekehrt, jedoch niemals Flüssigkeits-Zellen direkt in Gaszellen oder umgekehrt

(Abbildung 2.3). Diese Konversionen werden in einem zusätzlichen Flag-Feld gespeichert, wobei hier auch ein Quell- und Zielfeld vorliegen. Das Quellfeld enthält die Zellumwandlungen des vorangegangenen Zeitschritts und das Zielfeld die des aktuellen Zeitschritts.

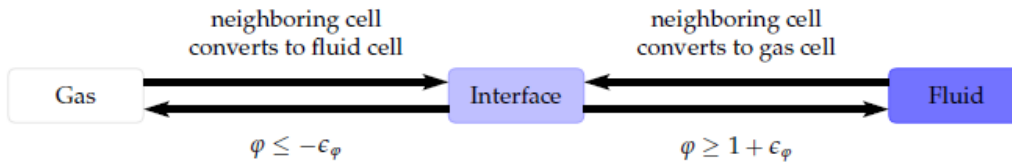


Abbildung 2.3: Darstellung der möglichen Zellkonversionen mit den jeweiligen nötigen Kriterien [5].

Das Fluidphasenmodell benötigt für die Ausführung des Stream-Collide Schrittes den Druck p für die Zellen, die zu einer Blase gehören. Der Druck ist für alle Zellen einer Blase gleich und wird im Gasphasenmodell berechnet. Außerdem können sich die Volumeneinhalte der Blasen im Stream-Collide Schritt ändern, weshalb diese Volumenänderungen ΔV den Blasen im Gasphasenmodell natürlich auch mitgeteilt werden müssen. Deswegen muss zwischen den beiden Modellen eine Verbindung hergestellt werden. Dies geschieht, indem jeder Interface- und Gaszelle über ein Integerfeld eine Identifikationsnummer zugeordnet wird. Dadurch ist eine Zuordnung einer gegebenen Zelle zu einer Blase möglich, umgekehrt kann man allerdings nicht direkt sagen welche Zellen zu einer gegebenen Blase gehören.

Hier wird die Problematik beim Trennen von Blasen in der freien Oberflächen Methode deutlich. Ein Trennen der Blase kann nur im Fluidphasenmodell erkannt werden, da dort die Information über die Topologie der Blase vorliegt. Ist eine Trennung erkannt muss eine neue Blase erstellt werden und das Volumen der Teilblasen auf dem Gitter vermessen werden. Das Blasentrennungproblem lässt sich somit in zwei Probleme unterteilen: die Blasentrennungserkennung und die Volumenvermessung. Beim Problem der Blasentrennungserkennung geht es darum den Zeitpunkt einer Blasentrennung zu finden. Für die Volumenvermessung muss festgestellt werden welche Region auf dem Gitter eine Blase umfasst und der Volumeninhalt mittels Addition der einzelnen Zellvolumen festgestellt werden. Beide Problemstellungen sind hierbei alleinig im Fluidphasenmodell zu lösen. Die Anzahl der zu erstellenden Blasen sowie die berechneten Volumina müssen im Anschluss an das Gasphasenmodell übermittelt werden.

Die Blasenvereinigung gestaltet sich weitaus einfacher: Berühren sich zwei Interface-Zellen mit unterschiedlichen Identifikationsnummern, werden alle Interface-Zellen mit einer der beiden Nummern markiert und der Gasphase wird mitgeteilt welche beiden Blasen sich miteinander vereinigt haben. Die beiden Blasenvolumen werden addiert und die Blase, deren Identifikationsnummer nicht verwendet wurde, wird gelöscht.

3 Algorithmusentwicklung

In diesem Kapitel wird die Entwicklung eines Algorithmus beschrieben, der die Aufgabenstellung erfüllt. Es werden für jeden Teilaspekt jeweils verschiedene Ansätze beleuchtet und deren Tauglichkeit überprüft und diskutiert. Das Kapitel beginnt mit der Suche nach einem Kriterium, das eine wahrscheinliche Blasentrennung mit möglichst geringen Aufwand ermitteln soll. Die Einordnung der Blasentrennungserkennung und Volumenvermessung in die Klasse von Segmentierungsproblemen und die Auswahl eines passenden Algorithmus zur Lösung des Problems wird in Abschnitt 3.2 vorgenommen. Abschließend gilt es die Implementierung des Verfahrens in waLBerla genauer zu erläutern.

3.1 Kriterium zur Blasentrennungserkennung

Eine Blasentrennung eindeutig zu identifizieren ist ein schwieriges Unterfangen und mit hohem Rechenaufwand verbunden. Es ist deshalb nötig die Aufrufe der Blasentrennungserkennung möglichst gering zu halten. Wünschenswert wäre es nur dann eine Überprüfung durchzuführen, wenn eine Blase sich auch wirklich getrennt hat. Da dies allerdings nicht möglich ist, soll stattdessen ein Kriterium gesucht werden, das eine große Menge an Zeitschritten ausschließt, in denen Blasen auf potentielle Aufspaltungen untersucht werden müssen. Dabei darf das Kriterium keine Blasentrennung übersehen, soll aber gleichzeitig so wenige Trennungen wie möglich falsch erkennen. Die Überprüfung des Kriteriums muss außerdem einen sehr viel geringeren Zeitaufwand benötigen als die eindeutige Blasentrennungserkennung. Nur bei Erfüllung der oben genannten Voraussetzungen ist eine Implementierung einer solchen Voruntersuchung sinnvoll und effizient. Die nun folgenden Abschnitte sollen zwei solcher Kriterien genauer untersuchen und auf Tauglichkeit für eine Umsetzung prüfen.

3.1.1 Blasenkrümmung

In diesem Abschnitt wird ein Ansatz für ein Ausschlusskriterium über die Blasenkrümmung diskutiert. Als Ausgangspunkt für die Wahl dieses Kriteriums dient die Beobachtung einer Gasblase, die sich in zwei separate Blasen aufteilt. Kurz vor der Aufspaltung der Blase in zwei Teile befinden sich die beiden Blasenhälften in einer Art Tropfenform (Abbildung 3.1). Diese Form kommt durch das Zusammenwirken von Kräften, die für den Trennungsvorgang verantwortlich sind, und der Oberflächenspannung zustande. Die Oberflächenspannung versucht dabei die Oberfläche der Blase zu minimieren und in die ideale Form einer Kugel zu bringen. Die von außen wirkenden Kräfte ziehen die Blase auseinander. Dadurch entsteht im Trennungsbereich eine Einschnürung, die zum Reißen der Blase führt. Die Oberflächenkrümmung der Tropfenspitzen erreicht hierbei, relativ zur restlichen Blasenoberfläche, sehr hohe Werte. Eben jene Werte sollen ein Kriterium für die Blasentrennungserkennung liefern.

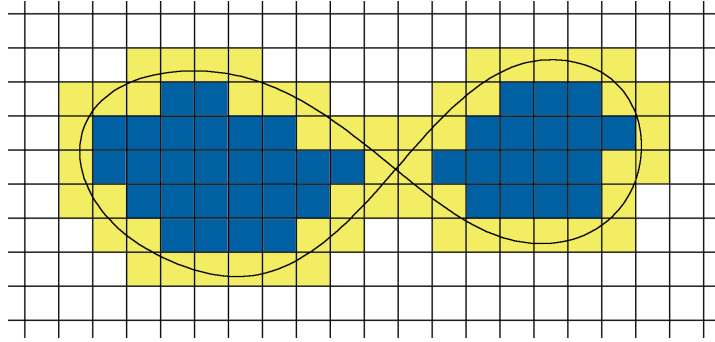


Abbildung 3.1: Blase in idealisierter Form vor der Trennung.

