Mesoscopic simulations of multiphase flows

Harting, J.D.R.

Published: 01/01/2008

Document Version
Publisher’s PDF, also known as Version of Record (includes final page, issue and volume numbers)

Please check the document version of this publication:

• A submitted manuscript is the author's version of the article upon submission and before peer-review. There can be important differences between the submitted version and the official published version of record. People interested in the research are advised to contact the author for the final version of the publication, or visit the DOI to the publisher's website.
• The final author version and the galley proof are versions of the publication after peer review.
• The final published version features the final layout of the paper including the volume, issue and page numbers.

Link to publication

Citation for published version (APA):
Mesoscopic Simulations of Multiphase Flows
Contents

I Zusammenfassung in deutscher Sprache 1

1 Einführung 3

2 Anwendungen 5
   1 Gitter-Boltzmann Simulationen von Mesosphasen und Mikroemulsionen 5
   2 Simulation von Anwendungen aus der Mikrofluidik 7
   3 Simulation teilchenbeladener Strömungen 9

3 Diskussion und Ausblick 13

II Overview and summary 15

1 Introduction 17

2 Mesoscopic modeling of multiphase flows 21
   1 Motivation 21
   2 Mesoscopic simulation methods for fluid dynamics 24
      2.1 Dissipative particle dynamics (DPD) 24
      2.2 Stochastic rotation dynamics (SRD) 24
      2.3 Lattice gas cellular automata (LGCA) 25
      2.4 The lattice Boltzmann method (LBM) 26
   3 Mesoscopic simulation methods for particle-laden flows 30
      3.1 Particle-particle interactions 30
      3.2 Brownian dynamics 32
      3.3 Stochastic rotation dynamics 32
      3.4 Lattice Boltzmann 33
      3.5 Conventional Navier-Stokes solvers 34
      3.6 Inertia-less coarse-grained gas method 34

3 Lattice Boltzmann simulations of mesophases and microemulsions 35
IV CONTENTS

4 Simulations of microfluidic applications: probing the boundary condition 41
5 Simulations of particle-laden flows 47
6 Discussion and Outlook 53

III Mesophases and microemulsions 59

1 Steering in computational science: mesoscale modelling and simulation
   J. Chin, J. Harting, S. Jha, P. V. Covenev, A. R. Porter, and S. M. Pickles

2 Large-scale grid-enabled lattice-Boltzmann simulations of complex fluid flow in porous media and under shear
   J. Harting, M. Venturoli, and P. V. Covenev

3 Detection and tracking of defects in the gyroid mesophase
   J. Harting, M. Harvey, J. Chin, and P. V. Covenev

4 Large-scale lattice Boltzmann simulations of complex fluids: advances through the advent of computational grids
   J. Harting, J. Chin, M. Venturoli, and P. V. Covenev

5 Stress response and structural transitions in sheared gyroidal and lamellar amphiphilic mesophases: lattice-Boltzmann simulations
   N. González-Segredo, J. Harting, G. Giupponi, and P. V. Covenev

6 Emergence of rheological properties in lattice Boltzmann simulations of gyroid mesophases
   G. Giupponi, J. Harting, and P. V. Covenev

7 Structural transitions and arrest of domain growth in sheared binary immiscible fluids and microemulsions
   J. Harting, G. Giupponi, and P. V. Covenev
## Microfluidic applications

1. **Lattice Boltzmann simulations of apparent slip in hydrophobic microchannels**  
   J. Harting, C. Kunert, and H. J. Herrmann  

2. **On the effect of surfactant adsorption and viscosity change on apparent slip in hydrophobic microchannels**  
   C. Kunert and J. Harting  

3. **Roughness induced boundary slip in microchannel flows**  
   C. Kunert and J. Harting  

4. **Simulation of fluid flow in hydrophobic rough microchannels**  
   C. Kunert and J. Harting  

5. **Slip flow over structured surfaces with entrapped microbubbles**  
   J. Hyväläluoma and J. Harting  

## Simulation of particle-laden flows

1. **Transport phenomena and structuring in shear flow of suspensions near solid walls**  
   A. Komnik, J. Harting, and H. J. Herrmann  

2. **Anomalous distribution functions in sheared suspensions**  
   J. Harting, H. J. Herrmann, and E. Ben-Naim  

3. **Simulation of claylike colloids**  
   M. Hecht, J. Harting, T. Ihle, and H. J. Herrmann  

4. **Shear viscosity of claylike colloids in computer simulations and experiments**  
   M. Hecht, J. Harting, M. Bier, J. Reinshagen, and H. J. Herrmann  

5. **Stability diagram for dense suspensions of model colloidal Al₂O₃-particles in shear flow**  
   M. Hecht, J. Harting, and H. J. Herrmann  
<table>
<thead>
<tr>
<th>Section</th>
<th>Title</th>
<th>Authors</th>
<th>Source</th>
<th>Page</th>
</tr>
</thead>
</table>

**Bibliography** 323

**Acknowledgments** 355
Zusammenfassung in deutscher Sprache
Einführung


Mikroskopische Simulationsmethoden beschreiben in der Regel jedes einzelne Molekül oder Atom in einem System und basieren auf klassischer Molekulardynamik. Für größere Probleme bedeutet dies, dass der Rechenaufwand stark ansteigt und dadurch die derzeit simulierbaren Längen- und Zeitskalen auf den Bereich von Nanometern und Nanosekunden limitiert sind. Selbst die derzeit schnellsten Großrechner stellen nicht ausreichend Rechenleistung bereit, um dieses Problem zu lösen. Im Gegensatz zu Simulationsmethoden, die auf der Mikroskala operieren, haben makroskopische Ansätze oft nicht das Potenzial, die mikroskopischen Details korrekt zu beschreiben, die dem charakteristischen Verhalten eines Systems zugrunde liegen.

Daher existiert ein großes Interesse an sogenannten mesoskaligen Modellen. Diese Methoden vereinfachen die atomaren oder molekularen Details unter Erhalt jener physikalischen Eigenschaften, die für die Beschreibung der relevanten Vorgänge notwendig sind. Sie sind geeignet, Systeme auf Längenskalen zwischen einigen Nanometern und einigen Millimetern, sowie Prozesse auf verschiedenen Zeit- und Längenskalen zu beschreiben.

In der Regel machen sich mesoskopische Simulationsmethoden die praktische Tatsache zu Nutze, dass es nicht notwendig ist, jedes individuelle Molekül einer Flüssigkeit zu verfolgen, um ihr hydrodynamisches Verhalten zu beschreiben. Tatsächlich reicht es aus, eine große Gruppe von Molekülen oder "Flüssigkeitspakete" zu betrachten und diese wie einzelne Partikel zu beschreiben. Anstatt nun für jedes Paar aus Molekülen die entsprechenden Wechselwirkungen zu berechnen, kann dies für die mesoskopischen Teilchengruppen geschehen. Unter der Voraussetzung, dass die physikalischen Gesetze richtig implementiert sind, ist das Verhalten des resultierenden Systems vergleichbar mit einem System, bei dem die einzelnen Moleküle individuell modelliert werden [37, 135].

Vor- und Nachteile. Daher muss vor der Simulation genau abgewägt werden, welche Methode am besten geeignet ist. Nur so können die Rechnerressourcen so effizient wie möglich genutzt werden.

Sollen in Flüssigkeit suspendierte Teilchen modelliert werden, verwendet man mesoskopische oder Kontinuumslöser für die Flüssigkeit, um die zeitabhängige langreichweitige Hydrodynamik korrekt zu beschreiben. Die Teilchenbewegung wird dann mit Hilfe einer Molekulardynamik simuliert, die auch für die Einbeziehung von Teilchen-Teilchen Wechselwirkungen verantwortlich ist. Diese entstehen beispielsweise durch Oberflächenladungen.


Im Rahmen der vorliegenden Arbeit werden verschiedene mesoskopische Simulationsmethoden angewendet, um Fragestellungen aus drei unterschiedlichen Gebieten zu behandeln:


Für die ersten beiden Gebiete verwenden wir die Gitter-Boltzmann Methode, da hier besonders einfach Wechselwirkungen zwischen verschiedenen Flüssigkeiten und komplexen Randbedingungen implementiert werden können. Weiterhin ist diese Methode intrinsisch parallel, sodass beispielsweise die Simulation sehr großer amphiphiler Systeme möglich ist. Im Fall der Mikrofluidik können wir unsere Simulationen aus demselben Grund direkt den Experimenten gegenüber stellen.

Anwendungen

1 Gitter-Boltzmann Simulationen von Mesophasen und Mikroemulsionen


I-2. Anwendungen


Kapitel III-4, “Large-scale lattice Boltzmann simulations of complex fluids: advances through the advent of computational grids”, fasst die Ergebnisse, die im Rahmen des RealityGrid Projektes und des TeraGyroid Experiments erlangt wurden, zusammen. Insbesondere werden sowohl die technischen als auch die wissenschaftlichen Ergebnisse beleuchtet.


Das letzte Kapitel in Teil III, “Structural transitions and arrest of domain growth in sheared binary immiscible fluids and microemulsions”, stellt systematische Studien des Einflusses
von Amphiphilen auf die spinodale Entmischung zweier nicht mischbarer Flüssigkeiten dar.
Wir weisen die Möglichkeit der Bildung von stabilen Mikroemulsionen und den linearen Zu-
sammenhang der Domänengröße mit der Lösungsmittelkonzentration nach. Weiterhin zeigen
wir die Bildung einer lamellaren Phase unter konstanter Scherung und von zylinderförmigen
Strukturen unter oszillierender Scherung.

2 Simulation von Anwendungen aus der Mikrofluidik

In den letzten Jahrzehnten hat die Miniaturisierung technischer Geräte große Fortschritte
gemacht. Insbesondere wurden in den 1980er Jahren sogenannte Mikroelektromechanische
Systeme (MEMS) entwickelt. Bekannte Beispiele für solche MEMS sind Beschleunigungs-
sensoren in Airbags oder Druckköpfe in Tintenstrahldruckern. Diese neuartigen Geräte ver-
wenden Strömungen von Flüssigkeiten und Gasen unter bisher unerforschten Bedingungen.
Dadurch entstand in den 1990er Jahren eine neue wissenschaftliche Disziplin, die Mikrofluid-
dik [447].

Neben diesen „praktischen Anwendungen“ hat die Mikrofluidik das Potenzial, fundamentale
Antworten auf Fragen der Physik zu beantworten. Hierzu zählen beispielsweise Fragen zum
Verhalten einzelner Moleküle oder Teilchen in Strömungen [426] oder zur Gültigkeit der
Haftbedingung [272]. Letzteres ist von besonderem Interesse für die vorliegende Arbeit, da die Validierung der Haftbedingung der Hintergrund für Kapitel IV-1 bis IV-5 ist.

Die Reynoldszahlen in Mikrofluidiksystemen sind in der Regel klein. Zudem sind die Kanä-
le, durch die die Flüssigkeiten fließen, sehr klein, was ein großes Verhältnis von Oberfläche
to Volumen zur Folge hat. Daher dürfen Oberflächeneffekte, wie z.B. die Benetzung oder
Oberflächenladungen, nicht vernachlässigt werden. Auch die mittlere freie Weglänge \( \lambda \)
 eines Moleküls kann von ähnlicher Größe sein wie die charakteristische Längenskala \( l \) des
Systems. Für Gase kann dieser Einfluss mit der Knudsen Zahl

\[
Kn = \frac{\lambda}{l}.
\]

charakterisiert werden [245]. Wird diese Zahl zu groß, werden Verarmungseffekte dominant
und die Navier-Stokes Gleichungen müssen durch eine passendere Beschreibung ersetzt wer-
den. Insbesondere kommt es bei höheren Knudsen Zahlen zu einem Wandschlupf.

Während die Knudsen Zahl einen guten Anhaltspunkt gibt, wann mit Verarmungseffekten
in Gasströmungen zu rechnen ist, würde man bei Flüssigkeiten annehmen, dass die relative
Geschwindigkeit einer Strömung an einer Oberfläche immer verschwindet. Diese Annahme
kann als eines der Kernkonzepte der Fluidmechanik bezeichnet werden. Vor einigen Jahrhun-
derten war diese These allerdings nicht so anerkannt wie heute. Der wesentliche Grund der
generellen Anerkennung der Haftbedingung war das Fehlen von experimentellen Beob-
achtungen, die diese widerlegten. Daher verwundert es nicht, dass bereits Navier 1823 eine
Randbedingung vorschlug, bei der die Geschwindigkeit an der Oberfläche proportional zu
der Sherrate ist [326]. Es gilt (bei \( x = x_0 \))

\[
v_z(x_0) = \beta \frac{\partial v_z(x)}{\partial x},
\]

wobei die Proportionalitätskonstante \( \beta \) die sogenannte “Schlupflänge” ist. In den letzten Jah-
ren wurde es möglich, sehr gut kontrollierte Experimente durchzuführen, die einen Wand-
schlupf in Mikrokanälen finden. Seitdem wurden hauptsächlich experimentelle [21, 41,


Unsere Forschung zu Mikrofluidsystemen und insbesondere zum Effekt des Wandschlupfes ist in Teil IV dargestellt.


Wenn typische Längenskalen eines Experimentes in der Größenordnung der Oberflächenrauhigkeit sind, kann der Einfluss der Rauhigkeit nicht mehr vernachlässigt werden. Diesen Effekt untersuchen wir in Kapitel IV-3, “Roughness induced boundary slip in microchannel flows”. Wir zeigen, dass ein gemessener Schlupf seinen Ursprung in der falschen Annah-


3 Simulation teilchenbeladener Strömungen


Zusätzlich zu den oben beschriebenen Wechselwirkungen, können gelöste Teilchen auch über elektrostatische Abstoßung, von der Waals Anziehung oder Verarmungskräfte miteinander

Im Rahmen dieser Arbeit sind besonders tonartige Kolloide interessant, wobei hier modellhaft Al$_2$O$_3$ Pulver in Wasser verwendet wird. Unsere Kollegen führen im Rahmen der DFG Forschergruppe “Peloide” makroskopische Experimente mit diesen Suspensionen durch. Da das makroskopische Verhalten durch die Teilchen-Teilchen Wechselwirkungen auf mikroskopischer Ebene direkt beeinflusst wird, tragen wir zum Verständnis dieser Systeme auf mikroskopischer Skala bei. Mit Hilfe von Computersimulationen sind wir in der Lage, mikroskopische Strukturen von Sedimenten zu reproduzieren und den direkten Einfluss einer Änderung von pH-Wert oder Salzkonzentration zu studieren.

Optische Pinzetten eröffnen die Möglichkeit, die Dynamik kolloidaler Systeme auf sehr kontrollierte Weise zu studieren. Experimentell wird ein Laserstrahl verwendet, um ein Kolloid in seinem Fokus zu “fangen”. Optische Pinzetten haben den Vorteil, dass sie es ermöglichen, Kolloide mit festgelegter Geschwindigkeit zu bewegen und Kräfte in Piconewton Auflösung zu messen [179, 333]. In Computersimulationen können Experimente mit solche Pinzetten in derselben Art wie Suspensionen beschrieben werden, d.h. die durch die optische Pinzette auf ein Teilchen ausgeübte Kraft kann zusätzlich in den Molekulardynamiklöscher eingeführt werden.


Wenn die Teilchen kleiner werden, können brownsche Fluktuationen nicht mehr vernachlässigt werden und die Gitter-Boltzmann Methode ist in ihrer einfachen Form nicht mehr die Methode der Wahl. Stattdessen wenden wir hier die stochastische Rotationsdynamik an, um dichte Suspensionen von Al$_2$O$_3$ Teilchen in Wasser zu simulieren.

In Kapitel V-3, “Simulation of claylike colloids”, führen wir eine Simulationsmethode ein, die neben dem Flüssigkeitslöser auch einen Molekulardynamiklöser enthält, der van der Waals und Coulomb Kräfte beschreiben kann. Weiterhin verwenden wir ein von uns entwickeltes Skalierungsschema, um die hydrodynamischen Parameter so zu wählen, dass der Rechenaufwand möglichst gering wird. Zu den hier präsentierten Ergebnissen zählen Sedi-
mentationsgeschwindigkeiten, Geschwindigkeitsverteilungen, Diffusionskonstanten, sowie Korrelationsfunktionen.


I-2. Anwendungen
Diskussion und Ausblick

In diesem Kapitel sollen eine kurze Diskussion der Ergebnisse und daraus folgende interessante Fragestellungen für zukünftige Forschungsvorhaben erläutert werden.


Die Gitter-Boltzmann Methode ist nicht in der Lage, Materialeigenschaften atomar aufzulösen. Allerdings sind es gerade diese Eigenschaften, die oftmals für einen gemessenen Schlupf verantwortlich sind. Daher sollen gekoppelte Simulationsmethoden entwickelt werden, die die Oberfläche und die Flüssigkeit in ihrer direkten Nähe mit Molekulardynamik beschreiben und die weniger relevanten Bereiche mit Hilfe eines mesoskopischen Ansatzes modellieren.

Weiterhin sind wir an der Entwicklung und Optimierung von Mikromischern interessiert. Mit Hilfe der Gitter-Boltzmann Methode sollen solche Bauteile simuliert werden und wir entwickeln neuartige Techniken, um die Qualität solcher Mischer zu bestimmen.


Insbesondere in biologischen Systemen hat man es in der Regel nicht mit harten Kugeln, sondern mit deformierbaren, nicht kugelförmigen Teilchen zu tun, die in einem Lösungsmittel
Overview and summary
Introduction

Numerous systems occurring in our daily life can be considered as “multiphase flows”. With this term, mixtures of different liquids, liquids and gases, or a solid phase consisting of individual particles solved in a fluid can be described. Examples for such systems are ubiquitous: for a shower in the morning, we use shampoo or soap which contain amphiphilic molecules reducing the surface tension of water. Tooth paste consists of fine particles in a liquid. The same is true for the morning coffee containing finely ground coffee beans solved in water, or the paint used to color the walls of our flat. Leaving home on a rainy day, we might find splashy streets covered with mud consisting of soil particles and water. Being back in the evening, a glass of wine containing alcohol and water is a typical example for a mixture of liquids again. Extreme examples can be found on beaches and in deserts: sand dunes have formed due to the wind moving around the sand.

Many of these systems have in common that their complex macroscopic behavior is determined by microscopic interactions. A typical example which is of interest for the current thesis is a mixture of two immiscible liquids like oil and water. The attempt to mix these liquids in a container by shaking it is destined to fail since a phase separation starts just thereafter. By adding so-called amphiphiles, i.e., molecules that arrange at the interface between oil and water and which lower the surface tension, this mixture can be stabilized. Then, the two otherwise immiscible liquids can form an almost homogeneous system, where the macroscopically observable properties are determined by the microscopic interactions between the amphiphiles and the oil and water molecules.

There are numerous examples where such mixtures are kept in confined geometries as it is the case in industrial production processes or microfluidic applications. Here, also the interaction between the liquids and the confining geometries itself may be of relevance. For example, the macroscopically observed flow velocity of a fluid in a channel depends on the surface structure of the channel itself, i.e., a smooth channel will be less resistant to the flow than a channel with very rough surfaces. Also, chemical reactions taking place in the vicinity of the surface depend to a large extent on the particular materials used for the channel and the kind of fluid mixture used.

Another very descriptive example of a system containing a solid phase and a liquid phase can be observed in the kitchen and easily be reproduced by the reader: due to the microscopic interactions between starch molecules in cornflour, a mixture of cornflour and water becomes more difficult to stir if one stirs it quickly.

These examples demonstrate that improving the understanding of such ubiquitous systems is a problem of general interest. Experimentally, it is often easy to study their macroscopic behavior. Microscopic scales, however, are either very hard or even impossible to be investigated in the required detail. As an example, let us consider the mixture of cornflour and water again: it is easy to determine macroscopic observables like the temperature of the system or the volume concentration of cornflour and the dependence of the stirring velocity on the resistance of the mixture against this disturbance. However, for a detailed understanding, the microscopic interactions between the starch molecules need to be investigated. Experimentally, these interactions are not easily accessible and although they can be measured in some
cases, experimental approaches often do not allow to tune a single parameter only. For example, modifying the interaction strength between the molecules by introducing some additive to the solution might also change the properties of the solving water itself.

Computer simulations allow to model the microscopic interactions in a detailed way and also provide the possibility to change individual parameters only without influencing any others. However, typical microscopic models consider every atom or molecule within a system and are based on classical molecular dynamics. Unfortunately, for increasing system sizes, such models become very complex and computationally demanding. Thus, even on today’s fastest supercomputers, typical length scales are restricted to nanometers and timescales to the order of nanoseconds. On the other hand, macroscopic models operate on relevant scales, but are not able to resolve the molecular details needed to describe processes operating on several length and time scales which give rise to the characteristic macroscopic behavior.

A possible solution of this problem is to coarse grain most of the atomic or molecular details but to retain the essential physics needed to describe the phenomena of interest. Such an approach is followed by so-called mesoscale methods which are intended to model systems at intermediate length scales between several nanometres and a few millimetres. Relevant physical processes can operate on multiple length and time scales. A typical example for such an approach is the description of a fluid: it is not necessary to keep track of every single molecule of a fluid in order to reproduce its hydrodynamic behavior. Instead, very large numbers of molecules are grouped into Lagrangian “packets of fluid”. Interactions can be coarse grained as well and take place between such mesosopic groupings of particles, rather than their constituent molecules. Provided that restrictions like isotropy and conservation laws are taken into account properly, the resulting large scale behavior is similar to the one expected from a microscopic description. Typical examples for such models are “dissipative particle dynamics”, “stochastic rotation dynamics”, “lattice gas cellular automata”, or the “lattice Boltzmann method”. As common methods operating on micro- or macroscales, all mesoscale methods have their inherent strengths and drawbacks. For example, while some of them can easily treat the effect of thermal fluctuations, others only include those after substantial extensions of the model. Thus, for every problem to be simulated, one has to choose beforehand the best suitable simulation method.

If the system of interest is not only a mixture of different liquids or gases, but also contains solid constituents, these models need to be extended. In order to simulate for example colloidal suspensions, the colloidal particles and their electrostatic interactions are described by a molecular dynamics algorithm. The fluid solvent, however, is treated by one of the approaches above. By applying suitable coupling mechanisms, hybrid methods are obtained which allow the proper integration of long-range hydrodynamic interactions between the colloidal particles.

Even though, these recent methods allow to reduce the computational effort by orders of magnitude, modeling three dimensional systems in a realistic way still is a typical high performance computing problem. Thus, it is of particular importance, not only to choose the method which is best suited to treat the problem of interest, but also to take care of efficient code implementation and parallelization. In the framework of the current thesis, a strong focus is given to the importance of high performance computing and also to the improvements expected from the advent of new techniques appearing in this field. These are “computational steering” and the concept of “grid computing”. “Computational steering” allows to interact with remotely running simulations to change parameters and to observe the state of the simulation without interfering with it. The concept of “computational grids” has become popular during recent years and can be seen as a modern way of distributed computing – with the
major extension in that it enables the transparent sharing and collective use of resources such as CPUs, data storage facilities, or visualization engines.

Within the current thesis, different mesoscopic simulation methods are applied in order to simulate known effects and recent experimental results from three different subjects. The work presented can be summarized as follows:

- Lattice Boltzmann simulations of amphiphilic fluid mixtures including microemulsions and the self-assembly of cubic mesophases and in particular the “gyroid” mesophase. These “gyroid” mesophases are obtained if amphiphiles force a fluid mixture into an equilibrium structure, where a layer of molecules forms a single continuous and minimal surface dividing space into two multiply connected and interpenetrating volumes. In our case, the volumes contain two simple immiscible fluid species.

- Lattice Boltzmann simulations of microfluidic applications. In particular, we investigate an experimentally observed violation of the no-slip boundary condition and study the influence of the interaction between fluids and confining geometries.

- The simulation of particle-laden flows. Here, we utilize coupled lattice Boltzmann and molecular dynamics algorithms to study hard sphere suspensions. For the treatment of colloidal suspensions, where electrostatic interactions and Brownian motion are important, the “stochastic rotation dynamics” technique is more suitable than the lattice Boltzmann method. Brownian dynamics simulations are applied to model experiments where optical tweezers are used to drag a large colloid through a polymer solution.

As stated above, a careful selection of the most suitable simulation method is essential. For the first two topics, we choose the lattice Boltzmann method. Here, it is particularly easy to implement interactions between different species and to use complex boundary conditions. Also, it is inherent of this method that very efficient parallel implementations are possible. Since our implementations scale on thousands of processors, we are able to reach length and time scales which can be directly compared to experimental data. In the case of amphiphilic systems, we are able to reproduce known behavior of microemulsions and study the properties of the cubic gyroid mesophase in detail. For microfluidic systems, we are able to resolve the full experimentally obtained data sets without compromising on the resolution of our simulation. This way, we can obtain results which are directly comparable to experiments.

For the simulation of particle-laden flows, the picture is less clear. Until recently, the correct treatment of Brownian fluctuations within the lattice Boltzmann framework was still a matter of discussion. Thus, it is mainly used for non-Brownian suspensions. In this thesis, the system of consideration is a hard sphere suspension, e.g., as a model system of glass spheres in sugar solution, where we reproduce experimentally observed structuring effects under creeping shear flow. Also, the method has demonstrated to be a good candidate to investigate statistical properties of such suspensions. For example, the shape of the probability distribution function of particle velocities can be investigated.

Stochastic rotation dynamics contains thermal noise intrinsically and must not be used if the lengths scales of the solved particles are of the order of hundreds of micrometers or even millimeters. In such cases, a lot of averaging is needed causing unacceptably high demands of computing time. Therefore, this method is applied to study the properties of colloidal suspensions containing sub-micrometer sized particles suspended in water as a model system for clay and silt. We investigate the influence of salt concentration and pH-value of the solvent. Changing these parameters can cause the particles to form agglomerates or repulsive structures.
If the hydrodynamic interactions are less important as it is the case in very dense systems, a simplified approach to model a suspension can be chosen. We use Brownian dynamics. Here, only a Stokes damping and random noise are added to a molecular dynamics algorithm covering the motion of the particles. We find this method to be a suitable candidate to model optical tweezer measurements in dense polymer suspensions.

This thesis is of cumulative type and organized as follows. The current part II, “Overview and summary”, contains an introduction into the field of research and starts by focusing on typical mesoscale simulation techniques for fluids and for particle-laden flows in chapter II-2. The following chapters provide an insight into the physical problems of interest. Thus, in chapter II-3, a description of our lattice Boltzmann simulations of mesophases and microemulsions is given. We focus on amphiphilic fluid mixtures and a possible self-organisation of amphiphilic mesophases. Also, we relate our description to the improvements of our work due to emerging new technical achievements such as computational grids. Further, a summary of our research in this field which relates to the scientific papers presented in part III is provided.

Chapter II-4, “Simulations of microfluidic applications: probing the boundary condition”, gives a short introduction into the field of microfluidics. The main focus is laid on the validation of the no-slip boundary condition and experiments which have found a violation of this concept on microscopic scales. We try to answer questions arising in this context by means of lattice Boltzmann simulations. This chapter closes by providing a summary of our research papers which are presented in part IV.

The following chapter II-5, “Simulations of particle-laden flows”, gives examples of systems, which are in the scope of this thesis colloidal or hard-sphere suspensions, and introduces our applications of interest, namely hard spheres under creeping shear or AL₂O₃ powder suspended in water as a model system for soil or silt. It follows a summary of our research performed in this field and a short introduction to the related papers presented in part V.

We finalize the introductory part with a discussion of the results obtained so far and provide an outlook on future projects and already ongoing research tasks.

As already stated above, the following parts of this thesis contain reprints of related research papers. The order and naming is chosen such that it relates to the chapters in the current introduction (part II).

At the end of this booklet, an extensive collection of bibliographic references can be found which is meant to be a useful collection for a researcher trying to work through the literature relevant to the topics covered in this thesis. The list includes all papers cited within the introductory part as well as the reprints given in chapters III-1 to V-8.
2 Mesoscopic modeling of multiphase flows

1 Motivation

In condensed matter physics, many phenomena operate at length and time scales which are too large for detailed microscopic modeling. Typical microscopic models consider every atom or molecule within a system and are based on classical molecular dynamics. For increasing problem sizes, this results in rapidly increasing complexity which restricts the currently treatable length scales to the order of several nanometers and the timescales to the order of nanoseconds. Even today’s fastest supercomputers are not able to provide the computing power required to overcome this limitation.

As an example, a microscopic description of a fluid would track the position, momentum, and energy of every single fluid molecule. On the other hand, macroscopic models usually deal with a many orders of magnitude smaller number of variables, which are often more closely related to physical observables. For example, a macroscopic model of a fluid would describe its velocity, density and temperature at various points.

Thermodynamic descriptions can be obtained from the underlying microscopic picture by applying standard methods known in statistical physics [375]. This contraction of description for systems at thermodynamic equilibrium is in principle straightforward. However, within this thesis we are primarily interested in time-dependent, non-equilibrium systems for which there is less widespread agreement about their statistical mechanical description. Additionally, only few methods exist for the treatment of such complex systems containing processes involving several length and time scales. This situation presents a general problem for both, microscopic and macroscopic models. Whereas microscopic descriptions require vast amounts of computational effort to model systems spanning several length and time scales, macroscopic treatments omit the microscopic details which may give rise to the characteristic behavior of the system.

Therefore, there is currently considerable interest in mesoscale computer models. These models coarse grain most of the atomic or molecular details but retain the essential physics needed to describe the phenomena of interest. They are intended to treat systems at intermediate length scales between several nanometres and a few millimetres, and processes operating on multiple length and time scales. Examples of such systems can be found in everyday life: the macroscopic behavior of detergents, shampoos, cacao drinks, milk, blood, and paint is induced by their microscopic and/or mesoscopic properties.

A number of typical mesoscale situations involve mixtures of fluids which exhibit complicated behavior due to the interactions of their individual molecules. The attempt to model such systems results in modeling both the bulk flow, or hydrodynamic behavior, and the interactions. As described above, hydrodynamics is very difficult and expensive to treat on an atomistic level, but relatively straightforward to handle at the continuum level. On the other hand, the fluid interactions can be examined at the atomistic level, but are usually not straightforward to incorporate at the continuum level. There are numerous examples in the literature where interactions between microscopic particles may give rise to macroscopic
II-2. Mesoscopic modeling of multiphase flows

flow behavior – for example, Marangoni flow [411]. Here, gradients in surface tension due to an inhomogeneous distribution of surfactant at an interface between two immiscible fluids induce macroscopically observable fluid flow [173].

As already stated in the introduction, mesoscale methods exploit the surprising and convenient fact that it is not necessary to keep track of every single molecule of a fluid in order to reproduce its hydrodynamic behavior. Instead, it is sufficient to group very large numbers of molecules into Lagrangian “packets of fluid”, and treat these packets as self-contained particles themselves. Interactions take place between such mesoscopic groupings of particles, rather than their constituent molecules. Provided that certain restrictions, such as isotropy and conservation of mass and momentum, are adhered to, the resulting large-scale behavior is similar to the one which is expected from treating each molecule individually [37, 135].

In the next section, a short introduction to a number of common mesoscale simulation methods is given. These include “dissipative particle dynamics”, “stochastic rotation dynamics”, “lattice gas cellular automata”, as well as the “lattice Boltzmann method”. In particular, we extend our description to the treatment of multiphase or multicomponent fluid flows. The focus of the next section is on the methods applied within the scope of this thesis, i.e., “stochastic rotation dynamics” and the “lattice Boltzmann method”.

The phrases “multiphase” or “multicomponent” do not necessarily relate to fluids only. In fact, in this thesis we also consider particle-laden flows. These include hard sphere suspensions or colloidal dispersions such as mixtures of aluminium oxide powder and water as a model system for soils. Such particle suspensions are ubiquitous in our daily life, but are not well understood due to their complexity. They can be seen as a mixture of granular materials and a fluid, where each component alone is a challenge to be modeled numerically. When they are combined in a suspension, neither component can be neglected, so all the difficulties of both fluids and grains must be solved, in addition to the new problem of describing the interaction between the two.

Long-range fluid-mediated hydrodynamic interactions often dictate the behavior of particle-fluid mixtures. The majority of analytical results for the particle scale behavior of suspensions has been obtained in the regime of vanishing Reynolds numbers (viscous flow). For large systems, scientists aim at an average, continuum description of the large-scale behavior. However, this requires time-consuming and sometimes very difficult experimental measurements of phenomenological quantities such as the mean settling speed of a suspension, the stress contributions in the system of the individual components (solid and fluid) as functions of, e.g., the solid volume fraction of the constituents.

Computer simulation methods are indispensable for many-particle systems, for the inclusion of inertia effects (Reynolds numbers \( > 1 \)) and Brownian motion (Peclet number of order 1). These systems often contain a large number of important time scales which differ by many orders of magnitude, but nevertheless have to be resolved by the simulation, leading to a large numerical effort. However, simulations have the potential to increase the knowledge of elementary processes and enable to yield the aforementioned relations from simulations instead of experiments.

During the last twenty years, various simulation methods have been developed to simulate particle-fluid mixtures. All of them have their inherent strengths but also some disadvantages. For example, simplified Brownian dynamics does not contain long-ranged hydrodynamic interactions among particles at all [214]. Brownian dynamics with full hydrodynamic interactions utilizes a mobility matrix which is based on tensor approximations which are exact in the limit of zero Reynolds number and zero particle volume fraction [8, 355]. However, the computational effort scales with the cube of the particle number due to the in-
Motivation

Pair-Drag simulations have been proposed by Silbert et al. [425], and include hydrodynamic interactions in an approximative way. They have focused on suspensions with high densities (up to 50%) of uncharged spherical colloidal particles. Stokesian dynamics has been presented by Bossis and Brady in the 80s and applied by many authors [42, 46, 358, 422]. For example, Melrose and Ball have performed detailed studies of shear thickening colloids using Stokesian dynamics simulations [313, 314]. However, this method is limited to Reynolds numbers close to zero and the computational effort is very high for dynamical simulations. Even with today’s powerful computers it is not possible to study the dynamics of more than a few hundred particles because the computational effort scales with the particle number \( N \) as \( O(N^3) \). The method is still widely used due to its physical motivation and its robustness, but is complicated to code. Accelerated Stokesian dynamics partly solves the limitation to small particle numbers by reducing the computational time to \( O(N \ln N) \) [422]. Boundary-element methods are more flexible than Stokesian dynamics and can also be used to simulate non-spherical or deformable particles, but the computational effort is even higher [262, 286].

All these methods assume that hydrodynamic interactions are fully developed and that the dynamics of the fluid and of the dispersed particles can be treated as fully separated. In reality, this is not the case. Hydrodynamic interactions are time dependent due to local stresses at the fluid-particle interfaces. Utilizing a mesoscopic or continuum solver for the fluid field allows to describe the time dependent long-range hydrodynamics properly. The particle movement is then described by a molecular dynamics algorithm which is also responsible for the modeling of particle-particle interactions due to, e.g., surface charges.

Molecular dynamics originally was developed to simulate the motion of atoms and molecules. However, the method can be extended by including rotation of rigid bodies and various types of interaction forces developing it into a more general simulation method to solve equations of motion of rigid bodies. In this broader context it is sometimes called “discrete element method” (DEM).

The computational effort of these coupled methods consisting of a fluid solver and a molecular dynamics algorithm scales linearly with the number of particles. However, for small Reynolds numbers, the computational gain of these methods can be lost due to the additional effort needed to describe the motion of the fluid.

Depending on the particular problem to be modeled, it is of importance to choose the best suitable simulation algorithm in order to use the available compute resources most effectively with resolving the system as well as needed. For example, it has to be decided if thermal fluctuations are of importance, if long-range hydrodynamic interactions are needed, if the flow field around a single particle is of interest or if one only needs to obtain averaged properties. Also, it is important to decide beforehand if large particle counts are needed or whether only a few of them are sufficient to tackle the problem of interest. In addition, questions about the shape of the particles and their interactions which are not related to hydrodynamics have to be answered [186, 187, 188, 198, 199, 200, 201, 209, 247, 408].

