Neues vom FreeWiHR-Projekt: Simulation von Metallschaum mittels der Lattice-Boltzmann Methode

Nils Thürey, Thomas Pohl, Carolin Körner und Ulrich Rüde

Einleitung: Das Free WiHR-Projekt beschäftigt sich mit der Simulation von Metallschäumen. Da der Produktionsprozess dieser Schäume momentan noch nicht vollständig verstanden ist, sollen mithilfe numerischer Simulation neue Erkenntnisse gewonnen, und die Produktionsparameter, wie Druck, Gaskonzentration und die Beschaffenheit des Rohmaterials optimiert werden. Für zweidimensionale Probleme wurden bereits erfoglreich Simulationen durchgeführt [1]. Dieser Artikel beschreibt die Erweiterung des dort vorgestellten Algorithmus für drei Dimensionen – dabei ist momentan der zugrundeliegende Strömungs-Solver implementiert, der für die Simulation der Bewegung der flüssigen Metall-Phase zuständig ist.

Das Verfahren benutzt die Lattice-Boltzmann Methode (LBM)mit einem D3Q19 Gitter. Im Gegensatz zu anderen Mehrphasen-Algorithmen soll die Gas-Phase jedoch nicht vollständig als Fluid simuliert werden, da sie, abgesehen vom Druck, nur einen vernachlässigbaren Einfluss auf das flüssige Metall hat. Aufgrund der großen Gas-Volumina die in Metallschäumen auftreten, kann dadurch viel Rechenzeit eingespart werden.

Die Grundidee der LBM ist die Simulation des komplexen Verhaltens von Fluiden mit sehr einfachen Regeln, die dennoch den Navier-Stokes-Gleichungen entsprechen [6]. Die zu simulierende Flüssigkeit wird als Gitter aus identischen Zellen repräsentiert, in welchen sich die Flüssigkeitsmoleküle nur entlang bestimmter Richtungen bewegen können. In der hier verwendeten Methode sind 19 Richtungen erlaubt. Ausgehend vom Mittelpunkt eines dreidimensionalen Würfels gibt es ruhende Moleküle und Moleküle die sich in Richtung aller Nachbar-Zellen bewegen, mit Ausnahme der acht Eck-Nachbarn. Da jede der 19 Richtungen eine feste Länge hat (entweder 0,1 oder $\sqrt{2}$), sind auch die Geschwindigkeiten der Moleküle definiert. Für jede Zelle und Richtung des Gitters wird gespeichert, mit welcher Wahrscheinlichkeit es Moleküle gibt, die sich entlang dieser Richtung bewegen (die Partikel-Verteilungs-Funktionen). Addiert man alle Verteilungsfunktionen einer Zelle auf, so erhält man durch die Normierung der Zellgröße und Partikelmasse auf 1, die Dichte der Zelle. Ähnlich einfach lässt sich auch die makroskopische Geschwindigkeit einer Zelle ausrechnen. Der Algorithmus besteht nun aus zwei sich wiederholenden Schritten: dem Ström- und dem Kollisions-Schritt. Beim Ström-Schritt werden die Verteilungsfunktionen entlang ihrer Richtung in die Nachbarzellen kopiert, wie in Abbildung 1 dargestellt. Das Kopieren der Verteilungsfunktionen entspricht dabei gerade der Bewegung der Moleküle während eines Zeitschritts. Der darauf folgende Kollisions-Schritt verändert nun die Verteilungsfunktionen mit einer einfachen Formel so, dass sie auch die Richtungsund Geschwindigkeitsänderungen durch Kollisionen mit anderen Molekülen beinhalten.



Abbildung 1: Ablauf des Ström-Schritts für eine einzelne Zelle eines 2D-Gitters.

Für diese Berechnung sind die Dichte und Geschwindigkeit, die wie oben beschrieben berechnet werden können, nötig. Danach folgt wieder ein Ström-Schritt, der die Verteilungsfunktionen in die umliegenden Zellen kopiert.

Der Vorteil der LBM ist vor allem die Möglichkeit einer einfachen, effizienten und paralellen Implementierung. Weiterhin ist die Behandlung von Interaktionen mit festen Wänden als *slip*- oder *no-slip*-Randbedingung einfach und ermöglicht die Simulation von komplizierten Geometrien. Daher bietet sich der Algorithmus auch zur Repräsentation der Topologie eines Metallschaumes an.