Die Tauglichkeit des Kriteriums wird anhand einer Simulation festgestellt. Es handelt sich dabei um eine Simulation die zum Testen des fertigen Algorithmus in Abschnitt 4.1 angewendet wird. Eine genauere Beschreibung ist dort zu finden. Die Simulation besteht daraus, dass eine längliche Blase, ohne Einfluss von Gravitation, einer Einströmung ausgesetzt wird. Die Einströmung geht dabei in einen schmalen Streifen über die gesamte Tiefe des simulierten Raumes und trifft mittig auf die Blase. Nach wenigen Zeitschritten teilt sich die Blase an dieser Stelle in zwei Blasen auf. Um das Kriterium in der Simulation zu verifizieren wird auf die freie Oberflächen Methode zurückgegriffen. Überschreitet hier ein Krümmungswert einen bestimmten Grenzwert, tritt dieser an die Stelle des Krümmungswertes. Diese Eigenschaft der freien Oberflächen Methode wird nun benutzt um die Werte der Krümmung kurz vor der Trennung zu untersuchen. Ziel dieses Vorgehens ist es Koordinaten von Interface-Zellen zu erhalten, bei denen solch einen Austausch direkt zeitgleich mit der Aufspaltung der Blase stattfindet. Dies stellt einen klaren Vorteil des Verfahrens dar, da kein zusätzlicher Rechenaufwand benötigt wird, um es zu überprüfen.

Die Auswertung der Simulation zeigt, dass kurz vor der Blasentrennung die Grenzwerte zwischen den Blasenhälften überschritten werden. Auch der Ort korreliert mit dem Aufspaltungs-ort. Allerdings schlägt das Kriterium nicht im Zeitschritt der vollständigen Trennung an. Erklären lässt sich dieses Phänomen mit Hilfe der Betrachtung des Vorgangs der Blasenspaltung. Kurz vor der Trennung der beiden Hälften der Blase ist eine zwei Interface-Zellen große Brücke entstanden, die die Blasen noch zusammenhält. Die Krümmung der Oberfläche der Blase ist an der Brücke sehr hoch und führt zu den Meldungen einer eventuellen Trennung. Die Endgültige Abspaltung erfolgt durch Auflösung eben jener Interface-Zellen. Es entstehen zwei Grenzflächen mit viel niedriger Oberflächenkrümmung als zuvor. Deshalb wird auch der Grenzwert nicht mehr überschritten.

Das Kriterium erfüllt somit nicht die Auflagen, dass es eine Blasentrennung erkennt. Es kann lediglich als ein Hinweis für eine mögliche, bald stattfindende Spaltung aufgefasst werden.

3.1.2 Umwandlung von Interface-Zellen in Liquid-Zellen und Überprüfung gegenüberliegender Normalen

Nähere Betrachtung des Blasentrennungsvorgangs zeigt, dass zwei Teilblasen vollständig voneinander gelöst sind, wenn die Oberflächen keine Verbindung mehr aufweisen. In Abbildung 3.2 wird eine Blase kurz nach der Aufspaltung auf dem Gitter in waLberla dargestellt. Die Grenzfläche der Blase ist eindeutig durch eine einzellige Schicht von Flüssigkeitszellen getrennt. Es muss also im Zeitschritt davor eine Interface-Zelle gegeben haben, die sich in eine Flüssigkeitszelle umgewandelt hat, sonst wären die Teilblasen schon vorher voneinander abgelöst gewesen. Die Untersuchung eben jener Interface-Zelle soll ein weiteres Indiz für eine Blasentrennung ergeben.

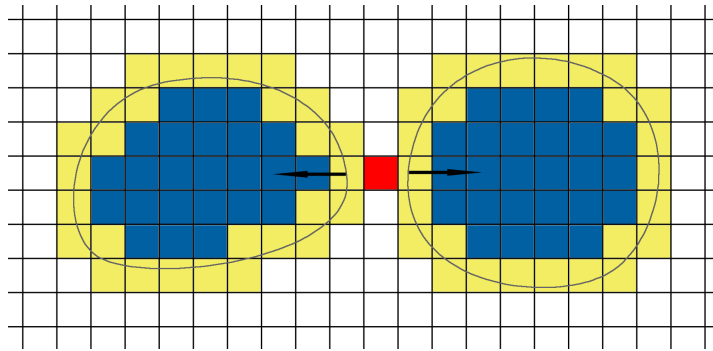


Abbildung 3.2: Blasen kurz nach der Trennung mit in Flüssigkeitszelle umgewandelter Interface-Zelle(rot) und den Oberflächennormalen der direkten Nachbarzellen.

Durch das in Kapitel 2 vorgestellte Flag-Feld, das die Konversionen des letzten Zeitschritts speichert kann dies sehr einfach realisiert werden, indem die Umwandlungen von Interface- in Flüssigkeitszellen überprüft werden. Allerdings reicht dieses Kriterium nicht aus, um einen Trennungsvorgang genau zu identifizieren. Die Blase bewegt sich in jedem Zeitschritt fort und verformt sich, wodurch natürlich auch Zellumwandlungen stattfinden. Die Folge ist eine fehlerhafte Erkennung.

Dabei gilt es auch die jeweilige Identifikationsnummer aufeinander abzugleichen. Interface-Zellen sind nur dann zu untersuchen, wenn sowohl die umgewandelte als auch die umliegenden Zellen die gleiche Nummer besitzen. Eine Lösung für dieses Problem ist, die Nachbarschaft um die umgewandelte Interface-Zelle genauer zu betrachten. In der direkten Umgebung der Zelle befinden sich mindestens zwei weitere Interface-Zellen, die sich gegenüberliegen. Grund dafür sind die beiden Grenzflächen der nun getrennten Blase. Um auch sicher zu gehen, dass es sich wirklich um Interface-Zellen handelt, die nicht über eine Grenzfläche miteinander verbunden sind, wird überprüft, ob die entsprechenden Oberflächennormalen beinahe antiparallel sind. Außerdem ist es nun auch notwendig die Identifikationsnummer der jeweiligen Interface-Zelle zu überprüfen. Eine Untersuchung eines Zellenpaares findet nur dann statt, wenn sowohl die umgewandelte als auch die beiden umliegenden Zellen die gleiche Nummer besitzen. So wird

sicher gestellt, dass die Zellen der gleichen Blase angehören. Die Überprüfung der Antiparallelität lässt sich mit Hilfe des Skalarprodukts der beiden Normalen einfach erreichen:

$$\vec{n}_1 \cdot \vec{n}_2 = |\vec{n}_1| |\vec{n}_2| \cos \sphericalangle (\vec{n}_1, \vec{n}_2) \quad (3.1)$$

Wobei $\sphericalangle (\vec{n}_1, \vec{n}_2)$ für den Winkel zwischen den Normalen und $|\vec{n}_1|, |\vec{n}_2|$ jeweils für die Länge einer Normalen steht. Der Kosinus des Winkels ergibt sich damit zu

$$\frac{\vec{n}_1 \cdot \vec{n}_2}{|\vec{n}_1| |\vec{n}_2|} = \cos \sphericalangle (\vec{n}_1, \vec{n}_2) \quad (3.2)$$

und ist somit nur abhängig von dem Skalarprodukt der normierten Normalen. Um die Antiparallelität der Oberflächennormalen gewährleisten zu können, sollte sich der Winkel im Bereich von 170° bis 180° befinden. Der entsprechende Kosinus liegt bei kleiner -0.98 .

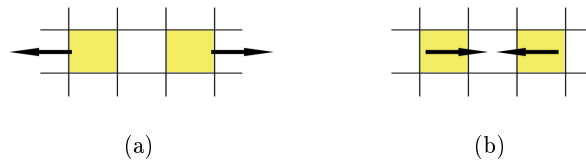


Abbildung 3.3: Interface-Zellen mit Normalenpaaren, die erkannt werden, obwohl nur die Erkennung von (a) korrekt ist.

Die Problematik bei der alleinigen Betrachtung des Winkels ist darin zu sehen, dass auch solche Normalenpaare als Treffer erkannt werden, die aufgrund der Lage der zugehörigen Interface-Zellen aufeinander zu deuten (Abbildung 3.3). Solche Paare lassen sich durch Einführen eines zusätzlichen Vektors ausschließen. Der Vektor geht hierbei von der einen zu untersuchenden Interface-Zelle zur anderen. Angenommen der Anfangspunkt wäre die Zelle I_1 und der Endpunkt die Zelle I_2 mit den zugehörigen Normalen \vec{n}_1 und \vec{n}_2 , so ist der gesuchte Vektor $\vec{v} = -\vec{I}_1 + \vec{I}_2$, wobei die Vektoren \vec{I}_1 und \vec{I}_2 vom Koordinatenursprung hin zu den jeweiligen Interface-Zellen zeigen. Um nun die unerwünschten Normalenpaare auszuschließen, wird auch hier jeweils das Skalarprodukt zwischen dem normierten Vektor \vec{v} und den Normalen \vec{n}_1 und \vec{n}_2 gebildet. Danach werden die Ergebnisse in Abhängigkeit von der jeweiligen Zelle weiter behandelt. Das Resultat des Skalarprodukts zwischen \vec{v} und \vec{n}_1 wird daraufhin untersucht, ob es kleiner 0 ist, das mit \vec{n}_2 , ob es größer 0 ist. Werden beide Überprüfungen als wahr erachtet, steht aufgrund der Richtung des Vektors \vec{v} fest, dass die Normalen in entgegengesetzte Richtungen zeigen (Abbildung 3.4).