Section 3 introduces some typical methods to couple solvers for moving particles with descriptions for hydrodynamic interactions. These include “Brownian dynamics” as well as coupling algorithms for some of the mesoscopic fluid solvers described in section 2. We close that section with an outlook on a special method for massive particles in a gas flow and conventional Navier-Stokes solvers.
2 Mesoscopic simulation methods for fluid dynamics

2.1 Dissipative particle dynamics (DPD)

The technique of dissipative particle dynamics (DPD) was originally introduced by Hoogerbrugge and Koelman [211]. Later, Español and Warren modified the model in order to produce a Gibbsian equilibrium distribution [126]. The algorithm essentially solves a Langevin equation with momentum conservation. Mesoscopic fluid “particles” with continuous positions $r_i$ and continuous momenta $p_i$ are moved with discrete timesteps $\delta t$. The algorithm consists of two stages: in the first, particle positions are advected according to their momenta, so

$$r_i \rightarrow r_i + p_i \delta t.$$  \hspace{1cm} (2.1)

In the second stage, the momentum of each particle is updated according to the force acting upon it, so

$$p_i \rightarrow p_i + \sum_{j \neq i} F_{ij}^C + \sum_{j \neq i} F_{ij}^D + \sum_{j \neq i} F_{ij}^R,$$  \hspace{1cm} (2.2)

where $F_{ij}^C$ is a conservative interaction force between different particles $i$ and $j$,

$$F_{ij}^D = -\gamma \omega_D (r_{ij}) (e_{ij} \cdot v_{ij}) e_{ij}$$  \hspace{1cm} (2.3)

is a dissipative, viscous force, and

$$F_{ij}^R = \sigma \omega_R (r_{ij}) e_{ij} \zeta_{ij}$$  \hspace{1cm} (2.4)

is a random force to introduce stochastic fluctuations. $e_{ij}$ is the unit vector pointing from particle $i$ to particle $j$, $v_{ij}$ their relative velocity, and $\zeta_{ij}$ a Gaussian noise term. $\omega_D$ and $\omega_R$ are weighting functions defining the range of the dissipative and random interactions. The frictional coefficient $\gamma$ and the amplitude of the white noise $\sigma$ are related to the temperature of the system as

$$\sigma = \sqrt{k_B T \gamma}.$$  \hspace{1cm} (2.5)

While DPD permits a continuous range of values for $r_i$ and $p_i$, more simplified models may be created discretizing the position, the momentum, or both. The algorithm has been extended to cover multicomponent fluids [94, 303] and was successfully applied to domain growth and phase separation in binary immiscible fluids with and without shear [96, 232], amphiphilic mesophases [231], and colloidal suspensions [36].

2.2 Stochastic rotation dynamics (SRD)

Like DPD, the stochastic rotation dynamics method (SRD) introduced by Malevanets and Kapral [297, 298] is a promising tool for a coarse-grained description of a fluctuating solvent. The method is also known as “real-coded lattice gas” [222] or as “multi-particle-collision dynamics” (MPCD) [382]. It can be seen as a “hydrodynamic heat bath”, whose details are not fully resolved but which provides correct hydrodynamic interactions among embedded particles [266]. SRD is especially well suited for flow problems with Péclet numbers of order one and Reynolds numbers on the particle scale between 0.05 and 20 for ensembles of many particles. Also similar to DPD, the method is based on mesoscopic fluid particles with continuous positions and velocities and each time step is composed of two simple steps: one streaming step and one interaction step. In the streaming step the positions of the fluid
2.3 Lattice gas cellular automata (LGCA)

particles are updated as in the Euler integration scheme known from molecular dynamics simulations,

$$\mathbf{r}_i(t + \delta t) = \mathbf{r}_i(t) + \delta t \mathbf{v}_i(t),$$

(2.6)

where \( \mathbf{r}_i(t) \) denotes the position of the particle \( i \) at time \( t \), \( \mathbf{v}_i(t) \) its velocity at time \( t \) and \( \delta t \) is the time step used for the SRD simulation.

After updating the positions of all fluid particles, they interact collectively in an interaction step which is constructed to preserve momentum, energy and particle number. The fluid particles are sorted into cubic cells of a regular lattice and only the particles within the same cell are involved in the interaction. First, their mean velocity

$$u_j(t') = \frac{1}{N_j(t')} \sum_{i=1}^{N_j(t')} \mathbf{v}_i(t)$$

(2.7)

is calculated, where \( u_j(t') \) denotes the mean velocity of cell \( j \) containing \( N_j(t') \) fluid particles at time \( t' = t + \delta t \). Then, the velocities of each fluid particle in cell \( j \) are updated as

$$\mathbf{v}_i(t + \delta t) = u_j(t') + \Omega_j(t') \cdot [\mathbf{v}_i(t) - u_j(t')]$$

(2.8)

Here, \( \Omega_j(t') \) is a rotation matrix, which is randomly chosen for each time step and each cell. One out of many possible choices is using rotations about one of the coordinate axes by an angle \( \pm \alpha \), with \( \alpha \) fixed. This has been suggested by M. Strauß in [472]. The coordinate axis as well as the sign of the rotation are chosen randomly, resulting in six possible rotation matrices. The mean velocity \( u_j(t) \) in the cell \( j \) can be seen as streaming velocity of the fluid at the position of the cell \( j \) at the time \( t \), whereas the difference \( [\mathbf{v}_i(t) - u_j(t')] \) entering the interaction step can be interpreted as a contribution to the thermal fluctuations.

In order to remove low temperature anomalies and to achieve exact Galilean-invariance, the original algorithm was modified: all particles are shifted by the same random vector with components in the interval \([-a/2, a/2]\) before the collision step. Particles are then shifted back by the same amount after the collision. The random vectors of consecutive iterations are uncorrelated. Ihle and Kroll have shown that this method leads to transport coefficients independent of an imposed homogeneous flow field [218, 219]. In [220] and [242] analytical calculations of the transport coefficient of this method are presented.

The method is mainly used as a fluctuating solvent to simulate colloidal suspensions or polymer solutions as shown later in this chapter. Hashimoto et al. generalized this model to treat binary immiscible fluid mixtures [194] by adjusting the rotation matrix \( \Omega_j \) for each different phase in such a way that they repel. Sakai et al. extended the model to include amphiphiles, and successfully modeled the formation of micelles [396]. Recently, an extension of the SRD algorithm to binary immiscible and ternary amphiphilic mixtures based on collections of sub-cells was introduced by Tuzel et al. [471]. The authors observed both droplets and micelles, as well as a bicontinuous phase.

### 2.3 Lattice gas cellular automata (LGCA)

The lattice gas cellular automaton (LGCA) method was proposed in 1986 by Frisch, Hasslacher, and Pomeau [147], as well as by Wolfram [499]. Here, mesoscopic particles are only permitted to occupy points on a Bravais lattice, discretizing space, and are only permitted to travel along the lattice vectors, discretizing momentum. The algorithm consists of two steps: during the “advection” step, particles travel along their velocity vectors to adjacent
lattice sites; in the “collision” step, particles at each individual lattice site undergo collisions during which their velocities are redistributed in a manner conserving the total mass and momentum at each site. Provided the lattice is carefully chosen to ensure isotropy of the fluid, the large-scale behavior will be hydrodynamically correct. A substantial advantage of the LGCA algorithm is that it only requires Boolean operations, and its unconditional numerical stability [146].

Fluid mixtures may be simulated by introducing different species onto the lattice, often denoted by color. Interactions between the different species may then be introduced by the particles carrying a “color charge”. Thus, they experience a force due to the color field generated by surrounding particles [389]. The different fluids may be made immiscible by introducing a force which compels, e.g., “red” particles to travel up the color-field gradient towards regions of higher “red” density, and compels “blue” particles to travel towards regions of higher “blue” density. Hence, a mixture of “red” and “blue” particles will separate into separate single-color regions.

The “colored particle” description may be extended to cover the case of amphiphiles, by introducing a new species of mesoscopic particles, which have an orientational degree of freedom. A single mesoscopic particle of this species can be regarded as a dumbbell consisting of a red particle and a blue particle bolted together. The whole particle possesses no net color charge, but will tend to align itself with the color field. The orientation of the mesoscopic particle represents an averaged orientation of its constituent molecules. Lattice gas models with amphiphile particles have been used to simulate the effect of surfactants on oil-water mixtures in porous media [290], and the self-assembly of micelles [38, 39].

2.4 The lattice Boltzmann method (LBM)

The lattice Boltzmann (LBM) method is a simplification of the lattice gas method: particles have discretized positions and momenta, but rather than individual Boolean particles being tracked around a lattice, their real-valued population is stored, resulting in a less noisy method. Much of the power of this algorithm lies in the ease to impose boundary conditions, and to extended the model in order to describe mixtures of interacting complex fluids. As in the previously described methods, not individual atoms and molecules are tracked. However, in contrast to DPD or LGCA coarse graining is not obtained by tracking individual discrete mesoscopic “packets of fluid” either. Instead, the lattice Boltzmann method describes the dynamics of the single-particle distribution function of mesoscopic fluid packets. Since the lattice Boltzmann method has been applied to all problems described in parts III and IV, as well as some problems in part V, it will be described in more detail here.

In a continuum description, the single-particle distribution function \( f_1(r, v, t) \) represents the density of fluid particles with position \( r \) and velocity \( v \) at time \( t \), such that the density and velocity of the macroscopically observable fluid are given by \( \rho(r, t) = \int f_1(r, v, t) dv \) and \( u(r, t) = \int f_1(r, v, t) v dv \), respectively. In the non-interacting, long mean free path limit, with no externally applied forces, the evolution of this function is described by Boltzmann’s famous equation,

\[
(\partial_t + v \cdot \nabla) f_1 = \Omega[f_1].
\] (2.9)

The left hand side describes changes in the distribution function due to free particle motion and the right hand side contains the collision operator \( \Omega \), describing changes due to pairwise collisions. Typically, this is a complicated integral expression, which is often simplified to
the linear Bhatnagar-Gross-Krook, or BGK form [31]

\[ \Omega[f] \simeq -\frac{1}{\tau}[f - f^{(eq)}]. \]  

(2.10)

This simple collision operator describes the relaxation towards a Maxwell-Boltzmann equilibrium distribution \( f^{(eq)} \) at a time scale set by the characteristic time \( \tau \). Even though this is a drastic simplification, distributions governed by the Boltzmann-BGK equation conserve mass, momentum, and energy, and obey a non-equilibrium form of the second law of thermodynamics [279]. Moreover, the well-known Navier-Stokes equations for macroscopic fluid flow are obeyed on coarse length and time scales by these distributions [66, 279].

In a lattice Boltzmann formulation, the single-particle distribution function is discretized in time and space. The positions \( r \) on which \( f_i(r, v, t) \) is defined are restricted to points \( r_i \) on a lattice, and the velocities \( v \) are restricted to a set \( c_i \) joining points on the lattice. Hence, \( f_i(r, t) = f_i(r_i, c_i, t) \) represents the density of particles at lattice site \( r \) travelling with velocity \( c_i \), at timestep \( t \). The density and velocity of the simulated fluid are now given by

\[ \rho(r, t) = \sum_i f_i(r, t) \]  

(2.11)

and

\[ u(r, t) = \sum_i f_i(r, t)c_i. \]  

(2.12)

The lattice has to be chosen carefully to ensure isotropic behavior of the simulated fluid [367]. The lattice Boltzmann equation may be rigorously derived by discretizing the continuum Boltzmann equation [196]. Alternatively, it may be regarded as a Boltzmann-level approximation of its ancestor, the LGCA [310].

In a manner similar to the previously described methods, the discretized description of the fluid may now be evolved according to a two-step procedure consisting of a collision step and an advection step. In the collision step, particles at each lattice site are redistributed across the velocity vectors: this process corresponds to the action of the collision operator, and in the most simple case takes the BGK form. In the advection step, values of the post-collisional distribution function are propagated to adjacent lattice sites: this corresponds to particles streaming along their velocity vectors, and is the discretized equivalent of the left-hand side of the continuum Boltzmann equation. Overall, the system obeys the lattice Boltzmann equation (LBE), produced by combining the two evolution steps:

\[ f_i(r, t + 1) - f_i(r, t) = \Omega[f] = -\frac{1}{\tau}[f_i(r, t) - N_i(\rho, u)], \]  

(2.13)

where \( N_i = N_i(\rho(r), u(r)) \) is a polynomial function of the local density and velocity, which may be found by discretizing the well-known Maxwell-Boltzmann equilibrium distribution. It can be shown that the resulting macroscopic density and velocity fields obey the Navier-Stokes equations [69]. A well-known drawback of the lattice Boltzmann method is that it is typically not guaranteed to be numerically stable, and will crash or produce physically unreasonable results if, for example, the forcing rate applied to a fluid is too high or if the inter-particle interaction strength is set too high.

There are several schemes for generalizing the lattice Boltzmann algorithm to treat multi-component fluids, including analogies with LGA [177], discretization of a modified form of the continuum Boltzmann equation [293], imposition of free-energy functionals [444], or inclusion of an explicit forcing term in the collision operator [415]. The most widely used
methods are the so-called Oxford model by Swift et al. [444] and the approach introduced by Shan and Chen [415]. In the framework of this thesis, only the latter one is used, but we will also give a short introduction into the alternative approach based on free-energy functionals.

The multiphase lattice Boltzmann model introduced by Swift et al. is designed to produce a well defined isothermal equation of state and utilizes a free energy functional of the density $\rho(r)$

$$
\Psi(\rho) = \int \left[ \frac{\kappa}{2} |\nabla \rho(r)|^2 + \psi(\rho(r)) \right] \, dr.
$$

(2.14)

The first term is controlled by the constant $\kappa$ and gives the contribution from density gradients. The remaining term describes the bulk free energy. A pressure can then be defined as

$$
p(r) = \rho \frac{\delta \Psi}{\delta \rho} - \Psi = p_0 - \kappa \rho \nabla^2 \rho - \frac{\kappa}{2} |\nabla \rho|^2.
$$

(2.15)

This results in a well defined equation of state and the equilibrium distribution function can be constructed to conserve mass and momentum. The method has the convenient advantage that it is possible to impose parameters such as the surface tension, by choosing an appropriate form of $\Psi(\rho)$. Additionally, the method is easily extended to multiphase flows [342, 444].

The drawback of this approach becomes important if large density gradients are present because the viscosity term in the macroscopic equations is not Galilean invariant [444].

Also very popular to introduce multiphase or multicomponent flows into lattice Boltzmann models is the model by Shan and Chen. Here, to distinguish between different components, the single-particle distribution function $f_i$ may be extended to the form $f_{\sigma}^i$, where each component is denoted by a different value of the superscript $\sigma$, so that the density and momentum of a single component $\sigma$ are given by $\rho^\sigma = \sum_i f_i^\sigma$ and $\rho^\sigma \mathbf{u}^\sigma = \sum_i f_i^\sigma \mathbf{c}_i$, respectively [415, 416]. The lattice BGK equation (2.13) now takes the form

$$
f_{\sigma}^i(r, t + 1) - f_{\sigma}^i(r, t) = -\frac{1}{\tau^\sigma} [f_{\sigma}^i - N_i(\rho^\sigma, \mathbf{v}^\sigma)]
$$

(2.16)

The velocity $\mathbf{v}^\sigma$ is found by calculating a weighted average velocity

$$
\mathbf{u}' = \frac{\sum_\sigma \frac{\rho^\sigma}{\tau^\sigma} \mathbf{u}^\sigma}{\sum_\sigma \frac{\rho^\sigma}{\tau^\sigma}},
$$

(2.17)

and then adding a term to account for external forces,

$$
\mathbf{v}^\sigma = \mathbf{u}' + \frac{\tau^\sigma}{\rho^\sigma} \mathbf{F}^\sigma.
$$

(2.18)

The force term $\mathbf{F}^\sigma$ can take the form $g_\sigma^\sigma \hat{z}$ to produce a gravitational force acting in the $z$-direction. In order to produce nearest-neighbor interactions between components, it chosen to be

$$
\mathbf{F}^\sigma = -\psi^\sigma(r) \sum_\sigma g_{\sigma\bar{\sigma}} \sum_i \psi^\bar{\sigma}(r + \mathbf{c}_i) \mathbf{c}_i,
$$

(2.19)

where $\psi^\sigma(r) = \psi^\sigma(\rho^\sigma(r))$ is an effective mass for component $\sigma$; $g_{\sigma\bar{\sigma}}$ is a coupling constant controlling the strength of the interaction between two components $\sigma$ and $\bar{\sigma}$. If $g_{\sigma\bar{\sigma}}$ is set to zero for $\sigma = \bar{\sigma}$, and to a positive value for $\sigma \neq \bar{\sigma}$, in the interface between bulk regions of each component, particles experience a force in the direction away from the interface, producing immiscibility. In two-component systems, it is usually the case that $g_{\sigma\bar{\sigma}} = g_{\bar{\sigma}\sigma}$. Amongst other things, this model has been used to simulate spinodal decomposition [77,
163, 183, 192], polymer blends [305], liquid-gas phase transitions [416], and flow in porous media [192, 304]. Of particular interest for the current thesis is the possibility to apply the interaction force at boundaries in order to tune the wettability of surfaces [25, 189]. This approach has found applications in microfluidics and is used in part IV of this thesis.

As with many other mesoscale fluid methods, fluids containing surfactant may be treated in the LB framework by introducing a new species $s$ with an orientational degree of freedom [68]. The particles of this species are each given a vector dipole moment $\mathbf{d}$ which has maximum magnitude $d_0$, corresponding to complete alignment of the constituent molecules. This is introduced in the model by a dipole field $\mathbf{d}(\mathbf{r}, t)$ representing the average orientation of any amphiphile present at site $\mathbf{r}$. During advection, values of $\mathbf{d}(\mathbf{r}, t)$ are propagated around the lattice according to

$$
\rho^s(\mathbf{r}, t + 1)\mathbf{d}(\mathbf{r}, t + 1) = \sum_i \tilde{f}_i^s(\mathbf{r} - \mathbf{c}_i, t)\tilde{\mathbf{d}}(\mathbf{r} - \mathbf{c}_i, t),
$$

(2.20)

where tildes denote post-collision values. During collision, the dipole moments evolve in a BGK process controlled by a dipole relaxation time $\tau_d$ as

$$
\dot{\mathbf{d}}(\mathbf{r}, t) = \mathbf{d}(\mathbf{r}, t) - \frac{1}{\tau_d} \left[ \mathbf{d}(\mathbf{r}, t) - \mathbf{d}^{(eq)}(\mathbf{r}, t) \right],
$$

(2.21)

where the equilibrium dipole moment $\mathbf{d}^{(eq)}$ is aligned with the color field $\mathbf{h}$,

$$
\mathbf{d}^{(eq)} \simeq \frac{\beta d_0}{3} \mathbf{h}.
$$

(2.22)

The color field contains a component $\mathbf{h}^c$ due to colored species, and a part $\mathbf{h}^s$ due to dipoles. The former can be found from the populations of surrounding lattice sites,

$$
\mathbf{h}^c = \sum_\sigma q^\sigma \sum_i \rho^\sigma(\mathbf{r} + \mathbf{c}_i)\mathbf{c}_i,
$$

(2.23)

where $q^\sigma$ is a color charge, such as $+1$ for red particles, $-1$ for blue particles, and $0$ for amphiphile particles. The field due to other dipoles turns out to be given by

$$
\mathbf{h}^s(\mathbf{r}, t) = \sum_i \left[ \sum_{j \neq 0} f_i^s(\mathbf{r} + \mathbf{c}_i, t)\theta_j \cdot \mathbf{d}_i(\mathbf{r} + \mathbf{c}_j, t) + f_i^s(\mathbf{r}, t)\mathbf{d}_i(\mathbf{r}, t) \right],
$$

(2.24)

where the second-rank tensor $\theta_j$ is defined in terms of the unit tensor $\mathbf{I}$ and lattice vector $\mathbf{c}_j$ as

$$
\theta_j = \mathbf{I} - \frac{D}{c^2} \mathbf{c}_j\mathbf{c}_j.
$$

(2.25)

Here, $c$ is the modulus of $\mathbf{c}_j$ and $D$ the spatial dimension. In the presence of an amphiphilic species, the force on other species includes an additional term $\mathbf{F}^{\sigma,s}$ to account for the color field due to the amphiphiles. By treating an amphiphilic particle as a pair of oil and water particles with a very small separation $\mathbf{d}$, and Taylor-expanding in $\mathbf{d}$, it can be shown that this term is given by

$$
\mathbf{F}^{\sigma,s}(\mathbf{r}, t) = -2\psi^\sigma(\mathbf{r}, t)g_{\sigma s} \sum_{i \neq 0} \mathbf{d}(\mathbf{r} + \mathbf{c}_i, t) \cdot \theta_i \psi^s(\mathbf{r} + \mathbf{c}_i, t),
$$

(2.26)
where \( g_{\sigma \bar{\sigma}} \) is a constant controlling the strength of the interaction between amphiphiles and non-amphiphiles. While they do not possess a net color charge, the amphiphiles also experience a force as a result of the color field, consisting of a part \( F^{s,c} \) due to ordinary species, and a part \( F^{s,s} \) due to other amphiphiles. These terms are given by

\[
F^{s,c} = 2 \psi^{s}(r, t) \tilde{d}(r, t) \cdot \sum_{\sigma} g_{s\bar{\sigma}} \sum_{i \neq 0} \theta_{i} \psi^{\sigma}(r + c_{i}, t) \tag{2.27}
\]

\[
F^{s,s} = -\frac{4D}{c^{2}} g_{ss} \psi^{s}(r) \sum_{i} \left\{ \tilde{d}(r + c_{i}) \cdot \theta_{i} \cdot \tilde{d}(r) c_{i} \right\} \psi^{s}(r + c_{i}). \tag{2.28}
\]

To summarize, the interactions between fluid components are governed by the coupling constants \( g_{\sigma \bar{\sigma}} \), controlling the interaction between different sorts of colored particles, between colored particles and amphiphiles, and between the amphiphiles, respectively. While the form of the interactions seems straightforward at a mesoscopic level, it is essentially phenomenological. Thus, it is not necessarily easy to relate the interaction scheme or its coupling constants to either microscopic molecular characteristics, or to macroscopic phase behavior. Different values of these parameters will give rise to a wide variety of different phases as described in section II-3.3 and part III, such as spherical and wormlike micelles, sponges, lamellae, or droplets: the phase behavior can be very difficult to predict beforehand from the simulation parameters, and brute-force parameter searches are often resorted to [39, 79].

3 Mesoscopic simulation methods for particle-laden flows

All mesoscopic methods describing fluid flow can be extended to model suspensions or polymer solutions. In particular for polymers, but also for membranes and vesicles, the coupling between fluid solver and the solved constituent is implemented as a velocity coupling of mass-less particles or contact points. Since such systems are not studied in this thesis, only the case of colloidal suspensions is explained. Here, the individual particles are usually treated by a molecular dynamics algorithm and momentum is transferred between them and the fluid after a sufficiently small number of timesteps. In the following, an introduction to the treatment of particle-particle interactions is given and different coupling algorithms are described. In order to highlight that every method has its distinct areas of applicability and that often a correct description of the long-range hydrodynamic effects is not needed, we also introduce popular methods to describe such systems.

3.1 Particle-particle interactions

Inter-particle interactions which are not due to hydrodynamic forces are usually taken care of within the molecular dynamics solver. For example, in the case of the claylike colloids studied in part V, we include effective electrostatic interactions and van der Waals attraction, a lubrication force and Hertzian contact forces.

The interaction between the particles is described by DLVO theory (after Derjaguin, Landau, Verwey and Overbeek) [214, 278, 393]. If the colloidal particles are suspended in a solvent,
typically water, ions move into solution, whereas their counter ions remain in the particle due to a different resolvability. Thus, the colloidal particle carries a charge. The ions in solution are attracted by the charge on the particles and form an electric double layer. The resulting electrostatic interaction between two of these particles can be described by an exponentially screened Coulomb potential

$$V_{\text{Coul}} = \pi \varepsilon_r \varepsilon_0 \left[ \frac{2 + \kappa d}{1 + \kappa d} \cdot \frac{4 k_B T}{z e} \tan \left( \frac{z e \zeta}{4 k_B T} \right) \right]^2 \times \frac{d^2}{r} \exp(-\kappa [r - d]).$$

(2.29)

d denotes the particle diameter and r is the distance between the particle centers. e is the elementary charge, T the temperature, $k_B$ the Boltzmann constant, and z is the valency of the ions of added salt. Within DLVO theory one assumes linear screening, mainly by one species of ions with valency z (e.g. $z = +1$ for NH$_4^+$ with a corresponding counter-ion Cl$^-\). The effective surface potential $\zeta$ is the electrostatic potential at the border between the diffuse layer and the compact layer, it may therefore be identified with the $\zeta$-potential. It includes the effect of the bare charge of the colloidal particle itself, as well as the charge of the ions in the Stern layer, where the ions are bound permanently to the colloidal particle. In other words, DLVO theory uses a renormalized surface charge. This charge can be related to the pH value of the solvent within Debye-Hückel theory [198]. $\varepsilon_0$ is the permittivity of the vacuum, $\varepsilon_r$ the relative dielectric constant of the solvent. $\kappa$ is the inverse Debye length defined by $\kappa^2 = 8 \pi \ell_B I$, with the ionic strength $I$ and the Bjerrum length $\ell_B$. In this thesis, we use $\varepsilon_r = 81$ for water, which implies $\ell_B = 7$ Å. The Coulomb term of the DLVO potential competes with the attractive van der Waals term

$$V_{\text{vdW}} = -\frac{A_H}{12} \left[ \frac{d^2}{r^2 - d^2} + \frac{d^2}{d^2} + 2 \ln \left( \frac{r^2 - d^2}{r^2} \right) \right].$$

(2.30)

$A_H = 4.76 \cdot 10^{-20}$ J is the Hamaker constant [213] which involves the polarizability of the particles. The primary minimum has to be modeled separately, as discussed below.

It is common to all simulation methods incorporating detailed solvers for the flow field that long range hydrodynamic interactions can only be reproduced correctly down to a certain length scale. On shorter distances, a lubrication force has to be introduced explicitly in the molecular dynamics simulation. The most dominant mode, the so-called squeezing mode, is an additional force

$$F_{\text{lub}} = -(\mathbf{v}_r, \hat{r}) \hat{r} \frac{6 \pi \eta r_{\text{red}}^2}{r - r_1 - r_2},$$

(2.31)

with $r_{\text{red}} = \frac{r_1 r_2}{r_1 + r_2}$

(2.32)

between two spheres with radii $r_1$, $r_2$ and the relative velocity $\mathbf{v}_r$. $\hat{r}$ is the unit vector in the direction connecting the two colloids, $\eta$ is the dynamic viscosity of the fluid. In contrast to the DLVO potentials the lubrication force is a dissipative force.

To avoid that the particles penetrate each other, one needs a repulsive force depending on their overlap. We are using a Hertz force described by the potential

$$V_{\text{Hertz}} = K (d - r)^{5/2} \quad \text{if} \quad r < d,$$

(2.33)

where $K$ can be expressed by the elastic modulus of Al$_2$O$_3$. The Hertz force also contains a damping term in normal direction,

$$F_{\text{Damp}} = -(\mathbf{v}_r, \hat{r}) \beta \sqrt{r - r_1 - r_2},$$

(2.34)
with a damping constant $\beta$ and for the transverse direction a viscous friction proportional to the relative velocity of the particle surfaces is applied.

For the integration of the translational motion a velocity Verlet algorithm [13] is utilized to update the velocity and position of particle $i$ according to the equations

$$x_i(t + \delta t) = x_i(t) + \delta t v_i(t) + \frac{\delta t^2}{m} F_i(t),$$  \hspace{0.5cm} (2.35)

$$v_i(t + \delta t) = v_i(t) + \delta t \frac{F_i(t) + F_i(t + \delta t)}{2m}.$$  \hspace{0.5cm} (2.36)

### 3.2 Brownian dynamics

The Brownian dynamics algorithm is not able to treat long-range hydrodynamics. However, the method is used in chapter V-8 and particularly easy to implement. Further, due to the missing hydrodynamic interactions, the time of computation is lower than compared to the more advanced techniques. Thus, it is still an attractive method in cases where hydrodynamics is not of importance or where very large systems need to be simulated. Only the two most important aspects of hydrodynamics felt by the suspended particles are taken into account in the Brownian dynamics algorithm, namely the Stokes friction and the Brownian motion. Correspondingly, this is done by adding to a molecular dynamics simulation two additional forces. A Langevin equation can be utilized to describe the motion of a Brownian particle with radius $R$ at position $r(t)$:

$$m \ddot{r}(t) = 6 \pi \eta R \dot{r}(t) + F_{\text{rand}}(t) + F_{\text{ext}}(r, t),$$  \hspace{0.5cm} (2.37)

where the first term models the Stokes friction in a solvent of viscosity $\eta$, $F_{\text{ext}}(r, t)$ is the sum of all external forces like gravity, forces exerted by other suspended particles. $F_{\text{rand}}(t)$ describes the thermal noise which gives rise to the Brownian motion. The random force on different particles is assumed to be uncorrelated, as well as the force on the same particle at different times. It is further assumed to be Gaussian with zero mean. The method is very efficient and widely used [372, 373, 374]. However, if one tries to model suspensions which are not sufficiently dilute, a non-neglectable error is being introduced. There are Brownian dynamics schemes available where people have tried to correct these errors by adding additional force components due to the surrounding particles. Often, analytic two-body solutions of the flow field are used.

### 3.3 Stochastic rotation dynamics

A number of different methods coupling stochastic rotation dynamics and the molecular dynamics simulation have been introduced in the literature. Inoue et al. proposed a way to implement no slip boundary conditions on the particle surface [222], whereas Padding and Louis implement slip boundaries by introducing modified Lennard-Jones potentials for the interaction between solved particles and SRD particles [347, 348]. A more coarse grained method was originally designed to couple the monomers of a polymer chain to the fluid [129, 299], but is also well suitable for colloidal suspensions.

In this approach, the colloidal particles are sorted into the SRD boxes and their velocities are included in the rotation step. This technique has been used to model protein chains suspended in a liquid [129, 497] and the claylike colloids as presented in part V. Since the mass of the fluid particles is much smaller than the mass of the colloidal particles, one has to use the mass
of each particle—colloidal or fluid particle—as a weight factor when calculating the mean velocity

\[
\mathbf{u}_j(t') = \frac{1}{M_j(t')} \sum_{i=1}^{N_j(t')} \mathbf{v}_i(t) m_i, \tag{2.38}
\]

with

\[
M_j(t') = \sum_{i=1}^{N_j(t')} m_i, \tag{2.39}
\]

where we sum over all colloidal and fluid particles in the cell, so that \(N_j(t')\) is the total number of both particles together. \(m_i\) is the mass of the particle with index \(i\) and therefore \(M_j(t')\) gives the total mass contained in cell \(j\) at the time \(t' = t + \tau\).

### 3.4 Lattice Boltzmann

For larger particles, thermal fluctuations are undesirable. Here, the lattice Boltzmann method and its extension to particle suspension is a very good candidate, for example to study the dynamics of glass spheres in a sugar solution [247]. The method has been applied to suspensions of spherical and non-spherical particles by various authors. Recently, it has been extended to include Brownian motion as well [5, 121]. Pioneering work on the development of an extension to the lattice-Boltzmann method in order to incorporate particles into the model fluid and to impose the correct boundary condition at the solid/fluid interface has been done by Ladd et al. [259, 260, 262].

To simulate the hydrodynamic interactions between solid particles in suspensions, the lattice-Boltzmann model has to be modified to incorporate the boundary conditions imposed on the fluid by the solid particles. Stationary solid objects are introduced into the model by replacing the usual collision rules at a specified set of boundary nodes by the “link-bounce-back” collision rule [334]. When placed on the lattice, the boundary surface cuts some of the links between lattice nodes. The fluid particles moving along these links interact with the solid surface at boundary nodes placed halfway along the links. Thus, a discrete representation of the surface is obtained, which becomes more and more precise as the surface curvature gets smaller and which is exact for surfaces parallel to lattice planes. Since the velocities in the lattice-Boltzmann model are discrete, boundary conditions for moving suspended particles cannot be implemented directly. Instead, we can modify the density of returning particles in a way that the momentum transferred to the solid is the same as in the continuous velocity case. This is implemented by introducing an additional term

\[
\Delta_{b,i} = \frac{2 \omega c_s \rho \mathbf{u}_b \cdot \mathbf{c}_i}{c_s^2}, \tag{2.40}
\]

in the discrete Boltzmann equation [259], with \(c_s\) being the velocity of sound and coefficients \(\omega c_s\) being the lattice weights; \(\mathbf{u}_b\) is the velocity of the boundary, \(\rho\) the fluid density and \(\mathbf{c}_i\) are the discrete velocities. To avoid redistributing fluid mass from lattice nodes being covered or uncovered by solids, one can allow interior fluid within closed surfaces. Its movement relaxes to the movement of the solid body on much shorter time scales than the characteristic hydrodynamic interaction [259]. However, improved schemes are available where interior fluid is allowed [203, 334].
3.5 Conventional Navier-Stokes solvers

A more computational demanding and not as easy to code method is a Navier Stokes solver for the fluid coupled to particles. This method has been successfully applied to the simulation of sedimentation processes of spherical or non-spherical particles and profits from its well established physical background and long standing experience with similar fluid solvers in engineering disciplines [139, 140, 209, 257, 409, 487]. Finite element or finite difference methods need a proper meshing of the computational domain which is not trivial for complicated boundary conditions as in the case of dense suspensions. Therefore, many authors only simulated a limited number of static configurations rather than the full dynamics of the system. Advances in remeshing techniques as well as more powerful computers have allowed to overcome these problems. Also, in order to avoid remeshing at all, uniform grids can be used [138, 209, 409].

3.6 Inertia-less coarse-grained gas method

Systems with very massive particles solved in a fluid of very low density do not require a full hydrodynamic treatment of the solvent. A possible way to describe such a system is based on a coarse-grained description of the fluid, so that it is resolved on a length scale larger than the particles. Much larger systems can be treated this way, but the coarse-graining is justified only in certain situations. As an example the method was applied to model the pneumatic transport of a powder in a pipe which is a common process in many industrial applications [311, 437, 438].

The model for the gas simulation was first introduced by McNamara and Flekkøy [311] and is based on the mass conservation of the gas and of the granular medium corresponding to the solved particles. If the fluid density is small compared to the density of the particles, and the Reynolds number of the particles is also small, a simplified equation for the pressure can be given. The pressure drop enters as a drag force into the molecular dynamics algorithm.

\[
\frac{\partial P'}{\partial t} = \frac{P_0}{\phi} \nabla (\kappa(\phi) \nabla P') - \frac{P_0}{\phi} \nabla u ,
\]

where \( P_0 \) is the atmospheric pressure, \( P' \) the relative pressure, \( P = P_0 + P' \), \( \eta \) the dynamic viscosity, \( u \) the granular velocity, \( \phi \) the local porosity, and \( \kappa \) the local permeability. This differential equation can be interpreted as a diffusion equation with a diffusion constant \( D = \frac{\phi \kappa(\phi)}{\eta} \). This method is as easy to implement as for example Brownian dynamics, but only works for gas flow. It is not suitable to simulate particles in liquids and Brownian motion cannot be resolved.
Lattice Boltzmann simulations of mesophases and microemulsions

The food, cosmetic and chemical industries often need to stabilize otherwise immiscible fluids. A good example is a barbecue sauce containing large fractions of water and oil or fat. Without any additives the constituents would phase separate, entering a not very appealing de-mixed state in which the fat accumulates in a thick layer on top of the remainder. Adding an emulsifier or surfactant (from "surface active agent") helps to stabilize the sauce. These molecules are often called amphiphiles, where their name follows the Greek words αµϕι (“both”) and φιλια (“love”, “friendship”).

In their simplest form, amphiphiles are comprised of a polar hydrophilic (i.e., water-loving) head group and a hydrophobic (i.e., oil-loving) tail as depicted in Fig. 3.1. The surfactant molecules self-assemble on the surface of oil droplets and reduce the surface tension. Thus, the droplets stabilize and remain suspended within the bulk water. A typical emulsifier used by the food industry is egg yolk lecithin. Proteins and emulsifiers with low molecular weight are also very common.

Figure 3.1: Amphiphiles (surfactants) are comprised of a hydrophilic (water-loving) head group and a hydrophobic (oil-loving) tail.

A solution of water and surfactant is easily produced in the kitchen sink by just adding a few drops of washing-up liquid or soap to the water. Such solutions show complicated and rich behavior as summarized for example by Gompper and Schick [157] or Seddon and Templer [412, 413]. Contributing factors to this behavior are the temperature, surfactant volume concentration, lengths of the polar and non-polar groups as well as the kind and strength of their interaction. The simplest modeling approaches to study surfactant properties consider only two effects, namely the minimization of the interfacial area between water-like and oil-like regions due to hydrophobic repulsion, and geometrical effects due to relative interaction strengths of the head and tail groups.

A binary mixture of otherwise immiscible fluids (for example oil and water) undergoes phase separation. Adding surfactant to this mixture can cause the de-mixing process to slow down. If the surfactant concentration is sufficiently high, the de-mixing process might eventually arrest completely.

It has been shown by Langevin, molecular dynamics, lattice gas, and by lattice Boltzmann simulations that the temporal growth law for the size of oil and water domains in a system without amphiphiles follows a power law $t^\alpha$ [47, 163]. Adding amphiphiles, it crosses over
to a logarithmic growth law \((\ln t)^\theta\), where \(\alpha, \theta\) are fitting parameters and \(t\) is the time [125, 159, 239]. A further increase of the surfactant concentration can lead to growth which is well described by a stretched exponential form \(A - B \exp(-Ct^D)\), where capital letters denote fitting parameters [125, 159].

If the domain growth stops, the so-called sponge mesophase might be reached. In the context of our simulations it is called a bicontinuous microemulsion since it is formed by the amphiphilic stabilization of a phase-separating binary mixture, where the immiscible fluid constituents occur in equal proportions. Here, the oil and water phases interpenetrate and percolate and are separated by a monolayer of surfactant at the interface.