Die freie Oberfläche: Mit der normalen LBM lassen sich nur Bereiche simulieren, die vollständig mit Flüssigkeit gefüllt sind. Zur Unterscheidung der Flüssigkeits- und Gasvolumina wurden drei verschiedene Zelltypen eingeführt:

- *Gaszellen* enthalten keine Flüssigkeit und werden in der LBM nicht berücksichtig.
- *Flüssigkeitszellen* sind komplett gefüllt und werden während der Simulation wie oben beschrieben behandelt.
- Interface-Zellen enthalten sowohl Gas als auch Flüssigkeit und erfordern deshalb eine gesonderte Behandlung.

Die Veränderung der freien Oberflächen und Effekte wie Gasdruck und Oberflächenspannung fließen nur in die Behandlung der Interface-Zellen mit ein. Für diese Zellen muss zusätzlich zu den Verteilungsfunktionen noch die Masse der Flüssigkeit in der Zelle gespeichert werden. Die Veränderung der Masse kann vor dem Ström-Schritt berechnet werden. Das Kopieren der Verteilungsfunktionen im Ström-Schritt entspricht dabei genau der Bewegung der durch die Verteilungsfunktion angegebenen Masse in die angrenzenden Zellen. Addiert man also, ausgehend von einer InterfaceZelle, die Differenz des Austauschs entlang aller Verteilungsfunktions-Richtungen mit den Nachbarzellen auf, so erhält man die Veränderung der Flüssigkeitsmasse für diesen Zeitschritt. Da diese Massenveränderung nur für Interface-Zellen durchgeführt wird, ist es für die Massenerhaltung wichtig, dass keine Fluid-Zellen mit angrenzenden Gas-Zellen existieren. Im Laufe einer Simulation können sich die Interface-Zellen komplett leeren oder füllen, was einer Masse von 0 bzw. einer Masse gleich einer vollen Zelle entspricht. Daraufhin wird der Typ der Interface-Zelle entsprechend geändert, und die umliegenden Zellen müssen so initialisiert werden, dass wieder eine zusammenhängende Schicht von Interface-Zellen entsteht.

Für eine solche Behandlung der Interface-Zellen wären Verteilungsfunktionen aus der Gas-Phase nötig, die im vorgestellten Algorithmus jedoch nicht existieren. Es ist deshalb eine Rekonstruktion dieser Partikel-Verteilungsfunktionen nötig, wofür nur Dichte und Geschwindigkeit des Gases bekannt sein müssen. Die Dichte der Atmosphäre wird auf 1 gesetzt, die einer Blase kann anhand des Volumes und der Masse der Blase errechnet werden. Für die Geschwindigkeit des Gases wird angenommen, dass es, aufgrund des großen Viskositätsunterschiedes, nach je-



Abbildung 2: Beispiel einer Flüssigkeitskonfiguration und ihre Repräsentation im Gitter.

dem Zeitschritt dieselbe Geschwindigkeit wie die Flüssigkeit hat. Diese Rekonstruktion wird für alle aus Gas-Zellen kommenden Verteilungsfunktionen vorgenommen, wodurch alle Interface-Zellen wieder einen vollen Satz an Verteilungsfunktionen besizten und wie oben beschrieben behandelt werden können.

Oberflächenspannung:

Zur Berücksichtigung der Oberflächenspannung muss die Krümmung der Oberfläche bei jeder Interface-Zelle ermittelt werden. In 2D lässt sich die Krümmung anhand von drei Punkten auf einer Kurve leicht ermitteln, indem man den Kreis berechnet, der alle drei Punkte berührt. Damit ist der Radius des Kreises R und somit auch die Krümmung $\sigma = 1/R$ bekannt. Die Krümmung einer Oberfläche kann entlang einer beliebigen Kurve berechnet werden; für die Metallschaumsimulation werden die durch zwei rechtwinklige Normalschnitte entstehenden Kurven zur Krümmungsberechnung herangezogen. Die gemittelte Krümmung der beiden Kurven wird dann in der Simulation als Druckveränderung bei der Rekonstruktion der Verteilungsfunktionen für Interface-Zellen verwendet. Dadurch wird je nach Vorzeichen der Krümmung eine Kraft in Richtung des Fluids bzw. des Gases auf die Oberfläche ausgeübt.