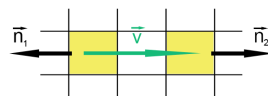


Abbildung 3.4: Hilfsvektor \vec{v} und Normalenpaar, das als Treffer erkannt werden muss.

Dieser Abschnitt soll eine kurze Abschätzung über den Aufwand, des vorgestellten Kriteriums liefern. Für die Überprüfung des Kriteriums müssen im Flag-Feld alle Zellen gefunden werden, die von Interface- in Flüssigkeitszellen umgewandelt wurden. Dies entspricht einem Durchlauf über ein Feld mit der Problemgröße. Desweiteren ist es nötig im Anschluss die jeweiligen Nachbarschaften für jede gefundene Zelle auf dem Gitter zu überprüfen. Aufgrund der Tatsache, dass pro zu untersuchender Zelle jeweils die Normalen von 26 Zellen miteinander verglichen werden müssen, ist Schritt zwei mit viel Aufwand verbunden. Die Effizienz des Verfahrens ist deshalb sehr stark von der Anzahl der Zellen abhängig, die im zweiten Teil überprüft werden müssen. Festzustellen ist hierbei, dass im Normalfall Interface-Zellen der am wenigsten vorkommende Zellentyp auf dem Gitter ist. Außerdem treten innerhalb eines Zeitschrittes meistens auch nur wenige Zellkonversionen von Interface- in Flüssigkeitszellen auf, was die Zahl der zu überprüfenden Zellen noch weiter einschränkt. Logischerweise werden deswegen auch nur wenige Nachbarschaften untersucht. Zusätzlichen Speicherplatz benötigt das Verfahren nicht, da alle verwendeten Daten bereits in der freien Oberflächen Methode gespeichert werden. Es ist lediglich nötig eine Variable einzuführen, die die relevanten Blasen für die anschließende Blasentrennungserkennung und Volumenvermessung markiert.

Auch dieses Kriterium wurde anhand von Simulationen getestet, die in Kapitel 4 näher beschrieben werden. Hier sollen nur die relevanten Ergebnisse präsentiert werden. Es zeigte sich, dass das Hinzufügen der Überprüfung des Kriteriums nur einen geringen Einfluss auf die Gesamtlaufzeit hat. Außerdem wurden nur sehr wenige falsche Blasentrennungen erkannt, die sich alle im Zeitraum kurz vor der eigentlichen Blasentrennung befinden. Dies erklärt sich dadurch, dass kurz vor der Trennung Interface-Zellen zwischen den beiden entstehenden Grenzflächen in Flüssigkeitszellen umgesetzt werden und sich deshalb in deren Nachbarschaft viele Interface-Zellen befinden, von denen wiederum einzelne das Kriterium erfüllen. Zusätzlich wurde noch eine Testreihe zum Winkel zwischen den Normalen durchgeführt. Bereits ab einem Winkel von 110° ist für die verwendeten Simulationen festzustellen, dass keine zusätzlichen Falschmeldungen über mögliche Blasentrennung berichtet werden. Es ist also nicht nötig einen Winkel zu wählen der nahe 180° liegt. Ein Grund dafür könnte sein, dass in den Simulationen nur relativ simple Geometrien verwendet und somit nur ein kleiner Teil von möglichen Blasentrennungsabläufen untersucht wurden. Es ist nicht auszuschließen, dass sich im späteren Einsatz Szenarien ergeben, in denen bei kleinem Winkel mehr Zeit benötigt wird, da vermehrt falsche Treffer gefunden werden. Außerdem lässt sich keine Aussage über einen optimalen Winkel treffen. Dieser muss erst in einer längeren Versuchsreihe mit komplexeren Simulationen genauer bestimmt werden.

Schlussendlich erfüllt das Kriterium alle in Abschnitt 3.1 gestellten Bedingung und kann als Voruntersuchung für die aufwändigere Blasentrennungserkennung und Volumenvermessung eingesetzt werden.

3.2 Segmentierungsalgorithmen

In diesem Abschnitt werden Ansätze aus der Bildverarbeitung vorgestellt und diskutiert, die eine eindeutige Blasentrennungserkennung und anschließende Volumenvermessung ermöglichen sollen.

Bei der Blasentrennungserkennung müssen Regionen von Zellen mit gleicher Identifikationsnummer auf Zusammenhänge hin untersucht werden. Besteht zwischen zwei Regionen keine direkte Verbindung, hat sich die Blase geteilt. Für die Volumenvermessung müssen daraufhin die jeweiligen Zellen der Teilblasen lokalisiert und die jeweiligen Volumeninhalte aufaddiert werden. Im Idealfall sollte ein Ansatz deshalb möglichst eine Blasentrennung nicht nur Erkennen, sondern die Teilbereiche unterschiedlich markieren. Die Volumenvermessung kann anschließend durch Abzählen der Gaszellen und Addieren der Füllstände der Interface-Zellen erreicht werden.

Ein Teilbereich der digitalen Bildverarbeitung, die so genannte Segmentierung, beschäftigt sich mit derartigen Problemen. Der Begriff Segmentierung steht für das Erzeugen von inhaltlich zusammenhängenden Regionen gemäß eines Homogenitätskriteriums. Dabei wird zwischen pixel-, kanten und regionenorientierten Verfahren unterschieden [4, 6]:

Pixelorientierte Verfahren bestimmen die Zugehörigkeit eines Bildpunktes zu einem bestimmten Objekt einzig über den Farbwert eines Punktes. Die meist verwendete Methode dieser Kategorie ist das so genannte Schwellwertverfahren. Hier werden alle Punkte, die einen bestimmten Wert über- bzw. unterschreiten, entweder dem Objekt oder dem Hintergrund zugeordnet.

Kantenorientierte Verfahren benutzen verschiedene Arten von Filtern, um Kanten im Bild zu erkennen. Daraufhin wird mittels eines Konturverfolgungsalgorithmus die Kontur aus dem Kantenbild extrahiert. Alle Punkte die nun innerhalb der durch die Kontur begrenzten Region liegen, gehören dann zum Segment. Die wichtigsten verwendeten Kantendetektionsfilter sind der Laplace- und Sobel-Filter.

Regionenorientierte Verfahren versuchen Gruppen von benachbarten Bildpunkten zu finden, die in Hinblick auf ein bestimmtes Kriterium homogen sind. Das Homogenitätskriterium wird dabei so gewählt, dass die entstehende Region einer Fläche eines Objekts entspricht. Eine typische regionenorientierte Methode ist die Regionenexpansion. Ausgehend von einem Anfangspunkt werden benachbarte Punkte gesucht, die das gewählte Homogenitätskriterium erfüllen. Wird ein Punkt ausgewählt, werden auch dessen Nachbarzellen überprüft. Der Algorithmus terminiert, wenn keine weiteren Punkte das Kriterium mehr erfüllen, und gibt die markierte Fläche zurück. Dieses Vorgehen ist in der Computergrafik auch unter dem Begriff Flutfüllung bekannt. Hier wird es allerdings dazu verwendet monochrome Flächen einzufärben.

Die Problemstellung der Blasentrennungserkennung umfasst jedoch nur bestimmte Teilaspekte der Segmentierung. Die Zellen einer Blase sind bereits über die Identifikationsnummer eindeu-

tig einer Blase zuzuordnen. Es ist also nicht nötig die Bereiche erneut einzugrenzen. Vielmehr geht es darum herauszufinden, ob die Zellen vollständig miteinander verbunden sind. Es kommen deshalb nur Ansätze der Bildsegmentierung in Frage, die Zusammenhänge zwischen den einzelnen Bildpunkten zur Generierung der Regionen verwenden. Dementsprechend sind alle Algorithmen, die auf der alleinigen Betrachtung der Pixelwerte beruhen, nicht geeignet für die Lösung des Problems.