Considering amphiphiles in a single solvent only, at low concentrations, they may be found as independent monomers. After the so-called Krafft point, the excess surfactant will precipitate, or the amphiphiles will start to assemble into aggregates. In the latter case, the critical aggregation concentration (CAC) is reached.

Examples of such aggregates are illustrated in Fig. 3.2. For relatively compact tail groups, closely packed amphiphiles prefer to form spherical micelles as shown in Fig. 3.2a. With this shape, the tail groups hardly get in contact with any water. If the tails groups are larger, it is energetically more favorable for the amphiphiles to form cylindrical aggregates. Such structures are called wormlike micelles and are depicted in Fig. 3.2b. Wormlike micelles are of great industrial importance since they are very common and they occur in various structures: they may be stacked into ordered hexagonal phases or form interpenetrating branched structures. In addition, such phases show pronounced non-Newtonian behavior due to a constant breakup and reformation of wormlike micelles. If the tail groups become even larger, it is energetically most favorable for the surfactants to assemble into flat sheets and to form lamellar phases (see Fig. 3.2c). As illustrated in Fig. 3.2d and Fig. 3.2e, a non-polar solvent will cause inverse structures to form such as inverse wormlike micelles or inverse spherical micelles. Here, the tail groups are pointing outwards.
By adjusting temperature, fluid composition or pressure, amphiphiles can self-assemble and force the fluid mixture into a number of equilibrium structures or mesophases. These are in contrast to the disconnected aggregates described above. The particular case of bicontinuous structures was suggested by Scriven [410]. Here, a surfactant layer forms a single continuous surface dividing space into two multiply connected and interpenetrating volumes. “Minimal surfaces”, i.e., surfaces minimizing their area might repeat throughout space and are then called “periodic minimal surfaces”. Such minimal surfaces have zero mean curvature everywhere [287]. Triply periodic surfaces are minimal surfaces with cubic symmetry, i.e., they repeat in all Carthesian directions. Until 1970, only five such surfaces were known which were discovered by H. A. Schwarz et al. in the late 19th century [405]. These include two of their most celebrated members, namely the \( P \), or “plumber’s nightmare”, and \( D \), or “diamond”, surfaces as shown in Fig. 3.3 (left) and Fig. 3.3 (middle). Schoen discovered twelve more surfaces in 1970 [401], but they did not attract much interest until their existence was proven by Karcher [237]. One of the surfaces discovered by Schoen was the so-called \( G \) or “gyroid” surface as depicted in Fig. 3.3 (right). These three surfaces can be transformed into each other through the single parameter known as the Bonnet angle [137].

A possible simple representation of the \( P \), \( D \), and \( G \) surfaces can be given by a nodal representation, where the surface is defined as the region where the following functions become zero:

\[
\Xi_P(r) = \cos x + \cos y + \cos z \tag{3.1}
\]

\[
\Xi_D(r) = \sin x \sin y \sin z + \sin x \cos y \cos z + \cos x \sin y \cos z + \cos x \cos y \sin z \tag{3.2}
\]

\[
\Xi_G(r) = \cos x \sin y + \cos y \sin x + \cos z \sin x \tag{3.3}
\]

Within the present thesis, we will focus on the \( G \) surface. The “gyroid” is the only known such surface with triple junctions. A numerical study of the gyroid was made by Grosse-Brauckmann [172]. The gyroid has symmetry group \( \text{Ia}\overline{3}d \); the unit cell consists of 96 copies of a fundamental surface patch, related through the symmetry operations [149] of this space group. Channels run through the gyroid labyrinths in the (100) and (111) directions; passages emerge perpendicular to any given channel as it is traversed, the direction at which they do so gyrating down the channel, giving rise to the “gyroid” name [172]. The labyrinths are chiral, so that the channels of one labyrinth gyrate in the opposite sense to the channels of the other. It was shown by Schwarz and Gompper that the gyroid has the lowest free energy

![Figure 3.3: Three common cubic triply periodic surfaces: the \( P \), or “plumber’s nightmare” surface (left); the \( D \), or “diamond” surface (middle); the \( G \), or “gyroid” surface (right) (from [75]).](image)
of all known triply periodic minimal surfaces [403, 404]. However, their analysis has shown that the lamellar phase should be even more stable giving rise to a metastable gyroid.

Experimentally, gyroids can be regarded as the most commonly occurring cubic triply periodic minimal surfaces in nature. For example, they were observed in triblock copolymers [418], lipid-water mixtures [412, 413], and it was even suggested that they might exist in the endoplasmic reticulum [268].

Computationally, complex fluids as they are described in this chapter are too expensive to tackle with atomistic methods such as molecular dynamics, yet they require too much molecular detail as it is usually available with continuum Navier-Stokes solvers. Algorithms which work at an intermediate or “mesoscale” level of description have produced promising results during the last twenty years. In particular, the lattice Boltzmann method has been found highly useful for simulating complex fluid flows in a wide variety of systems.

The purely kinetic lattice Boltzmann method we use is able to model complex flows whose rheological properties are emergent from the mesoscopic kinetic processes without any imposed macroscopic constraints [68]. See chapter II-2 for details on our implementation.

Varieties of the lattice Boltzmann method have been used successfully to study the behavior of multi-phase flows in the past. A number of authors have investigated spinodal decomposition [9, 61, 76, 77, 158, 163, 240, 241, 343, 349, 489, 501] and the same phenomenon has also attracted some interest in the presence of shear, where structural transitions from isotropic to lamellar or tubular phases may occur [183, 192, 435, 488, 490, 500].

There have been only limited investigations of the influence of amphiphiles on the domain growth of otherwise immiscible fluids undergoing spinodal decomposition within the lattice Boltzmann method, despite the fact that ternary amphiphilic fluids have been studied by a number of authors [267, 291, 330, 455]. For example, it has been shown that the lattice Boltzmann method can be used to describe the self-assembly and the rheological properties of mesophases including the primitive $P$-phase [329] and the gyroid phase [155]. The gyroid mesophase in particular has been of major interest during the last few years, where the phase formation and structural properties [159, 161], the influence of defects [78, 184, 185, 360], as well as its properties under shear [155, 162] have been investigated. Some of these works are part of the current thesis since our research on multiphase lattice Boltzmann simulations of mesophases and microemulsions is presented in part III.

A substantial part of the results described in this chapter and part III was obtained within the RealityGrid project [371]. A central theme of this project was the facilitation of distributed and collaborative steering of parallel simulation codes. This allows the transparent access to a running simulation over the network and to change parameters or visualize output data on the fly from the user’s workstation. We were involved in the development of new approaches to computational steering and possibly one of the few groups from natural sciences being involved in the development of early computational grids. Therefore, a large fraction of the following chapters do not only cover questions related to physics or chemistry, but focus on technical aspects and the usability of emerging techniques as tools to improve our scientific knowledge.

Our massively parallel 3D lattice Boltzmann code (LB3D) was the software base for the work presented in the following chapters. LB3D is based on Shan and Chen’s multi-phase lattice Boltzmann model [415, 416] and can be utilized to simulate two miscible or immiscible fluids. In addition, amphiphilic molecules have been added to the model [68]. Collaborations with computer scientists and software developers have led to a large number of improvements to the simulation code. These include computational steering facilities as well as platform
independent checkpoint and restart techniques allowing to migrate a running simulation from one supercomputer to another. Also, the code has been ported to most available supercomputer platforms and shows a very good performance and scaling behavior. The Edinburgh parallel computer centre has awarded LB3D with its gold medal for scaling almost linearly to 1024 processors.

In chapter III-1, “Steering in computational science: mesoscale modelling and simulation”, we describe the benefits of computational steering for high performance computing applications. To our knowledge, this is one of the few papers in the literature which focuses on the applicability of computational steering and not only on the implementation of a particular steering environment. Lattice-Boltzmann mesoscale fluid simulations of binary and ternary amphiphilic fluids in two and three dimensions are used to illustrate the substantial improvements which computational steering offers in terms of resource efficiency and time to discover new physics. We discuss details of our first steering implementations and describe their future outlook with the advent of computational grids.

The following chapter III-2, “Large-scale grid-enabled lattice-Boltzmann simulations of complex fluid flow in porous media and under shear”, focuses on our three dimensional implementation of the lattice Boltzmann algorithm and shows first results from simulations of binary immiscible fluid flows through a porous medium derived from digitized x-ray microtomographic data of Bentheimer sandstone. In addition, we present studies of the same fluids under shear. While in the case of the porous media simulations, it is of importance to resolve the pores of the sandstone image sufficiently well, simulations of sheared fluids can easily suffer from finite size effects. Thus, in both cases very large scale simulations were needed reaching the limits of the available computing facilities. Simulations on these scales can benefit considerably from the use of computational steering and we describe our implementation of steering within the lattice-Boltzmann code making use of the RealityGrid steering library. The simulations presented in this chapter benefit from the new concept of capability computing, designed to prioritize the execution of big jobs on major supercomputing resources. The advent of persistent computational grids promises to provide an optimal environment in which to deploy these mesoscale simulation methods, which can exploit the distributed nature of compute, visualisation and storage resources to reach scientific results rapidly; we discuss our work on the grid-enablement of lattice-Boltzmann methods in this context.

The TeraGyroid experiment [33, 359] took place at the Supercomputing 2003 conference, where the main machines of the UK’s national HPC centres were coupled with the TeraGrid facilities in the US through a custom high-performance network. In total, about 5000 CPUs were part of this prototype computational grid. Collaborative steering sessions with active participants on two continents and observers worldwide were made possible through this approach. The scientific objective was to study the self-assembly, defect pathways and dynamics of liquid crystalline cubic gyroid mesophases. In particular, we were interested in the time dependent behavior of defects and grain boundaries. In order to obtain a valuable statistics and make sure the simulations are virtually free of finite size effects, we used the largest set of lattice-Boltzmann (LB) simulations ever performed, involving in some cases lattices of over one billion sites and for highly extended simulation times. Altogether we were able to use about 400000 CPU hours and generate two terabytes of simulation data within a period of two weeks.

Different visualization techniques such as volume rendering or isosurfacing of fluid density distributions allow the human eye to detect and track defects in liquid crystals because humans are easily capable of finding imperfections in repetitive spatial structures. However,
manual data analysis becomes too time consuming and algorithmic approaches are needed when there are large amounts of data as it was the case in the TeraGyroid project. In chapter III-3, “Detection and tracking of defects in the gyroid mesophase”, we present and compare two different approaches we have developed to study defects in gyroid mesophases of amphiphilic ternary fluids. While the first method is based on a pattern recognition algorithm, the second uses the particular structural properties of gyroid mesophases to detect defects.

In chapter III-4, “Large-scale lattice Boltzmann simulations of complex fluids: advances through the advent of computational grids”, we present a review paper on our achievements on computational steering and grid computing as performed within the RealityGrid framework. In addition, we report on our results of three dimensional simulations of ternary amphiphilic fluids. We focus on the description of the TeraGyroid project and a summary of the main scientific and technical results. This is of particular interest since we were one of the few scientific groups involved in the development of computational grids. Since smoothly working production grids are not yet available, we have tried to influence the direction of software development and grid deployment.

While the previous chapters mainly focus on technical issues, the following contributions address scientific aspects of binary and ternary amphiphilic fluid mixtures with a main focus being on the gyroid mesophase.

In chapter III-5, “Stress response and structural transitions in gyroid and lamellar amphiphilic mesophases: lattice-Boltzmann simulations”, we employ two gyroidal mesophases, the gyroid per se (above the sponge-gyroid transition, of high crystallinity) and the molten gyroids (within such a transition, of shorter-range order). We find that both mesophases exhibit shear-thinning, more pronounced and at lower strain rates for the molten gyroid. We also find that, at late times after the onset of shear, the skeleton of the longer-range gyroid becomes a structure of interconnected irregular tubes and toroidal rings, mostly oriented along the velocity ramp imposed by the shear. We also compare the shear stress and deformation of lamellar mesophases with and without amphiphile in a steady Couette flow applied in normal direction to the lamellae. We find that the presence of amphiphile cause the shear stress at late times to increase and allows the formation of rich patterns on the shear interface.

The following chapter III-6, “Emergence of rheological properties in lattice Boltzmann simulations of gyroid mesophases”, can be seen as an extension of chapter III-5. Here, we investigate in detail the occurrence of shear thinning in the self-assembled cubic gyroid mesophase and demonstrate that viscoelastic flow can be reproduced using our purely kinetic approach.

In the last chapter of part III, “Structural transitions and arrest of domain growth in sheared binary immiscible fluids and microemulsions”, we present systematic investigations of spinodal decomposition and structuring effects in binary immiscible and ternary amphiphilic fluid mixtures under shear. We show that the growth of individual fluid domains can be arrested by adding surfactant to the system, thus forming a bicontinuous microemulsion. We demonstrate that the maximum domain size and the time of arrest depend linearly on the concentration of amphiphile molecules. In addition, we conclude that for a well defined threshold value of amphiphile concentration, the maximum domain size and time of complete arrest do not change. For systems under constant and oscillatory shear we analyze domain growth rates in directions parallel and perpendicular to the applied shear. We find a structural transition from a sponge to a lamellar phase by applying a constant shear and the occurrence of tubular structures under oscillatory shear. The size of the resulting lamellae and tubes depends strongly on the amphiphile concentration, shear rate/amplitude and shear frequency.
Simulations of microfluidic applications: probing the boundary condition

During the last few decades the miniaturization of technical devices down to submicrometric sizes has made considerable progress. In particular, during the 1980s, so-called microelectromechanical systems (MEMS) became available for chemical, biological and technical applications. Popular examples of widely spread MEMS are impact detectors for airbag activation or inkjet printer heads. Such apparatuses employ flows of gases and liquids under so far unexplored conditions leading to the rise of a new discipline called “microfluidics” in the 1990s [447]. The first miniaturized gas chromatography system was already created in 1979 [452], but it took until 1991 for these new applications to take off at a large scale. In addition to chromatography [301], a wide variety of microfluidic systems was built [447]. These include electrophoretic separation systems [122], micromixers [244], DNA amplifiers [252], and chemical reactors [236]. Next to those “practical applications”, microfluidics was used to answer fundamental questions in physics including the behavior of single molecules or particles in fluid flow [426] or the validity of the no-slip boundary condition [272]. The latter is of particular interest for the current thesis as it provides the background for chapters IV-1 to IV-5.

Some years ago, it became possible to use cheap materials such as PDMS (polydimethylsiloxane) instead of silicon to combine even multiple devices onto a single chip. Such “labs on a chip” can be used to detect biological molecules, transport and mix fluids or analyse raw samples such as blood or food samples. “Labs on a chip” are also referred to as µTAS (micro-total analysis systems). During the last decade a substantial number of functioning devices has been presented in the literature. Indeed, quite a few companies started to commercialize such “labs on a chip”. For example, devices are available that transport a drop of blood by capillary forces though a filter into a functionalized microchannel. Here, myocardial proteins are detected in order to determine if a patient has suffered from a heart attack [447]. This way, the amount of blood needed can be reduced by orders of magnitude and the analysis time can be substantially shorter. However, most systems today are not completely integrated onto a single chip, but still require ordinary computer equipment and read-out machinery to be used.

Miniaturization favors heat exchange. Therefore, microfluidic devices are of special interest for the chemical industry since in miniaturized systems strongly endo- or exothermic reactions are less difficult to handle than in conventional devices. However, in order to achieve a throughput comparable to chemical factories as known today, massive parallelization would be needed. Therefore, so far microfluidic devices are limited to small scale applications. A more detailed introduction to the field of microfluidics can be found in the recent books [238, 447] and review articles [272, 332] on this topic.

Reynolds numbers in microfluidic systems are usually small, i.e., assuming flow velocities do not exceed 0.01 m/s and channel widths are of the order of tens of micrometers, the Reynolds number stays below 0.1. In addition, due to the small scales of the channels, the surface to volume ratio is high causing surface effects like wettability or surface charges to be more important than in macroscopic systems. Also, the mean free path of a fluid molecule might be of the same order as the characteristic length scale of the system. For gas flows, this effect
can be characterized by the so-called Knudsen number [245]

\[ Kn = \frac{\lambda}{l}. \] (4.1)

In practice, different regimes can be defined [238]:

- At small Knudsen number \((Kn < 0.01)\), the ordinary hydrodynamic equations are valid and the flow is described by the Navier-Stokes equations (All results presented in the following chapters are obtained for liquids, i.e., for small \(Kn\)).

- If the Knudsen number is between 0.01 and 0.3, one enters the “slip” regime. While the Navier-Stokes equations can be applied to the bulk flow, at the surface the last molecular layers have a velocity different from that of the surface. A typical way to model flows in this regime is to solve the Navier-Stokes equations with modified slip boundary conditions. This regime is easily reached in microfluidic gas flow applications.

- In the “intermediate regime” \((Kn\) between 0.3 and 10), the effects of rarefaction of volume become significant and the Navier-Stokes equations have to be replaced with more appropriate descriptions like the Burnett equations. Those are extremely hard to solve even for very simple systems.

- If \(Kn > 10\), we have a rarefied gas and the system has to be described using the Boltzmann equation.

While the Knudsen number provides a good estimate for when to expect rarefaction effects in gas flows, for liquids one would naively assume that its velocity close to a surface always corresponds to the actual velocity of the surface itself. This assumption is called the no-slip boundary condition and can be counted as one of the generally accepted fundamental concepts of fluid mechanics. However, some centuries ago there were long debates about the velocity of a Newtonian liquid close to a surface and the acceptance of the no-slip boundary condition was mostly due to the fact that no experimental violations could be found, i.e., a so-called boundary slip could not be detected. Thus, it is not surprising that already in 1823, Navier proposed a boundary condition where the fluid velocity at a surface is proportional to the shear rate at the surface [326] (at \(x = x_0\)), i.e.,

\[ v_z(x_0) = \beta \frac{\partial v_z(x)}{\partial x}. \] (4.2)

In other words, the slip length \(\beta\) can be defined as the distance from the surface where the relative flow velocity vanishes. Assuming a typical Poiseuille setup consisting of a pressure driven flow of an incompressible liquid between two infinite planes, the velocity in flow direction \((v_z)\) at position \(x\) between the planes is given by

\[ v_z(x) = \frac{1}{2\mu} \frac{\partial P}{\partial z} \left[ h^2 - x^2 - 2h\beta \right], \] (4.3)

where \(2h\) is the distance between the planes, and \(\mu\) the viscosity. \(\partial P/\partial z\) is the pressure gradient. In contrast to a no-slip formulation, the last term in Eq. 4.3 linearly depends on the slip length \(\beta\) (see figure 4.1).

In recent years, it became possible to perform very well controlled experiments that have shown a violation of the no-slip boundary condition in sub-micron sized geometries. Since then, mostly experimental [21, 41, 67, 72, 80, 81, 90, 92, 97, 272, 331, 461, 480, 484, 508],
but also theoretical works [151, 481], as well as computer simulations [19, 83, 249, 250, 441, 456, 457, 462] have been performed to improve our understanding of boundary slip. The topic is of fundamental interest because it has practical consequences in the physical and engineering sciences as well as for medical and industrial applications. Interestingly, also for gas flows, often a slip length much larger than expected from classical theory can be observed. Extensive reviews of the slip phenomenon have recently been published by Lauga et al. [272] and Neto et al. [332].

The reason for such findings is that the behavior of a fluid close to a solid interface is very complex and involves the interplay of many physical and chemical properties. These include the wettability of the solid, the shear rate or flow velocity, the bulk pressure, the surface charge, the surface roughness, as well as impurities and dissolved gas. Since all those quantities have to be determined very precisely, it is not surprising that our understanding of the phenomenon is still very unsatisfactory. Due to the large number of different parameters, a significant dispersion of the results can be observed for ostensibly similar systems [272, 332]. For example, observed slip lengths vary between a few nanometres [81] and micrometers [461] and while some authors find a dependence of the slip on the flow velocity [80, 97, 508], others do not [72, 461].

Most recent computer simulations apply molecular dynamics and report increasing slip with decreasing liquid density [250, 456] or liquid-solid interactions [83, 325], while slip decreases with increasing pressure [19]. These simulations are usually limited to some tens of thousands of particles, length scales of nanometres and time scales of nanoseconds. Also, shear rates are usually orders of magnitude higher than in any experiment [272]. Due to the small accessible time and length scales of molecular dynamics simulations, mesoscopic simulation methods as described in chapter II-2 are well applicable for the simulation of microfluidic experiments.

The experimental investigation of apparent slip can be based on different setups: either a fluid is pumped through a microchannel and the measured mass flow rate at the end of the channel is compared to the theoretical value with no slip boundary conditions. From the deviation of the two values, the magnitude of slip can be computed [460]. Another possibility is to measure the slip length directly using optical methods like particle image velocimetry [462]. Very popular is the modification of an atomic force microscope by adding a silicon sphere to the tip of the cantilever. In vicinity of the boundary, the sphere is vibrated with a given frequency and the gap between sphere and wall is constantly growing and shrinking. The fluid has to be pushed out of the gap when the sphere moves in the direction of the wall and will flow back when the sphere moves in the opposite direction. It is possible to measure
the amount of slip at the wall by comparing the force needed to move the sphere with its theoretical value [97, 331, 482, 484, 508].

During the last few years, the substantial scientific research invested in the slip phenomenon has lead to a more clear picture which can be summarized as follows: One can argue that many surprising results published were only due to artefacts or misinterpretation of experiments. In general, there seems to be an agreement within the community that slip lengths larger than a few nanometers are usually due to experimental artefacts. Small slip lengths are experimentally even harder to determine and require sophisticated setups such as the modified atomic force microscopes as described above. Here, small variations of the apparatus such as choosing a different shape of the cantilever or modifying the control circuit of the sample holder can lead to substantial variation of the measurements. Also, the theoretical equations correlating the measured force to the slip length are only valid for perfect surfaces and infinitely slow oscillations of the sphere. Therefore, it is of importance to perform computer simulations which have the advantage that most parameters can be changed independently without modifying anything else. Thus, the influence of every single modification can be studied in order to present estimates of expected slip lengths. Our research on microfluidic applications and in particular the slip phenomenon as observed in flows of simple liquids in microchannels follows this argument and is presented in part IV.

In chapter IV-1, “Lattice Boltzmann simulations of apparent slip in hydrophobic microchannels”, we model the interaction between hydrophobic channel walls and the fluid by means of a multi-phase lattice Boltzmann model. A few groups have applied the lattice Boltzmann method for the simulation of microflows and to study boundary slip. A popular approach is to introduce slip by generalizing the no-slip bounce back boundary conditions in order to allow specular reflections with a given probability [441, 462]. Another possibility is to modify the fluid’s viscosity, i.e. the fluid viscosity is modified due to local density variations in order to model slip [336]. In both cases, the parameters determining the properties at the boundaries are “artificial” parameters and they do not have any obvious physical meaning. Therefore, they are not easily mappable to experimentally available values. Our approach overcomes this problem by applying a mesoscopical force between the walls and the fluid. This force can be linked to the contact angle which is commonly used by experimentalists to quantitatively describe the wettability of a material [101, 504]. A similar approach is used by Benzi et al. [25]. Our mesoscopic approach is able to reach the small flow velocities of known experiments and reproduces results from experiments and other computer simulations, namely an increase of the slip with increasing liquid-solid interactions, the slip being independent of the flow velocity, and a decreasing slip with increasing bulk pressure. Within our model we develop a semi-analytical approximation of the dependence of the slip on the bulk pressure.

In chapter IV-2, “On the effect of surfactant adsorption and viscosity change on apparent slip in hydrophobic microchannels”, we extend these works in order to find a possible explanation for some experiments observing a slip length depending on the flow velocity which is contradictory to many theoretical results and simulations. Our explanation is that a velocity dependent slip can be detected if the flow profile is not fully developed within the channel, but in a transient state. Further, we demonstrate the effect of adding surfactant to a fluid flow in a hydrophobic microchannel. The addition of surfactant can shield the repulsive potential of hydrophobic walls, thus lowering the amount of slip with increasing surfactant concentration.

If typical length scales of the experimental system are comparable to the scale of surface roughness, the effect of roughness cannot be neglected anymore. This effect is studied in detail in chapter IV-3, “Roughness induced boundary slip in microchannel flows”. Figure 4.2 shows a typical example of a simulation setup: Poiseuille flow between two rough surfaces.
Figure 4.2: A typical simulated system: Poiseuille flow between two rough surfaces showing random surface variations. Streamlines depict a two dimensional cut and illustrate the parabolic velocity profile. This profile is distorted in the vicinity of the rough surfaces.

As can be observed in the figure, the stream lines of the flow are getting disturbed or trapped between the obstacles at the surfaces. We show that an apparent boundary slip can have its origin in the misleading assumption of perfectly smooth boundaries. We introduce an “effective no-slip plane” at an intermediate position between peaks and valleys of the surface.

Our lattice Boltzmann simulations show good agreement with analytical results for sinusoidal boundaries, but can be extended to arbitrary geometries and experimentally obtained surface data which cannot be treated analytically. We applied our simulations to data obtained from an atomic force microscope measurement of a gold coated glass surface. These data were provided by O. I. Vinogradova and H. Gong. We find that the detected apparent slip is independent of the detailed boundary shape, but only given by the distribution of surface heights. Further, we show that the slip diverges as the amplitude of the roughness increases and the flow field gets more restricted which highlights the importance of a proper treatment of surface variations in very confined geometries.

As an obvious extension of chapters IV-1 to IV-3, we study the interplay of surface roughness and hydrophobic fluid-surface interactions in chapter IV-4, “Simulation of fluid flow in hydrophobic rough microchannels”. The knowledge about the interplay of the two effects is still very limited because these properties cannot be decoupled easily in experiments. We present preliminary results of the dependence of a detected boundary slip on roughness-hydrophobicity coupling and show that roughness creates a non-linear effect on the slip length for hydrophobic boundaries. In addition, we extend the studies of flow along surfaces with Gaussian distributed roughness as presented in chapter IV-3 to different variations of the Gaussian. We find that the position of the effective surface depends linearly on the width of the distribution.

The last chapter in part IV, “Slip flow over structured surfaces with entrapped microbubbles”, covers the transition to a superhydrophobic state where gas bubbles or layers at the surface can have a strong impact on a detected slip. By presenting two-phase lattice-Boltzmann simulations of a Couette flow over structured surfaces we show that, in contrary to the common view, bubbles can lead to increased friction on the surface and even to negative slip. Our simulations indicate that this friction results from the increased roughness due to the bubbles even though the bubbles add slippery surfaces to the flow channel. The simulation method used allows the bubbles to deform due to the viscous stresses. Our results show shear-rate dependent slip, but the behavior is opposite to that found in some experiments. In fact, the slip lengths measured by us decrease with increasing shear due to a deformation of the bubbles. Possible applications of such surfaces in microfluidic devices are discussed.
II-4. Simulations of microfluidic applications: probing the boundary condition
Simulations of particle-laden flows

Mixtures of particles and fluid are ubiquitous in our daily life. Possible examples include the cacao drink which keeps separating into its constituents, tooth paste and wall paint which are mixtures of finely ground solid ingredients in fluids or blood which is made up of red and white blood cells suspended in a solvent. An extreme example is the sand on the beach which can be blown away by the wind. Adding a fluid to a dry granulate causes the behavior of the mixture to change dramatically and a host of unexpected phenomena arises. A typical example can be observed on the beach: whereas it is impossible to build a sand castle from dry sand, once just a little bit of water has been stirred into the sand, one can shape the resulting material almost arbitrarily into surprisingly complex arrangements. Adding even more fluid might result in the material loosing this stability. If we stir such a mixture, it behaves like a liquid of increased viscosity.

Long-range fluid-mediated hydrodynamic interactions often dictate the behavior of particle-laden flows. A typical example is a suspension consisting of hard spheres in a solvent which is subject to constant shear forces. Here, due to the external shear forces and particle-particle interactions, a particle concentration gradient orthogonal to the shear plane arises which might generate an overall particle drift orthogonal to the flow direction. This particle drift might cause inhomogeneities in the suspension causing the phenomenon of pseudo wall slip to arise and a particle free region near the wall forms.

The behavior of particles solved in fluids does not only depend on the material properties. Indeed, it can be tailored by various different techniques including surface manipulation by, e.g., coating the particles with polymers, adjusting their interactions, or by adding nano-particles, salt, or acid to the solvent. In addition, the volume fraction is an important parameter: at sufficiently high volume concentration, even a hard sphere suspension can crystallize. It can be shown that in three dimensions hard sphere suspensions crystallize above 54% and that on two dimensions even 90% of volume fraction are required.

If one considers particles in solution which have a diameter of the order of nanometers or a micrometer, one calls such a mixture a colloidal suspension. Colloidal particles are much larger than atoms or molecules, but small enough for Brownian motion and diffusion being of importance. For example, thermal fluctuations can prevent particles to sediment in the field of gravity. The field of colloid science has gained substantial interest during the last years and closely connects physics, chemistry, material science, biology, and several branches of engineering. Next to a vast amount of research papers, several books have been published covering the physics and chemistry of colloids [212, 264, 295, 323, 393, 400, 417].

When investigating the behavior of such systems, a number of different effects and interactions have to be taken into account which often are in a subtle interplay and act on different scales: first of all, while the particle sizes are on the microscopic scale, hydrodynamics also plays a role on the mesoscopic or even macroscopic scale. In addition to the Brownian motion and gravitational forces described above, depending of the particular material and concentration, particles can interact via electrostatic repulsion, van der Waals attraction, or depletion forces. The actual state of a colloidal suspension strongly depends on the balance of these forces. Especially the attractive van der Waals and depletion forces have drawn scientific
interest recently [364, 395, 459].

These interactions not only depend on the material properties of the solved particles, but also on the solvent itself. For example, changing the $p\text{H}$-value or the salt concentration of a suspension can cause the particle interaction forces to vary substantially. This strongly influences the local order of the particles, i.e., the microstructure and its dynamics: the particles might form clusters, be in a suspended state or even form repulsive structures and glasses. For industrial applications, the shear viscosity or cluster size distribution can be of interest in order to obtain high quality work pieces for example in ceramics processing.

Of particular interest for the research presented in this thesis are the properties of claylike colloids. As a model system for silt and clay particles, our collaborators within the DFG research group “Peloide” have chosen $\text{Al}_2\text{O}_3$ powder suspended in water [380, 381]. This powder is industrially available in large quantities and the salt concentration and $p\text{H}$-value can be easily controlled in the laboratory. A possible experiment utilizes a number of vessels filled with initially identical suspensions, i.e., the volume fraction of suspended particles is the same in all vessels. After adjusting the $p\text{H}$-value differently in the individual containers, sediments of different heights form over a time span of several weeks. From the different height one can conclude that the porosity strongly depends on the $p\text{H}$-value. This observation can be justified by a compression experiment: loose packings with high porosity are less resistant against a bearing pressure than closely packed particle beds [202].

However, such experiments can only observe macroscopic properties of the material. The question remains, why different types of sediments appear and how to quantify the variations. A well funded understanding of such properties is of particular importance for the construction of buildings, bridges, and streets. Already during the planning of such a construction one has to understand how much load can be deposited on the soil, or how large a foundation has to be.

In order to answer these questions, the knowledge at the level of the interactions between the individual soil particles has to be improved. Their surface chemistry, the salt concentration of the water contained in the pores of the soil, as well as the $p\text{H}$-value are the most important ingredients for an adequate description and determine the microstructure of the suspension. By computer simulations we can reproduce the cluster formation process and thus obtain the microscopic structure of the sediment. This process and its dependence on the interactions among the particles plays the key role for the formation of sediments, and essentially determines its mechanical properties.

Optical tweezers open a highly controlled way to study the dynamics of colloidal systems. Such optical tweezers trap a colloid (or even an atom) in the focus of a laser beam. They have been very successful for studying micrometer-sized objects in materials science, biological sciences as well as soft matter research [179, 333]. It is possible to handle single isolated colloids, to move them at a given velocity and to measure the drag force on the colloid with $p\text{N}$-resolution at the same time. Such a tool offers the possibility to control the colloidal system with very high accuracy.

A possible application is to use a colloidal suspension as a model system to understand effects from atomic and molecular physics as well as solid state physics. This is particularly convenient since the typical timescale (seconds) and length scale (micrometers) are far easier accessible then those of atoms and molecules (femtoseconds and angstroms). At the same time, the interaction potential between colloids can be tuned in many ways, e.g., by coating them with polymers or adding salt to the suspension.

Theoretically or in computer simulations, optical tweezer experiments can be described along
the same lines as suspensions are modeled. Within the scope of the current thesis, an optical
tweezer is applied to a colloid-polymer mixture, where a comparably large colloid is dragged
through a polymer solution. Such colloid-polymer mixtures have important applications in
technological processes such as oil recovery, food science, as well as in most biological sys-
tems. Here, non-equilibrium properties are of importance and interface effects can play an
important role. In particular, the structure of fluids in general changes in the vicinity of inter-
faces. In equilibrium situations this is known to lead to fluid structure mediated interactions,
i.e., the so-called solvation force or depletion interactions. Long-ranged structural changes
in the fluid cause an enhanced polymer density in front of the colloid, and a reduced polymer
density in its back. As a result, these interactions become long ranged when moving from
equilibrium to non-equilibrium. In addition, the inhomogeneous polymer distribution leads
to an enhanced friction felt by the colloid.

Most analytical results for the particle scale behavior of suspensions have been obtained at
vanishing Reynolds number ($Re = 0$). For large systems, scientists aim at an average, con-
tinuum description of the large-scale behavior. However, this requires time-consuming and
sometimes very difficult experimental measurements of phenomenological quantities such as
the mean settling speed of a suspension, the stress contributions in the system of the individ-
ual components (solid and fluid) as functions of, e.g., the solid volume fraction of the con-
stituents. Additionally, if one is interested in understanding microscopic effects which might
lead to macroscopic properties of a system, computer simulations are the most promising
approach. In chapter II-2 various techniques for the simulation of particle suspensions have
been presented. It is important to note that there is no perfect candidate that is able to simu-
late all systems of interest and to utilize the available resources as efficiently as possible. For
each individual problem, one has to choose the method of choice carefully: while stochastic
rotation dynamics is well suited to simulate systems like clay-like colloids where Brown-
ian motion is important, the lattice Boltzmann method is not able to resolve the stochastic
motion of the particles without modifications of the method. However, in cases where ther-
modynamic fluctuations are neglectably small, this approach is much more efficient than
stochastic rotation dynamics. Like conventional Navier-Stokes solvers, the fluid flow can be
resolved in great detail, but the lattice Boltzmann method is much easier to implement and to
parallelize than a conventional solver. It is of particular advantage if complicated boundary
conditions like non-spherical particles or complex channel geometries come into play. The
implementation of Navier-Stokes solvers on the other hand can be based on a long-standing
and widespread experience with these techniques allowing to create very efficient codes. If
long range hydrodynamic effects are not important as for example in very dense colloidal
systems, Brownian dynamics can be a very efficient method which includes thermal fluctua-
tions as well as a Stokes friction. As soon as the system becomes macroscopic and if heavy
particles are considered in a gas flow, the exact properties of the flow field are not necessary
to understand experimentally observable parameters. A typical example for such a system is
the movement of granular particles in air as occurring in pneumatic transport applications.
Here, computationally much less demanding techniques like a coarse-grained description of
the fluid should be applied.

Our numerical research on particle-laden flows is presented in part V. The first two chapters
utilize the lattice Boltzmann method for the fluid solver, whereas chapters V-3 to V-7 are
based on stochastic rotation dynamics. Only in chapter V-8 the Brownian dynamics technique
is applied.

In chapter V-1, “Transport phenomena and structuring in shear flow of suspensions near solid
walls”, we model an experiment which has been performed in the group of H. Buggisch in
Karlsruhe [56, 324]. In the experiment, a Couette setup is used to study structuring effects of
glass spheres in sugar solution under creeping shear. We apply the lattice-Boltzmann method
and the extension to particle suspensions as introduced by Ladd et al. to study transport phe-
nomena and structuring effects of particles suspended in a fluid near sheared solid walls. We
find that a particle free region arises near walls, which has a width depending on the shear rate
and the particle concentration. The wall causes the formation of parallel particle layers at low
concentrations, where the number of particles per layer decreases with increasing distance to
the wall. While extending our lattice Boltzmann simulations to large particle counts in order
to study dense suspensions in confined geometries, we observed that the velocity probability
distribution functions (PDF) of sheared hard-sphere suspensions can show pronounced
deviations from a Maxwell-Boltzmann distribution.

These velocity probability distributions are the subject of interest of chapter V-2, “Non-
equilibrium velocity distributions in sheared hard-sphere suspensions”. Here, we investigate
such suspensions by means of simulations using the same method as in chapter V-1 and a
single-particle theory. We demonstrate that the distribution function is symmetric around
zero velocity and shows a Gaussian core and an exponential tail over more than six orders
of magnitude of probability. To our knowledge, no other group has been able to obtain data
of such numerical accuracy so far allowing us to close the question about the actual shape of
the probability distribution function. Previous studies have reported on transitions between
Gaussian to exponential or stretched exponential distributions, but these findings were due to
insufficient statistics only. The single particle theory developed by E. Ben-Naim from Los
Alamos supports our findings. Following the excellent agreement of theory and simulation
data, we show that the probability distribution functions scale with the shear rate as well as
the particle volume concentration, and the kinematic viscosity.