Um die drei benötigten Punkte auf jeder Kurve zu finden wird der Marching-Cubes-Algorithmus verwendet [3]. Er stammt ursprünglich aus der Visualisierung und wird zur Darstellung von Isoflächen in Skalar-Feldern verwendet. Ausgehend von einem Würfel mit acht Skalar-Werten werden die Schnittpunkte der Isofläche mit den Kanten des Würfels berechnet (siehe Abbildung 3), indem angenommen wird, dass sich die Werte des Feldes entlang jed-Kante nur mit einem linearen Ver- \mathbf{er} lauf ändern. Zur Visualisierung der Isofläche kann aus den Schnittpunkten ein zusammenhängendes Dreiecksnetz generiert werden. Dies ist für die Krümmungsberechnung jedoch nicht notwendig. Um Punkte auf der Flüssigkeitsoberfläche zu erhalten, werden die Füllstände der Lattice-Boltzmann-Zellen als Skalar-Feld verwendet. Für jede Interface-Zelle werden dann in zwei geeigneten Koordinatenebenen Punkte auf der Isofläche in der Nachbarschaft der Zelle gesucht und zur Krümmungsberechnung verwendet. Falls mehr als drei Punkte gefunden werden, müssen einzelne wieder verworfen werden.

Ergebnisse und Ausblick: Neben Standard-Testproblemen wie dem Breaking-Dam-Problem, sind für Metallschäume vor allem Probleme mit aufsteigenden und koaleszierenden Blasen interessant. Abbildung 4 zeigt drei Bildreihen von aufsteigenden Blasen verschiedener Größe. Während die kleinste Blase in der oberen Reihe im letzten Zeitschritt Oberfläche noch nicht durchbrochen die hat, haben die größeren Blasen höhere Aufstiegsgeschwindigkeiten und erreichen die Oberfläche. Ein komplizierteres Problem mit sechs aufsteigenden Blasen und Koaleszenz ist in Abbildung 5 gezeigt. Die Visualisierungen wurden in beiden Fällen mit einem eigenen Raytracer erstellt, der die vom Marching-Cubes-Algorithmus berechneten Oberflächen benutzt [2].

Die hier vorgestellte LBM erlaubt die Simulation und Visualisierung des dynamischen Verhaltens von Flüssigkeiten mit Oberflächenspannung und deren Interaktion mit Gasen. Wie vor allem an den zurückbleibenden Gaseinschlüssen in den Bildern aus Abbildung 5 zu sehen ist, können die Randbedingungen für die freie Oberfläche jedoch noch verbessert werden. Vor allem bei größeren Geschwindkeiten kann es vorkommen, dass einzelne Interface-Zellen nicht schnell genug geleert oder gefüllt werden können. Zur Zeit wird die Simulation erweitert, um auch die Gasdiffusion im flüssigen Metall und dessen Erstarrung zu berücksichtigen. Aufgrund der



Abbildung 3: Der Marching-Cubes-Algorithmus.

großen benötigten Rechengebiete ist auch eine optimierte, MPI- und OpenMP-paralelle C-Version für den Hitachi SR8000 Computer in München in Arbeit. Animationen der hier vorgestellten Ergebnisse und weitere Bilder können auf den folgenden Internetseiten heruntergeladen werden [4, 5].

Literatur

- M. Arnold, M. Thies, C. Körner, and R.F. Singer. Experimental and numerical investigation of the formation of metal foam. *Materialsweek*, 2000.
- [2] Henrik W. Jensen. Realistic Image Synthesis Using Photon Mapping. A K Peters Ltd, 2001.
- [3] William Lorensen and Harvey Cline. Marching cubes: A high resolution 3d surface reconstruction algorithm. In *Comput*er Graphics Vol. 21, No. 4, pages 163–169, August 1987.
- [4] Thomas Pohl. FreeWIHR, 2003. http://www10.informatik.uni-erlangen.de/en/Research/Projects/FreeWiHR.
- [5] Nils Thürey. Fluid simulation with LBM, 2003. http://www.ntoken.com/fluid.
- [6] Dieter A. Wolf-Gladrow. Lattice-Gas Cellular Automata and Lattice Boltzmann Models. Springer, 2000.



Abbildung 4: Drei aufsteigende Blasen unterschiedlicher Größe.



Abbildung 5: Sechs aufsteigende Blasen in einem Container.