Die Lösung der Blasentrennungserkennung mittels eines kantenorientierten Verfahrens ist denkbar. In waLBERla ist die Kontur einer Blase gegeben über ein Verbinden der Interface-Zellen, die zur Blase gehören. Bildet man ausgehend von einer Interface-Zelle die Kontur einer Blase ist feststellbar, ob eine Blasentrennung stattgefunden hat. Der Grund dafür ist, dass alle Interface-Zellen in der Kontur enthalten sein müssen, wenn die Blase aus einer Region besteht, ansonsten würden mehrere Konturen und somit auch Blasen vorliegen. Hat sich eine Blase gespalten müssen anschließend die Volumeninhalte der Zellen, die von den einzelnen Konturen umschlossen sind, gezählt werden, um das Volumen der Teilblasen zu erhalten.

Es ist auch möglich die Problematik mit Hilfe der Flutfüllung zu lösen. Dazu muss ausgehend von einer Blasenzone die Blase rekursiv eingefärbt werden. Findet der Algorithmus keine Zellen mehr zum Einfärben, müssen alle Interface- und Gaszellen der Blase markiert sein. Andernfalls sind Blasenzone räumlich nicht miteinander verbunden und es kann auf eine Blasentrennung geschlossen werden. Das Einfärben aller Teilblasen mit einer eigenständigen Farbe kann erreicht werden, indem die rekursive Flutfüllung mit einer Untersuchung des Gitters auf noch nicht gefärbte Zellen verbunden wird. Dazu wird immer wenn eine Zelle keiner Farbe zugeordnet ist, die Teilregionen rekursiv mit einer neuen Farbe gefüllt. Sind alle Zellen der Blase markiert, sind so viele Teilblasen entstanden wie Farben benötigt wurden. Gleichzeitig kann während des Einfärbens auch das Volumen der Zellen, die mit der jeweiligen Farbe gefüllt sind, vermessen werden. Dieser Vorteil gegenüber einem Ansatz mittels Konturuntersuchung ist deshalb auch ausschlaggebend dafür, dass eine iterative Variante des Flutfüllungsalgorithmuses für die Implementierung in waLBERla ausgewählt wurde.

3.3 Implementierter Algorithmus zur Blasentrennungserkennung und Volumenvermessung

3.3.1 Bounding Box

Ein Großteil der Rechenzeit beim Untersuchen von Blasen in der freien Oberflächen Methode wird darauf verwendet, für eine Blase die zugehörigen Interface- und Gaszellen zu suchen. In waLberla ist es nicht möglich von den Blasendaten direkt auf die Blasenzellen zuzugreifen. Der Weg führt immer über eine Iteration über die gesamte Größe des Gitters. Da eine Gasblase im Allgemeinen nur einen kleinen Teil des Raumes einnimmt, ist es unsinnig so vorzugehen. Den direkten Zugriff von Blasendaten auf die einzelnen Blasenzellen zu ermöglichen, ist allerdings schwierig. Für jede Blasenzone müssten die x -, y - und z -Koordinate der Zelle auf dem Gitter, also drei Integer-Werte gespeichert werden. Da die Identifikationsnummern in einem Integer-Feld hinterlegt sind, erhöht sich der Speicherplatzbedarf um das Dreifache. Außerdem benötigt keine andere Funktion in der freien Oberflächen Methode die Zellen einer einzelnen Blase zur Bearbeitung. Im Normalfall iterieren die Methoden über das Gitter und greifen über die Identifikationsnummern auf die Blasendaten zu. Der erhöhte Speicherplatzbedarf stünde deshalb in keinem Verhältnis zum Nutzen. Der Rechenaufwand lässt sich mit der Einführung von Bounding Volumes für die Blasen auf dem Gitter minimieren. Aufgrund der kartesischen Gitterstruktur eignen sich vor allem Bounding Boxen, um den Rauminhalt, in dem nach Blasen gesucht werden, zu begrenzen. Die Bounding Box entspricht somit einem kleinen Ausschnitt aus dem Gitter. Um die Bounding Box rekonstruieren zu können genügt es, den minimalen und maximalen Wert der Ausbreitung der jeweiligen Blasen zu speichern. Dies bedeutet für jede Blase müssen sechs zusätzliche Integer-Werte gespeichert werden. Zu beachten ist, dass die begrenzenden Werte der Bounding Box, bei jeder Bewegung der Blase, neu überprüft werden müssen.

3.3.2 Iteratives Flutfüllungsverfahren zur Blasentrennungserkennung und Volumenvermessung

In Abschnitt 3.2 wurde bereits erläutert wie die Blasentrennung mittels eines rekursiven Flutfüllungsalgorithmus gelöst werden kann. Für die eigentliche Implementierung wurde allerdings eine iterative Variante ausgewählt. Die Gründe hierfür liegen zum einen an der hohen Rekursionstiefe der Flutfüllung und zum anderen an der einfacheren Parallelisierbarkeit iterativer Verfahren. Die Rekursionstiefe ist bei der Flutfüllung von der Anzahl der zu färbenden Zellen sowie den möglichen Richtungen, die von einer Zelle aus weiter untersucht werden müssen, abhängig. Im dreidimensionalen Fall ergeben sich pro gefärbter Zelle sechs Aufrufe der Flutfüllungsmethode. Bei großen Blasen folgt deshalb ein großer Speicherplatzbedarf des Laufzeitkellers. Außerdem beeinträchtigt der Aufbau und die Abarbeitung der hohen Menge an Rekursionsebenen die Performanz des Algorithmus. Obwohl es sich bei dem hier vorgestellten Algorithmus um eine Einkernprozessorlösung handelt, soll trotzdem auf die Parallelisierbarkeit

des Verfahrens geachtet werden. In der Praxis werden sowohl große als auch sehr fein aufgelöste Gitter betrachtet, die nur mit parallelen Algorithmen in angemessener Zeit bewältigt werden können. Deswegen erhält die iterative Variante auch hier den Vorzug vor der rekursiven, da sich das Gebiet, auf dem iterative Methoden arbeiten, leicht in mehrere Teilregionen aufspalten lässt. Die gewonnenen Gebiete können dann von jeweils einem Prozessor bearbeitet werden. Allerdings wird in diesem Abschnitt nicht weiter auf eine Mehrkernprozessorlösung eingegangen, weil sich die Kommunikation zwischen den Prozessoren als schwierig erweist.

Für die Implementierung der iterativen Flutfüllung ist es nötig neue Speicherstrukturen einzuführen. Ein weiteres dreidimensionales Datenfeld in der Größe des Gitters, das sogenannte Farbfeld, ist erforderlich, um die durch den Algorithmus mit Farbe markierten Bereiche zu speichern. Die einzelnen Farben werden hierbei über unterschiedliche Integerwerte repräsentiert. Je nach vorhandener Geometrie werden unterschiedlich viele Farben zum Einfärben der Regionen benötigt. Die Verwaltung der Farben übernimmt ein Datenfeld, das die folgenden Parameter einer Farbe zuordnet:

- Initialfarbe
- aktuelle Farbe
- Volumen

Die Initialfarbe bezeichnet den anfänglichen Integerwert der Farbe und entspricht dem jeweiligen Platz der Farbe im Datenfeld. Die Ausführung des Algorithmus kann dazu führen, dass sich der Wert einer Farbe ändert, deshalb wird der aktuelle Wert in einer eigenständigen Variablen gespeichert (Pseudocode 1). Markierungen von Zellen werden immer mit der aktuellen Farbe durchgeführt. Das Volumen gibt das jeweilige durch die Initialfarbe eingefärbte Volumen an. Das bedeutet, wenn der aktuelle Wert nicht mehr dem Anfangswert entspricht, wird das Volumen einer neu markierten Zelle der Farbe mit dem dementsprechenden Platz im Datenfeld zugeschrieben.

Pseudocode 1: Tabelle zur Verwaltung der einzelnen Farben

```
1 Datenfeld fv mit Einträgen Farbe
2   Eintrag Farbe besteht aus:
3     int initFarbe;
4     int aktFarbe;
5     double Volumen;
```

Der Flutfüllungsalgorithmus wird für jede Blase, die als Kandidat für eine mögliche Blasen-trennung durch das in Abschnitt 3.1 vorgestellte Kriterium markiert wurde, aufgerufen.