If the particles become smaller, thermal fluctuations cannot be neglected anymore and one
has to use a simulation method that properly treats Brownian motion. While one possibility
would be to add fluctuations to an existing lattice Boltzmann code, we choose the stochastic
rotation dynamics technique to investigate properties of dense suspensions and sediments of
small spherical silt particles. In chapter V-3, “Simulation of claylike colloids”, we introduce
our combined molecular dynamics and stochastic rotation dynamics method which includes
van der Waals and effective electrostatic interactions between the colloidal particles. The
assumption of spherical particles in the molecular dynamics part of the simulation is an ap-
proximation, but it suggests an isotropic surface charge distribution which allows to assume
isotropic interactions between the particles. In absence of other charges, the electrostatic poten-
tial would decay with $1/r$, where $r$ is the distance between the centers of two particles.
However, equally charged co-ions are repelled and oppositely charged counter-ions are at-
tracted by the electrostatic potential leading to a charge separation and thus an exponential
screening of the electrostatic potential. In combination with attractive van der Waals forces
one obtains the well known DLVO potential [111, 479] (after Derjaguin, Landau, Vervey and
Overbeek). In addition to the particle-particle interactions, one has to adjust the hydrody-
namic parameters in the simulation code. Therefore, we introduce a scaling scheme which
improves the efficiency of the algorithm under the constraint that characteristic dimensionless
numbers like the Péclet number which expresses the importance of thermal motion for the
dynamics of the system, stay constant. In chapter V-3, we present the simulation technique
and first results. We show velocity distributions, diffusion coefficients, sedimentation veloc-
ities, and spatial correlation functions. In addition, we explore the phase diagram depending
on the parameters of the potentials and on the volume fraction.

In chapter V-4, “Shear viscosity of claylike colloids in computer simulations and experi-
ments”, simulations of dense suspensions and sediments of small strongly interacting Al$_2$O$_3$
in a shear cell are presented. In particular, we study structuring effects and the dependence
of the suspension’s viscosity on the shear rate and shear thinning for systems of varying salt concentration and pH-value. To show the agreement of our results to experimental data of our collaborators from Karlsruhe, the relation between bulk pH-value and surface charge of spherical colloidal particles is modeled by Debye-Hückel theory in conjunction with a 2pK charge regulation model. This model relates the microscopic particle-particle interactions to macroscopic parameters, i.e. pH-value and ionic strength $I$. The experimentally determined “ζ-potential” is used to adjust the remaining model parameters. In addition, a lubrication correction is introduced into the molecular dynamics part of the simulations in order to take the break down of short ranged hydrodynamic forces into account. Such lubrication corrections are known from other methods and are commonly used to solve the intrinsic problem that every numerical scheme can only resolve the hydrodynamics between two particles down to a certain scale.

In the following chapter V-5, “Stability diagram for dense suspensions of model colloidal Al$_2$O$_3$ particles in shear flow”, we investigate the fact that in such suspensions, depending on the experimental conditions, very different microstructures can be found, comprising fluid like suspensions, a repulsive structure, and a clustered microstructure. For technical processing in ceramics, the knowledge of the microstructure is of importance, since it essentially determines the stability of a workpiece to be produced. To enlighten this topic, we investigate these suspensions under shear. We observe cluster formation on two different length scales: on the scale of the distance of nearest neighbors and on the length scale of the system size. We find that the clustering behavior does not depend on the length scale of observation. If inter-particle interactions are not attractive the particles form layers in the shear flow. The results are summarized in a stability diagram. We demonstrate that the strengths of our simulations are that we are not only able to reproduce what is already known from experiments, but that simulations allow to study values which are difficult or even impossible to measure. A good example here is the structure factor. In principle, one can extract it from experimental scattering data. However, for the material of interest and in particular for high volume concentrations, scattering intensities in small angle X-ray scattering (SAXS) measurements are low. Our simulations can provide detailed data of the microstructure which are quasi inaccessible by experiments [202].

In particular the dynamical properties of the clustered microstructure are of interest in chapter V-6, “Formation and growth of clusters in colloidal suspensions”. In this chapter we investigate the clustered microstructure in order to gain statistics within a system containing numerous clusters. Therefore, large simulation volumes are needed. We present our parallel implementation of the simulation algorithm as well as a newly developed cluster detection and tracking algorithm. We then show first results of measured growth rates and cluster size distributions to validate the applicability of our method.

The last chapter covering the simulations of claylike colloids using coupled molecular dynamics and stochastic rotation dynamics methods is chapter V-7, “Computational steering of cluster formation in Brownian suspensions”. Due to the rich phase behavior of the systems of interest, many simulations are needed to explore the whole parameter space, when investigating the properties of the suspension depending on the experimental conditions. We have developed a steering approach to control a running simulation and to detect interesting transitions from one region in the configuration space to another. The advantages of the steering approach and the restrictions of its applicability due to physical constraints are illustrated by several example cases.

In the remaining chapter in part V, “Colloids dragged through a polymer solution: experiment, theory and simulation”, we model the experimental optical tweezer setup of C. Gutsche
and F. Kremer in Leipzig. As most other simulations of optical tweezer experiments we are aware of, we do not model long range hydrodynamic interaction, but utilize the Brownian dynamics technique in a similar manner as Reichhardt et al. [372, 373, 374]. We compare the results with a simplified dynamic density functional theory (DDFT) by M. Rauscher and M. Krüger. In the experiment, single isolated colloids in λ-DNA solutions are dragged using optical tweezers in order to study the dynamics of the polymer solution. This technique allows to measure the drag force on the colloid with pN-resolution at the same time. Our algorithm models the polymers and the colloid as hard spheres. The polymers are assigned the density of water and the volume of a sphere with the polymer’s radius of gyration. The colloid is trapped in a moving parabolic potential mimicking the optical tweezer. To our knowledge, we are the only authors simulating the trap as a parabolic potential. Other authors assume the driven colloid to move with a constant velocity. Since the fluctuations of the particle inside the trap contain important information on the dynamical properties of the solvent, those authors are not able to resolve them. For high DNA concentrations we find a significantly higher drag force than predicted by the Stokes-Einstein theory for the homogeneous solution, however, the force scales approximately linearly with the velocity. Our model reproduces very well the experimental and theoretical results.
Discussion and Outlook

Within the previous chapters and following parts of this thesis, a summary of our research is given. These projects have led to a number of results and answered some of our questions. However, they also opened new possibilities and drew our attention to new scientific problems. Therefore, in this chapter, we comment on our results and report on ongoing projects as well as ideas for future research.

In part III, we report on our achievements in the fields of high performance computing and in particular new concepts from the fields of grid computing and computational steering. We developed and applied our large scale multiphase lattice Boltzmann solver in order to study the behavior of fluid mixtures under shear or the rheological properties and self-assembly of the cubic gyroid mesophase. In particular the latter has lead to many interesting results including an improvement of our understanding of the defect behavior and the occurrence of shear thinning and viscoelasticity. Before, it was not clear that our simple numerical model is able to reproduce such features. Since it was shown that the lattice Boltzmann model is able to simulate mesophases, such as the $P$ and $G$ phases, it is an obvious future step to find the correct parameters in order to simulate other triply periodic minimal surfaces as they were described in chapter II-3. We started to study the properties of ternary mixtures consisting of two immiscible fluids and amphiphiles in more detail. Results regarding the influence of surfactant concentration on domain growth are already presented in chapter III-7. The results given here are for bicontinuous systems only. It is of further interest to investigate such systems with substantially different concentrations of the two immiscible fluids. In such asymmetric systems, it is expected that tubular structures or lamellar structures and related transitions due to external shear forces are more pronounced. It would be interesting to find out if our lattice Boltzmann model is able to capture the known features of such phases.

From a computational point of view, we have benefited from the recent achievements in the field of grid computing. Especially the investigations of the cubic gyroid mesophases were on this scale only possible due to a combined effort and close collaboration with computer scientists and computing centers. The Teragyroid project allowed us to harness 400000 CPU hours in two weeks in order to investigate formation and growth of the gyroid as well as the behavior of defects in such phases. Within this project, we demonstrated the usefulness, but also the problems of current implementations of steering software and grid middleware: especially if large scale computing resources are required, it is of importance to invest these resources in a responsible way. Steering may help to reduce the amount of CPU time wasted for parameter searches or to find regions of interest more quickly. On the other hand, being able to use supercomputers in various centers and even on different continents in a fully transparent manner can improve the efficiency of a scientist substantially. However, in its current state, the available systems cannot fulfill this dream: all systems known so far are only in a prototype state and not general and stable enough to be used on a daily basis. In fact, since there is no fully accepted standard used by many groups yet, it is expected to take another few years until grid computing or steering become widely used tools.

In part IV, we present our applications of the lattice Boltzmann method to microfluidic problems. The focus of our research is laid on the validation of the no-slip boundary condition. By
introducing a new model for hydrophobic fluid-surface interactions and studying pressure-driven flow in microchannels, we show that an experimentally detected slip can have its origin in hydrophobic interactions, but is constant with varied shear rates and decreases with increasing pressure. Adding surfactant to such microflows, causes the amphiphiles to align at the surface and to shield the hydrophobic interactions, thus lowering the detected slip. Another effect that was not fully understood so far is the influence of surfaces roughness. We show that ignoring roughness can lead to large errors in a detected slip. In fact, we propose that roughness alone could often be the reason for apparent boundary slip. In these chapters, we also demonstrate the suitability of the lattice Boltzmann method for modeling microfluidic applications: in contrast to molecular dynamics, it is able to reach experimentally available time and length scales. This allows us to compare our results to experimental data directly and to simulate flow along surface data obtained from AFM measurements of “real” samples.

Typical surface force apparatus based slip measurements as performed by many groups [97, 331, 482, 484, 508] can be modeled as follows: a single particle is oscillated next to a surface and when it comes close to the surface it has to push away the fluid at the boundary. The force required for this is dependent on the presence of a slip and can be measured in our simulations. In these simulations we combine our knowledge obtained from the projects presented in parts IV and V, i.e., from simulations of fluid flow in confined geometries and particle suspensions in order to model a sphere moving in the vicinity of a surface and the corresponding flow field. Thus, we are able to resolve all important details of the experiment. We reproduce the theoretical force versus distance curves if the surface is perfectly smooth and extend our work to surfaces consisting of regular sphere packings. Also, first studies of flow along surface data obtained from AFM measurements are being performed in collaboration with O. I. Vinogradova. A typical snapshot from such a simulation is given in Fig. 6.1. Currently, we study the influence of surface properties, particle size and velocity as well as the influence of the particle’s surface properties on the measured slip length. We are able to study the influence of an accelerated movement of the suspended particle and the possible misalignments of the experimental apparatus on the slip length. In addition, we are improving our understanding of the exact flow properties around spherical particles close to a wall with slip.

A natural continuation of our previous works on roughness induced apparent boundary slip and the collaboration mentioned above is the analysis of flow along superhydrophobic surfaces. While in typical experiments, slip lengths of a few tens of nanometers can be observed, it would be preferable for technical applications to increase the throughput of fluid in a microchannel, i.e., to obtain substantially larger slip. Superhydrophobic surfaces are promising in this context, since it has been recently predicted [91] and experimentally reported [353]
that the so-called Fakir effect or Cassie state considerably amplifies boundary slippage. Such a situation can be achieved with highly rough hydrophobic surfaces, where instead of entering the area between the rough surface elements, the liquid remains at the top of the roughness, thus trapping some air in the interstices, and therefore leading to a very small liquid-solid contact area. We presented some studies on slip measurements of flow over bubble mattresses in chapter IV-5. As a continuation of this project, it is planned to determine the influence of bubbles on an AFM based slip measurement and to study the detected slip in dependence on the deformation of a bubble due to the moving sphere.

Figure 6.2: Snapshot of a hybrid molecular dynamics (bottom) and stochastic rotation dynamics (top) simulation.

As already described in this thesis, the properties of fluid flows in micro- or nanoscale geometries are strongly determined by the detailed material properties of the fluid as well as the surface. The lattice Boltzmann method cannot resolve the full atomic details of the involved materials and the method of choice here would be the molecular dynamics algorithm. However, molecular dynamics simulations are not able to reach the system sizes and simulation times we are interested in to compare our simulations to experimental data. Therefore, hybrid or multiscale methods are promising: the molecular details are needed only for the surface itself and the fluid in its very close vicinity. Our idea is to simulate the really relevant volume and surface areas using molecular dynamics and to describe the bulk fluid with a mesoscale method like the lattice Boltzmann method or stochastic rotation dynamics. At the boundary between both domains, a proper treatment of mass and momentum transfer is required. There are already methods available to tackle the boundary between domains simulated using different methods. However, some of them need large overlapping regions to be simulated by both methods resulting in unnecessarily increased computational effort. Others utilize the mass and momentum flux at the interface and are probably more suitable in our context. Thus, we are working on the development and implementation of such new hybrid approaches in order to properly model the interaction between fluid and surface. A preliminary example is depicted in Fig. 6.2. Here, the grey part shows a region modeled using molecular dynamics for both the surface at the bottom and the fluid above. The particles interact via Lennard-Jones potentials. As soon as a particle crosses the boundary to the red region, it is treated using the stochastic rotation dynamics method. In this bulk area, the computational effort is substantially lower since the interactions are only described in a very simplified way. In addition, a much longer timestep can be chosen.

A further interesting application are mixing processes in microchannels. Since the flow at these scales is laminar and no turbulence occurs, it is a highly non-trivial task to mix two fluids. In the past, a number of specially designed geometries have been proposed in order to enhance the mixing process. A common example is the so-called herring bone mixer [439], where the streamlines of a fluid follow a chaotic motion. We are simulating such devices using the lattice Boltzmann method (see Fig. 6.3 for an example) and optimize geometries.
In addition, we try to find new methods in order to determine the mixing quality of such devices.

In part V, we demonstrate that the lattice Boltzmann method and its extension to particle suspensions can be used to reproduce experimentally measured transport phenomena and structuring effects in sheared hard sphere suspensions. Further, we present high precision measurements of the probability distribution of the particle velocity and show that it has a Gaussian core and exponential tails. These distributions are a consequence of the irreversible nature of the driving process. On average, particles gain energy by external shear forces causing particle-particle collisions but lose energy due to viscous damping.

In the following chapters of part V, we apply combined molecular dynamics and stochastic rotation dynamics simulations to study Brownian, electrostatically interacting suspensions. We investigate Al₂O₃ powder suspended in water as a model system for clay and silt and propose a stability diagram, where the occurrence of clustered, repulsive or freely suspended phases depends on the pH-value and salt concentration of the suspension. A particular focus in these chapters is on the cluster formation processes as well as a comparison to experimentally obtained shear viscosities. Future projects may involve the investigation of experimentally observed viscoelasticity in such suspensions and an extension of our models to asymmetric particles in order to study the properties of laponite.

We demonstrated that mesoscopic simulation methods are valuable tools to model optical tweezer experiments in the last chapter of part V. Here, we reproduce drag forces obtained by dragging a large colloid through a polymer solution. It is now planned to extend these simulations in order to study the influence of oscillatory moving tweezers. Currently, we apply the same approach based on Brownian dynamics in order to study the properties of colloidal crystals. As shown in Fig. 6.4, we are able to investigate the occurrence of defects and the relaxation of the crystal after the dragged colloid passed by.
While hard particles suspended in a fluid have many applications, soft and deformable objects such as vesicles or membranes are common in biological systems. One advantage of the mesoscopic methods described in this thesis is the ability to resolve the flow field around such objects and the possibility to study not only single objects in flow, but large numbers of them. We are extending our lattice Boltzmann codes in order to study these systems by introducing deformable membranes instead of spherical and hard particles. The final goal of this project is to study the behavior of red blood cells in clogged arteries, but the same method can also be applied to model a vesicle. A typical example of a simulation showing a vesicle performing tank treading is shown in Fig. 6.5.
Lattice Boltzmann simulations of mesophases and microemulsions
Steering in computational science: mesoscale modelling and simulation

J. Chin, J. Harting, S. Jha, P. V. Coveney, A. R. Porter, and S. M. Pickles
Steering in computational science: mesoscale modelling and simulation

J. Chin†, J. Harting‡, S. Jha†, P. V. Coveney†, A. R. Porter‡ and S. M. Pickles††

This paper outlines the benefits of computational steering for high performance computing applications. Lattice-Boltzmann mesoscale fluid simulations of binary and ternary amphiphilic fluids in two and three dimensions are used to illustrate the substantial improvements which computational steering offers in terms of resource efficiency and time to discover new physics. We discuss details of our current steering implementations and describe their future outlook with the advent of computational grids.

1. Introduction
Many phenomena in condensed matter physics operate at length and time scales which are too large for detailed microscopic modelling. In microscopic models, based on classical molecular dynamics, usually every atom or molecule within a system is considered, resulting in rapidly increasing complexity for increasing problem sizes. This tends to restrict the currently treatable length scales to the order of several nanometres and the timescales to the order of nanoseconds since the computing power required for larger length and timescales is unavailable today. For example, a microscopic description of a fluid would track the position, momentum, and energy of every single fluid molecule. At room temperature, a microscopic model of a single cubic millimetre of monatomic gas would have to deal with around $10^{17}$ variables. Macroscopic models, on the other hand, usually deal with a smaller number of variables, often more closely related to physical observables. For example, a macroscopic model of a fluid would describe its velocity, density and temperature at various points.

Statistical mechanics is used to extract macroscopic, thermodynamic descriptions from the underlying microscopic representation. While standard methods exist for performing this contraction of description for systems at thermodynamic equilibrium [1], we shall be primarily interested in time-dependent, non-equilibrium systems for which there is less widespread agreement about their statistical mechanical description. In addition, comparatively few established methods exist for the treatment of complex systems which contain processes operating on several length and time scales. This situation presents a general problem for both microscopic and macroscopic models. Whereas in microscopic descriptions, a vast amount of computational effort is required to model mesoscopic systems spanning several length and time scales, macroscopic treatments omit fine details which may give rise to the characteristic behaviour of the system.

Because of the shortcomings of microscopic and macroscopic descriptions, and the tremendous importance of fluid dynamics, there is currently considerable interest in mesoscale models. These models coarse grain most of the atomic or molecular details but retain enough of the essential physics to describe the phenomena of interest. They are intended to treat systems at intermediate length scales between several nanometres and a few millimetres, and processes operating on multiple length and time scales. Examples of such systems can be found in everyday life: detergents, shampoos, milk, blood, and paint. These are materials whose macroscopic behaviour is induced by their microscopic and/or mesoscopic properties. A very descriptive example can be observed in the kitchen and can easily be reproduced by the reader: due to the microscopic interactions between starch molecules in cornflour, as the starch molecules jam into one another, a mixture of cornflour and water becomes more difficult to stir quickly.

Computer modelling forms a valuable tool in understanding the behaviour of mesoscale systems [2–5], and
much time and effort is currently being invested in such
techniques. In this article, we provide a brief overview of a
few such mesoscale modelling techniques, and focus in
particular on the lattice-Boltzmann fluid dynamical meth-
od. We describe its implementation, and the problems
which arise when the computer implementation is run
’statically’, without interaction with either the user or other
computational components. These problems may be solved
by adding extra functionality which permits the interopera-
tion of the lattice-Boltzmann code with other programs,
such as separate code to monitor and readjust the simulation
in real-time, or a user interface to permit a
human to ‘steer’ the simulation as it runs. We give examples
of two such functionalities: in one, the functionality of the
simulation code is made available, through the use of a
wrapper layer, to a high-level language, which permits
versatile control of simulations; in the other, the code is
connected to a general-purpose steering library which
permits users to remotely control many kinds of simulation
as they run.

Mesoscale simulations often require access to high-end
computational and visualisation resources; we therefore
proceed to discuss computational steering in such a general
context, and how it may permit much more efficient use of
such resources.

The literature on computational steering is quite
substantial [6]. However, most papers [7,8] have focussed
on the design and architectural details of computational
steering and then go on to give a prototype implementation
of their steering system. Our paper represents a different
approach, in that our motivation is a scientific problem for
which computational steering is shown to be an effective
tool. We do this by highlighting the advantages that
computational steering brings over traditional non-steered
simulations.

Computational grids are an increasingly popular
paradigm of computation, somewhat akin to traditional
distributed computing, yet with a major extension in
that they enable the transparent sharing and collective
use of resources (anything from spare PC CPUs to
databases or high-end hardware), which would otherwise
be individual and isolated facilities. Therefore, we
discuss how the advent of computational grids is
expected to considerably facilitate modelling with high-
performance computers in general, and computational
steering in particular.

2. Mesoscale modelling and simulation methods

Many mesoscale situations of interest involve fluids,
particularly mixtures of fluids which exhibit complicated
behaviour due to the interactions of their individual
molecules. When attempting to model such systems, one
must treat both the bulk flow, or hydrodynamic behaviour,
requires Boolean operations, and that it is unconditionally numerically stable [18].

Fluid mixtures may be simulated by introducing different species onto the lattice, often denoted by colour: red particles may represent oil, and blue particles water. Interactions between the different species may then be introduced by imagining that particles carry a ‘colour charge’, and experience a force due to the colour field generated by surrounding particles [19].

The red and blue fluids may be made immiscible by introducing a force which compels red particles to travel up the colour-field gradient towards regions of higher red density, and compels blue particles to travel towards regions of higher blue density; a mixture of red and blue particles will then separate into separate single-colour regions.

Porous media may be simulated by blocking off some lattice sites: any particles which would travel into the blocked sites during the advection step are bounced back to travel in the opposite direction, producing a no-slip boundary. The flow of oil-water mixtures in porous media has been successfully modelled using this technique [20].

An amphiphilic particle, such as a detergent molecule, typically contains two parts: a water-loving head, and an oil-loving tail. The behaviour of such molecules can be very complicated. In a mixture of oil and water, such a particle will seek out regions of oil-water interface, and reduce the interfacial tension. Solutions of an amphiphile in water may spontaneously assemble to produce a variety of different phases, ranging from simple spherical or wormlike clusters called ‘micelles’, to extensive sponge-like phases [21].

The ‘coloured particle’ description may be extended to cover the case of amphiphiles, by introducing a new species of mesoscopic particle, which has an orientational degree of freedom. A single such mesoscopic particle can be regarded as consisting of a red particle and a blue particle bolted together, so that the whole particle possesses no net colour charge, but will tend to align itself with the colour field; the orientation of the mesoscopic particle represents some sort of average of the orientations of its constituent molecules. Lattice gas models with amphiphile particles have been used to simulate the effect of surfactants on oil-water mixtures in porous media [22], and the self-assembly of micelles [23,24].

In the model proposed by Malevanets and Kapral [25], sometimes called the ‘Real-coded Lattice Gas’, or ‘Discrete Simulation Automaton’ (DSA) model, the mesoscopic particles occupy discrete cells in space, but are permitted to have real-valued velocities. During the collision step of the algorithm, the velocities of the particles in each cell are transformed according to

\[ v_i \rightarrow V + \omega_i (v_i - V), \]

where \( \omega_i \) is a random rotation and \( V \) the centre of mass velocity of particles in a cell. The total momentum in each cell remains unchanged, producing hydrodynamic behaviour, while the randomization of the velocities produces dissipative behaviour. Particles then travel along their velocity vectors to nearby cells. This model may be generalized to treat immiscible fluid mixtures [26] and amphiphiles [27] in much the same manner as with LGA models.

The lattice Boltzmann (LB) method is a simplification of LGA: particles have discretized positions and momenta, but rather than individual Boolean particles being tracked around a lattice, their real-valued population is stored, resulting in a less noisy method. We examine LB in more detail in section 3.

3. Lattice Boltzmann models of immiscible and amphiphilic fluids

The lattice Boltzmann algorithm is a powerful method for simulating fluid flow. Much of its power lies in the ease with which boundary conditions can be imposed, and with which the model may be extended to describe mixtures of interacting complex fluids. Rather than tracking the state of individual atoms and molecules, as is done in molecular dynamics, or tracking individual discrete mesoscopic ‘packets of fluid’, as in LGA or DSA algorithms, the lattice Boltzmann method describes the dynamics of the single-particle distribution function of mesoscopic fluid packets.

3.1. The Continuum Boltzmann Equation

In a continuum description, the single-particle distribution function \( f_i(\mathbf{r}, \mathbf{v}, t) \) represents the density of fluid particles with position \( \mathbf{r} \) and velocity \( \mathbf{v} \) at time \( t \), such that the density and velocity of the macroscopically observable fluid are given by \( \rho(\mathbf{r}, t) = \int f_i(\mathbf{r}, \mathbf{v}, t) \, d\mathbf{v} \) and \( \mathbf{u}(\mathbf{r}, t) = \int f_i(\mathbf{r}, \mathbf{v}, t) \mathbf{v} \, d\mathbf{v} \) respectively. In the non-interacting, long mean free path limit, with no externally applied forces, the evolution of this function is described by the famous Boltzmann equation,

\[ (\partial_t + \mathbf{v} \cdot \nabla)f_i = \Omega[f_i]. \]

The left hand side of the equation describes changes in the distribution function due to free particle motion; the right hand side contains the collision operator \( \Omega \), describing changes due to pairwise collisions. Typically, this is an integral expression which can be hard to work with, so it is often simplified [28] to the linear Bhatnagar-Gross-Krook, or BGK form:

\[ \Omega[f] \simeq -\frac{1}{2} [f - f^{(0)}]. \]
The BGK collision operator describes the relaxation, at a rate controlled by a characteristic time $t$, towards a Maxwell-Boltzmann equilibrium distribution $f^{\text{eq}}$. While this is a drastic simplification, it can be shown that distributions governed by the Boltzmann-BGK equation conserve mass, momentum, and energy, and obey a non-equilibrium form of the Second Law of Thermodynamics [29]. Moreover, it can be shown [29,30] that the well-known Navier-Stokes equations for macroscopic fluid flow are obeyed on coarse length and time scales by such distributions.

### 3.2. The Lattice Boltzmann Equation

In a lattice Boltzmann formulation, the single-particle distribution function is discretized in time and space. The positions $\mathbf{r}$ on which $f_i(\mathbf{r},v,t)$ is defined are restricted to points $\mathbf{r}_i$ on a lattice, and the velocities $\mathbf{v}$ are restricted to a set $\mathbf{c}_i$ joining points on the lattice as shown in figure 1; hence, $f_i(\mathbf{r},t) = f_i(\mathbf{r},\mathbf{c}_i,t)$ represents the density of particles at lattice site $\mathbf{r}$ travelling with velocity $\mathbf{c}_i$ at timestep $t$. The density and velocity of the simulated fluid are now given by

$$
\rho(\mathbf{r}) = \sum_i f_i(\mathbf{r}), \quad (4)
$$

$$
\mathbf{u}(\mathbf{r}) = \sum_i f_i(\mathbf{r})\mathbf{c}_i, \quad (5)
$$

The lattice must be chosen carefully [31] to ensure isotropic behaviour of the simulated fluid. It can be shown [32] that the lattice Boltzmann equation may be rigorously derived by discretizing the continuum Boltzmann equation; alternatively, it may be regarded as a Boltzmann-level approximation of its ancestor, the LGA [33].

The discretized Boltzmann-level description of the fluid may now be evolved according to a two-step procedure. In the collision step, particles at each lattice site are redistributed across the velocity vectors; this process corresponds to the action of the collision operator, and usually takes the BGK form:

$$
f_i \leftrightarrow f_i - \frac{1}{t} [f_i - N_i],
$$

where $N_i = N_i[\rho(\mathbf{r}), \mathbf{u}(\mathbf{r})]$ is a polynomial function of the local density and velocity, which may be found by discretizing the well-known Maxwell-Boltzmann equilibrium distribution.

In the advection step, values of the post-collisional distribution function are propagated to adjacent lattice sites: this corresponds to particles streaming along their velocity vectors, and is the discretized equivalent of the left-hand side of the continuum Boltzmann equation:

$$
f_i(\mathbf{r} + \mathbf{c}_i) \leftrightarrow f_i(\mathbf{r}).
$$

Overall, the system obeys the lattice Boltzmann equation (LBE), produced by combining the two evolution steps:

$$
f_i(\mathbf{r},t+1) = f_i(\mathbf{r},t) + \frac{1}{t} \left[ f_i(\mathbf{r},t) - N_i[\rho(\mathbf{r}), \mathbf{u}(\mathbf{r})] \right].
$$

It can be shown that the resulting macroscopic density and velocity fields obey the Navier-Stokes equations [34].

A well-known drawback of the lattice Boltzmann method is that it is typically not guaranteed to be numerically stable, and will crash or produce physically unreasonable results if, for example, the forcing rate applied to a fluid is too high or if the interparticle interaction strength is set too high.

### 3.3. Multicomponent Interacting lattice Boltzmann Scheme

There are several schemes for generalizing the lattice Boltzmann algorithm to treat multicomponent fluids, including analogies with LGA [5], imposition of free-energy functionals [35], discretization of a modified form of the continuum Boltzmann equation [36], or inclusion of an explicit forcing term in the collision operator. The lattice Boltzmann algorithms described in this paper use the last approach, due to Shan and Chen [37].

The single-particle distribution function $f_i$ may be extended to the form $f_i^\sigma$, where each component is denoted by a different value of the superscript $\sigma$, so that the density

```
\rho(\mathbf{r}) = \sum_{\sigma} f_i^{\sigma}(\mathbf{r}),
```
and momentum of a single component $s$ are given by $\rho^s = \sum_i f_i^s$ and $\rho^s u^s = \sum_i f_i^s c_i$ respectively. The lattice BGK equation (8) now takes the form

$$f_i^s(x, t + 1) - f_i^s(x, t) = -\frac{1}{\tau_s} [f_i^s - N_s(\rho^s, \mathbf{v}^s)].$$  \hfill (9)

The velocity $\mathbf{v}^s$ is found by calculating a weighted average velocity

$$\mathbf{u}' = \left( \sum_{s} \rho^s \mathbf{v}^s \right) / \left( \sum_{s} \rho^s \right),$$  \hfill (10)

and then adding a term to account for external forces,

$$\mathbf{v}^s = \mathbf{u}' + \frac{\tau_s}{\rho^s} \mathbf{F}^s.$$  \hfill (11)

The force term $\mathbf{F}^s$ can take the form $g \rho^s \mathbf{z}$ to produce a gravitational force acting in the $z$-direction. In order to produce nearest-neighbour interactions between components, it assumes the form

$$\mathbf{F}^s = -\psi^s(\mathbf{x}) \sum_s g_{ss} \sum_i \psi^s(\mathbf{x} + \mathbf{c}_i) \mathbf{c}_i,$$  \hfill (12)

where $\psi^s(\mathbf{x}) = \psi^s(\rho^s(\mathbf{x}))$ is an effective charge for component $s$ and $g_{ss}$ is a coupling constant controlling the strength of the interaction between two components $s$ and $\bar{s}$. If $g_{ss}$ is set to zero for $s = \bar{s}$, and a positive value for $s \neq \bar{s}$ then, in the interface between bulk regions of each component, particles experience a force in the direction away from the interface, producing immiscibility. In two-component systems, it is usually the case that $g_{ss} = g_{ss} = g_{ss0}$.

Amongst other things, this model has been used to simulate spinodal decomposition [38,39], polymer blends [40], liquid-gas phase transitions [41], and flow in porous media [42].

### 3.4. Amphiphilic lattice Boltzmann

As with many other mesoscopic fluid methods, amphiphilic fluids may be treated in the LB framework by introducing a new species of particle with an orientational degree of freedom [43]. The particles of this species are each given a vector dipole moment $\mathbf{d}$ which has maximum magnitude $d_0$, corresponding to complete alignment of the constituent molecules. This is represented in the model by a dipole field $\mathbf{d}(\mathbf{x}, t)$ representing the average orientation of any amphiphilic present at site $\mathbf{x}$. In the advection step, values of $\mathbf{d}(\mathbf{x}, t)$ are propagated around the lattice according to

$$\rho^s(\mathbf{x}, t + 1) \mathbf{d}(\mathbf{x}, t + 1) = \sum_{\mathbf{a}} f_i^s(\mathbf{x} - \mathbf{c}_i, t) \mathbf{d}(\mathbf{x} - \mathbf{c}_i, t),$$  \hfill (13)

where tildes denote post-collision values. During the collision step itself, the dipole moments evolve in a BGK process controlled by a dipole relaxation time $\tau_d$:

$$\mathbf{d}(\mathbf{x}, t) = \mathbf{d}(\mathbf{x}, t) - \frac{1}{\tau_d} \left[ \mathbf{d}(\mathbf{x}, t) - \mathbf{d}^{ss}(\mathbf{x}, t) \right].$$  \hfill (14)

The equilibrium dipole moment $\mathbf{d}^{ss}$ is aligned with the colour field $\mathbf{h}$:

$$\mathbf{d}^{ss} \sim \frac{\beta d_0}{\chi} \mathbf{h}.$$  \hfill (15)

The colour field contains a component $\mathbf{h}^r$ due to coloured particles such as oil and water, and a part $\mathbf{h}^c$ due to dipoles. The former can be found from the populations of surrounding lattice sites,

$$\mathbf{h}^r(\mathbf{x}, t) = \sum_{\mathbf{a}} \sum_{\mathbf{c}_0} \rho^s(\mathbf{x} + \mathbf{c}_0) \mathbf{c}_0,$$  \hfill (16)

where $\rho^s$ is a colour charge, such as +1 for red particles, −1 for blue particles, and 0 for amphiphile particles. The field due to other dipoles turns out to be given by

$$\mathbf{h}^c(\mathbf{x}, t) = \sum_{\mathbf{a}} \sum_{\mathbf{c}_0} \rho^s(\mathbf{x} + \mathbf{c}_0) \mathbf{c}_0 \mathbf{d}(\mathbf{x}, t),$$  \hfill (17)

where the second-rank tensor $\theta_j$ is defined in terms of the unit tensor $\mathbf{l}$ and lattice vector $\mathbf{c}_i$ as

$$\theta_j = \mathbf{l} - \frac{\mathbf{c}_i \mathbf{c}_j}{c_i^2}.$$  \hfill (18)

In the presence of an amphiphilic species, the force on an oil or water particle includes an additional term $F_{\mathbf{z}}^{ss}$ to account for the colour field due to the amphiphiles. By treating an amphiphilic particle as a pair of oil and water particles with a very small separation $\mathbf{d}$, and Taylor-expanding in $\mathbf{d}$, it can be shown that this term is given by

$$F_{\mathbf{z}}^{ss}(\mathbf{x}, t) = 2 \psi^s(\mathbf{x}, t) g_{ss} \sum_{\mathbf{c}_0} \mathbf{d}(\mathbf{x} + \mathbf{c}_0) \cdot \theta_j \psi^s(\mathbf{x} + \mathbf{c}_0),$$  \hfill (19)

where $g_{ss}$ is a constant controlling the strength of the interaction between amphiphiles and non-amphiphiles. While they do not possess a net colour charge, the amphiphiles also experience a force due to the colour field, consisting of a part $F_{\mathbf{c}}^{ss}$ due to ordinary species, and a part $F_{\mathbf{z}}^{ss}$ due to other amphiphiles. These terms are given by

$$F_{\mathbf{c}}^{ss}(\mathbf{x}, t) \mathbf{d}(\mathbf{x}, t) \cdot \sum_{\mathbf{a}} g_{ss} \sum_{\mathbf{c}_0} \theta_j \psi^s(\mathbf{x} + \mathbf{c}_0),$$  \hfill (20)
\[ F^{s} = -\frac{4D}{c^2} g_{s,s}(x) \sum_i \left\{ \bar{d}(x + c_i) \cdot \theta_i \cdot \bar{d}(x) c_i + \bar{d}(x + c_i) \bar{d}(x) + \bar{d}(x) \bar{d}(x + c_i) \right\} \psi(x + c_i). \]  

(21)

To summarize, the interactions between fluid components are governed by the coupling constants \( g_{s,s} \) and \( g_{s,n} \), controlling the interaction between different sorts of coloured particles, between coloured particles and amphiphiles, and between the amphiphiles.

While the form of the interactions seems straightforward at a mesoscopic level, it is essentially phenomenological, and it is not necessarily easy to relate the interaction scheme or its coupling constants to either microscopic molecular characteristics, or to macroscopic phase behaviour. Some theoretical progress has been made in relating LGA amphiphile models to an underlying microscopic model [44], although macroscopic behaviour is very sensitive not only to the values of the coupling constants, but also to the concentrations of each species present, \textit{inter alia}. Different values of these parameters will give rise to a wide variety of different phases [21], such as spherical and wormlike micelles, sponges, lamellae, or droplets: the phase behaviour can be very difficult to predict beforehand from the simulation parameters, and brute-force parameter searches are often resorted to [24].