Pseudocode 2 skizziert den Ablauf des implementierten Verfahrens. Zu beachten ist, dass aktuelle Farben nur mithilfe des Zugriffs auf das Datenfeld fv (Pseudocode 1) abgefragt werden

können. Solche Abfragen finden statt, da die zur Markierung auf dem Farbfeld genutzten Werte sich im Laufe des Algorithmus geändert haben könnten.

Der Algorithmus lässt sich im Prinzip in zwei Teile aufspalten: zum einen die Markierung der Zellen der Blase auf dem Farbfeld und der Aufbau des Datenfeldes fv (Zeilen 1- 51), zum anderen die Auswertung der Blasentrennungserkennung und abhängig davon die Erstellung neuer Gasblasen (Zeilen 53-69).

Im ersten Teil läuft das Verfahren über den von der Bounding Box abgegrenzten Teil des Gitters und sucht nach Interface- sowie Gaszellen, die die gleiche Identifikationsnummer wie die zu untersuchende Blase besitzen. Das Gitter wird hierbei in jeweils positiver Richtung der einzelnen Koordinatenachsen abgelaufen: zuerst die x-Richtung, dann die y-Richtung und schlussendlich die z-Richtung (Zeilen 1-4).

Ist eine entsprechende Zelle gefunden, wird gegen die Laufrichtung in den jeweiligen Nachbarzellen überprüft, ob bereits eine Farbe gesetzt wurde. Die restlichen Nachbarzellen werden nicht untersucht, da der Algorithmus dort wegen der iterativen Abfolge noch keine Farbe gesetzt haben kann. Die als erstes gefundene Farbe wird zur Markierung der Zelle verwendet. Ist keine Farbe in der Nachbarschaft vorhanden, wird eine Neue erstellt, dem Datenfeld hinzugefügt und die Zelle mit dieser eingefärbt (Zeilen 6- 15).

Anschließend muss das Gasvolumen der Zellen zum Volumen der ausgewählten Farbe addiert werden. Das Volumen der Zellen hängt davon ab, ob es sich um eine Interface- oder Gaszelle handelt. Da Gaszellen vollkommen mit Gas gefüllt sind, ist ihr Volumen 1.0. Interface-Zellen hingegen sind nur Teilweise mit Gas gefüllt. Ihr Volumen ergibt sich zu 1.0 abzüglich des Füllstands der Zelle mit Flüssigkeit (Zeilen 17-20).

Befinden sich in der Nachbarschaft einer Zelle verschiedene Farben, so werden diese beim Markieren der Zelle miteinander verbunden. Dies wird erreicht, indem die aktuelle Farbe der benachbarten Zellen auf den selben Wert gesetzt wird. Im Algorithmus wird immer die Farbe als Farbe für den Verbund ausgewählt, die den niedrigsten aktuellen Wert besitzt. Diese Prozedur ist im Verfahren so realisiert, dass jeweils der Verbund zwischen zwei Farben durchgeführt wird. Deshalb überprüft der Algorithmus die in y- und z-Richtung liegenden Nachbarzellen getrennt. Die x-Richtung muss aufgrund der Vorgehensweise der Farbauswahl nicht beachtet werden. Ist die Nachbarzelle in x-Richtung nämlich gefärbt, so ist auch die Zelle selbst mit dieser Farbe markiert.

Pseudocode 2: Iterativer Flutfüllungsalgorithmus

```

1  for Zellen in durch Bounding Box begrenzten Bereich {
2
3  if Zelle(x,y,z) == Interface- oder Gaszelle and
4  if Identifikationsnummer der Zelle == Identifikationsnummer der Blase {
5
6  if Farbfeld(x-1,y,z) markiert mit Farbe
7  Farbfeld(x,y,z) := fv(Farbfeld(x-1,y,z)).aktFarbe;
8  else if Farbfeld(x,y-1,z) markiert mit Farbe
9  Farbfeld(x,y,z) := fv(Farbfeld(x,y-1,z)).aktFarbe;
10 else if Farbfeld(x,y,z-1) markiert mit Farbe
11 Farbfeld(x,y,z) := fv(Farbfeld(x,y,z-1)).aktFarbe;
12 else{
13     erstelle neue Farbe und füge diese fv hinzu;
14     Farbfeld(x,y,z):= neue Farbe;
15 }
16
17 if Interface-Zelle
18     Volumen gewählter Farbe += 1.0 - Füllstand(x,y,z);
19 else
20     Volumen gewählter Farbe += 1.0;
21
22 bool farbVerbund = false;
23 if Farbfeld(x,y-1,z) markiert mit Farbe and
24 if Farbfeld(x,y,z) != fv(Farbfeld(x,y-1,z)).aktFarbe {
25     farbVerbund = true;
26
27     if Farbfeld(x,y,z) < fv(Farbfeld(x,y-1,z)).aktFarbe {
28         if aktFarbe != initFarbe von Farbfeld(x,y-1,z)
29             fv(fv(Farbfeld(x,y-1,z)).aktFarbe).aktFarbe:= Farbfeld(x,y,z);
30
31         fv(Farbfeld(x,y-1,z)).aktFarbe := Farbfeld(x,y,z);
32     }
33     else {
34         if aktFarbe != initFarbe von Farbfeld(x,y,z)
35             fv(fv(Farbfeld(x,y,z)).aktFarbe).aktFarbe;
36             := fv(Farbfeld(x,y-1,z)).aktFarbe;
37
38         fv(Farbfeld(x,y,z)).aktFarbe:= fv(Farbfeld(x,y-1,z)).aktFarbe;
39     }
40 }
41
42
43 //analoges Vorgehen für Farbverbund mit Zelle(x,y,z-1), allerdings
44 //muss auch für Farbfeld(x,y,z) die aktuelle Farbe überprüft werden
45 ...
46
47
48

```

```

49     if farbVerbund == true {
50         for Farben in fv {
51             aktFarbe := fv(aktFarbe).aktFarbe;
52         }
53     }
54 }
55
56 if mehr als eine aktFarbe in Datenfeld fv {
57     for jede vorhandene aktFarbe bis auf kleinste {
58         Volumen der Teilblase:= addiertes Volumen von Farben mit gleicher aktFarbe;
59         Erstelle neue Blasen mit:d
60         Volumenverhältnis:= Volumen jeweiliger Teilblase / Gesamtblasenvolumen;
61         aktuelles Volumen:= Volumen jeweiliger Teilblase;
62         initialVolumen:= Volumenverhältnis * initialVolumen der Blase;
63         neuer Identifikationsnummer;
64     }
65
66     Verwende Originalblase für kleinste aktFarbe mit
67     entsprechenden aktuellen und initialen Volumen
68
69     for durch Bounding Box begrenztes Gebiet {
70         Setze Identifikationsnummern für neue Blasen
71     }
72 }

```

Die Zeilen 22-40 in Pseudocode 2 beschreiben den Farbverbund zwischen der Zelle und einer Nachbarzelle in y -Richtung. Ein solcher Verbund tritt ein, wenn sich in den beiden Zellen unterschiedliche Farben befinden. Die Auswahl der Verbundfarbe ist davon abhängig welche der beiden Farben den kleinsten aktuellen Wert besitzt. Die andere Farbe übernimmt diesen dann als aktuellen Farbwert. Die Problematik bei einer solchen Vereinigung liegt darin, dass sich bei Farben, deren anfänglicher und aktueller Farbwert voneinander abweichen, auch der aktuelle Wert der entsprechenden aktuellen Farbe ändern muss. Dazu muss die aktuelle Farbe aus dem Datenfeld fv gelesen und anschließend auf den richtigen Wert gesetzt werden (Zeilen 30-31 und 36-38).

Das Vorgehen für die z -Richtung läuft analog zur y -Richtung ab. Allerdings muss beim Überprüfen und Setzen des Farbwertes, der Zelle selbst, nun die aktuelle Farbe verwendet werden. Ursache hierfür ist, dass bereits ein Farbverbund in y -Richtung stattgefunden haben kann und somit der Initialwert vom aktuellem Wert abweicht (Zeilen 43-45).

Es entstehen beim Verbinden von Farben auch weitere Abhängigkeiten zwischen den Farben. Problematisch ist vor allem wenn eine große von einer kleinen Initialfarbe abhängt. Ändert sich nämlich der aktuelle Farbwert der kleinen Initialfarbe, muss sich auch gleichzeitig der aktuelle Wert der großen ändern. Dieses Problem wird gelöst indem nach dem Verbinden aller Farben um eine Nachbarzelle das Datenfeld fv durchlaufen wird und alle aktuellen Farben mit

der aktuellen Farbe der entsprechenden Initialfarbe ersetzt werden (Zeilen 49-54).

Ist der Bereich innerhalb der Bounding Box eingefärbt, ist der eigentliche Flutfüllungsalgorithmus beendet. Ein Teilbereich einer Blase ist nun mit einer einzigartigen aktuellen Farbe markiert. Die Anzahl der aktuellen Farben im Datenfeld fv entspricht der Anzahl von Blasen in die sich die Blase aufgespalten hat. Liegt nur eine aktuelle Farbe vor, hat sich die Blase nicht getrennt und der Algorithmus endet. Mehrere aktuelle Farben bedeuten hingegen, dass eine Blasentrennung stattgefunden hat. Für die aktuellen Farben müssen nun neue Blasen mit entsprechenden Volumina und Identifikationsnummer erstellt werden. Das Volumen einer Blase ergibt sich aus dem Verhältnis zwischen der Summe der Volumen der entsprechenden aktuellen Farbe und dem Volumen der Gesamtblase (Zeilen 56-64).

Abschließend wird ein weiteres Mal über das durch die Bounding Box begrenzte Gebiet iteriert um für die entsprechenden Regionen die neuen Identifikationsnummern zu setzen (Zeilen 69-72).

Eine Ausnahme bildet die aktuelle Farbe mit dem kleinsten Wert. Für diese wird lediglich das aktuelle und anfängliche Volumen der Originalblase angepasst. Das Setzen neuer Identifikationsnummern für diesen Teilbereich ist deshalb auch nicht notwendig (Zeile 66-67).

3.3.3 Performanz

Die Erkennung von Blasentrennungen und das Vermessen der Volumen lässt sich mittels Flutfüllung in einem Durchlauf über das von der Bounding Box begrenzte Gebiet bewerkstelligen. Die Performanz des Verfahrens ist dabei vor allem abhängig von der Anzahl der pro Zelle zu untersuchenden Verzweigungen. Das Markieren von Zellen mit einer entsprechenden Farbe benötigt dabei relativ wenige Überprüfungen von Verzweigungen. Das Verbinden von bis zu drei Farben hingegen ist aufgrund der Anzahl der auftretenden Verzweigungen sehr zeitaufwändig. Die Anzahl von Farbverbänden wird stark von der Geometrie der Gasblasen beeinflusst. Eine ideale Blasengeometrie stellt der Quader dar. Dieser lässt sich durchgängig mit einer Farbe markieren und somit würden keine Farbvereinigung auftreten (Abbildung 3.5a). Der zeitaufwändigste Fall ist in Abbildung 3.5b dargestellt. Hier wird für fast jede Nachbarzelle der Interface-Zellen am linken Rand jeweils eine Farbvereinigung benötigt. Beide Fälle werden im Regelfall allerdings nicht auftreten. Aufgrund der Oberflächenspannung streben die Gasblasen danach ihre Oberfläche zu minimieren und nehmen so kugelähnliche Formen an. Deshalb finden Farbvereinigungen auch meist nur in der unteren Hälfte von Blasen statt (Abbildung 3.5c).

Das in Abschnitt 3.1.2 vorgestellte Kriterium trifft eine sehr gute Vorauswahl, so dass der kostenintensive Blasentrennungserkennungs- und Volumenvermessungsalgorithmus nur in sehr seltenen Fällen aufgerufen wird. Für die Simulation in Abschnitt 4.1 wird so nur ein einziger Aufruf der Flutfüllung bei der Trennung der Blase benötigt. Der zusätzliche Zeitaufwand der Gesamtverfahrens ist somit hauptsächlich von der benötigten Laufzeit der Untersuchung

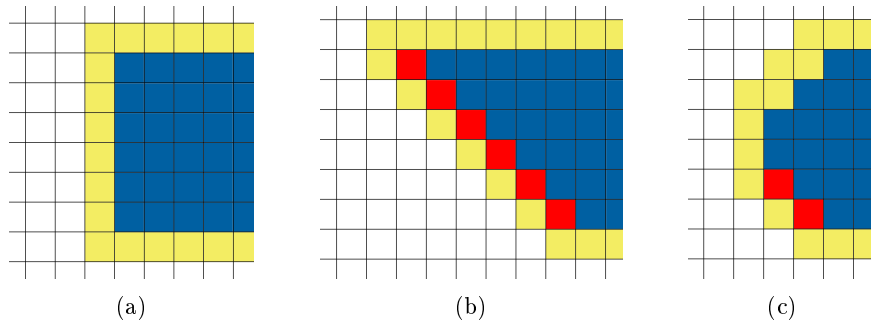


Abbildung 3.5: Geometrie des linken Randes für den besten (a), schlechtesten (b) und verbreitetsten (c) Fall von idealisierten Blasenformen, mit Zellen(rot), bei denen eine Farbvereinigung durchgeführt werden muss.

des Kriterium abhängig. Da die Überprüfung kaum Einfluss auf die Performanz der freien Oberflächen Methode nimmt, gilt dies auch für den gesamten Algorithmus.

4 Test des Algorithmus mittels Simulationen

Die korrekte Funktionsweise des implementierten Algorithmus soll in diesem Kapitel bestätigt werden. Dazu werden zwei unterschiedliche Arten von Simulationen durchgeführt. Die erste Simulation basiert auf keiner realen Ausgangssituation und dient lediglich dazu, den Unterschied des Verhaltens der freien Oberflächen Methode mit und ohne korrekter Blasentrennung aufzuzeigen. Die Zweite hingegen ist der Versuch ein reales Experiment nachzubilden, um die numerische Simulation auf qualitative Korrektheit zu überprüfen.

4.1 Längliche Blase in Einströmung

Dieser Abschnitt dient dazu den implementierten Algorithmus zur Blasentrennungserkennung und Volumenvermessung mittels einer einfachen Simulation zu testen. In der Simulation befindet sich eine längliche Blase in einem mit Flüssigkeit gefüllten Bereich. Es wirkt keine Gravitation, so dass die Blase nicht aufsteigt und sich mittig in der Region befindet. Von unten strömt Flüssigkeit mit konstanter Geschwindigkeit in einem dünnen Streifen, der sich über die gesamte Tiefe der Region erstreckt, in die Domäne ein. Dieser Strom trifft auf die Gasblase und führt dazu, dass sich die Blase nach einiger Zeit in zwei Teilblasen aufspaltet (Abbildung 4.1).

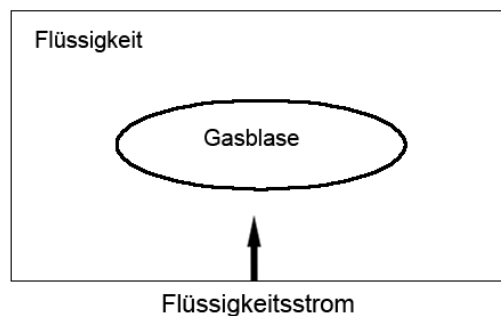


Abbildung 4.1: Längliche Gasblase in einer Flüssigkeitsströmung.

Um den Unterschied zwischen der Simulation mit und ohne einer funktionierenden Blasentrennung aufzuzeigen, sind die Ergebnisse beider Simulationen in Abbildung 4.2 gegenübergestellt. Beide Verfahrensweisen zeigen bis zum Zeitpunkt der Blasentrennung gleiche Ergebnisse. Nach der Aufspaltung der Blasen verhalten sich die beiden Methoden jedoch unterschiedlich. Im Laufe der Zeit nimmt bei der Simulation ohne implementierter Blasentrennung das Volumen der linken Blase immer mehr ab und die rechte bläht sich auf. Die Volumen der Blasen bei funktionierender Blasentrennung bleiben jedoch annähernd konstant.