### 4. The practicalities of the lattice Boltzmann method

If the sites of a lattice Boltzmann grid are evolved according to the algorithm described in section 3, then the state of each site at a given timestep depends only on its state and the state of the neighbouring cells at the previous timestep, so that LB can be considered a form of cellular automaton [3,4]. This spatial locality of the algorithm translates into memory locality in implementation, allowing for efficient performance on contemporary commodity computer architectures which use caching techniques to improve the speed of memory access, but also enables extremely efficient implementation on massively parallel computer architectures since, for a lattice split across CPUs (spatial domain decomposition), only the state of the lattice sites at the edge of each CPU’s chunk of the lattice must be communicated to other CPUs. A more detailed examination of LB performance and a comparison with LGA is available in [45].

The following sections describe how two existing lattice Boltzmann codes were modified to allow for different forms of computational steering. One code, LB2D, is a light-weight, single-CPU solver for two-dimensional problems; steering was not directly bolted on to this code, but instead a high-level scripting interface was added to allow simulations to be controlled at runtime, either through high-level scripts, interactive manipulation, or other processes. The other code, LB3D, is a solver for three-dimensional problems involving ternary amphiphilic fluids, designed for use on distributed-memory parallel processing architectures; steering was added by interfacing the code to a separate steering library.\(^1\)

#### 4.1. Design of a typical lattice Boltzmann code

The lattice Boltzmann codes we examine in this article each revolves around a single data structure, which encapsulates the entire state of a simulation at a single instant in time. Specifically, this data structure contains the complete state of the lattice, with the value of \( f^s_i(x) \) for all values of \( s, \), \( i, \), and \( x \), and also a set of simulation parameters. These parameters may be divided into two categories: parameters which are static and unchanging, such as the dimensions of the simulation lattice or parameters describing the initial state of the system, and parameters which could conceivably be changed during the course of the simulation, such as coupling constants and forcing rates. The code is then structured as a set of methods which act upon the data in this structure.

These methods can be loosely grouped into categories. ‘Constructor’ and ‘destructor’ methods allocate new simulation objects and free the memory associated with old ones; initialization methods initialize the state of a simulation object before commencing a given simulation run. IO methods write simulation data to disk, and also modify simulation data according to data saved on disk. These methods can load and save complete simulation states to and from disk, as well as loading, for example, porous media data, and saving information such as the fluid density field to disk. The save-data methods consist of two parts: one generic routine which produces a block of data corresponding to a physical field (such as density or pressure), and another routine which saves this block of data to disk in a specific format (such as raw binary, a portable binary format known as XDR [46], or a portable image format known as PNG).

Evolution methods perform advection or collision processes on a simulation object, and generally consume the majority of the CPU time in a given simulation run. Finally, boundary condition methods alter the lattice, for example, to maintain a constant fluid density or composition at the edges of the simulated region.

---

\(^1\)LB3D was recently awarded the gold-star rating for its excellent scaling properties with large models running on 1024 processors on HPCx, the UK’s fastest supercomputer.
4.2. Traditional simulation methodology and its drawbacks
Traditionally, large compute-intensive simulations are run non-interactively. A text file describing the initial conditions and parameters for the course of a simulation is prepared, and then the simulation is submitted to a batch queue, to wait until there are enough resources available to run the simulation. The simulation runs entirely according to the prepared input file, and outputs the results to a disk for the user to examine later.

This mode of working is sufficient for many simple investigations of mesoscale fluid behaviour; however, it has several drawbacks. Firstly, consider the situation where one wishes to examine the dynamics of the separation of two immiscible fluids: this is a subject which has been of considerable interest in the modelling community in recent years [39,47]. Typically, a guess is made as to how long the simulation must run before producing a phase separation, and then the code is run for a fixed number of timesteps. If a phase transition does not occur within this number of timesteps, then the job must be resubmitted to the batch queue, and restarted. However, if a phase transition occurs in the early stages of the simulation, then the rest of the compute time will be spent simulating an equilibrium system of very little interest; worse, if the initial parameters of the system turn out not to produce a phase separation, then all of the CPU time invested in the simulation will have gone to waste.

Secondly, the input file often takes the simple form of a list of parameters and their values, but this may not be sufficiently expressive to describe the boundary conditions one may wish to apply, or the conditions under which they are to be applied. For example, to simulate the flow of a fluid mixture through a porous medium, it is necessary to equilibrate the flow of a single component through the medium first, before introducing the fluid mixture, in order to prevent the behaviour of the mixture from being affected by transients present as the flow field develops.

4.3. High-level control of simulation codes: scripting
For every new and complicated boundary condition one wishes to impose, it is in principle possible to write a corresponding new subroutine in the simulation code, add an option in the input file to switch this boundary condition on or off, and recompile the simulation code. However, in practice this leads to redundancy and overcomplication, or “bloat”, in the simulation code, and also to excessively complicated or verbose input files. Bloated code will, in the long term, become difficult to maintain or change, and more complicated input file syntax makes it harder for new users to learn how to use the code.

An alternative strategy is to abandon the concept of an input file altogether, and instead to control the simulation from a script written in a high-level language. This has several advantages.

Firstly, provided that enough access to the simulation data structures is provided to the scripting layer, new boundary conditions may be formulated, tested, and run with ease, without requiring the code to be recompiled. The core of the number-crunching code stays small and maintainable as a result.

Secondly, a high-level language provides conditional and loop structures, so simulations may be given much more detailed instructions than simply to run for a fixed number of timesteps: for example, a simulation of fluid phase separation could be instructed to run until the fluid components have separated to a certain degree, and then to stop.

Thirdly, writing the core of the simulation code in a language like C or Fortran but controlling its behaviour through a higher-level language allows the programmer to easily interface the simulation code with other components (such as image generation libraries) via the high-level language, which avoids the necessity of dealing with tedious low-level details of interfacing to many third-party libraries. This strategy also avoids incurring the performance penalty that would result from writing the entire simulation code in a higher-level language.

The approach of making a piece of code such as the simulation solver available as a self-contained reusable object to some higher-level “glue” layer is often termed “componentization”. In this case, the glue layer is the high-level language; in the more general case, it could be a Grid fabric layer such as Web Services, allowing interoperation across the network of components running on different machines.

The high-level language chosen to script LB2D was Perl, a powerful language popular, amongst other things, for its ability to interface with, or “glue”, external components, and also for the wide variety of freely-available Perl code which can be easily accessed from scripts written in the language [48]. Constructing a functional Perl interface to the simulation code required little more than writing a formal description of the C subroutines comprising the simulation code [49]; interfacing code to other popular scripting languages such as TCL, Python, or Ruby is typically just as easy.

4.4. Parameter space exploration using high-level scripting facility
For an algorithm which runs the risk of encountering numerical instabilities, it is desirable to know the regions of parameter space in which one can operate without expecting to encounter such problems: for example, when
studying the behaviour of an interface between two fluid components, it is useful to know how high the surface tension can be set before numerical instabilities are introduced, resulting in a simulation crash.

A crude approach to map out this region is to guess the size and location of a region of parameter space which will contain a region of stability, and blindly launch many simulations over this space.

A slightly more versatile approach made possible by scripting is to start with a known-stable point in parameter space, and an initial direction in parameter space. A simulation is started at the known-stable point, and if it completes successfully, another one is started at an adjacent point in parameter space, until eventually a numerically unstable regime is found; successive simulations can then be launched to home in on the location of the stable/unstable boundary.

In the Shan-Chen LB model, the interaction force between components, described in equation (12), gives rise to a surface tension at the interface between regions of different components; the strength of this interaction, and therefore the magnitude of the surface tension, is controlled by the coupling constant $g_{br}$.

A simple parameter-space investigation is to take an interface between two immiscible fluids, and run simulations with increasing $g_{br}$, and therefore increasing surface tension, until numerical instability sets in. The script controlling this process ensures that each simulation runs long enough for the system to reach equilibrium. If a simulation succeeds, then the surface tension is raised; if not, then it is lowered, and the boundary is located using an interval bisection algorithm. The results of such an investigation are shown in figure 2, in which the surface tension was calculated for each value of $g_{br}$ for which a simulation could successfully be run.

4.5. Scripted boundary conditions

The use of scripting allows extremely versatile and dynamic specification of boundary conditions in a simulation. Consider a stream of droplets flowing through a channel until they meet an obstacle: the droplets accumulate on the obstacle and coalesce to form a larger droplet until the resulting droplet becomes too large and breaks free to travel further down the channel. Conventionally, one might investigate this situation by first running a simulation of single-component flow through the channel with the obstacle, until the velocity field equilibrates, and then starting a new simulation from this one in which droplets are added to the channel, either through manual intervention, or by writing an additional piece of code which periodically generates a new droplet near the entrance to the channel.

However, it is also possible to perform the simulation in one single run, using a script. This script equilibrates the single-component velocity field, and then automatically introduces a droplet to the channel entrance. It then waits, monitoring the simulation state, until the droplet has moved sufficiently far down the channel that a new one may be introduced without colliding with it. The results of such a simulation are shown in figure 3: the automatically-generated droplet stream is induced by the obstacle to coalesce into a single large droplet, which then breaks off shortly before timestep 7850. The advantage of this approach is that it does not require human intervention to restart the simulation after the velocity field has equilibrated, nor does it require a fixed droplet generation rate to be set: if such a rate were set slightly too high, then droplets may collide with one another before reaching the obstacle, a situation avoided by the use of a dynamically-specified boundary condition.
5. Steering lattice-Boltzmann simulations through a generic library interface

In this section, we present a discussion of the way in which we have implemented computational steering for LB3D within the ongoing RealityGrid project [50]. The RealityGrid project aims to enable the modelling and simulation of complex condensed matter structures at the molecular and mesoscale levels as well as the discovery of new materials using computational grids. The project also involves biomolecular applications and its long-term ambition is to provide generic computational grid based technology for scientific, medical and commercial activities.

5.1. Motivation

Within RealityGrid, the way in which computational steering is implemented is driven by a desire to enable existing scientific computer programs (often written in Fortran90 and designed for multi-processor/parallel supercomputers) to be made steerable while minimising the amount of work required. Minimising the number of changes that a scientist must make to an existing program is important since it encourages him to take responsibility for this work. Consequently, the scientist understands the changes that are required and can continue to maintain the software in the future.

In the light of these requirements, we chose to implement the steering software as a library written in C and thus able to be called from a variety of languages (including C, C++ and Fortran90). The library completely insulates the application from any implementation details. For instance, the process by which messages are transferred between the steering client and the application (e.g. via files or sockets) is completely hidden from the application code.

Different scientists favour various techniques for writing programs intended to run on the specialist architectures of large supercomputers. Our steering library therefore does not assume or prescribe any particular parallel-programming paradigm (e.g. message passing or shared memory).

Obviously, a scientist does not want a failure in the steering system (such as a loss of connection to the application) to result in valuable computing time being lost. We have therefore designed the steering protocol so that, insofar as is possible, the steering is never made critical to the simulation process. The protocol enables a steering client to attach and detach from a running application without affecting its state.

The scientist’s ability to monitor the state of his simulation and use this to inform his steering decisions plays a key role in computational steering. While a steering client provides some information via the simulation’s monitored parameters, a visualisation of some aspect of the simulation’s state is often required. In our architecture this visualisation is created by a second software component.

5.2. Requirements

In order to make use of the steering library, an application must satisfy certain requirements. In particular, the application must have a logical structure such that there exists a point (which we term a breakpoint) within a control loop at which it is possible to carry out the following steering tasks:

(i) emit a consistent representation of the state of the application’s steerable and monitored parameters;
(ii) accept a change to one or more steered parameters;
(iii) emit a consistent representation (data sample) of part of the system being simulated (e.g. for visualisation);
(iv) take a checkpoint or restart from an existing checkpoint.

While all of these things must, theoretically, be possible at the breakpoint, it is up to the scientist as to how many of them his application actually supports. For instance, enabling the application to restart from a checkpoint during execution might be a difficult task and therefore need only be attempted if the scientist particularly wants the functionality that that facility will bring.

5.3. The steering library

The steering library itself consists of two parts: an application side and a client side. The client side is intended to be used in the construction of a steering client. We have developed a generic steering client using C++ and Qt (a GUI toolkit) [51] which is capable of steering any application that has been ‘steering enabled’ using the library.

The steering library itself supports a variety of features. These include the facility for the application to register both monitored (read-only) and steerable (changed only through user interaction) parameters. Beyond this facility, the library supports a set of pre-defined commands such as ‘pause’, ‘resume’, ‘detach’ and ‘stop.’ In addition to these pre-defined commands, the library also allows the user to instruct the application to emit or consume any data sets that it has previously registered. Similarly, the user may instruct the application to take a checkpoint or restart from one which the application has registered.

The latter functionality is particularly important since it provides the basis of a system that allows the scientist to ‘rewind’ a simulation (by restarting from a previous
checkpoint). Having done so, it can then be run again, perhaps after having steered some parameter or altered the frequency with which data from the simulation are recorded. The GRASPARC project [52] is an example of another system with this functionality.

In order to maximise the flexibility of the library, we use a system of ‘reverse communication’ with the application. This means that, for most actions, the library simply notifies the application of what the latter needs to do. It is then the application’s responsibility to carry out the task, possibly using utility routines from the steering library. This is consistent with the philosophy mentioned earlier, of allowing the scientist to decide how much steering functionality he wishes to implement.

The steering library currently uses files for transmission of the steering messages. This means either that the application and the steering client must have access to the same disk or that some other software (known as ‘middleware’) takes responsibility for transferring the files between specific locations on the computers running the application and the client. Work in progress will lift this restriction by introducing direct communication between the application and client.

5.4. Computational steering with LB3D

As noted earlier, our parallel three-dimensional lattice-Boltzmann code (LB3D) has been interfaced to the RealityGrid steering library, which allows the user to steer all parameters of the simulation including coupling constants, fluid densities, relaxation times and even data dumping frequencies. Steerable data dumping frequencies enable the user to increase the amount of generated data for parts of the simulation where the effects of interest are happening. This helps to save an expensive resource, namely disk space.

In addition to the features the steering library provides, LB3D has its own logging and replay facilities which permit the user to ‘replay’ a steered simulation. This is an important feature since it allows the data from steered simulations to be reproduced without human intervention. Moreover, this feature can be used as an ‘auto-steerer’, i.e. multiple simulations which read different input-files at startup and are ‘steered’ in the same way can be launched without the need for human intervention during the simulation. One application of this particular feature is for studies of how changes in parameters affect a simulation that has evolved for a given number of timesteps. Another application is the automatic adaptation of data dumping or checkpointing frequencies. If the user has found from a manually steered simulation that no effects of interest are expected for a given number of initial timesteps, he can reduce the amount of data written to disk for the early times of the simulation.

All steered LB3D simulations that are reported in this paper were performed on 64 processors of an SGI Origin 3800 in Manchester, UK. For data visualisation we used the Visualization Toolkit (VTK) [53] on a workstation in London. We chose to run the steering client on the same workstation.

5.4.1. Spinodal decomposition

As an example of a typical steered simulation with LB3D, let us consider the miscibility of a binary fluid mixture. We are interested in the behaviour of the system for different values of the coupling constant $g_{br}$ which controls the strength of the interaction between both fluids, which we call ‘blue’ and ‘red’ here [see equation (12)]. By interacting with a single ongoing simulation, we can change $g_{br}$ ‘on the fly’ and immediately see how the fluid mixture behaves. Depending on the phenomena we are interested in, we can ‘steer’ the fluid into miscible or immiscible states. This technique can as well be utilised to find optimal values of $g_{br}$ to study spinodal decomposition. Spinodal decomposition takes place if an incompressible binary fluid mixture is forced into thermodynamically unstable regions of its phase diagram, i.e. below its spinodal temperature. In this case, the mixture starts to phase separate into domains of the two fluids. This effect is important in various industries because phase separations in products like paints or cosmetics have to be controlled carefully and many researchers have studied spinodal decomposition in detail [38,39,47,54 – 61].

Figure 4 shows snapshots of volume rendered ‘colour’ fields (see sections 2 and 3 for details). The colour field describes the net force of different fluid species on a given lattice site. A value of zero is obtained if forces caused by both fluid types cancel each other. This takes place at the interface between the fluids. A ‘colour’ field greater than zero corresponds to areas where the ‘blue’ fluid dominates while negative values correspond to a domination of ‘red’ particles respectively. In figures 4(a-f), areas of ‘blue’ dominance are rendered in blue and ‘red’ areas close to the interface are visualised in red. In the remaining snapshots, the colouring is utilized to visualise the diffusion of one fluid species into areas where the other species is dominant.

As initial condition for the simulation of the $64^3$ system, we chose a mixture, where for both species, each vector on each lattice site is assigned a random occupation number between zero and 0.7. Relaxation times and masses are set to unity and the initial value of $g_{br}$ is 0.001. This value is too low for phase separation to occur. Therefore, $g_{br}$ is slowly raised to a value of 0.01. Within a few thousand timesteps, both phases start to separate and after 11000 timesteps of the simulation, a clear structure in the fluid densities can be observed. It takes until timestep 24000 for the mixture to reach a fully separated state (figure 4a-f). At timestep 25000...
we start to reduce $g_{br}$ again in order to force the fluids to mix again. The minimum value of the coupling strength used is $-0.001$. Soon after reducing $g_{br}$, red fluid particles start to diffuse into areas of blue dominance and vice versa. At timestep 29000 the system has arrived in a nearly fully mixed up state again (figures 4g-i).

5.4.2. Parameter searching

The second example we give to demonstrate the usefulness of computational steering of three dimensional lattice-Boltzmann simulations is focused on parameter searches.

As noted previously, our simulations are very resource intensive. Single simulations might take between hours and days on a large number of processors of a parallel computer, and storing the generated data requires tens or hundreds of gigabytes of disk space. Typical system sizes of our simulations include $64^3$, $128^3$ or $256^3$ lattices. The data written to disk for a single measuring timestep of a $256^3$ lattice requires about one gigabyte of disk space. For typical simulation lengths of 20000 timesteps and a measuring frequency of 100 timesteps, 200GB are needed. By reducing the lattice size to $64^3$, one is able to substantially reduce the amount of data to 3GB. However, due to the possible occurrence of finite size effects, such small lattice sizes are often not appropriate. Most of these data might turn out not to include physically interesting results or the data might be of limited use because the simulation parameters were not chosen correctly. Moreover, like other mesoscopic models, the lattice-Boltzmann method contains a number of free parameters (see section 3), resulting in high-dimensional parameter spaces, although only limited areas may be of interest. In addition,
the phenomena of interest might occur within a limited time interval in the simulation only. In all these situations, very expensive compute resources are wasted.

The free parameters of our lattice-Boltzmann ternary amphiphilic fluid algorithm include the coupling constants between different fluid types (see section 3.3). The relationship between these parameters and the experimentally available fluid properties is not well understood. Therefore, it is important to choose these parameters carefully so as to study a wide range of phenomena with one or a small number of parameter sets.

Traditionally, such ‘optimal’ parameter sets have been determined by ‘task farming’ approaches, that is by performing large numbers of small simulations concurrently, on a large parallel machine or on a large number of small individual machines [24]. This technique allows one at least in principle to ‘scan’ the entire parameter space. In practice, only subspaces can usually be investigated in detail, although these can be distributed. No human interaction is required after the jobs have been submitted, which makes it easy to use script-based approaches for the generation of input files and job submission. However, the available computing resources are not used very efficiently. Not only is CPU time wasted in a task farm simulation, the amount of disk space needed to store the simulation data can be immense.

For example, we did large-scale parameter searches for binary water-surfactant mixtures. The system size was $64^3$ and the parameters studied were the surfactant-surfactant coupling constant $g_{ss}$, the surfactant-water coupling constant $g_{sw}$, and initial fluid densities. Masses and relaxation times were kept fixed at unity. We were only able to study small regions of the available parameter space, i.e. $g_{sw}$ was varied between $-0.001$ and $-0.006$ and values for $g_{sw}$ varied between $-0.004$ and $-0.008$. The initial conditions were set as in section 5.4.1, but the maximum occupation numbers were varied from 0.2 to 0.7 for each fluid individually. In practice, we launched a number of simulations with different values for $g_{ss}$, $g_{sw}$ and initial densities and analysed the generated data afterwards. This analysis gave us an idea of interesting values for the parameters studied and, in principle, on the basis of these findings one could launch more simulations in order to investigate the system in more detail.

However, within a few wall-clock weeks of simulation time, we generated about 300 GB of data and used about 30000 CPU hours on an SGI Origin 3800. While the simulations were performed in a highly automated manner within a couple of weeks, data analysis has been ongoing for months. Automation of the analysis of the generated data is much harder because it can be difficult to define the effects sufficiently well, or impossible to anticipate the effects in advance, or simply not worthwhile to invest additional effort in the development of algorithms to automate the process.

By considering this example, the disadvantage of conventional parameter searches is apparent: a significant fraction of the simulations performed in such a search employ parameters which do not admit interesting phenomena. Nonetheless, these simulations generate data that has to be analysed. In fact, it is not the elapsed computing time that makes parameter searches very time consuming, but rather the extraction of the information from the data produced.

This analysis time can be significantly reduced by introducing human intuition into the simulation-analysis loop. The scientist doing the simulations and analysing the data is usually able to decide whether a parameter set is in a region of interest long before any given simulation finishes. By providing the scientist with the possibility to change simulation parameters on the fly, two goals can be achieved. First, he might be able to ‘steer’ the simulation into areas of interest: this improves the effective use of CPU cycles and reduces the amount of produced data. Second, analysing the reduced simulation output data is much less time consuming.

Figure 5 depicts a steered parameter search using the LB3D code. Instead of trying to cover the full parameter space, we only perform a limited number of steered simulations. We start with a random water-surfactant mixture with the surfactant-surfactant coupling constant $g_{ss}$ set to $-0.003$ and the water-surfactant coupling constant $g_{sw}$ set to $-0.006$ (figure 5a). The lattice size is $64^3$ and the initial maximum occupation numbers in this case are 0.7 for surfactant and 0.4 for water. The insets of figure 5 show volume rendered surfactant densities, where densities higher than the average are coloured in green. We monitor the state of the system while the simulation evolves and find a stable micellar phase after about 10000 timesteps (figure 5e), i.e. a large fraction of surfactant molecules forms spherical micelles.

Since we are not able to detect any drastic changes for later timesteps, we ‘rewind’ the simulation back to timestep 5000 and change $g_{sw}$ to $-0.002$. In this way we lower the interaction between surfactant molecules. Rewinding to an earlier simulation step is necessary because due to the low fluid velocities it takes a very long time in a system that is close to equilibrium for parameter changes to take effect. Steering the values of the coupling constants more drastically is not a solution to this problem because the simulation might become unstable and produce unphysical results or even crash.

Five thousand timesteps after ‘rewinding’, the system arrives in a state that is only slightly different from the $g_{sw} = -0.003$ case at timestep 10000. In order to investigate whether any more changes occur, we let the simulation evolve for a further 5000 timesteps and...
74 III-1. Steering in computational science: mesoscale modelling and simulation

discover more well defined structures than before. A closeup of the dataset (see inset of figure 5d) allows us to investigate the self-assembled spherical micelles in our system in more detail: the surfactant molecules visualised by the blue cones are pointing away from regions of low surfactant density, that is away from regions of high water density. The more well defined structure than in figure 5c is due to a larger fraction of surfactant molecules being involved in forming micelles.

In order to investigate areas of the parameter space where the absolute value of $g_{ss}$ is higher, we ‘rewind’ the simulation again and change $g_{ss}$ to $-0.005$. At timestep 10000, the system now finds itself in a drastically different state from the previous cases: water and surfactant form lamellae, and the surfactant molecules align in parallel bilayers between areas filled with water. The colour purple is used here to visualise the interface between water and surfactant and the inset of figure 5f depicts how the surfactant molecules are aligned in this case.

Of course, a single steered simulation cannot by itself replace a full task farm parameter search, but a small number of steered simulations can provide a coarse grained overview of the available parameter space. By steering into areas of interest, one is able to dramatically reduce the resources required. Most of the analysis takes place during the simulation time itself and therefore required additional effort for offline analysis is reduced. Moreover, CPU time is further reduced because not every simulation has to start from timestep zero again. In the example given here, three conventional simulations of 15000 timesteps each sum up to 45000 simulation steps in total. In contrast, the steered simulations use only about 25000 timesteps because we do not have to rewind to timestep zero and can stop as soon as we cannot detect any further changes. Since the scientist interacts with the simulations, data dumping rates can be adapted during the run, thus reducing the requirements for disk storage even further.

6. Computational steering

As alluded to in previous sections, the problems associated with high performance computational science in general and large scale simulations in particular are not confined to merely finding resources with larger numbers of processors.
or memory. Simulations that require greater computational resources also require increasingly sophisticated and complex tools for the analysis and management of the output of the simulations. In sections 4 and 5, we highlighted the limitations of a simple simulate-then-analyse approach and indicated how more sophisticated approaches help alleviate some of the problems. We then sketched some specific advantages of controlling the evolution of a computation, based upon real time analysis and visualisation of the status of a simulation, as opposed to post facto analysis and visualisation of the output of a computation. The functionality that we refer to as computational steering enables the user to influence the otherwise sequential simulation and analysis phases by merging and interspacing them.

What additional requirements does computational steering place on computer systems? In order to computationally steer a simulation, one needs an interface to communicate with the simulation, which may be running on a remote machine. In addition to allowing parameters to be monitored and changed, this interface needs to offer the possibility of visualising complex data sets, for instance 3D isosurfacing and volume rendering. To enable intuitive interaction with a simulation, it is essential that visualisation can be performed sufficiently fast compared to changes taking place in the simulation. Visualisation of large and complex data sets typically requires high-end graphics hardware, which, like high-end computing resources, is not always available locally. Therefore, visualisation should be treated as a distributed resource as well, the need for which stems not only from computational steering but also from the requirements of high performance visualisation. The requirement to use more than one distributed resource simultaneously in turn raises more subtle issues associated with the requirements of sophisticated scheduling algorithms and techniques. Typical supercomputer centres currently make no provision for coallocation of resources, for example a compute node and a visualisation node. It would be desirable to be able to request resources for a computational steering session in advance and be assured of a certain quality of service during a session. Equally important is the requirement to be able to reserve substantial compute resources with small turn around time.

What advantages does computational steering provide the application scientist with in return? We have described a few specific examples of how computational steering can increase a scientist’s productivity. This increase in productivity is due to an increase in the throughput of hardware resources but equally important is the enhanced productivity due to a more effective computational science workflow bench (simulation-analysis loop) as a consequence of being able to use computational steering. There have been attempts to use computational steering as a novel approach to studying outstanding and important problems in biomolecular systems [62]. At the very least, computational steering complements existing techniques [63]. However, computational steering should not simply be thought of as an effective tool in the production and analysis phase of a simulation. It can provide the application scientist (often also the application developer) with greater flexibility in the development, debugging and validation phases of an application [8,64,65], where it complements rather than replaces other well established tools. Computational steering can also be extended to collaborative environments where several geographically distributed scientists can simultaneously interact with one or more simulations from separate locations [66–70].

This sets the stage for a few remarks on what kind of computational science applications are suitable candidates for the use of computational steering. If an application requires barely a few seconds of computing time to simulate a physical process or effectively finish the simulation (say a fixed small number of iterations in a minimization routine), then the advantages of user intervention while the simulation is in progress are limited. Any overhead associated with interrupting such an inexpensive simulation will not be worth the gain bought by interactivity. At the other end of the spectrum, simulations that ‘take forever’ for any discernible changes to manifest as a consequence of user interaction are also not good candidates as the advantage from such interactivity is typically limited. A case in point are ab initio quantum mechanical molecular dynamics simulations, where even when only a small number of atoms are of interest, each step of the calculation may take several hours on a multiprocessor machine [71,72]. Any changes initiated dynamically by the user for such simulations would take many hours to become manifest, clearly limiting their use. Thus it appears to be the case that simulations with a run time from several minutes to several hours (irrespective of the resources used), are ideally suited for interactive aspects of computational steering. It is important to distinguish between the role of steering in long running simulations as opposed to simulations with long response times to a perturbation. We have discussed the limited role of steering in the latter, but in the former case, steering can be useful for checking the progress of a long running simulation by connecting to the simulation, getting a sample and visualising it and then disconnecting after checking all is correct, thus enabling the scientist to use computational steering as ‘simulation monitor’ and as a safeguard against possible wastage of computational resources.

A few cautionary remarks are in place. Many physical systems have long equilibration time scales and suffer from finite size effects. In such cases, changing the parameters and taking the state immediately following the change to be the putative true state might be misleading. Moreover, many physical systems exhibit hysteresis, that is their
properties at a given point in parameter space are
dependent upon their history. Before computationally
steering a simulation, it is imperative to determine if it
displays hysteresis and, if so, how the use of steering may
influence the analysis. Computational steering of diverse
applications may involve different challenges, but in all
cases an antidote to possible problems will be careful and
consistent study rather than a refusal to adopt new analysis
techniques.

We end this section by discussing why there has been
comparatively limited acceptance or use of computational
steering in scientific applications until now. If there is a
lesson to be learnt from the evolution of computational
science, it is that the complexity of doing something new
and exciting has to be well hidden from the application
scientist, i.e. it is essential to minimize the amount of
implementation, learning, disruption and changes to the
user interface, until the advantages of the new features are
well established and very clearly seen to offset the cost of
implementation. Ideally one would like, if possible, to just
slip in the functionality where the user never knows or
notices, but this may not be achievable in a computational
steering context, when the application scientist often has to
actually execute the functionality. Thus maximal effort will
have to be invested into reducing the ‘barrier of entry’.

Until now, implementing computational steering has
required a high degree of customization, but most scientists
typically are not in a position to invest in the time-
consuming task of developing the necessary tools; indeed,
cooperation with specialists in visualisation and interfacing
techniques has hitherto been vital. This problem is not
helped by the fact that the requirements of a scientist might
change during a project because new results from simul-
ations lead future investigations in different directions than
initially planned. This could result in the steering tools
having to be adapted, which can be very costly [64,65].

The question that logically follows is what can be done to
address the relatively low acceptance of computational
steering in computational science. We believe that most
scientists are not cognisant of the advantages computa-
tional steering offers and thus unaware how they might
benefit from steerable applications. Therefore, their simul-
ations are done in the ‘traditional’ way, invoking long batch
jobs and subsequently lengthy offline data analysis. In this
paper we have outlined the advantages that computational
steering has brought to our LB studies. Part of the purpose
of this article is to encourage computational scientists to
think about the enhanced ability and benefits that
computational steering capabilities would bring to their
scientific productivity, along the lines of our LB studies, but
specific to their own applications. It is obvious that not all
computational science problems are amenable to, or for
that matter require, computational steering. However, we
believe that documenting the advantages of computational
steering in widely differing applications and areas will help
bring greater acceptance of steering as a valid paradigm for
computational science research. We also wish to emphasize
that, as shown by the RealityGrid steering framework, by
using the correct abstractions and good software engineer-
ing practices, implementing the required changes is much
less effort than might otherwise be expected. Indeed, many
generic tools and libraries useful in program steering and
data visualisation are now readily available [7,8,73,74].

7. Steering on Computational Grids: Current Status and
Future Outlook

Significant effort is being invested worldwide in Grid
computing [75]. A basic premise of grid computing is to
provide the infrastructure required to facilitate the colla-
borative sharing of resources. The grid aims to present the
elements required for a computational task (e.g. calculation
engine, filters, visualisation capability) as components
which can be effectively and transparently coupled through
the grid framework. In this scenario, any application or
simulation code can be viewed simply as a data producing/
consuming object on the grid and computational steering is
a way of allowing a user to interact with such objects. As
discussed in previous sections, a scientist using steering has
heterogeneous and dynamic computational resource re-
requirements, making the stated ability of the grid to marshal
collectively and transparently diverse resources comple-
mentary to the primary motivation of computational
steering. Thus a grid infrastructure that permits the
coordination of heterogeneous and distributed computing
resources provides a natural environment as well as a
testbed for demonstrating the effectiveness of steering in
computational science.

In our description of the RealityGrid steering framework
in section 5, we did not mention the use of a computational
grid or dependence on any underlying middleware require-
ment. This is because the RealityGrid steering framework is
capable of being used on stand alone middleware as well
as the most ambitious computational grids available in the
world today. Equally important is the fact that our steering
framework is not critically dependent on any one particular
middleware although it does adhere to the best practises
and the open standards currently being discussed within the
Global Grid Forum (GGF) [75]. In the remainder of this
section we will describe how the RealityGrid project uses
the grid to implement computational steering.

In the same way that a high-fidelity simulation of a
physical system often requires a supercomputer, so the
visualisation of the (potentially) large data sets that these
simulations produce also requires specialist hardware that
few scientists have direct access to. Consequently, the
visualisation component of the RealityGrid steering may
well be on a machine other than the one the scientist is
sitting in front of. This is consistent with our earlier proposal that visualisation be treated like a distributed resource. This then requires that the output (images) of the visualisation component be returned to the user’s workstation quickly enough to allow for full interactivity (e.g. to rotate, zoom, etc.). We currently use SGI’s OpenGL VizServer software to perform this task; it takes the images directly from the rendering hardware on the visualisation machine, compresses them, transports them to the user’s machine, decompresses and displays them [76]. This allows the scientist to interact with a remote visualisation, even over network links with relatively low bandwidth.

The outline traced above has been the basis for several successful computational steering demonstrations that we have performed within the past year. In our first demonstration at the UK e-Science All Hands Conference in Sheffield in September 2002, we used this technique to interact with a visualisation produced on an SGI Onyx 300 in Manchester from a laptop in Sheffield with the computation performed in London. For this demonstration and the subsequent two others, we used Unicore [77] as the underlying middleware to manage the file transfer aspects of the demonstration. At Supercomputing 2002, we used a trans-Atlantic link to interact with a visualisation on the Onyx 300 in Manchester from a laptop in Baltimore, USA, the computation being performed on the SGI Origin 3800 in Manchester. In February of 2003 at the SGI VizSummit, we used a laptop in Paris to interact with simulations on 128 processors of the Origin 3800 in Manchester, visualising and steering being performed locally using the Onyx 300 facilities provided on the demonstration floor. Thus we have in the process performed computational steering using three different albeit transient grid scenarios: within the UK, trans-Atlantic and UK-continental Europe.

The grids used in these demonstrations were assembled especially for each event. However, the UK e-Science community has constructed an ambitious Level 2 Grid [78] that aims to provide the user community with a persistent grid. We have already deployed a preliminary RealityGrid LB3D application involving computational steering, using Globus (as opposed to Unicore, confirming the flexibility of our steering framework) on this Level 2 Grid, thus being amongst the first groups in the world to use a persistent grid for routine science requiring high performance computing and computational steering.

At the time of writing, the RealityGrid steering library supports both file-based and streaming (based on globus.io from the Globus project [79]) data transfer between the application and visualisation components. Communications between the application and the client is currently implemented by exchanging XML documents through a shared file system (XML is a widely accepted language specifying the syntax to mark-up data in computer documents).

We are in the process of implementing a more flexible architecture, based on the Open Grid Services Infrastructure [80]. As shown in figure 6, communications between the application and client are routed through an intermediate steering grid service (SGS). The SGS provides the public interface through which clients can steer the application. In our architecture, the visualisation and application components appear on equal footing, and a visualisation can possess its own SGS. Each SGS publishes information about itself in a registry, which is used by clients to discover and bind to running applications, and can also be instrumental in bootstrapping the communications between the application and visualisation components. We note that the approach of exposing steering controls as grid services in a standard way could bring profound benefits in the form of interoperability between different implementations of computational steering.

8. Conclusions

This paper has described the work we have done to incorporate computational steering in mesoscale lattice Boltzmann simulations of binary and ternary immiscible and amphiphilic fluids. The scale and efficiency of these studies is set to increase dramatically with the advent of computational grids which are now becoming widely available within the UK, Europe, the USA and the Pacific Rim.
Acknowledgements

We are grateful to ESPRC for funding much of this research through RealityGrid grant GR/R67699 and for providing access to SGI Origin 3800, Origin 2000, and CRAY T3E supercomputers at Computer Services for Academic Research (CSAR), Manchester, UK. We would also like to thank the University of Manchester for access to their SGI Onyx 300 and HEFCE for funding the 16 processor dual pipe SGI Onyx 2 at University College London. Jonathan Chin acknowledges Huntsman and Queen Mary, University of London, for funding his Ph.D. studentship. Jens Harting wishes to thank the European Commission Access to Research Infrastructures action of the ‘Improving Human Potential Programme’ for supporting his stay at Italy’s national supercomputer centre in Bologna (CINECA) and the use of their local IBM SP4, SGI Origin 3800 and SGI Onyx 2 facilities.

References

Jonathan Chin is a postgraduate student in Peter Coveney’s group at the University College London. In 1999, while an undergraduate at the University of Oxford, he took a summer placement in the group, where he wrote the LBIB parallel lattice Boltzmann code and enjoyed it so much that he joined the group in 2000 to do a PhD. His interests include complex fluids, visualization, and premature micro-optimization.

Jens Harting studied physics at the Carl von Ossietzky University in Oldenburg, Germany. After a Diploma thesis on Bose-Einstein Condensation he worked on path integral Monte Carlo simulations of few electron systems such as semiconductor quantum dots and received his PhD in December 2001. Being interested in simulations of fluids, computational steering and high performance computing, he joined the RealityGrid project as a Research Fellow in Peter Coveney’s group in March 2002.