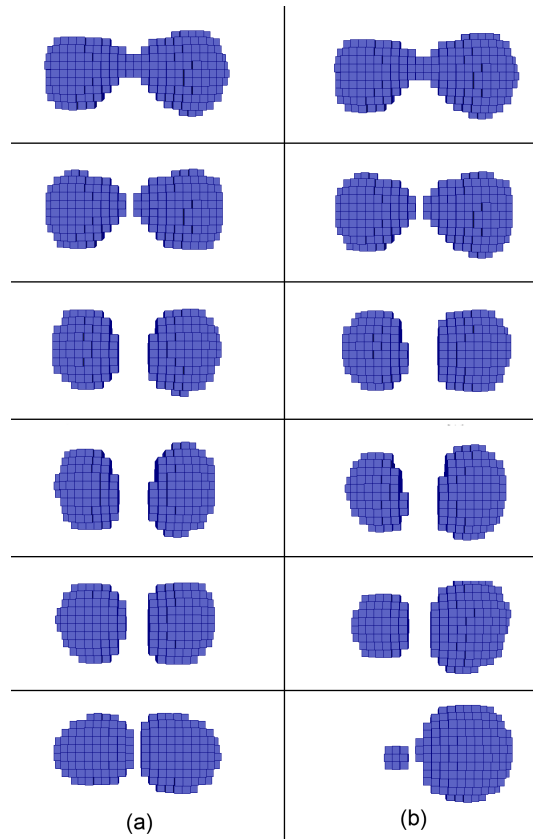


Abbildung 4.2: Verhalten der entstandenen Blasen nach der Trennung mit (a) und ohne (b) implementierter Blasentrennung.

Der Grund hierfür liegt darin, dass die Oberflächenspannung einen Druck auf die Blasen ausübt. Je größer die Oberflächenspannung desto größer der Druck auf eine Blase. Die Oberflächenspannung ist für kleinere Blasen wegen der höheren Krümmung größer. Daraus resultiert folglich auch ein höherer Druck. Die beiden Blasen sind zwar faktisch auf dem Gitter voneinander getrennt, werden aber aufgrund der nicht funktionierenden Blasentrennung als eine Blase angesehen. Also sind die Blasen für die freie Oberflächen Methode eigentlich immer noch miteinander verbunden. Da das Volumen der linken Blase in der hier vorliegenden Simulation kleiner ist als das der Rechten, herrscht auf ihr auch ein höherer Druck. Aufgrund des Druckunterschiedes und mittels der vom Verfahren wahrgenommenen Verbindung fließt das Gasvolumen von der linken Blase in die Rechte. Die linke Blase wird also immer kleiner und die Rechte um das gleiche Volumen größer, bis die Linke vollkommen verschwunden ist und die Rechte das Volumen beider Blasen enthält. Bei funktionierender Blasentrennung tritt dieses Phänomen nicht auf, weil diese Verbindung zwischen den beiden Blasen nicht mehr existiert.

Der Vergleich zwischen der Summe der vermessenen Teilvolumina und des Gesamtvolumens aus den Blasendaten zum Zeitpunkt der Trennung, das von der freien Oberflächen Methode berechnet wird, ergibt, dass beide Ergebnisse identisch sind. Der Algorithmus führt also sowohl die Blasentrennung als auch das Vermessen der Teilvolumen korrekt aus.

4.2 Blasenbildung an einer überfluteten Düse

Für die in der Einleitung erwähnten Blasensäulen ist es wichtig das Einbringen des Gases in die Flüssigkeit genauer zu untersuchen. Das Einströmen des Gases geschieht meist über Düsen, die im Boden des jeweiligen mit Flüssigkeit gefüllten Behälters, angebracht sind. Die Untersuchung der Blasenbildung an solchen überfluteten Düsen stellt einen wichtigen Bestandteil der Forschung dar und wird mithilfe von Experimenten genauer betrachtet. In diesem Abschnitt soll ein solches Experiment in einer Simulation nachgestellt und ein qualitativer Vergleich mit einem anderen Simulationsverfahren angestrebt werden.

Der experimentelle Aufbau umfasst eine einzelne Düse die am Boden eines mit Flüssigkeit gefüllten Gefäßes angebracht ist. Durch die Düse strömt Gas mit einer konstanten Geschwindigkeit ein. Die dabei entstehenden Blasen werden mittels einer Kamera aufgezeichnet und die Aufzeichnungen im Nachhinein genauer untersucht.

Die Umsetzung des Experiments in waLberla gestaltet sich aufgrund einer fehlenden Randbedingung, die es erlaubt Gas in die Domäne einströmen zu lassen, als etwas komplizierter. Um einen ähnlichen Simulationsansatz zu haben, wird auf dem Grund der Domäne eine Schicht aus Hinderniszellen geschaffen, in deren Zentrum sich ein zylindrischer Freiraum befindet. In diesen Raum wird eine exakt passende Gasblase eingebracht und das Volumen dieser Blase schrittweise erhöht. Die Blase wird dabei durch die Oberflächenspannung und den Hydrostatischen Druck am Aufsteigen gehindert. Durch dieses Vorgehen wird eine ähnliche Blasenbildung wie die an einer überflutete Düse erreicht.

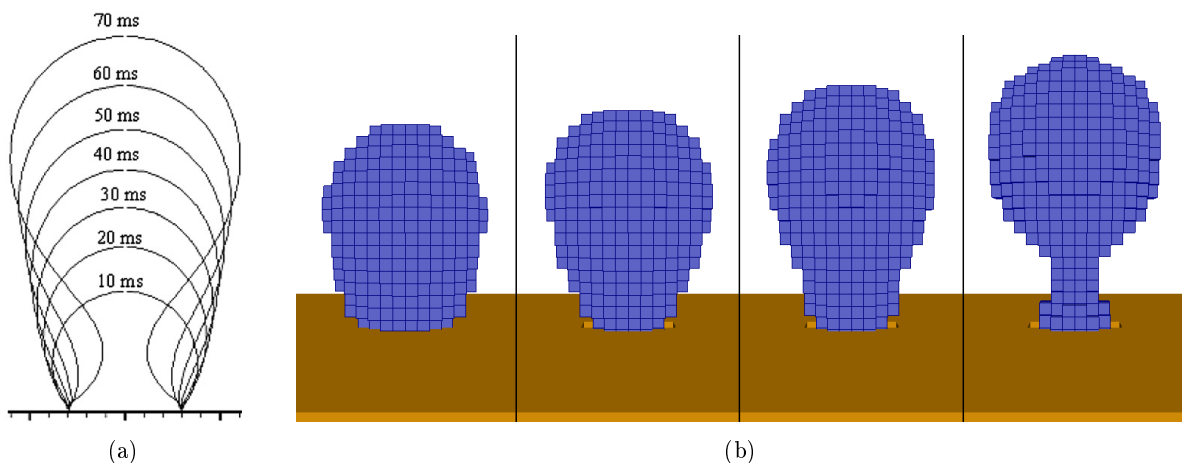


Abbildung 4.3: Blasenkonturen für die Blasenbildung mit (a) CLSVOF Methode [1] und in (b) waLberla.

Die Simulation umfasst dabei nicht nur das Blasenwachstum, sondern auch die Trennung und den Aufstieg der Blase in der Flüssigkeit. Zu beachten ist dabei, dass das Blasenwachstum möglichst langsam ablaufen sollte [3]. Bei schnellem Aufpumpen der Blase treten beim Abreißen einer Blase hohe Geschwindigkeiten auf, die zu Instabilitäten in der freien Oberflächen

Methode und zum Abbruch der Simulation führen. Volumenänderungen $\Delta V \ll 1.0$ pro Zeitschritt stellen sich dabei als sehr robust heraus.

Zur qualitativen Validierung der Simulation werden Ergebnisse, die mithilfe der combined level set and volume-of-fluid (CLSVOF) Methode entstanden sind, herangezogen [1, 2, 3]. Die Simulationen in waLBerla wurden ausschließlich mit implementierter Blasentrennung durchgeführt, da eine Simulation ohne Blasentrennung vergleichbare Ergebnisse wie in Abschnitt 4.1 liefern würde.

In Abbildung 4.3 ist die qualitative Ähnlichkeit der Konturen beider Verfahren beim Blasenwachstum gut zu erkennen.