Shantena Jha is a Research Fellow with RealityGrid at the Centre for Computational Science, UCL, London. His graduate work is in Physics and Computer Science from Syracuse University, New York, USA. His interests are in computational physics, high performance and distributed computing and politics.

Peter Coveney holds a Chair in Physical Chemistry in the Department of Chemistry and is Director of the Centre for Computational Science (CCS) at University College London. His group performs internationally leading research in the area of atomistic and mesoscale modelling and simulation, including molecular dynamics, dissipative particle dynamics, lattice-gas and lattice-Boltzmann techniques and exploits state of the art high performance computing and visualisation methods. He has published numerous theoretical and modelling/simulation papers on lattice-gas and lattice-Boltzmann automata, dissipative particle dynamics and molecular dynamics inter alia. Professor Coveney is currently leading the large RealityGrid research programme, funded by the UK’s Engineering and Physical Science Research Council, aimed at grid enablement of supercomputing, visualisation and computational steering (http://www.realitygrid.org). He previously held the Chair in Physical Chemistry in the Department of Chemistry at Queen Mary, University of London, before which he was with the Schlumberger Cambridge Research, where he held a number of scientific and management positions.

Andrew Porter is a software engineer in the Supercomputing, Visualization and e-Science group in Manchester Computing at the University of Manchester. He graduated with a PhD in computational condensed-matter physics from the University of Cambridge in 2000 and, after a spell as an IT consultant in industry, he joined the RealityGrid project in March 2002. His interests include computational steering and scientific visualisation.

Stephen Pickles is Software Infrastructure Manager for the RealityGrid project, and co-leader of Manchester Computing’s e-Science team at the University of Manchester. He has been engaged in grid computing since 1999. After nearly a decade as programmer and systems analyst with ICL (Australia), Stephen graduated from Macquarie University in 1994 with a BSc (Hons I) in Physics, gained his PhD in lattice quantum chromodynamics from the University of Edinburgh in 1998 and then joined the CSAR service at Manchester Computing as Senior Applications Consultant.
Large-scale grid-enabled lattice-Boltzmann simulations of complex fluid flow in porous media and under shear

J. Harting, M. Venturoli, and P. V. Coveney
Large-scale grid-enabled lattice Boltzmann simulations of complex fluid flow in porous media and under shear

By Jens Harting¹, Maddalena Venturoli¹,² and Peter V. Coveney¹

¹Centre for Computational Science, Christopher Ingold Laboratories, University College London, 20 Gordon Street, London WC1H 0AJ, UK (p.v.coveney@ucl.ac.uk)
²Schlumberger Cambridge Research, High Cross, Madingley Road, Cambridge CB3 0EL, UK

Published online 3 June 2004

Well-designed lattice Boltzmann codes exploit the essentially embarrassingly parallel features of the algorithm and so can be run with considerable efficiency on modern supercomputers. Such scalable codes permit us to simulate the behaviour of increasingly large quantities of complex condensed matter systems. In the present paper, we present some preliminary results on the large-scale three-dimensional lattice Boltzmann simulation of binary immiscible fluid flows through a porous medium, derived from digitized X-ray micro-tomographic data of Bentheimer sandstone, and from the study of the same fluids under shear. Simulations on such scales can benefit considerably from the use of computational steering, and we describe our implementation of steering within the lattice Boltzmann code, called LB3D, making use of the RealityGrid steering library. Our large-scale simulations benefit from the new concept of capability computing, designed to prioritize the execution of big jobs on major supercomputing resources. The advent of persistent computational grids promises to provide an optimal environment in which to deploy these mesoscale simulation methods, which can exploit the distributed nature of computer, visualization and storage resources to reach scientific results rapidly; we discuss our work on the grid-enablement of lattice Boltzmann methods in this context.

Keywords: lattice Boltzmann; porous media; complex fluids under shear; grid computing; computational steering

1. Introduction

The length- and time-scales that can be modelled using microscopic modelling techniques such as molecular dynamics are circumscribed by the limited computational resources available today. Even with today’s fastest computers, the accessible length-scales are of the order of nanometres and the time-scales are restricted to the nanosecond range. Mesoscopic models open the way to studies of time-dependent, nonequilibrium phenomena occurring in much larger systems and on orders of magnitude

One contribution of 21 to a Theme ‘Connecting scales: micro, meso and macro processes’.

longer time-scales, thus bridging scales between microscopic models and macroscopic or continuum approaches.

In this paper, we use the lattice Boltzmann method to model binary fluids under shear and flow in a porous medium. In the porous medium case, we are now able to reach length-scales for the simulated fluid flow which can be compared directly with data gleaned from magnetic resonance imaging (MRI) experiments. In the case of fluids under shear, one must take care of finite size effects which make it undesirable to study cubic systems, but rather preferable to study systems of high aspect ratio. In this paper we present preliminary results, but expect to return with extensive descriptions of these applications in the future. Both problems are very computationally demanding and require today’s top-of-the-range supercomputers and large-scale data storage facilities. Since these resources are expensive, we have to handle them with care and minimize wastage of central-processing-unit (CPU) time and disk space. A good initial position is to make sure that our simulations do not last longer than needed and do not produce more data than necessary. But more than this, computational steering allows us to interact with a running simulation and adjust simulation parameters and data-dumping rates; it also enables us to monitor the state of our simulations and react immediately if they do not behave as expected, as we shall discuss later.

The worldwide effort to develop reliable computational grids gives us hope to run our simulations in an even more efficient way. Computational grids are a collection of geographically distributed and dynamically varying resources, each providing services such as compute cycles, visualization, storage or even experimental facilities. It is hoped that computational grids will offer for information technology what electricity grids offer for other aspects of our daily life: a transparent and reliable resource that is easy to use and conforms to commonly agreed standards (Foster & Kesselman 1999, pp. 15–25; Berman et al. 2003). Then we shall be able to use the available computational resources in a transparent way, leaving to smart middleware the task of finding the best available machines to run simulations on, and migrate them to other platforms if necessary to ensure optimal performance. Grids should also allow storage, computation and visualization resources to be widely distributed without our having to care about their location.

The main purpose of the present paper is to introduce the concepts of computational steering and grid computing to an audience of computational scientists, concerned here with simulation of fluid dynamics. The paper is structured as follows. After a short introduction to our lattice Boltzmann method in §2, we give a description of our implementation of computational steering in §3 and explain the advantages we expect to gain from the advent of computational grids in §4. Sections 5 and 6 contain our preliminary results on large-scale grid-enabled simulations of fluids under shear and in porous media. We present our conclusions in §7.

2. A lattice Boltzmann model of immiscible fluids

During the last decade, many authors have shown that the lattice Boltzmann algorithm is a powerful method for simulating fluid dynamics. This success is due to its simplicity and to facile computational implementations (Chin et al. 2003; Love et al. 2003; Nekovee et al. 2001; Succi 2001). Instead of tracking individual atoms
Grid-enabled LB simulations of complex fluid flow

or molecules, the lattice Boltzmann method describes the dynamics of the single-particle distribution function of mesoscopic fluid packets.

In a continuum description, the single-particle distribution function \( f_1(\mathbf{r}, \mathbf{v}, t) \) represents the density of fluid particles with position \( \mathbf{r} \) and velocity \( \mathbf{v} \) at time \( t \), such that the density and velocity of the macroscopically observable fluid are given by

\[
\rho(\mathbf{r}, t) = \int f_1(\mathbf{r}, \mathbf{v}, t) \, d\mathbf{v} \quad \text{and} \quad \mathbf{u}(\mathbf{r}, t) = \int f_1(\mathbf{r}, \mathbf{v}, t) \mathbf{v} \, d\mathbf{v},
\]

respectively. In the non-interacting, long mean free path limit, with no externally applied forces, the evolution of this function is described by the Boltzmann equation

\[
(\partial_t + \mathbf{v} \cdot \nabla) f_1 = \Omega[f_1].
\]

(2.1)

While the left-hand side describes changes in the distribution function due to free particle motion, the right-hand side models pairwise collisions. The collision operator \( \Omega \) is an integral expression that is often simplified (Bhatnagar et al. 1954) to the linear Bhatnagar–Gross–Krook (BGK) form

\[
\Omega[f] \simeq -\frac{1}{\tau}[f - f^{(eq)}].
\]

(2.2)

This collision operator describes the relaxation, at a rate controlled by a characteristic time \( \tau \), towards a local Maxwell–Boltzmann equilibrium distribution \( f^{(eq)} \). It can be shown that distributions governed by the simple Boltzmann–BGK equation conserve mass, momentum and energy (Succi 2001). They obey a non-equilibrium form of the second law of thermodynamics (Liboff 1990) and the Navier–Stokes equations for macroscopic fluid flow are obeyed on coarse length- and time-scales (Chapman & Cowling 1952; Liboff 1990).

By discretizing the single-particle distribution function in space and time, one obtains the usual lattice Boltzmann formulation, where the positions \( \mathbf{r} \) on which \( f_1(\mathbf{r}, \mathbf{v}, t) \) is defined are restricted to points \( \mathbf{r}_i \) on a Bravais lattice. The velocities \( \mathbf{v} \) are restricted to a set \( \mathbf{c}_i \) joining points on the lattice and the density of particles at lattice site \( \mathbf{r} \) travelling with velocity \( \mathbf{c}_i \), at time-step \( t \) is given by \( f_i(\mathbf{r}, t) = f(\mathbf{r}, \mathbf{c}_i, t) \).

The fluid’s density and velocity are given by

\[
\rho(\mathbf{r}) = \sum_i f_i(\mathbf{r}),
\]

(2.3)

\[
\mathbf{u}(\mathbf{r}) = \sum_i f_i(\mathbf{r}) \mathbf{c}_i.
\]

(2.4)

The discretized Boltzmann description can be evolved as a two-step procedure. In the collision step, particles at each lattice site are redistributed across the velocity vectors; this process corresponds to the action of the collision operator. In the advection step, values of the post-collisional distribution function are propagated to adjacent lattice sites.

By combining the two steps, one obtains the lattice Boltzmann equation (LBE)

\[
f_i(\mathbf{r}, t+1) - f_i(\mathbf{r}, t) = \Omega[f] = -\frac{1}{\tau}[f_i(\mathbf{r}, t) - N_i(\rho, \mathbf{u})],
\]

(2.5)

where \( N_i = N_i(\rho(\mathbf{r}), \mathbf{u}(\mathbf{r})) \) is a polynomial function of the local equilibrium density and velocity, and can be found by discretizing the Maxwell–Boltzmann equilibrium distribution.

Our lattice Boltzmann implementation uses the Shan & Chen (1993) approach, by incorporating an explicit forcing term in the collision operator in order to model multi-component interacting fluids. Shan & Chen extended the single-particle distribution function $f_i$ to the form $f_i^\sigma$, where each component is denoted by a different value $\sigma$, so that the density and momentum of a single component $\sigma$ are given by

$$\rho^\sigma = \sum_i f_i^\sigma \quad \text{and} \quad \rho^\sigma \mathbf{u}^\sigma = \sum_i f_i^\sigma \mathbf{c}_i,$$

respectively. The fluid viscosity $\nu^\sigma$ is proportional to $\left(\tau^\sigma - \frac{1}{2}\right)$ and the mass of each particle is $m^\sigma$. This results in a lattice BGK equation (2.5) of the form

$$f_i^\sigma(\mathbf{r}, t + 1) - f_i^\sigma(\mathbf{r}, t) = -\frac{1}{\tau^\sigma} \left[ f_i^\sigma - N_i(\rho^\sigma, \mathbf{u}^\sigma) \right]. \quad (2.6)$$

The velocity $\mathbf{v}^\sigma$ is found by calculating a weighted average velocity

$$\mathbf{u}' = \left( \sum_\sigma \frac{\rho^\sigma}{\tau^\sigma} \mathbf{u}^\sigma \right) \left( \sum_\sigma \frac{\rho^\sigma}{\tau^\sigma} \right)^{-1}, \quad (2.7)$$

and then adding a term $\mathbf{F}^\sigma$ to account for additional forces:

$$\mathbf{v}^\sigma = \mathbf{u}' + \frac{\tau^\sigma}{\rho^\sigma} \mathbf{F}^\sigma. \quad (2.8)$$

To produce nearest-neighbour interactions between fluid components, this term assumes the form

$$\mathbf{F}^\sigma = -\psi^\sigma(\mathbf{x}) \sum_\sigma g_{\sigma \bar{\sigma}} \sum_i \psi^\bar{\sigma}(\mathbf{x} + \mathbf{c}_i) \mathbf{c}_i, \quad (2.9)$$

where $\psi^\sigma(\mathbf{x}) = \psi^\sigma(\rho^\sigma(\mathbf{x}))$ is an effective charge for component $\sigma$, set equal to the fluid component density, that is $\psi^\sigma(\mathbf{x}) = \rho^\sigma(\mathbf{x})$; $g_{\sigma \bar{\sigma}}$ is a coupling constant controlling the strength of the interaction between two components $\sigma$ and $\bar{\sigma}$. If $g_{\sigma \bar{\sigma}}$ is set to zero for $\sigma = \bar{\sigma}$, and to a positive value for $\sigma \neq \bar{\sigma}$, then, in the interfacial region between bulk domains of each component, particles experience a force in the direction away from the interface, producing immiscibility. For two-component systems, we use the notation $g_{\sigma \bar{\sigma}} = g_{\sigma \sigma} = g_{br}$. External forces are added in a similar manner. For example, in order to produce a gravitational force acting in the $z$-direction, the force term $\mathbf{F}^\sigma$ can take the form $g \rho^\sigma \mathbf{zh}$.

A convenient way to characterize binary fluid mixtures is in terms of the order parameter or colour field

$$\phi(\mathbf{x}) = \rho^r(\mathbf{x}) - \rho^b(\mathbf{x}). \quad (2.10)$$

The order parameter is positive in areas of high concentration of ‘red’ fluid and negative in areas of ‘blue’ dominance; the isosurface $\phi(\mathbf{x}) = 0$ denotes the interface between both fluid constituents.

The model has been extended to handle amphiphiles, which are treated as massive point like dipoles with different interaction strengths on each end and an orientational degree of freedom (Chen et al. 2000). Our code, LB3D, can handle binary and ternary fluid mixtures with or without amphiphiles. But since we only discuss simulations of binary fluids in this paper, we refer the reader to other papers (Chen et al. 2000; Love et al. 2003; Nekovee et al. 2001) for a more comprehensive description of the amphiphilic case.
Grid-enabled LB simulations of complex fluid flow

3. Computational steering of lattice Boltzmann simulations

This section outlines the benefits of computational steering for high-performance computing applications. Our three-dimensional lattice Boltzmann code (LB3D) and its computational steering implementation are used to illustrate the substantial improvements which computational steering offers in terms of resource efficiency and time to discover new physics.

Traditionally, large, compute-intensive simulations are run non-interactively. A text file describing the initial conditions and parameters for the course of a simulation is prepared, and then the simulation is submitted to a batch queue on a large compute resource. The simulation runs entirely according to the prepared input file, and outputs the results to disk for the user to copy to his local machine and examine later.

This mode of working is sufficient for many simple investigations of mesoscale fluid behaviour, but has several drawbacks. Firstly, consider the situation where one wishes to examine the dynamics of the separation of two immiscible fluids: this is a subject which has been of considerable interest in the modelling community in recent years (González-Segredo et al. 2003; Kendon et al. 2001). Typically, a guess is made as to how long the simulation must run before producing a phase separation, and then the code is run for a fixed number of time-steps. If a phase transition does not occur within this number of time-steps, then the job must be resubmitted to the batch queue and restarted. However, if a phase transition occurs in the early stages of the simulation, then the rest of the computation time will be spent simulating an equilibrium system of very little interest. Even worse, the initial parameters of the system might turn out not to produce a phase separation at all and all of the CPU time invested in the simulation will have been wasted.

Computational steering is a way to overcome these drawbacks. It allows the scientist to interact with a running simulation and to change or monitor simulation parameters on the fly. Examples of monitored parameters are the time-step, surface tension, density distributions or ‘colour’ fields. Steerable parameters are data-dumping frequencies, relaxation times or shear rates. One can also ‘stop’, ‘pause’ or ‘restart’ from a previously saved checkpoint. A ‘checkpoint’ is a set of files representing the state of the simulation and allowing the code to be restarted without rerunning earlier steps of the simulation. The ‘restart’ functionality is particularly important, since it provides the basis of a system that allows the scientist to ‘rewind’ a simulation. Having done so, it can then be run again, perhaps after having steered some parameter or altered the frequency with which data from the simulation are recorded.

We have implemented computational steering within the LB3D code with the help of colleagues at Manchester Computing as part of the ongoing RealityGrid project (http://www.realitygrid.org) (Brooke et al. 2003; Chin et al. 2003; Coveney 2003). The RealityGrid project aims to enable the modelling and simulation of complex condensed matter structures at the molecular and mesoscale levels as well as the discovery of new materials using computational grids. The project also involves biomolecular applications and its long-term ambition is to provide generic computational-grid-based technology for scientific, medical and commercial activities.

Within RealityGrid, computational steering has been implemented in order to enable existing scientific computer programs (often written in Fortran90 and designed...
for multi-processor/parallel supercomputers) to be made steerable while minimizing
the amount of work required. The steering software has been implemented as a library
written in C and is thus callable from a variety of languages (including C, C++ and
Fortran90). The library completely insulates the application from any implemen-
tation details. For instance, the process by which messages are transferred between
the steering client and the application (e.g. via files or sockets) is completely hid-
then from the application code and the steering library does not assume or prescribe
any particular parallel-programming paradigm (such as message passing or shared
memory). The steering protocol has been designed so that the use of steering is
never critical to the simulation. Thus, a steering client can attach and detach from
a running application without affecting its state.

The ability to monitor the state of a simulation and use this to make steering
decisions is very important. While a steering client provides some information via
the simulation’s monitored parameters, a visualization of some aspect of the simul-
ation’s state is often required. In our case this is usually a three-dimensional dataset,
visualized by a second software component using isosurfacing or volume rendering.

The steering library itself consists of two parts: an application side and a client
side. Using a generic steering client written in C++ and Qt (a graphical user interface
toolkit (http://www.trolltech.org)), one is capable of steering any application that
has been ‘steering enabled’ using the library.

The RealityGrid steering library and client are generic enough to be interfaced to
by almost any simulation code. Usually only a couple of hours have to be invested
in adapting a code to do simple parameter steering and monitoring. Indeed, since
the initial version of the steering library was written at least four other codes have
been made steerable in this way (these include molecular dynamics, Monte Carlo
and other lattice Boltzmann codes).

In addition to the features the steering library provides, LB3D has its own logging
and replay facilities which permit the user to ‘replay’ a steered simulation. This
is an important feature, since it allows the data from steered simulations to be
reproduced without human intervention. Moreover, this feature can be used as an
‘auto-steerer’. Thus multiple simulations, which read different input files at start-
up and are ‘steered’ in the same way, can be launched without the need for human
intervention during the simulation. One application of this particular feature appears
in studies of how changes in parameters affect a simulation that has evolved for
a given number of time-steps. Another application is the automatic adaptation of
data-dumping or checkpointing frequencies. If the user has found from a manually
steered simulation that no effects of interest are expected for a given number of initial
time-steps, he can reduce the amount of data written to disk for early times of the
simulation.

A more detailed description of computational steering and its implementation
within RealityGrid can be found in recently published papers (Brooke et al. 2003;
Chin et al. 2003). Chin et al. (2003) also contains an example demonstrating the use-
fulness of computational steering of three-dimensional lattice Boltzmann simulations:
parameter searches are a common task we have to handle because our lattice Boltz-
mann method has a range of free parameters. Only by choosing them correctly can
one simulate effects of physical interest. Previously, these parameter searches have
been performed using a taskfarming approach: many small simulations with different
parameters have been launched. In such cases we have used up many thousands of

CPU hours and needed hundreds of gigabytes of simulation data to be stored for a single large-scale parameter search. Computational steering offers the possibility to ‘joystick’ through parameter space in order to find regions of interest. In this way, the resources needed can be substantially reduced. The main benefit, however, is the reduced amount of data that has to be analysed subsequently, since this is the most time-consuming and demanding task. While simulations can be completed within days or weeks, analysis usually takes months. We have also found that if the amount of data to be searched for interesting effects exceeds certain bounds, it is almost impossible for a human to keep track of it. One might suggest automation of the analysis process, but the human eye turned out to be the only reliable tool for our simulations. It is often very easy to spot effects occurring in datasets by looking at isosurfaces or volume-rendered visualizations; by contrast automation of the analysis of the generated data is much harder because it can be difficult to define the effects sufficiently well, impossible to anticipate the effects sought in advance, or simply not worthwhile to invest additional effort in the development of algorithms to automate the process.

4. Capability computing and terascale computational grids

Three-dimensional lattice Boltzmann simulations are very computationally demanding and need high performance computing resources (i.e. supercomputers). In order to reach length- and time-scales which can be compared with experimental data and to eliminate finite size effects, one needs large lattices, for example, $512^3$ or $1024^3$. Simulations also have to run for several thousands or tens of thousands of time-steps, thus pushing the required computation and storage resources beyond what is typically available to users on medium scale supercomputers today. In the case of LB3D, the main restriction is the per-CPU memory available, which, on all machines we have access to currently, is not more than 1 Gb. For example, we require at least 1024 CPUs to simulate a $1024^3$ system.

Obviously, computational steering becomes an even more useful tool here because large-scale simulations are very expensive; it is essential that the simulation does not generate useless data, and that the expensive resources are used as efficiently as possible. The need for vast computational power has brought with it the concept of ‘capability computing’. We understand this term as a description of how large jobs are handled by supercomputing centres: large jobs are favoured and assigned a higher priority by the queueing system. In these terms, ‘large’ refers to jobs that request at least half of the total number of CPUs available. With standard job-queue configurations operating on batch systems, there is a strong disincentive to submit large jobs: if a user submits a ‘large’ job, turnaround times can be very long, making such high-end resources uncompetitive compared with modern commodity clusters, which are becoming widely available locally. In some cases, supercomputing centres can offer discounts for large (capability computing) jobs if the simulation code can be shown to scale well. LB3D has recently been awarded a gold-star rating for its excellent scaling capabilities by the HPCx Consortium (http://www.hpcx.ac.uk) allowing us to run simulations on 1024 CPUs (the full production partition) of their 1280 CPU IBM SP4, with a discount of 30% (Harting et al. 2003). The flow in porous media simulations described later in this section have been done on up to 504 CPUs of a 512 CPU SGI Origin 3800 at the CSAR service in Manchester, UK.
J. Harting, M. Venturoli and P. V. Coveney
(http://www.csar.cfs.ac.uk). LB3D scales linearly on all platforms available to us. In addition to those mentioned above, these include a CRAY T3E, a 3000 CPU Compaq Alpha cluster ‘Lemieux’ (at Pittsburgh Supercomputing Center), various Linux clusters and SGI Origin 2000 and 3800 systems.

We expect it to become easier to simulate systems of a size which is comparable with the experimental data with the advent of computational grids (Foster & Kesselman 1999, pp. 15–25; Berman et al. 2003). Grids are geographically distributed and dynamically varying collections of resources such as supercomputers, storage facilities or advanced experimental instruments that are connected by high-speed networks, thus allowing widespread human collaborators to work together closely. In the same way that large-scale lattice Boltzmann simulations require a supercomputer, the visualization of the large and complex datasets that these simulations produce also require specialist hardware that few scientists have direct access to. As we shall see, visualization engines can also be treated as distributed resources on the grid.

Computational grids are related to traditional distributed computing, with the major extension that they enable the supposedly transparent sharing and collective use of resources across multiple administrative domains which would otherwise be individual and isolated facilities. With growing intensity, significant effort is being invested worldwide in grid computing (http://www.gridforum.org). The grid aims to present the elements required for a computational task (e.g. calculation engine, filters, visualization capability) as components which can be effectively and transparently coupled through the grid framework using middleware. In this scenario, any application or simulation code can be viewed simply as a data-producing or data-consuming object on the grid and computational steering as a way of allowing users to interact with such objects.

Our LB3D code is now a fully grid-enabled, steerable application. LB3D simulations can then be launched and steered on a remote machine, with the visualizations being performed in other geographic locations. One or more users control the workflow from a laptop running a steering client and client software to interact remotely with the computation and visualization engines. Behind the scenes, the ‘grid middleware’ moves files, simulation data and commands between the resources involved. We have used both Globus (http://www.globus.org) (Foster & Kesselman 1999, p.259) and Unicore (http://www.unicore.org) as the basic middleware fabric in this work, and digital certificates provided by the UK e-science certification authority (http://www.grid-support.ac.uk).

The grids being used in these demonstration activities have been assembled especially for each event. By contrast, the UK e-science community has constructed an ambitious level-two grid (http://www.grid-support.ac.uk/l2g) and more recently still a National Grid Service (http://www.ngs.ac.uk) that aim to provide the user community with a persistent grid of heterogenous resources. LB3D and the RealityGrid steering framework have already been deployed on this level-two grid, which uses Globus GT2 as middleware. Thus we are amongst the first groups in the world to use a persistent grid for scientific research requiring high-performance computing and computational steering.

In a major US–UK grid project leading up to and including the Supercomputing 2003 conference, we studied the defect formation and dynamics within a self-assembled gyroid mesophase (González-Segredo & Coveney 2004) using a fast network between the ‘Extended Terascale Facility’ (http://www.teragrid.org) in the
USA and the national supercomputing centres at Manchester (CSAR) and Daresbury (HPCx) in the UK. This gives us access to machines in the US including Lemieux and various Itanium (IA64) systems. In the UK, access to the total combined resources of CSAR and HPCx provides us with a 1280 CPU IBM SP4 machine, a 504 CPU SGI Origin 3800, and a 256 CPU SGI Altix. For visualization there are resources available on both sides of the Atlantic as well, including various SGI Onyx machines and commodity clusters which use ‘Chromium’ (http://chromium.sourceforge.net) for parallel rendering. The Visualization Toolkit (VTK, http://www.kitware.com) and more specialized ray-tracing codes allow us to generate isosurfaces or volume-rendered visualizations of even our largest datasets on these platforms (see http://www.realitygrid.org/TeraGyroid.html).

However, the vision of a computational grid that furnishes for information technology what electricity (and other utility) grids have achieved in terms of almost universal and transparent access to energy (and other resources) within modern civilized societies is still a dream today. Many problems remain to be addressed before computational grids become easy to use. At the time of writing, it remains very awkward to both access and use grid-enabled resources, and the much vaunted advantages are yet to be realized. In fact, our own experiences indicate that real progress towards usability will be achieved most quickly with the development and deployment of lightweight middleware, in marked contrast with the existing heavyweight behemoths (Chin & Coveney 2004).

Moreover, for effective use of computational steering together with large-scale simulations, it is very important that supercomputing centres change their policy of job scheduling since advanced reservation for the co-allocation of computation and visualization resources becomes essential. Today, this is possible for small scale simulations which do not run on the grid if turnaround times are short, but for large-scale jobs one needs special arrangements with the resource owners. It is also important that users will be able to request access to resources at convenient times, i.e. during working hours rather than in the middle of the night. We expect that the huge effort currently being invested in the development of grid standards will result in a satisfactory solution of these issues. We believe that computational grids will revolutionize the way scientific simulations are performed in the future because they should then offer an easy and effective way to access distributed resources in an optimal way for the scientific problem under investigation.

5. Immiscible fluid mixtures under shear

Lees & Edwards published their method for the application of shearing in molecular dynamics simulations in 1972 (Lees & Edwards 1972). Since then Lees–Edwards boundary conditions have become a popular method for simulating fluid rheology under shear using a variety of different methods and have been implemented in lattice Boltzmann codes before (Wagner & Pagonabarraga 2002; Wagner & Yeomans 1999).

The method can be described as an extension of the use of standard periodic boundary conditions and is illustrated in figure 1. While with periodic boundary conditions particles that arrive at a system boundary leave the simulation volume and ‘re-enter’ it on the opposite side, for a sheared system this is only true for the boundaries not subject to shear. Particles crossing the shear planes, which are the \( x = 0 \) and \( x = nx \) planes in our case, get their \( z \)-velocities altered by \( \pm \Delta u_z \) and
are displaced in the $\pm z$-direction by $d = t\Delta u_Z$ ($t$ is the simulation time and $nx$ is the lattice size in the $x$-direction). The corresponding shear rate $s$ is $\Delta u_Z/nx$. This algorithm can be extended for simulations of fluids under oscillatory shear by multiplying $\Delta u_Z$ with a time-dependent cosine function of frequency $\omega$: $\Delta u'_Z = \cos(\omega t)\Delta u_Z$.

We have applied our model to study spinodal decomposition under shear, also referred to as Couette flow. The phase separation of binary immiscible fluids without shear has been studied in detail by different authors and LB3D has been shown to model the underlying physics successfully (González-Segredo et al. 2003). It has been shown in the non-sheared studies of spinodal decomposition that lattice sizes need to be large in order to overcome finite size effects, i.e. $128^3$ has been found the minimum acceptable number of lattice sites (González-Segredo et al. 2003). For high shear rates, systems also have to be very long because, if the system is too small, the domains interconnect across the $z = 0$ and $z = nx$ boundary and form interconnected lamellae in the direction of shear. Such artefacts need to be eliminated from our simulations.

Computational steering is a very useful tool for checking on finite size effects in an ongoing sheared fluid simulation. By virtue of its ability to constantly monitor volume-rendered colour fields or fluid densities, the human eye turned out to be very reliable in spotting the moment when these simulations become unphysical. In this way, we were able to keep the computational resources required to a minimum.

From our studies we found that to avoid finite size effects $64 \times 64 \times 512$ systems are sufficient for low shear rates and short simulation times, but $128 \times 128 \times 1024$ lattices are needed for higher shear rates and/or very long simulations.

The results presented in this section were all obtained for a $64 \times 64 \times 512$ system with all relaxation times and masses set to unity, i.e. $\tau^\sigma = 1.0$, $m^\sigma = 1.0$. The initial oil and water fluid densities $f_r$ and $f_b$ were given by a random distribution between 0.0 and 0.7 (in lattice units). All simulations were performed on 64 CPUs of a SGI Origin 3800 at CSAR in Manchester, UK. Shear rates $s$ were set to $0$, $7.8 \times 10^{-4}$, $1.6 \times 10^{-3}$, $3.1 \times 10^{-3}$ and $4.7 \times 10^{-3}$ (in lattice units) in order to study the influence of shear on the domain growth. In order to compare different simulations, we define the time-dependent lateral domain size $L_i(t)$ along direction $i = x, y, z$ as

$$L_i(t) \equiv \frac{2\pi}{\sqrt{\langle k^2_i(t) \rangle}}, \quad (5.1)$$
Grid-enabled LB simulations of complex fluid flow

Figure 2. The time-dependent lateral domain size $L$ in the (a) $x$- and (b) $z$-directions for different shear rates. Initial fluid densities are randomly distributed between 0 and 0.7, the system size is $64 \times 64 \times 512$, and $\tau^\sigma = m^\sigma = 1$ (all quantities are reported in lattice units).

where

$$
\langle k_i^2(t) \rangle = \frac{\sum_k k_i^2 S(k, t)}{\sum_k S(k, t)}
$$

(5.2)

is the second-order moment of the three-dimensional structure function

$$
S(k, t) \equiv \frac{1}{V} |\phi_k'(t)|^2
$$

(5.3)

with respect to the Cartesian component $i$. $\langle \cdot \rangle$ denotes the average in Fourier space, weighted by $S(k)$ and $V$ is the number of nodes of the lattice, $\phi_k'(t)$ the Fourier transform of the fluctuations of the order parameter $\phi' \equiv \phi - \langle \phi \rangle$, and $k_i$ is the $i$th component of the wavevector.

Figure 2 shows the time-dependent lateral domain size in $x$- and $z$-directions for the above-mentioned parameters up to 4000 time-steps. At the beginning of simulations, there is a steep increase of $L$ due to rapid diffusion of mass to nearest neighbours before the domain growth starts. As expected, the behaviour of $L(t)$ is identical in $x$- and $z$-directions for $s = 0.0$, but is very different for $s > 0.0$. The average slope of $L_x(t)$ decreases for increasing $s$ until phase separation almost arrests and multiple peaks occur for $s = 1.6 \times 10^{-3}$ and $s = 3.1 \times 10^{-3}$ (in lattice units). These peaks arise very regularly at approximately every 700 time-steps in the former case and every 1500 time-steps in the latter case. For $s = 4.7 \times 10^{-3}$ these peaks cannot be observed. They can be explained as follows: if a domain reaches a substantial size, the probability of it coalescing with a similarly sized one becomes high, but the resulting very large domain will be unable to withstand the strain caused by the shear and will break up a few time-steps later. For higher shear rates, domain growth in the $x$-direction is slower than for lower shear rates and the peaks occur with a lower frequency. If $s$ becomes too high (as in the $s = 4.7 \times 10^{-3}$ case), the imposed strain prevents domains substantially larger than the average domain size from forming.

In the $z$-direction, shear causes elongated domains, resulting in increasing values of $L(t)$ for increasing shear rates. For $s = 4.7 \times 10^{-3}$, $L_z(t)$ grows rapidly until it saturates at $t = 1100$. A critical domain size is reached, after which domains still grow but are very elongated and tilted by an angle. Due to this tilting, $L_z$ saturates, since it measures only the size of the domains in the $z$-direction.
This effect is illustrated in figure 3, which shows volume-rendered snapshots of the order parameter $\phi$ for shear rates $s = 0.0, 1.6 \times 10^{-3}$, and $4.7 \times 10^{-3}$, all at time-step 3000. Areas of high density of ‘blue’ fluid are coloured blue and the interface between both fluids is coloured red. The figure shows how the domains develop evenly in the $x$- and $z$-directions for $s = 0.0$, but become tilted and elongated under shear. This effect increases with increasing shear rate, resulting in very long and slim lamellae in the $s = 4.7 \times 10^{-3}$ case.

The results presented in this section are preliminary. We hope to be able to report on more detailed studies of essentially finite-size-free simulations of complex fluid mixtures under shear in the near future. We are planning to use our code to quantify domain growth and compare our results with previous theoretical work and experimental results (Cates et al. 1999; Corberi et al. 2002; Wagner & Yeomans 1999). Applying oscillatory instead of constant shear is a natural extension of this work and simulations are already ongoing (Xu et al. 2003). We are also studying the properties of amphiphilic fluid mixtures under shear including the effect shear has on previously formed cubic mesophases such as the ‘P’-phase (Nekovee & Coveney 2001) and the
gyroid cubic mesophase (González-Segredo & Coveney 2004). Such complex fluids are expected to exhibit non-Newtonian properties.

6. Multiphase flow in porous media

The study of transport phenomena in porous media is of great interest in fields ranging from oil recovery and water purification to industrial processes such as catalysis. In particular, the oilfield industry uses complex, non-Newtonian, multi-component fluids (containing polymers, surfactants and/or colloids, brine, oil and/or gas), for processes like fracturing, well stimulation and enhanced oil recovery. The rheology and flow behaviour of these complex fluids in a rock is different from their bulk properties. It is therefore of considerable interest to be able to characterize and predict the flow of these fluids in porous media.

The flow of a single phase non-Newtonian fluid through two-dimensional porous media has been addressed with lattice Boltzmann methods (Aharonov & Rothman 1993; Chin et al. 2002), using a ‘top-down’ approach, in which the effective dynamic viscosity of the fluid, and hence the relaxation parameter in the BGK equation, \(\lambda\), explicitly depends on the strain rate tensor through a power law. However, from the point of view of a modelling approach, the treatment of complex fluids in three-dimensional complex geometries is an ambitious goal. In this paper we shall only consider binary (oil–water) mixtures of Newtonian fluids, since this is a first and necessary step in the understanding of multiphase fluid flow in porous media.

The advantage of using lattice Boltzmann (or lattice gas) techniques in studying flow in porous media is that complex geometries can be easily implemented and the flow problem solved therein, since the evolution of the particle distribution functions can be described in terms of local collisions with the obstacle sites using simple bounce-back boundary conditions. Synchrotron-based X-ray microtomography (XMT) imaging techniques provide high-resolution, three-dimensional digitized images of rock samples. By using the lattice Boltzmann approach in combination with these high-resolution images of rocks, not only it is possible to compute macroscopic transport coefficients, such as the permeability of the medium, but information on local fields, such as velocity or fluid densities, can also be obtained at the pore scale, thus providing a detailed insight into local flow characterization and assisting in the interpretation of experimental measurements (Auzerais et al. 1996).

The XMT technique measures the linear attenuation coefficient from which the mineral concentration and composition of the rock can be computed. From the tomographic image of the rock volume the topology of the void space can be derived, including pore-size distribution and tortuosity, and the permeability and conductivity of the rock can be computed (Spanne et al. 1994). The tomographic data are represented by a reflectivity greyscale value and are arranged in voxels in a three-dimensional coordinate system. The linear size of each voxel is defined by the imaging resolution, which is usually of the order of micrometres. By introducing a threshold to discriminate between pore sites and rock sites, these greyscale images can be reduced to a binary (zeros and ones) representation of the rock geometry. Using the lattice Boltzmann method, single phase or multiphase flow can then be described in these real porous media.

Lattice Boltzmann and lattice gas techniques have already been applied to study single and multiphase flow through three-dimensional micro-tomographic reconstruc-
The possibility of describing fluid flow in real rock samples furnishes the possibility of being able to make comparisons with experimental results obtained on the same, or similar, pieces of rock. Of course, to achieve a reasonable comparison, the size of the rock used in lattice Boltzmann simulations should be of the same order of magnitude as the system used in the experiments, or at least large enough to capture the rock’s topological features. The more inhomogeneous the rock, the larger the sample size needs to be in order to describe the correct pore distribution and connectivity. Another reason for needing to use large lattice sizes is the influence of boundary conditions and lattice resolution on the accuracy of the lattice Boltzmann method. It has been shown (see, for example, He et al. 1997; Chen & Doolen 1998; and references therein) that if the BGK (Bhatnagar et al. 1954) approximation of the lattice Boltzmann equation is used, the so-called bounce-back boundary condition at the wall sites actually mimics boundaries which move with a speed that depends on the relaxation parameter \( \tau \) of the collision operator in the BGK equation (2.5). The relaxation parameter determines the kinematic viscosity of the simulated fluid. This implies that the computed permeability is a function of the viscosity.

The accuracy of lattice Boltzmann simulations also depends on the Knudsen number, which represents the ratio of the mean free path of the fluid particles and the characteristic length-scale of the system (such as the pore diameter). To accurately describe hydrodynamic behaviour this ratio has to be small. If the pores are resolved with an insufficient number of lattice points, finite size effects arise, leading to an inaccurate description of the flow field.

The error in solving the flow field increases with increasing viscosity (or equivalently relaxation time), but this viscosity dependence becomes weak with increasing lattice resolution. Hence it is desirable to use a high resolution within the pore space in order to decrease the error induced by the use of bounce-back boundary conditions. However, increasing the resolution means increasing the lattice size, and hence the computational cost of the simulation.

Boundary conditions other than bounce-back have been proposed and shown to give correct velocities at the boundaries. However, these methods are either suitable only for flat interfaces (Inamuro et al. 1995) or cumbersome to implement (Verberg & Ladd 2002), reducing the efficiency of the lattice Boltzmann method.

Large lattices require a highly scalable code, access to high-performance computing, terascale storage facilities and high-performance visualization. LB3D provides the first of these, while the others are now offered by the UK High Performance Computing services, and are also accessible via the UK e-science grid using RealityGrid capabilities.

Using LB3D and capability computing services provided by the Manchester CSAR SGI Origin 3800 supercomputer, we were able to simulate drainage and imbibition processes in a \( 512^3 \) subsample of Bentheimer sandstone X-ray tomographic data. The whole set of XMT data represented the image of a Bentheimer sandstone of cylindrical shape with diameter 4 mm and length 3 mm. The XMT data were obtained at the European Synchrotron Research Facility (Grenoble) at a resolution of 4.9 \( \mu m \).
resulting in a dataset of approximately $816 \times 816 \times 612$ voxels. Figure 4 shows a snapshot of the $512^3$ subsystem.

The aim of this study is to compare velocity distributions with the ones measured by MRI of oil and brine infiltration into saturated Bentheimer rock core (Sheppard et al. 2003). The rock sample used in these MRI experiments had a diameter of 38 mm and was 70 mm long; three-dimensional images of the rock were acquired at a resolution of 280 $\mu$m. The system size we used in lattice Boltzmann simulations was smaller than the sample used in MRI experiments, but still of a similar order of magnitude and large enough to represent the rock geometry. On the other hand, the higher spatial resolution provided by the lattice Boltzmann method permits a detailed characterization of the flow field in the pore space, hence providing a useful tool with which to interpret the MRI experiments, for example, in identifying regions of stagnant fluid. Detailed results from these lattice Boltzmann simulations will be presented in a future paper.

(a) Binary flow in a digitized Bentheimer rock sample

Darcy’s (1856) law for binary immiscible (oil–water) fluid mixtures was investigated and the dependence of the relative permeability coefficients on water saturation was derived and compared with lattice gas studies. The extended Darcy law for binary flow takes the form

$$J_i = \sum_{j=1}^{2} k_{ij}(S) \frac{k}{\mu_i} X_j, \quad i = 1, 2,$$

where $J_i$ is the flux of the $i$th component and $X_j$ is the force acting on the $j$th component. $k_{ij}(S)$ is the relative permeability coefficient depending on the saturation $S$, $k$ is the permeability of the medium and $\mu_i$ the viscosity of component $i$.

Using the lattice Boltzmann method, it is easy to selectively force only one component in a binary mixture, leaving the other one unforced. In this way the diagonal terms of the relative permeability matrix ($k_{ii}$) can be computed by analysing the flux of one component when it is forced, while the cross terms ($k_{ij}$) can be computed from the flux of one component when the other one is forced.

Since we want to study the flow behaviour for different forcing levels and for forcing applied in turn to both fluids, a large number of simulations is required. Hence, we limited the size of our system to a subsample of $64 \times 64 \times 32$ voxels, mirrored in the $z$-direction (flow direction) to give a final size of $64^3$ lattice sites (see figure 5). Periodic boundary conditions were applied in all directions. Immiscible, binary mixtures of oil and water were flowed in this sample, at different forcing levels and by forcing in turn either the water or the oil component. In each of these numerical experiments, the system was initialized with a 50:50 mixture of oil and water, both given the same viscosity and the same initial density distributions. The rock walls were made fully water wettable, to reproduce experimental conditions and to discriminate between the two fluids which would otherwise be equivalent. The rock wettability is implemented by assigning each rock site a density distribution equal to the initial density of water. These density distributions do not flow, but exert a (repulsive) force on the oil component, pushing it away from the rock walls.

As the two components flow, they initially phase separate and, after some time, a steady state is reached. Here we are only interested in the velocity field at the steady
Figure 4. Rendering of an X-ray micro-tomographic image of a 512$^3$ sample of Bentheimer sandstone. The data have a resolution of 4.9 $\mu$m. The pore space is shown in red, while the rock is represented in blue.

state. If the total time allocated for the simulation is longer than the time needed to reach the steady state, we would waste CPU time performing calculations which are of no value. If we run a simulation for less time than needed to reach the steady state, using the checkpoint-restart facility within LB3D we can resume the simulation and continue the run until the steady state is reached. In both cases, steering improves the efficiency of the runs. By steering we can dump velocity and density fields at variable frequency, check whether the fields at two different times differ or not and, in case their difference is less than a given threshold, decide whether the steady state is reached and stop the simulation. In the simulations presented here an average of 15,000 time-steps is needed to reach the steady state. At this time the total flux (normalized by the pore volume) can be computed.

Figure 6$^a$ shows the force/flux dependence for the forced fluids. The linearity of force/flux holds for all the forcing levels considered. At any given forcing level, the wetting fluid flows less than the non-wetting one. This is due to the fact that the wetting fluid interacts with the rock walls, and adheres to them, hence exerting more resistance to flow, while the non-wetting fluid is lubricated by the wetting fluid. Similar results have been achieved in three-dimensional lattice gas studies of binary flow in Fontainebleau sandstone (Love et al. 2001; Olson & Rothman 1997). A difference between our results and those for the lattice gas is that in the latter the authors observed the presence of a capillary threshold, a minimum forcing level required to make the non-wetting fluid flow, while here we observe flow even at small forcing levels. The presence of this threshold is due to geometric constraints imposed by the distribution and size of rock pores and throats, which can trap bubbles of the non-wetting fluid. In the rock sample we used for this study there are no such narrow throats and hence we would not expect to observe any capillary threshold.

In figure 6$^b$ the force/flux relation is plotted for the non-forced fluids. In this case at any applied forcing the non-wetting fluid flows more than the wetting one. For the non-wetting fluid we observe viscous coupling, i.e. the fluid flows even if it is
Grid-enabled LB simulations of complex fluid flow

Figure 5. A $64^3$ sample of Bentheimer sandstone at a lattice resolution of 4.9 µm. The figure shows the isosurface delimiting the pore volume. The pore space is in the inner region of the isosurface.

Figure 6. Flux of binary fluid when (a) forced and (b) unforced. All quantities are given in lattice units and the flux is normalized by the pore volume.

not directly forced, and Darcy’s linearity is found for all forcing levels. On the other hand, the wetting fluid does not flow until a sufficiently high force is applied. This is due to the capillary forces. Hence Darcy’s law is observed to hold only for sufficiently high forcing.

From the linear regime regions in both graphs we computed the relative permeability coefficients: $k_{ww} = 1.6$, $k_{wo} = 3.3$, $k_{ow} = 0.77$, $k_{wo} = 0.57$, where the subscripts indicate water (w) and oil (o). The diagonal terms are one order of magnitude larger than the cross terms, which can be expected because the cross terms represent the response of one fluid when the other one is forced. This is also in agreement with the results from lattice gas studies (Love et al. 2001).

A previous much debated issue is whether the off-diagonal terms in the relative permeability matrix should satisfy a reciprocity relationship. The reciprocity of the

coefficients in macroscopic linear transport laws of the form

\[ J_i = \sum_j L_{ij} X_j, \]  

(6.2)

where \( J_i \) is a current and \( X_j \) is a force conjugate to the current, is a consequence of Onsager’s regression hypothesis and it holds for systems which are linearly perturbed from equilibrium (de Groot & Mazur 1985).

Our results show a linear dependence between force and flow, but the off-diagonal coefficients we obtained from the linear regime region have slightly different values for the wetting and non-wetting fluids. For binary immiscible fluid flow in porous media, where complex interfacial dynamics plays a major role, it is not clear if Onsager’s theory can be applied. In the aforementioned lattice gas studies, Onsager’s reciprocity was found, but no clear theoretical justification has been given for the reason that this should hold under the general nonlinear conditions pertaining (Flekkøy & Pride 1999).

It can also be observed that there are no error bars provided with our results. This is due to the fact that in lattice Boltzmann, as opposed to lattice gas simulations, there is no noise, and indeed this is one of the major computational advantages of the method. Nevertheless, we plan to perform the same set of simulations starting from different initial conditions, which may lead to different fluid–fluid interfacial structures and fluid transport coefficients, and then derive error bars from the flow computed in each of these simulations. More studies on different pore space geometries and larger systems need to be done to address the general validity of Darcy’s law for binary mixtures and Onsager’s reciprocity hypothesis.

7. Conclusions

This paper describes optimal implementation of large-scale lattice Boltzmann simulations of two-phase fluid dynamics through exploitation of capability computing and computational steering. Capability computing promotes the execution of scalable codes that use a large fraction and sometime the entire allocation of processors on a big supercomputer. Computational steering then ensures that this massive set of resources is used optimally to generate meaningful scientific data. We illustrated the use of these approaches by reporting preliminary results from two applications which benefit substantially from large-scale simulation. The first of these was concerned with Couette (shear) flow, where simulation cells of high aspect ratio are needed to eliminate finite size effects; the second described two-phase flow in large portions of digitized data obtained from X-ray micro-tomographic studies of Bentheimer sandstone. The most efficient use of computational steering of such large-scale simulations uses a computational grid. These grids are in their infancy today and much more work needs to be done to render them transparent to users. Nevertheless, important advances have already been made, and here we described the grid-enablement of our lattice Boltzmann codes. Our experience with grids to date leads us to conclude that much lighter middleware solutions will be required to foster their widespread use.

We thank J. Chin, N. González-Segredo and S. Jha (University College London), A. R. Porter and S. M. Pickles (University of Manchester), E. S. Boek, and J. Crawshaw (Schlumberger Cambridge Research) for fruitful discussions, and E. Breitmoser from the Edinburgh Parallel Computing Centre for her contribution to the Lees–Edwards routines in our code. We are grateful...
to EPSRC for funding much of this research through RealityGrid grant no. GR/R67699 and for providing access to SGI Origin 3800 and Origin 2000 supercomputers at Computer Services for Academic Research (CSAR), Manchester, UK.

We acknowledge the European Synchrotron Radiation Facility for provision of synchrotron radiation facilities and we thank Dr Peter Cloetens for assistance in using beamline ID19, as well as Professor Jim Elliott and Dr Graham Davis of Queen Mary, University of London, for their work in collecting the raw data and reconstructing the X-ray micro-tomography datasets used in our Bentheimer sandstone images.

References


Coveney, P. V. 2003 Phil. Trans. R. Soc. Lond. A 361, 1057.


J. Harting, M. Venturoli and P. V. Coveney

Succi, S. 2001 The lattice Boltzmann equation for fluid dynamics and beyond. Oxford University Press
Detection and tracking of defects in the gyroid mesophase

J. Harting, M. Harvey, J. Chin, and P. V. Coveney
Detection and tracking of defects in the gyroid mesophase

Jens Harting a, Matthew J. Harvey b, Jonathan Chin b, Peter V. Coveney b,*

a Institute for Computational Physics, University of Stuttgart, Pfaffenwaldring 27, D-70569 Stuttgart, Germany
b Centre for Computational Science, Christopher Ingold Laboratories, University College London, 20 Gordon Street, London WC1H 0AJ, UK

Received 6 May 2004; accepted 3 September 2004
Available online 8 December 2004

Abstract
Certain systems, such as amphiphile solutions or diblock copolymer melts, may assemble into structures called “mesophases”, with properties intermediate between those of a solid and a liquid. These mesophases can be of very regular structure, but may contain defects and grain boundaries. Different visualization techniques such as volume rendering or isosurfacing of fluid density distributions allow the human eye to detect and track defects in liquid crystals because humans are easily capable of finding imperfections in repetitive spatial structures. However, manual data analysis becomes too time consuming and algorithmic approaches are needed when there are large amounts of data. We present and compare two different approaches we have developed to study defects in gyroid mesophases of amphiphilic ternary fluids. While the first method is based on a pattern recognition algorithm, the second uses the particular structural properties of gyroid mesophases to detect defects.

© 2004 Elsevier B.V. All rights reserved.
PACS: 61.30.Jf; 61.72.Bb; 83.10.Lk
Keywords: Gyroid cubic mesophase; Liquid crystal; Defect analysis

1. Introduction
Molecules in an ordinary liquid will usually have random orientations. There is a certain interesting class of liquids, called liquid crystals, in which this is not the case: the molecules exhibit a tendency to align or order. Liquid crystals have some of the properties of crystals, since they exhibit long-range ordering and strong anisotropy, but retain the ability to flow, unlike ordinary crystals in which the molecules are locked onto well-defined positions on a lattice.

Amphiphiles are one class of molecule which may produce liquid crystalline behavior. These are molecules constructed from two parts, usually a water-loving head and a long tail which is attracted to oil. In a mixture of oil, water, and amphiphile, amphiphile molecules are attracted to the interface between the oil and water to minimize free energy, hence they are often termed “surface active agents”, or surfactants. Their precise behavior is strongly dependent on concentration, so they are termed “lyotropic” liquid crystals.
A random mixture of oil, water, and surfactant molecules will often spontaneously arrange itself into separate regions of oil and water, separated by a layer of surfactant at the interface, forming structures called surfactant mesophases. These structures can be very complex, and their geometry depends strongly upon the relative proportions of different molecules in the mixture, and the details of how they interact. Such structures can also be produced in binary systems, such as water–lipid mixtures, and often occur in biological systems [1–3].

Surfactant mesophases will often form minimal surfaces: for any boundary drawn on the surface, the surface lying inside the boundary will have the minimal possible surface area, and the surface will also have zero mean curvature. Triply periodic minimal surfaces [4] which extend through space with cubic symmetry [5] have been found in many systems, such as lipid–water mixtures [1], diblock copolymers [6], and in many biological systems [7].

The gyroid, also known as the “G surface”, is a particular minimal surface, discovered by Schoen [8]. It is embedded (has no self-intersections), triply periodic (repeats in \(x\), \(y\), and \(z\)-directions), and is the only known such surface with triple junctions (Fig. 1(b)). A numerical study of the gyroid was made by Große-Brauckmann [9]; while an analytical description was found by Gandy and Klinowski [10] a very close approximation of a gyroid is the surface \(\cos x \sin y + \cos y \sin z + \cos z \sin x = 0\). It is possible to transform the gyroid into two other well-known surfaces, the P and D surfaces, through the single parameter known as the Bonnet Angle [11]. The gyroid has symmetry group \(Ia\overline{3}d\); the unit cell consists of 96 copies of a fundamental surface patch, related through the symmetry operations [10] of this space group.
The gyroid surface divides space into two interpenetrating regions, or labyrinths. In the case of gyroids formed from a mixture of oil, water, and surfactant, one labyrinth contains mostly oil, the other mostly water, and the gyroidic boundary surface between the two labyrinths is populated with surfactant, as shown in Fig. 1(a) and (b).

Channels run through the gyroid labyrinths in the (100) and (111) directions; passages emerge perpendicular to any given channel as it is traversed, the direction at which they do so gyrating down the channel, giving rise to the “gyroid” name [9].

The labyrinths are chiral, so that the channels of one labyrinth gyrate in the opposite sense to the channels of the other, as seen in Fig. 1(c). Looking down the (111) direction of a gyroid shows a distinctive “wagon-wheel” pattern (Fig. 1(d)), which has been observed experimentally in transmission electron micrographs of gyroid phases [6].

As a Platonic or mathematical abstraction, the gyroid consists of perfect copies of the unit cell, repeating on a Bravais lattice extending through space. This is not the case for gyroid structures in the real world: various effects may give rise to regions where the structure deviates from a gyroid. Such deviations are called defects.

During the gyroid self-assembly process, several small, separated gyroid-phase regions or domains may start to form, and then grow. Since the domains evolve independently, the lattices describing them may not be identical, and can differ in orientation, position, or unit cell size. The interface between the domains will not be gyroidal: therefore, grain-boundary defects arise between gyroid domains. Inside a domain, there may be dislocations, or line defects, corresponding to the termination of a plane of unit cells; there may also be localized non-gyroid regions, corresponding to defects due to contamination or inhomogeneities in the initial conditions. See Fig. 2.

While equilibrium gyroid mesophases and their defects are observed experimentally, it is desirable to formulate a theoretical or computational model to better understand how and why they form and how they evolve.

Much effort has been invested in theoretical and computational modeling of liquid crystals. Nematic liquid crystals have been modeled using the Leslie–Erickson formulation [12]; Monte Carlo [13,14] and Molecular Dynamics [15] simulations have provided some insight, but reaching regimes where hydrodynamic effects are significant is currently computationally unfeasible with these techniques.

Lyotropic liquid crystals, of interest in this paper, have also been studied extensively [16] through techniques such as free-energy methods [17,18] and through consideration of the interface between the lipid and water phases [19]; again, most treatments have been limited to examination of the equilibrium state and its stability.

There have been recent attempts to take advantage of the lattice Boltzmann method for hydrodynamics [20,21], and modify it to take account of liquid-crystalline behavior. Lattice Boltzmann is a discrete-time and discrete-space algorithm; since only nearest-neighbor interactions take place between vertices on the simulation lattice, it is extremely fast, and extremely scalable on parallel computer hardware [22]. There are several such schemes, some of which [23] are based around the Leslie–Ericksen model for nematic liquid crystals, and others which use a free-energy-based approach for nematic [24–29] and lyotropic [30,31] phases. We used another kind of lattice Boltzmann model, which employs a “bottom-up” approach to model interactions between particles [32,33] including surfactants [34], postulating a form of interparticle potential rather than using a free-energy-based technique. Briefly, the single-particle distribution function \( f_\sigma(\mathbf{x}) \) for species \( \sigma \) with velocity \( \mathbf{c}_i \) is discretized onto lattice points \( \mathbf{x} \), and then evolved according to the lattice BGK equation [21]. Three species, called red, blue, and green, are used, corresponding to oil, water, and
surfactant. Immiscible fluids are modeled using an interparticle interaction force, controlled by a coupling constant $g_{cr}$: this force is calculated from the gradient of the order parameter, or “color field”, defined as the difference $\phi(x) = \rho^f(x) - \rho^b(x)$ between red and blue fluid densities. The mean surfactant director field $d(x)$ is also tracked on the lattice. A point-like surfactant molecule is modeled as being constructed from two different immiscible fluid molecules joined together, and therefore subject to dipole-like interactions with the other fluids, controlled by a coupling constant $g_{ct}$. Finally, interactions between surfactant molecules are controlled by a constant $g_{cs}$. It was recently shown [35,36] through simulations with the LB3D parallel lattice Boltzmann code that certain mixtures of specific composition (specified by the initial densities $f_r$, $f_b$, $f_g$ of oil, water, and surfactant) would spontaneously assemble from a randomized, disordered initial condition, into a gyroid mesophase whose lattice parameter is around 8–9 simulation lattice sites. Similar cubic phases have been observed experimentally with lattice parameter of order 50 nm, in polymer blends [6] and biological systems [7].

On a sufficiently small lattice, the gyroid may evolve to perfectly fill the simulated region, without defects. As the lattice size grows, it becomes more probable that multiple gyroid domains will emerge independently, so that grain boundary defects are more likely to appear and the time required for localized defects to diffuse across the lattice increases making it more likely that defects will persist. Therefore, examination of the defect behavior of surfactant mesophases requires the simulation of very large systems. This was achieved as part of the TeraGyroid project, where systems on lattice sizes of up to 1024$^3$ were simulated by linking together multiple geographically-distributed supercomputing resources to form a computational Grid [37–40].

2. Structure factor analysis of liquid gyroid mesophases

Simulation data from liquid crystal dynamics can be visualized using isosurfacing or volume rendering techniques. The human eye has a remarkable ability to easily distinguish between regions where the crystal structure is well developed and areas where it is not. For quantitative studies of large systems evolving over long intervals of time, computational methods for defect detection and tracking are required. Developing algorithms to detect and track defects is a non-trivial task, however, since defects can occur within and between domains of varying shapes and sizes and over a wide variety of length and time scales.

A standard method to analyze simulation data is the calculation of the three-dimensional structure function

$$S(k, t) = \frac{1}{V} |\phi_k(t)|^2,$$

(1)

where $V$ is the number of cites of the lattice, $\phi_k(t)$ the Fourier transform of the fluctuations of the order parameter $\phi' \equiv \phi - \langle \phi \rangle$, and $k$ is the wave vector [36,41]. $S(k, t)$ can easily be calculated, but only gives general information about the crystal development [35,42,43]. It does not allow one to detect where the defects are located or how many there are, nor does it furnish access to information about the number of differently oriented gyroid domains.

Fig. 3 gives an example of the three-dimensional structure factor calculated for the order parameter $\phi(x)$ at timesteps $t = 10,000, 100,000, and 700,000$. We use periodic boundary conditions, a 128$^3$ lattice and simulate for one million timesteps. This is more than an order of magnitude longer than any other simulation performed before the TeraGyroid [40] project using our lattice-Boltzmann code LB3D and took 300 wall clock hours on 128 CPUs of an IBM SP4 (namely HPCx in Daresbury, UK). We use data from this simulation throughout the present paper to demonstrate the properties of different defect detection and tracking algorithms. The initial condition of the simulation is a random mixture with maximum densities of 0.7 for the case of the immiscible fluids and 0.6 for surfactant. The surfactant–surfactant coupling constant is given by $g_{ss} = -0.0045$ and the coupling between surfactant and the other fluids is determined by $g_{cs} = -0.006$. In order to obtain a visualization that is comparable to experimentally obtained SAXS data (see, for example, [42]), we sum the structure factor in one of the Cartesian directions. The example here has been summed in the $x$-direction and $X_{\text{max}}$ denotes the value of the largest peak normalized by the number of lattice sites in the direction of summation (128 in this case). Results for the $y$- and $z$-directions are similar. At $t = 10,000$, gyroid assembly is just commencing,
which is evident due to the eight peaks of the structure factor which are already clearly distinguishable. $X_{\text{max}}$ is 11.85 in this case and is almost eight times bigger for $t = 100,000$. The peaks of the structure function correspond to a well-developed crystal which consists of differently oriented gyroid domains with defects at the domain boundaries. At $t = 700,000$, $X_{\text{max}}$ reaches 197.00 and most of the previously existing domains have merged into a single one. Only a few defects are left of which two can be spotted visually at the right corner of the volume rendered visualization and the centre of the top surface (denoted by the white arrows).

We are interested here in very well developed liquid crystals with defect domains covering only a minority of the total simulation volume. In order to distinguish between different defects and to study their evolution in time, we need to be able to clearly separate defect domains from regions where we find a perfect gyroid structure. An important property of these systems is that the variations of the structure function in time become very small and the system reaches a state close to equilibrium. Fig. 4 shows the time dependence of the maxima of the structure function in $x$-, $y$- and $z$-directions for up to 700,000 timesteps. To suppress short lived fluctuations within the fluid mixture, i.e. local variations that spontaneously form and disappear after up to a few thousand timesteps, we average every data point over 20,000 timesteps. The maximum value of $S(k, t)$ shows a generally increasing behavior in all three cases, but fluctuates greatly for the first 320,000
timesteps. Then, \(X_{\text{max}}(t)\) and \(Z_{\text{max}}(t)\) show a steep increase indicating that two major gyroid domains are merging into a single one. During this process, defects located at the boundaries between these domains disappear. At \(t = 400,000\), the fluctuations present in all three plots become very small indicating a very clean crystal with only a small number of defects. It is easy to detect this state numerically by defining a maximum allowed variation of the maximum values of the structure function. The remaining part of this paper will only discuss the analysis of data obtained for \(t \geq 340,000\).

Very long simulations like the one presented here can generate large amounts of data—especially if one measures physical quantities with a high resolution. Here, we measured the order parameter every 100 timesteps resulting in 10,000 data files or about 78 GB of data which we need to analyze. Filtering out data that is irrelevant for studying defect behavior using the method described here allows us to reduce the number of files to 6500.

3. Data reduction: From three dimensions to two dimensions

A first order approach to reduce the amount of data which needs to be analyzed in detail is to project the three-dimensional system onto a two-dimensional plane. If one volume renders the order parameter of a gyroid unit cell using a step function which fills out all areas above an appropriate threshold value and leaves all values below that threshold transparent, it is possible to "look through" the unit cell under various angles. Since in a perfect gyroid mesophase the individual unit cells assemble in a very regular way, it is then possible to look through the whole liquid crystal.

This can be implemented as a ray-tracing algorithm: First, select an appropriate projection direction, for example, \(y\). Define a projection plane \(P(x,z)\) to store the results and define a threshold value \(C\). For the gyroid mesophase, we use 66% of the maximum value of the order parameter, but this value depends on the system to be analyzed. For every point in the \(xz\)-plane, start at \(y = 0\) and check for all values of \(y\) if the order parameter \(\phi(x,y,z)\) is smaller than \(C\). As long as that is the case, we keep \(P(x,z) = 0\). If \(\phi(x,y,z)\) is greater or equal \(C\), we set \(P(x,z) = 1\), move to the next point in the \(xz\)-plane and start from \(y = 0\) again.

**Fig. 5** shows visualizations of \(P(x,z)\) for timesteps 50,000, 100,000, 200,000, 300,000, 400,000, 500,000, 600,000, 700,000. Black areas correspond to \(P(x,z) = 1\) and white areas to \(P(x,z) = 0\). For early simulation times most of \(P(x,z)\) is 1 and while the simulation evolves white spots start to occur in the images presented in **Fig. 5** until a very regular lattice-
like structure with some black islands appears for $t > 400,000$. Areas of regular lattice structure correspond to perfect crystal structures along the projection direction. Black islands can be interpreted as areas where the gyroid structure is disturbed or not existent at some point along the projection axis.

Obviously, projecting the full system makes it impossible to retain the three-dimensional structure of individual defects. Therefore, we apply the projection algorithm on slabs of the dataset only. For an optimal resolution of the defect detection, the slab thickness $l$ should be comparable to the size of a gyroid unit cell which corresponds to eight lattice sites in our case, resulting in 16 individual $128 \times 128 \times 8$ slabs for a $128^3$ lattice. We found that using overlapping slabs does not improve the defect detection rate. The positions and sizes of the defects detected in the two-dimensional projections can be used to reconstruct three-dimensional datasets which only include the defective areas. A defective area in the two-dimensional datasets is mapped to a volume of thickness $l$.

In order to further improve the reliability of the detection, we repeat the analysis for all three Cartesian directions. In this way, we can detect gyroid cells which are deformed in one direction only. For the reconstruction of the three-dimensional dataset, all three analysis runs are taken into account. Additional resolution can be obtained by distinguishing between how often and in which direction(s) a defective volume has been detected, indicating the particular kind of defect.

The human eye is easily able to accurately detect defective areas in the individual images of Fig. 5. In the following sections we will present two possible approaches which try to transfer this remarkable ability to a well defined algorithm that can be implemented on a computer. The first approach is based on a generic pattern recognition algorithm and should work with all liquid crystals that form a regular pattern, while the second has been developed with our particular problem in mind and is not known to work with systems other than the gyroid mesophase. However, it is about an order of magnitude faster and the general principles underlying it should be applicable to different systems as well.

4. A general pattern recognition based approach for the detection of defects

The first approach is based on the regularity or periodicity of patterns and was developed by Chetverikov and Hanbury in 2001 [44]. It is assumed that the defect-free pattern is homogeneous and shows some periodicity. The algorithm searches for areas which are significantly less regular (i.e., aperiodic) than the bulk of the dataset by computing regularity features for a set of windows and identifying defects as outliers in regularity feature space. The regularity is quantified by computing the periodicity of the normalized autocorrelation function in polar coordinates. In short, for every window a regularity value is computed. If this value differs by more than a defined threshold value from the median of all window regularity values, the area is accordingly classified as a defect. For a more detailed description of the algorithm see [44,45].

We have found this pattern recognition algorithm to be very robust and reliable in detecting defects. In Fig. 6 the detected defects for an example dataset at $t = 600,000$ are depicted by the white boxes. For good results, the regular pattern needs to occur multiple times within an analysis window. A window size of $17 \times 17$ lattice sites has been found to generate the best results. Defects located in the central area of

![Fig. 6. Two-dimensional projection of the three-dimensional color field for a gyroid system of size $128^3$ at $t = 600,000$. The white frames depict the areas detected by the pattern recognition algorithm of Chetverikov and Hanbury [44]. In order to detect small defects at the system's boundaries, we extend the dataset by mirroring 50% of its corresponding content from the opposite side. The size of the original dataset is depicted by the large box.](image-url)
the crystal are very well detected, but the algorithm fails at the boundaries. We overcome this problem by taking advantage of the fact that our model has periodic boundary conditions, and extend the dataset by mirroring 50% of its corresponding content from the opposite side. In this way we increase the number of analysis windows containing regular patterns and the defects at the boundaries differ more substantially from the surrounding regular pattern since they appear at their full size. The pattern recognition algorithm is then able to detect boundary defects and, depending on the maximum size of defects, the size of the additional “padded” regions can be adapted in order to limit the additional computational costs.

By applying the pattern recognition algorithm to all individual projected slabs of the dataset, we are able to reconstruct a three-dimensional volume that only consists of areas which have been detected as defective. Fig. 7 shows reconstructed datasets at \( t = 340,000, 500,000 \) and 999,000. Even at \( t = 340,000 \) a very large region of the system has not yet formed a well defined gyroid phase. 160,000 timesteps later, the main defects are pillar shaped ones at the centre and at the corners of the visualized systems. Due to the periodic boundary conditions, the corner defects are connected and should be regarded as a single one. As can be seen from the analysis at \( t = 999,000 \), defects in the gyroid mesophase are very stable in size as well as in their position.

5. A mesh generator as an alternative method for defect detection

The second approach presented in this paper also utilizes the two-dimensional slab projections and encapsulates knowledge about the patterns produced by regular and defect regions (see Fig. 8). As a consequence, it is an order of magnitude faster than the pattern recognition code.

For each slab image, the centroid and area of each dot is computed and dots below a certain threshold area are discarded. The four nearest neighbors of each dot are determined and a connection mesh is generated (Fig. 8(b)). Neighbors that lie more than one gyroid unit cell width away from a dot are discounted. Peri-

Fig. 7. Volume rendered visualization of the color field at \( t = 340,000, 500,000, 999,000 \) from the evolving gyroid system. Only the defects are shown as they have been isolated from the full datasets using the pattern recognition algorithm.

Fig. 8. Defect regions (c) are extracted from a mesh representation (b) of the two-dimensional projection (a) for \( t = 600,000 \) shown in Fig. 5.
Periodic boundary conditions are assumed when generating the mesh so that defects at the edge of the system may be reliably detected. Nodes which lie in defect-free regions of the lattice are distinguished by having four four-hop closed routes through neighboring nodes (denoted by the arrows in Fig. 9). Mesh nodes that describe the perimeter of defect regions lack this property. A tree-searching algorithm is used to search the data-structure representing the mesh and detect the presence of closed loops. The regions of regular mesh are discarded, leaving only mesh that describes the perimeters of defect regions (emboldened regions in Fig. 8(b)).

A flood-fill algorithm is applied to a rasterized image of the defect perimeters to locate distinct defect regions and isolate them from the background (corresponding to defect-free regions). Thereafter, a mask image as presented in Fig. 8(c) is generated.

As with the pattern recognition approach, this procedure is repeated for each of the three Cartesian axes and the resultant mask images are assembled to produce a three-dimensional defect mask which is then applied to the original dataset. For comparison, Fig. 10 shows reconstructed visualizations of defective areas for the same parameters and timesteps as Fig. 7.

Since the algorithm defines the perimeter of a defect region by the nearest mesh nodes, there is a tendency to over-estimate the boundary of the defect region. However, this over-estimate is proportional to the defect surface area and the maximum error of the detected defect volume can be estimated as the volume of a one gyroid unit cell thick layer surrounding all individual defects.

---

Fig. 9. Mesh nodes which lie within regular regions (A) are distinguished by having four four-hop closed routes through their neighboring nodes. Nodes lacking this property (black) are categorized as defect boundary nodes.

Fig. 10. Volume rendered visualization of the color field at $t = 340,000$, $500,000$, $999,000$ from the evolving lattice-Boltzmann simulation. Only the defects are shown as they have been separated from the full datasets using the mesh generation algorithm.
Since the dots in the two-dimensional projections are sections of tube-like structure that run through the gyroid rather than spatially localized entities, the division of the dataset into slabs is an essential step. For this reason, no attempt was made to develop a three-dimensional mesh generator.

6. Comparison of the detection algorithms

From the reconstructed datasets we are able to compute the volume fraction of the simulation system that contains defects. This value is plotted in Fig. 11(a). The shades denote the original data, while the solid and dashed lines are averages over 4000 timesteps. As expected, the volume fraction detected by the mesh generator is larger than the area detected by the pattern recognition algorithm because the mesh generation algorithm’s resolution is limited by the size of a unit cell. In addition, the mesh generator detects very small and short-lived variations of the dataset which occur due to small local variations of the gyroid structure resulting in more noisy data for the volume fraction. The results of the pattern recognition algorithm are noisy because the shape of the detected regions is determined by the combination of overlapping squares corresponding to the analysis windows. Small variations of the defect shape can result in differently arranged overlapping windows and thus in varying defect volume fractions. Improvement is possible by using a higher resolution for the pattern recognition analysis, but at greater computational cost. However, both methods show the same general behavior.

Fig. 11(b) shows the averaged volume fraction contained in the large defect in the centre of the system. We compute the total volume of an individual defect by assuming that all detected areas that overlap belong to the same defect. While for early simulation times the mesh generator’s results fluctuate substantially more than the values obtained from the pattern recognition algorithm, they eventually converge for \( t > 600,000 \).

An important feature of both algorithms is the possibility they provide to track individual defects in time and so enable us to study their dynamics on the lattice. As an example, we plot the distance of the centre of mass of the large defect in the centre of the system in Fig. 12. Both methods generate the same general behavior and differences in these plots are caused by the slightly different volumes detected by each methods. The origin corresponds to the lower front corner in Figs. 7 and 10.

Finally, we analyze the number of defects detected by both methods for \( t > 34,000 \). As shown in Table 1, the pattern recognition algorithm has a minimum detection rate of 4 defects which corresponds to a value
III-3. Detection and tracking of defects in the gyroid mesophase


Fig. 12. Distance (in lattice sites) of the centre of the large defect in the centre of the simulated $128^3$ system from the origin. The data has been obtained using the pattern recognition algorithm (solid line) and the mesh generation algorithm (dashed line). The origin corresponds to the left front corner in Figs. 7 and 10.