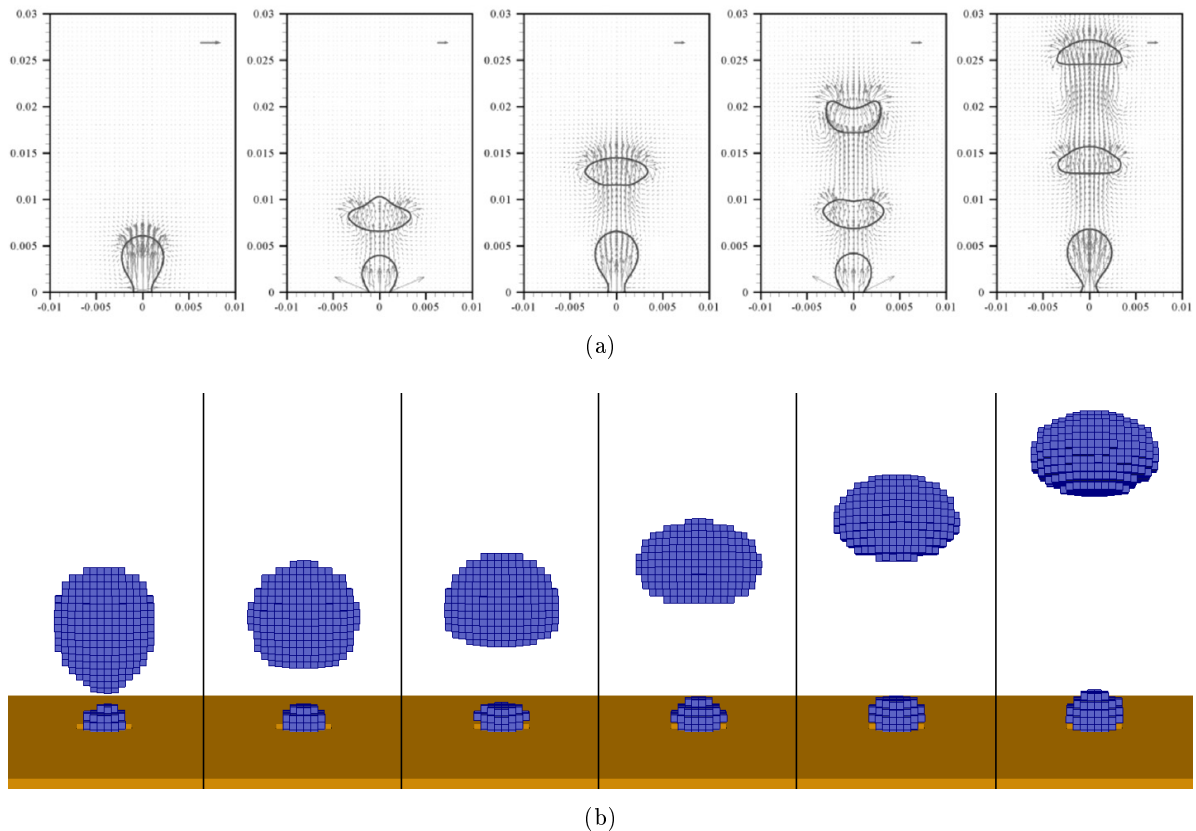


Abbildung 4.4: Blasenkonturen für die Blasen aufstieg mit (a) CLSVOF Methode [1] und in (b) waLBerla.

Nach dem Abreißen der Blasen hingegen sind Ähnlichkeiten nicht so stark ausgeprägt (Abbildung 4.4). Bei der Simulation mit der CLSVOF Methode sind die Blasen direkt nach der Blasentrennung stärker deformiert und gleichen sich erst im Laufe des Aufstiegs an die Form der waLBerla Blase an.

Die gefundenen qualitativen Ähnlichkeiten zwischen den beiden Simulationen zeigen, dass die freie Oberflächen Methode mit implementierter Blasentrennung dazu in der Lage ist, Blasenbildungen an überfluteten Düsen zu simulieren.

5 Zusammenfassung und Ausblick

In dieser Arbeit wurde die Entwicklung und Implementierung eines Algorithmus für die Blasentrennung in waLBerla gezeigt. Zuerst wurde ein Kriterium gesucht, um Blasen auf mögliche Blasentrennungen hin zu überprüfen. Diese Voruntersuchungen dienen der Erhöhung der Performanz, da die genaue Erkennung einer Blasentrennung mit hohem Aufwand verbunden ist. Die ausgewählten Blasen werden daraufhin mittels eines iterativen Flutfüllverfahrens genauer untersucht. Die Implementierung der iterativen Flutfüllung stellt den Kern dieser Arbeit dar. Mit ihrer Hilfe lassen sich sowohl Blasentrennungen erkennen, als auch gleichzeitig die entstanden Volumen der Teilblasen vermessen.

Die Performanz des Verfahrens ist dabei sehr abhängig von den vorhandenen Blasenformen und der Ausschlussquote des Kriteriums. Außerdem wurde der Algorithmus qualitativ in Simulation getestet und mit einer CLSVOF Methode verglichen. In den hier verwendeten Simulationen wurden alle Blasentrennungen korrekt erkannt. Da das Kriterium allerdings rein auf Überlegungen zur Form der Blasen direkt nach dem Trennungsvorgang beruht, ist nicht auszuschließen, dass Trennungen bei komplexeren Trennvorgängen möglicherweise nicht erkannt werden können. Eine mathematische Evaluation konnte leider nicht durchgeführt werden. Die iterative Flutfüllung kann aber auch ohne eine Voruntersuchung verwendet werden. Eine Blasentrennung wird dann in jedem Fall erkannt, da die Flutfüllung den Zusammenhang von Blaszellen auf dem Gitter untersucht und deshalb unabhängig von der Blasenform arbeitet. Die Performanz des Gesamtverfahrens bricht dadurch jedoch ein.

Die in dieser Arbeit implementierte Blasentrennung motiviert weitere Untersuchungen und Erweiterungen des Verfahrens:

- Um die freie Oberflächen Methode für die Simulation realer Experimente zu verwenden, muss die Korrektheit der Blasentrennung nachgewiesen werden. Eine quantitative Validierung kann mittels weiterer Simulationen zu überfluteten Düsen durchgeführt werden.
- Das verwendete Kriterium muss weiter getestet und validiert werden, um es entweder unter allen Umständen als tauglich zu befinden oder es zu verwerfen. Die Entwicklung und Untersuchung weiterer Kriterien sollte hierbei auch nicht außer Acht gelassen werden.
- Die Simulation größerer und feinerer Rechengitter ist sehr zeitaufwendig und kann nur durch Parallelisierung des vorgestellten Verfahrens in angemessener Zeit gelöst werden. Beim Parallelisieren des Verfahrens muss vor allem auf eine korrekte Kommunikation zwischen den Teilgebieten geachtet werden. Erstreckt sich eine Blase nämlich über mehrere Gebiete, so bedeutet eine Trennung der Blase in einem Gebiet, nicht unbedingt eine Trennung der Gesamtblase.

Literatur

- [1] BUWA, Vivek V. ; GERLACH, D. ; DURST, F. ; SCHLÜCKER, E.: Numerical simulations of bubble formation on submerged orifices: Period-1 and period-2 bubbling regimes. In: *Chemical Engineering Science* 62 (2007), S. 7119–7132
- [2] GERLACH, D. ; ALLEBORN, N. ; BUWA, V. ; DURST, F.: Numerical simulation of periodic bubble formation at a submerged orifice with constant gas flow rate. In: *Chemical Engineering Science* 62 (2007), S. 2109–2125
- [3] GERLACH, Daniel: *Analyse von kapillar-dominanten Zweiphasenströmungen mit einer kombinierten Volume-of-Fluid und Level-Set Methode*, Universität Erlangen-Nürnberg, Dissertation, 2008
- [4] GONZALEZ, Rafael C. ; WOODS, Richard E.: *Image Segmentation*. S. 689–794. In: *Digital Image Processing*, Pearson Education, Inc., 2008
- [5] POHL, Thomas: *High Performance Simulation of Free Surface Flows Using the Lattice Boltzmann Method*, Universität Erlangen-Nürnberg, Dissertation, 2007
- [6] PRATT, William K.: *Image Segmentation*. S. 579–622. In: *Digital Image Processing*, John Wiley & Sons, 2007