Table 1
Statistics for the number of defects within the system obtained from the pattern recognition algorithm, the mesh generator and the mesh generator with added time averaging over 4000 timesteps

<table>
<thead>
<tr>
<th>Algorithm</th>
<th>Minimum</th>
<th>Maximum</th>
<th>Mean</th>
<th>σ</th>
</tr>
</thead>
<tbody>
<tr>
<td>Pattern recognition</td>
<td>4</td>
<td>34</td>
<td>20.17</td>
<td>3.48</td>
</tr>
<tr>
<td>Mesh generator</td>
<td>28</td>
<td>78</td>
<td>48.70</td>
<td>7.52</td>
</tr>
<tr>
<td>Mesh generator (avg)</td>
<td>7</td>
<td>36</td>
<td>22.76</td>
<td>3.80</td>
</tr>
</tbody>
</table>

at an early simulation time ($t = 346,200$) and a maximum value of 34 ($t = 578,600$). The mean is 20.17 and the standard deviation $\sigma$ is 3.48. All values are substantially smaller than the results obtained from the mesh generator (see Table 1). This is because of the mesh generator’s ability to detect very small variations of the gyroid. Furthermore, since the pattern recognition’s resolution is limited due to the rectangular shape of the analysis windows, the resulting detection areas are not as flexible in shape as the ones from the mesh generator. Thus, the mesh generator might detect multiple small defects which are very close to each other while the pattern recognition algorithm would detect those as a single defect. We can enhance the comparability of both methods by applying a filter to the mesh generator and only taking defects into account that have a lifetime of at least 1000 timesteps. For our data, the new minimum drops to 7 which is only 3 defects more than obtained using the pattern recognition algorithm. The resulting maximum is 36 and the mean becomes 22.76.

In order to make the advantages of the two-dimensional projections apparent, we have extended the pattern recognition algorithm to three dimensions and applied it to the three-dimensional dataset directly. For the case presented here, the computational effort needed to analyze a single dataset is 2.7 times higher than the analysis of all individual projections ($s = 8$) in all three directions. Furthermore, the three-dimensional approach does not allow a greater resolution than the size of the analysis window which in our case is $17 \times 17 \times 17$. The projection based approach allows us to improve the resolution since the individual two-dimensional windows are mapped to $17 \times 17 \times 8$ volumes. By taking all three Cartesian directions into account, we are able to achieve an effective resolution of $8 \times 8 \times 8$. Furthermore, due to the well defined structures produced by the projection algorithm, the detection rate of the pattern recognition algorithm is higher than if one uses it on a non-processed dataset.

7. Conclusions

We first briefly described the most spectacular liquid crystalline mesophases which may arise in amphiphilic fluids, and our capability to simulate these. The specific liquid crystalline mesophase of interest has been the cubic gyroid phase. We then described two powerful algorithms based on a pattern recognition algorithm and a mesh generation method to detect and track defects in liquid crystals and applied them to simulation data of a gyroid mesophase obtained during the TeraGyroid project [37]. Both algorithms are superior to fully three-dimensional approaches since they exploit basic properties of the system to be analyzed. The two-dimensional projection of slabs which have the thickness of a crystal unit cell allows us to reduce the computational analysis effort substantially. Since the pattern recognition algorithm was not developed with gyroid mesophases in mind, it should be applicable to many different regular structures. Additionally, it is more robust in detecting defective areas than the mesh generation algorithm. However, the latter is about ten times faster and thus saves a substantial amount of CPU time if one has to analyze large amounts of simulation data. In addition, we found that
time averaging is efficient in filtering out short time fluctuations or artefacts.

The methods described in this paper are most powerful if they are applied in a combined fashion and it would be a natural extension to perform parts of it during an ongoing simulation. For checking if a gyroid phase has formed, observing the variation of the maxima of the projected structure function of the order parameter is most efficient. Efficient parallel FFT implementations are widely available and can be implemented within the simulation code. If the variation of the maxima of the projected structure function drops below a threshold value, one should apply the mesh generation algorithm and track values like the total defect volume or the number of defects. Since our simulation code uses spatial domain decomposition, each CPU can generate the two-dimensional projections of the order parameter individually following which the mesh generation algorithm can be applied locally. The computational effort for this analysis is negligible compared to the actual simulation time, and moreover would allow the scientist to use computational steering techniques [38, 46, 47] to monitor the state of the simulation while it is running.

The pattern recognition algorithm is less efficient than mesh generation, but is the only choice if one is not limited to simulations of gyroid mesophases. In the gyroid case, it is more efficient to use the results from the mesh generator to select a smaller number of datasets for post-processing using the pattern recognition algorithm since the computational effort involved in the pattern recognition can be substantial. For very large datasets, a promising approach is to determine regions of interest within a single dataset using the mesh generation algorithm and then to analyze subdomains of the system utilizing the pattern recognition approach.

Articles are currently in preparation which make extensive use of the detection and tracking algorithms described here in order to understand the dynamics and properties of amphiphilic gyroid phases.

Acknowledgements

We would like to thank Nélio González-Segredo for fruitful discussions and Dmitry Chetverikov for providing the two-dimensional version of the “The IPAN Structural Defect Detector”. We would also like to thank him for his support when we extended it to three dimensions.

We are grateful to the U.K. Engineering and Physical Sciences Research Council (EPSRC) for funding much of this research through RealityGrid grant GR/R67699 and to EPSRC and the National Science Foundation (NSF) for funding the TeraGyroid project.

References

III-3. Detection and tracking of defects in the gyroid mesophase

Large-scale lattice Boltzmann simulations of complex fluids: advances through the advent of computational grids

J. Harting, J. Chin, M. Venturoli, and P. V. Coveney
Large-scale lattice Boltzmann simulations of complex fluids: advances through the advent of computational Grids

BY JENS HARTING¹, JONATHAN CHIN², MADALENA VENTUROLI²,³ AND PETER V. COVENY²

¹Institute for Computational Physics, University of Stuttgart, Pfaffenwaldring 27, 70569 Stuttgart, Germany
²Centre for Computational Science, Christopher Ingold Laboratories, University College London, 20 Gordon Street, London WC1H 0AJ, UK (p.v.coveney@ucl.ac.uk)
³Schlumberger Cambridge Research, High Cross, Madingley Road, Cambridge CB3 0EL, UK

During the last 2.5 years, the RealityGrid project has allowed us to be one of the few scientific groups involved in the development of computational Grids. Since smoothly working production Grids are not yet available, we have been able to substantially influence the direction of software and Grid deployment within the project. In this paper, we review our results from large-scale three-dimensional lattice Boltzmann simulations performed over the last 2.5 years. We describe how the proactive use of computational steering, and advanced job migration and visualization techniques enabled us to do our scientific work more efficiently. The projects reported on in this paper are studies of complex fluid flows under shear or in porous media, as well as large-scale parameter searches, and studies of the self-organization of liquid cubic mesophases.

Keywords: lattice Boltzmann; complex fluids; Grid computing; computational steering

1. Introduction

In recent years there has emerged a class of fluid dynamical problems, called ‘complex fluids’, which involve both hydrodynamic flow effects and complex interactions between fluid particles. Computationally, such problems are too large and expensive to tackle with atomistic methods such as molecular dynamics, yet they require too much molecular detail for continuum Navier–Stokes approaches.

Algorithms which work at an intermediate or ‘mesoscale’ level of description in order to solve these problems have been developed in response, including dissipative particle dynamics (Español & Warren 1995; Jury et al. 1999; Flekkøy et al. 2000), lattice gas cellular automata (Rivet & Boon 2001), the stochastic rotation dynamics of Malevanets & Kapral (Malevanets & Kapral 1998; One contribution of 27 to a Theme ‘Scientific Grid computing’. 2005 The Royal Society

1895
Hashimoto et al. 2000; Sakai et al. 2000) and the lattice Boltzmann equation (LBE; Benzi et al. 1992; Succi 2001; Love et al. 2003). In particular, the lattice Boltzmann method has been found highly useful for simulation of complex fluid flows in a wide variety of systems. This algorithm, described in more detail below, is extremely well suited to implementation on parallel computers, which permits very large systems to be simulated, reaching hitherto inaccessible physical regimes. We describe some of these calculations, and also attempt to take parallel computing to a new scale, by coupling several supercomputers together into a computational Grid, which in turn permits easy use of techniques such as computational steering, code migration and real-time visualization.

The term ‘simple fluid’ usually refers to a fluid which can be described to a good degree of approximation by macroscopic quantities only, such as the density field $\rho(x)$, velocity field $v(x)$ and perhaps temperature $T(x)$. Such fluids are governed by the well-known Navier–Stokes equations (Faber 1995), which, being nonlinear, are difficult to solve in the most general case, with the result that numerical solution of the equations has become a common tool for understanding the behaviour of ‘simple’ fluids, such as water or air. Conversely, a ‘complex fluid’ is one whose macroscopic flow is affected by its microscopic properties. A good example of such a fluid is blood: as it flows through vessels (of order millimetres wide and centimetres long), it is subjected to shear forces, which cause red blood cells (of order micrometres wide) to align with the flow, so that they can slide over one another more easily, causing the fluid to become less viscous; this change in viscosity in turn affects the flow profile. Hence, the macroscopic blood flow is affected by the microscopic alignment of its constituent cells. Other examples of complex fluids include biological fluids such as milk, cell organelles and cytoplasm, as well as polymers and liquid crystals. In all of these cases, the density and velocity fields are insufficient to describe the fluid behaviour, and in order to understand this behaviour, it is necessary to treat effects which occur over a very wide range of length and time-scales. This length and time-scale gap makes complex fluids even more difficult to model than simple fluids. While numerical solutions of the macroscopic equations are possible for many simple fluids, such a level of description may not exist for complex fluids, yet simulation of every single molecule involved is computationally infeasible.

In a mixture containing many different fluid components, an amphiphile is a kind of molecule which is composed of two parts, each part being attracted towards a different fluid component. For example, soap molecules are amphiphiles, containing a head group which is attracted towards water, and a tail which is attracted towards oil and grease; analogous molecules can also be formed from polymers. If many amphiphile molecules are collected together in solution, they can exhibit highly varied and complicated behaviour, often assembling to form amphiphile mesophases, which are complex fluids of significant theoretical and industrial importance. Some of these phases have long-range order, yet remain able to flow, and are called liquid crystal mesophases. Of particular interest to us are those with cubic symmetry, whose properties have been studied experimentally (Seddon & Templer 1993; Seddon & Templer 1995; Czeslik & Winter 2002) in lipid–water mixtures (Seddon & Templer 1995), diblock copolymers (Shefelbine et al. 1999), and in many biological systems (Landh 1995).
Over the last decade, significant effort has been invested in understanding complex fluids through computational mesoscale modelling techniques. These techniques do not attempt to keep track of the state of every single constituent element of a system, nor do they use an entirely macroscopic description; instead, an intermediate, mesoscale model of the fluid is developed, coarse-graining microscopic interactions enough that they are rendered amenable to simulation and analysis, but not so much that the important details are lost. Such approaches include lattice gas automata (Frisch et al. 1986; Rothman & Keller 1988; Rivet & Boon 2001; Love 2002), LBE (McNamara & Zanetti 1988; Higuera & Jiménez 1989; Higuera et al. 1989; Benzi et al. 1992; Shan & Chen 1993; Lamura et al. 1999; Chen et al. 2000; Succi 2001; Chin & Coveney 2002), dissipative particle dynamics (Hoogerbrugge & Koelman 1992; Español & Warren 1995; Jury et al. 1999), or the Malevanets–Kapral real-coded lattice gas (Malevanets & Kapral 1998; Malevanets & Yeomans 2000; Sakai et al. 2000). Recently developed techniques (Garcia et al. 1999; Delgado-Buscalioni & Coveney 2003) which use hybrid algorithms have shown much promise.

All simulations described in this paper use the lattice Boltzmann algorithm, which is a powerful method for simulating fluid dynamics. This is due to the ease with which boundary conditions can be imposed, and with which the model may be extended to describe mixtures of interacting complex fluids. Rather than tracking the state of individual atoms and molecules, the method describes the dynamics of the single-particle distribution function of mesoscopic fluid packets.

In a continuum description, the single-particle distribution function $f_1(r, v, t)$ represents the density of fluid particles with position $r$ and velocity $v$ at time $t$, such that the density and velocity of the macroscopically observable fluid are given by $\rho(r, t) = \int f_1(r, v, t) dv$ and $u(r, t) = \int f_1(r, v, t) v dv$, respectively. In the non-interacting, long mean free path limit, with no externally applied forces, the evolution of this function is described by Boltzmann’s equation

$$\left(\partial_t + v \cdot \nabla\right) f_1 = \mathcal{Q}[f_1].$$

(1.1)

While the left-hand side describes changes in the distribution function due to free particle motion, the right-hand side models pairwise collisions. This collision operator $\mathcal{Q}$ is an integral expression that is often simplified (Bhatnagar et al. 1954) to the linear Bhatnagar–Gross–Krook (BGK) form

$$\mathcal{Q}[f] \approx -\frac{1}{\tau} [f - f^{eq}].$$

(1.2)

The BGK collision operator describes the relaxation, at a rate controlled by a characteristic time $\tau$, towards a Maxwell–Boltzmann equilibrium distribution $f^{eq}$. While this is a drastic simplification, it can be shown that distributions governed by the Boltzmann–BGK equation conserve mass, momentum and energy (Succi 2001), and obey a non-equilibrium form of the second law of thermodynamics (Liboff 1990). Moreover, it can be shown (Chapman & Cowling 1952; Liboff 1990) that the well-known Navier–Stokes equations for macroscopic fluid flow are obeyed on coarse length and time-scales (Chapman & Cowling 1952; Liboff 1990). In a lattice Boltzmann formulation, the single-particle distribution function is discretized in time and space. The positions $r$ on which $f_1(r, v, t)$ is defined are restricted to points $r_i$ on a lattice, and the velocities
\( v \) are restricted to a set \( c_i \) joining points on the lattice. The density of particles at lattice site \( r \), travelling with velocity \( c_i \), at time-step \( t \) is given by \( f_i(r, t) = f(r, c_i, t) \), while the fluid’s density and velocity are given by \( \rho(r) = \sum_i f_i(r) \) and \( u(r) = \sum_i f_i(r)c_i \). The discretized description can be evolved in two steps: the collision step, where particles at each lattice site are redistributed across the velocity vectors, and the advection, where values of the post-collisional distribution function are propagated to adjacent lattice sites.

By combining these steps, one obtains the LBE

\[
f_i(r, t + 1) - f_i(r, t) = \Omega[f] = -\frac{1}{\tau} [f_i(r, t) - N_i(\rho, u)],
\]

where \( N_i = N_i(\rho(r), u(r)) \) is a polynomial function of the local density and velocity, which may be found by discretizing the well-known Maxwell–Boltzmann equilibrium distribution. Our implementation uses the Shan–Chen approach (Shan & Chen 1993), by incorporating an explicit forcing term in the collision operator in order to model multicomponent interacting fluids. Shan and Chen extended \( f_i \) to the form \( f_i^\sigma \), where each component is denoted by a different value of the superscript \( \sigma \), so that density and momentum of a component \( \sigma \) are given by \( \rho^\sigma = \sum_i f_i^\sigma \) and \( \rho^\sigma u^\sigma = \sum_i f_i^\sigma c_i \). The fluid viscosity \( \nu^\sigma \) is proportional to \((\tau^\sigma - 1/2)\) and the particle mass is \( m^\sigma \). This results in a lattice BGK equation (1.3) of the form

\[
f_i^\sigma(r, t + 1) - f_i^\sigma(r, t) = -\frac{1}{\tau^\sigma} [f_i^\sigma(r, t) - N_i(\rho^\sigma, u^\sigma)].
\]

The velocity \( u^\sigma \) is found by calculating a weighted average velocity \( u' \) and then adding a term to account for external forces

\[
u^\sigma = u' + \tau^\sigma \psi^\sigma.
\]

In order to produce nearest-neighbour interactions between components, the force term assumes the form

\[
\psi^\sigma = -\frac{\psi^\sigma(x)}{\psi^\sigma(\rho^\sigma(x))} \sum_i g_{\sigma\bar{\sigma}} \sum_j \psi^\bar{\sigma}(x + c_i)c_i,
\]

where \( \psi^\sigma(x) = \psi^\sigma(\rho^\sigma(x)) \) is an effective charge for component \( \sigma \); \( g_{\sigma\bar{\sigma}} \) is a coupling constant controlling the strength of the interaction between two components \( \sigma \) and \( \bar{\sigma} \). If \( g_{\sigma\bar{\sigma}} \) is set to zero for \( \sigma = \bar{\sigma} \), and a positive value for \( \sigma \neq \bar{\sigma} \) then, in the interface between bulk regions of each component, particles experience a force in the direction away from the interface, producing immiscibility. In two-component systems, it is usually the case that \( g_{\bar{\sigma}\sigma} = g_{\sigma\bar{\sigma}} = g_{\sigma\sigma} \). Amongst other things, this model has been used to simulate spinodal decomposition (Chin & Coveney 2002; Gonza`lez-Segredo et al. 2003), polymer blends (Martys & Douglas 2001), liquid–gas phase transitions (Shan & Chen 1994) and flow in porous media (Martys & Chen 1996). Amphiphilic fluids may be treated by introducing a new species of particle with an orientational degree of freedom, which is modelled by a vector dipole moment \( d \) (Chen et al. 2000) with magnitude \( d_0 \). The dipole field \( d(x, t) \) represents the average orientation of any amphiphile present at site \( x \).
During advection, values of \( d(x, t) \) are propagated according to (tildes denote post-collision values)

\[
\rho^a(x, t + 1) d(x, t + 1) = \sum_i \tilde{f}^s_i(x - c_i, t) \tilde{d}(x - c_i, t).
\]

(1.7)

During collision, the dipole moments evolve in a BGK process controlled by a dipole relaxation time \( \tau_d \)

\[
\tilde{d}(x, t) = d(x, t) - \frac{1}{\tau_d} [d(x, t) - d^{(eq)}(x, t)].
\]

(1.8)

The equilibrium dipole moment \( d^{(eq)} = \beta q d h/3 \) is aligned with the colour field \( h \) which contains a component \( h^c \) due to coloured particles, and a part \( h^s \) due to dipoles. With \( q^s \) being a colour charge, such as +1 for red particles, −1 for blue particles and 0 for amphiphile particles, one gets

\[
h^c = \sum \rho (x + c_i) c_i,
\]

(1.9)

\[
h^s(x, t) = \sum_{i \neq 0} \left[ \sum_j f^s_i(x + c_i, t) \theta_j \cdot d_i(x + c_j, t) + f^s_i(x, t) d_i(x, t) \right].
\]

(1.10)

The second-rank tensor \( \theta_j \) is defined in terms of the unit tensor \( I \) and lattice vector \( c_j \) as \( \theta_j = I - D c_j c_j / c^2 \). In the presence of an amphiphilic species, the force on a coloured particle includes an additional term \( F^{s,s} \) to account for the colour field due to the amphiphiles. By treating an amphiphilic particle as a pair of oil and water particles with a very small separation \( d_i \), introducing a constant \( g_{s,s} \) to control the strength of the interaction between amphiphiles and non-amphiphiles and Taylor-expanding in \( d_i \), it can be shown that this term is given by

\[
F^{s,s}(x, t) = -2 \psi^s(x, t) g_{s,s} \sum_{i \neq 0} \tilde{d}(x + c_i, t) \cdot \theta_i \psi^c(x + c_i, t).
\]

(1.11)

While amphiphiles do not possess a net colour charge, they also experience a force due to the colour field, consisting of a part \( F^{s,c} \) due to ordinary species, and a part \( F^{s,s} \) due to other amphiphiles

\[
F^{s,c} = 2 \psi^c(x, t) \tilde{d}(x, t) \cdot \sum_{\sigma} g_{s,s} \sum_{i \neq 0} \theta_i \psi^s(x, c_i, t).
\]

(1.12)

\[
F^{s,s} = -\frac{4D}{c^2} g_{s,s} \psi^s(x) \sum_i \{ \tilde{d}(x + c_i) \cdot \theta_i \cdot \tilde{d}(x) c_i + [\tilde{d}(x + c_i) \tilde{d}(x) + \tilde{d}(x) \tilde{d}(x + c_i)] \cdot c_i \} \psi^c(x + c_i).
\]

(1.13)

While the form of the interactions seems straightforward at a mesoscopic level, it is essentially phenomenological, and it is not necessarily easy to relate the interaction scheme or its coupling constants to either microscopic molecular characteristics, or to macroscopic phase behaviour. The phase behaviour can be very difficult to predict beforehand from the simulation parameters, and brute-force parameter searches are often resorted to (Boghosian et al. 2000).
2. Technical projects

Our three-dimensional lattice Boltzmann code, LB3D, is written in FORTRAN 90 and designed to run on distributed-memory parallel computers, using the Message Passing Interface (MPI) standard for communication. In each simulation, the fluid is discretized onto a cuboidal lattice, each lattice point containing information about the fluid in the corresponding region of space. Each lattice site requires about a kilobyte of memory per lattice site so that, for example, a simulation on a $128^3$ lattice would require around 2.2 GB memory. The high-performance computing machines on which most of the simulation work is performed are typically rather heavily used. The situation frequently arises that while a simulation is running on one machine, CPU time becomes available on another machine which may be able to run the job faster or cheaper. The LB3D program has the ability to ‘checkpoint’ its entire state to a file. This file can then be moved to another machine, and the simulation restarted there, even if the new machine has a different number of CPUs or even a completely different architecture. It has been verified that the simulation results are independent of the machine on which the calculation runs, so that a single simulation may be migrated between different machines as necessary without affecting its output. As a conservative rule of thumb, the code runs at over $10^4$ lattice site updates per second per CPU on a fairly recent machine, and has been observed to have roughly linear scaling up to order $10^5$ compute nodes. A $128^3$ simulation contains around $2.1 \times 10^6$ lattice sites; running it for 1000 time-steps requires about an hour of real time, split across 64 CPUs. The largest simulation we performed used a $1024^3$ lattice. The output from a simulation usually takes the form of a single floating-point number for each lattice site, representing, for example, the density of a particular fluid component at that site. Therefore, a density field snapshot from a $128^3$ system would produce output files of around 8 MB. Writing data to disk is one of the bottlenecks in large-scale simulations. If one simulates a $1024^3$ system, each data file is 4 GB in size. LB3D is able to benefit from the parallel filesystems available on many large machines today, by using the MPI-IO based parallel HDF5 data format (HDF5 2003). Our code is very robust regarding different platforms or cluster interconnects: even with moderate inter-node bandwidths it achieves almost linear scaling for large processor counts with the only limitation being the available memory per node. The platforms our code has been successfully used on include various supercomputers like the IBM pSeries, SGI Altix and Origin, Cray T3E, Compaq Alpha clusters, NEC SX6, as well as low cost 32- and 64-bit Linux clusters. However, due to compiler or machine peculiarities it is a time consuming task to achieve optimum performance on many different platforms. Porting a complex FORTRAN code like LB3D to new platforms is often very difficult and time-consuming without the assistance of well-trained staff at the corresponding computer centres. Some of these problems are due to portability issues with the FORTRAN language. Also, tuning a code to take full advantage of the machine on which it runs requires considerable knowledge of the local system’s quirks. It is hoped that some of the portability issues could be solved in future by well-designed middleware. Such issues include the fact that location, size, and duration of temporary filesystem change from machine to machine, as do the methods for invoking compilers and batch queues.
LB3D has successfully been used to study various problems like spinodal decomposition with and without shear (González-Segredo et al. 2003; Harting et al. 2004), flow in porous media (Harting et al. 2004), the self-assembly of cubic mesophases such as the ‘P’-phase (Nekovee & Coveney 2001) in binary water-surfactant systems, or the cubic gyroid phase in ternary amphiphilic systems (González-Segredo & Coveney 2004a, b). Before we were able to take advantage of computational steering techniques, our work usually involved large-scale parameter searches organized as taskfarming jobs, in order to find the areas of interest of the available parameter space. The technique of computational steering (Brooke et al. 2003; Chin et al. 2003; Love et al. 2003) has been used successfully in smaller-scale simulations to optimize resource usage. Typically, the procedure for running a simulation of the self-assembly of a mesophase would be to set up the initial conditions, and then submit a batch job to run for a certain, fixed number of time-steps. If the time-scale for structural assembly is unknown then the initial number of time-steps for which the simulation runs is, at best, an educated guess. It is not uncommon to examine the results of such a simulation once they return from the batch queue, only to find that a simulation has not been run for sufficient time (in which case it must be tediously resubmitted), or that it ran for too long, and the majority of the computer time was wasted on simulation of an uninteresting equilibrium system showing no dynamical behaviour. Another unfortunate scenario often occurs when the phase diagram of a simulated system is not well known, in which case a simulation may evolve away from a situation of interest, wasting further CPU time. Computational steering, the ability to watch and control a calculation as it runs, can be used to avoid these difficulties: a simulation which has equilibrated may be spotted and terminated, preventing wastage of CPU time. More powerfully, a simulation may be steered through parameter space until it is unambiguously seen to be producing interesting results: this technique is very powerful when searching for emergent phenomena, such as the formation of surfactant micelles, which are not clearly related to the underlying simulation parameters. Steering is performed using the RealityGrid steering library which has been developed by collaborators at the University of Manchester. The library was built with the intention of making it possible to add steering capabilities to existing simulation codes with as few changes as possible, and in as general a manner as possible. Once the application has initialized the steering library and informed it which parameters are to be steered, then after every timestep of the simulation, it is possible to perform tasks such as checkpointing the simulation, saving output data, stopping the simulation, or restarting from an existing checkpoint. When a steered simulation is started, a Steering Grid Service (SGS) is also created, to represent the steerable simulation on the Grid. The SGS publishes its location to a registry service, so that steering clients may find it. This design means that it is possible for clients to dynamically attach to and detach from running simulations.

Successful computational steering requires that the simulation operators have a good understanding of what the simulation is doing, in real time: this in turn requires good visualization capabilities. Each running simulation emits output files after certain periods of simulation time have elapsed. The period between output emission is initially determined by guessing a time-scale over which the simulation will change in a substantial way; however, this period is a steerable
parameter, so that the output rate can be adjusted for optimum visualization without producing an excessive amount of data. The LB3D code itself will only emit volumetric datasets as described above; these must then be rendered into a human-comprehensible form through techniques, including volume-rendering, isosurfacing, ray-tracing, slice planes and Fourier transforms. The process of producing such comprehensible data from the raw datasets is itself computationally intensive, particularly if it is to be performed in real time, as required for computational steering. For this reason, we use separate visualization clusters to render the data. Output volumes are sent from the simulation machine to the remote visualization machine, so that the simulation can proceed independently of the visualization; these are then rendered using the open source Visualization Toolkit (VTK) (Schroeder et al. 2003) visualization library into bitmap images, which can in turn be multicast over the AccessGrid, so that the state of the simulation can be viewed by scientists around the globe. In particular, this was demonstrated by performing and interacting with a simulation in front of a live worldwide audience, as part of the SCGlobal track of the SuperComputing 2004 conference. The RealityGrid steering architecture was designed in a sufficiently general manner that visualization services can also be represented by SGS: in order to establish a connection between the visualization process and the corresponding simulation, the simulation SGS can be found through the Registry, and then interrogated for the information required to open the link.

In order to be able to deploy the above described components as part of a usable simulation Grid, a substantial amount of coordination is necessary, so that the end user is able to launch an entire simulation pipeline, containing migratable simulation, visualization, and steering components, from a unified interface. This requires a system for keeping track of which services are available, which components are running, taking care of the checkpoints and data which are generated, and to harmonize communication between the different components. This was achieved through the development of a Registry service, implemented using the OGSI::Lite (McKeown 2003) toolkit. The RealityGrid steering library (Chin et al. 2003) communicates with the rest of the Grid by exposing itself as a ‘Grid Service’. Through the Registry service, steering clients are able to find, dynamically attach to, communicate with, and detach from steering services to control a simulation or visualization process.

Large lattices require a highly scalable code, access to high performance computing, terascale storage facilities and high performance visualization. LB3D provides the first of these, while the others are being delivered by the major computing centres. We expect to be able to run our simulations in an even more efficient way due to the significant worldwide effort being invested in the development of reliable computational Grids. These are a collection of geographically distributed and dynamically varying resources, each providing services such as compute cycles, visualization, storage, or even experimental facilities. The major difference between computational Grids and traditional distributed computing is the transparent sharing and collective use of resources, which would otherwise be individual and isolated facilities. Perhaps at some point computational Grids will offer information technology what electricity Grids offer for other aspects of our daily life: a transparent and reliable resource that is easy to use and conforms to commonly agreed standards (Foster & Kesselman 1999; Berman et al. 2003). Robust and smart middleware will find the
best available resources in a transparent way without the user having to care about their location. Unfortunately, reliable and robust computational Grids are not available yet. We used various different demonstration Grids which were assembled especially for a given event or were intended for use as prototyping platforms rather than usable production Grids. These mainly included Grids coupling major compute resources in the UK and the biggest effort took place within the TeraGyroid project (Blake et al. 2004; Blake et al. 2005) where the main machines of the UK’s national High Performance Computing (HPC) centres were coupled with the TeraGrid facilities in the US through a custom high-performance network. In total, about 5000 CPUs were part of this Grid. Collaborative steering sessions with active participants on two continents and observers worldwide were made possible through this approach.

3. Scientific projects

(a) Complex fluids under shear

In many industrial applications, complex fluids are subject to shear forces. For example, axial bearings are often filled with fluid to reduce friction and transport heat away from the most vulnerable parts of the device. It is very important to understand how these fluids behave under high shear forces, in order to be able to build reliable machines and choose the proper fluid for different applications. In our simulations we use Lees-Edwards boundary conditions, which were originally developed for molecular dynamics simulations in 1972 (Lees & Edwards 1972) and have been used in lattice Boltzmann simulations by different authors before (Wagner & Yeomans 1999; Wagner & Pagonabarraga 2002; Harting et al. 2004). We applied our model to study the behaviour of binary immiscible and ternary amphiphilic fluids under constant and oscillatory shear. In the case of spinodal decomposition under constant shear, the first results have been published in Harting et al. (2004). The phase separation of binary immiscible fluids without shear has been studied in detail by different authors, and LB3D has been shown to model the underlying physics successfully (González-Segredo et al. 2003). In the non-sheared studies of spinodal decomposition, it has been shown that lattice sizes need to be large in order to overcome finite size effects: $128^3$ was the minimum acceptable number of lattice sites (González-Segredo et al. 2003). For high shear rates, systems also have to be very long because, if the system is too small, the domains interconnect across the $z=0$ and $z=nz$ boundaries to form interconnected lamellae in the direction of the shear. Such artefacts need to be eliminated from our simulations. Figure 1 shows an example from a simulation with lattice size $128 \times 128 \times 512$. The volume rendered blue and red areas depict the different fluid species and the arrows denote the direction of shear. In the case of ternary amphiphilic fluid mixtures under shear we are interested in the influence of the presence of surfactant molecules on the phase separation. We also study the stress response and stability of cubic mesophases such as the gyroid phase (González-Segredo & Coveney 2004b) or the P-phase (Nekovee & Coveney 2001) under shear. Such complex fluids are expected to exhibit non-Newtonian properties (see below). Computational steering has turned out to be very useful for checking on finite size effects during a sheared fluid simulation, since the human eye is extremely...
good at spotting the sort of structures indicative of such effects. Implementing an algorithm to automatically recognize ‘unphysical’ behaviour is a highly non-trivial task in comparison.

(b) Flow in porous media

Studying transport phenomena in porous media is of great interest in fields ranging from oil recovery and water purification to industrial processes like catalysis. In particular, the oilfield industry uses complex, non-Newtonian, multicomponent fluids (containing polymers, surfactants and/or colloids, brine, oil and/or gas), for processes like fracturing, well stimulation and enhanced oil recovery. The rheology and flow behaviour of these complex fluids in a rock is different from their bulk properties. It is therefore of considerable interest to be able to characterize and predict the flow of these fluids in porous media. From the point of view of a modelling approach, the treatment of complex fluids in three-dimensional complex geometries is an ambitious goal since the lattice has to be large enough to resolve individual structures. The advantage of lattice Boltzmann (or lattice gas) techniques is that complex geometries can be modelled with ease. Synchrotron based X-ray microtomography (XMT) imaging techniques provide high-resolution, three-dimensional digitized images of rock samples. By using the lattice Boltzmann approach in combination with these high-resolution images of rocks, not only is it possible to compute macroscopic transport coefficients, such as the permeability of the medium, but also information on local fields, such as velocity or fluid densities, can be obtained at the pore scale, providing a detailed insight into local flow characterization and supporting the interpretation of experimental measurements (Auzerais et al. 1996). The XMT technique measures the linear attenuation coefficient from which the mineral concentration and composition of the rock can be computed. Morphological properties of the void space, such as pore size distribution and tortuosity, can be derived from the tomographic image of the rock volume, and the permeability and conductivity of the rock can be computed (Spanne et al. 1994). The tomographic data are represented by a reflectivity greyscale value, where the linear size of each voxel is defined by the imaging resolution, which is
Advances through the advent of computational grids usually on the order of micrometres. By introducing a threshold to discriminate between pore and rock sites, these images can be reduced to a binary (0’s and 1’s) representation of the rock geometry. Utilising the lattice Boltzmann method, single phase or multiphase flow can then be described in these real porous media.

Lattice Boltzmann and lattice gas techniques have already been applied to study single and multiphase flow through three-dimensional microtomographic reconstruction of porous media. For example, Martys & Chen (1996) and Ferréol

*Large-scale lattice Boltzmann simulations* 1905

Figure 2. Rendering of 4.9 μm resolution X-ray microtomographic data of a 512³ sample of Bentheimer sandstone. The pore space is shown in red, while the rock is represented in blue.

Figure 3. An originally fully fluid saturated rock is being invaded by another immiscible fluid using a body force $g_{accn}=0.003$. The oil slowly pushes the other fluid component out of the rock pores until the rock is fully saturated by oil at $t=30000$. For better visability only the invading fluid is shown. (Movie available in online version.)

usually on the order of micrometres. By introducing a threshold to discriminate between pore and rock sites, these images can be reduced to a binary (0’s and 1’s) representation of the rock geometry. Utilising the lattice Boltzmann method, single phase or multiphase flow can then be described in these real porous media.

Lattice Boltzmann and lattice gas techniques have already been applied to study single and multiphase flow through three-dimensional microtomographic reconstruction of porous media. For example, Martys & Chen (1996) and Ferréol
& Rothman (1995) studied relative permeabilities of binary mixtures in Fontainebleau sandstone. These studies validated the model and the simulation techniques, but were limited to small lattice sizes, of the order of 64³. Simulating fluid flow in real rock samples allows us to compare simulation data with experimental results obtained on the same, or similar, pieces of rock. For a reasonable comparison, the size of the rock used in lattice Boltzmann simulations should be of the same order of magnitude as the system used in the experiments, or at least large enough to capture the rock’s topological features. The more inhomogeneous the rock, the larger the sample size needs to be in order to describe the correct pore distribution and connectivity. Another reason for needing to use large lattice sizes is the influence of boundary conditions and lattice resolution on the accuracy of the lattice Boltzmann method. It has been shown (see for example He et al. 1997; Chen & Doolen 1998 and references therein) that the BGK (Bhatnagar et al. 1954) approximation of the LBE which is commonly used causes so-called bounce-back boundaries to become inaccurate, resulting in effects such as the computed permeability being a function of the viscosity. This effect can be limited by lowering the viscosity and increasing the lattice resolution. To accurately describe hydrodynamic behaviour using lattice Boltzmann simulations, the Knudsen number, which represents the ratio of the mean free path of the fluid particles and the characteristic length-scale of the system (such as the pore diameter), has to be small. If the pores are resolved with an insufficient number of lattice points, finite size effects arise, leading to an inaccurate description of the flow field. In practice, at least 5–10 lattice sites are needed to resolve a single pore. Therefore, in order to be able to simulate realistic sample sizes, we need large lattices of the order of 512³.

Using LB3D, we are able to simulate drainage and imbibition processes in a 512³ subsample of Bentheimer sandstone X-ray tomographic data. The whole set of XMT data represented the image of a Bentheimer sample of cylindrical shape with diameter 4 mm and length 3 mm. The XMT data were obtained at the European Synchrotron Research Facility (Grenoble) at a resolution of 4.9 μm, resulting in a data set of approximately 816 × 816 × 612 voxels. Figure 2 shows a snapshot of the 512³ subsystem. We compare simulated velocity distributions with experimentally obtained magnetic resonance imaging (MRI) data of oil and brine infiltration into saturated Bentheimer rock core (Sheppard et al. 2003). The rock sample used in these MRI experiments had a diameter of 38 mm and was 70 mm long and was imaged with a resolution of 280 μm. The system simulated was smaller, but still of a similar order of magnitude and large enough to represent the rock geometry. On the other hand, the higher space resolution provided by the simulations allows a detailed characterization of the flow field in the pore space, hence providing a useful tool to interpret the MRI experiments, for example in identifying regions of stagnant fluid. Figure 3 shows an example from a binary invasion study. A rock which is initially fully saturated with ‘water’ (blue), is being invaded by ‘oil’ (red) from the right side. The lattice size is 512³ and the forcing level is set to $g_{\text{accn}} = 0.003$. In figure 3, only the invading fluid component is shown, i.e. only areas where oil is the majority component are rendered. Periodic boundary conditions are applied, and fluid leaving the system on the left side is converted to oil before re-entering on the opposite side. After 5000 time-steps, the oil has invaded about one quarter of the system already and after 25 000 time-steps only small regions of the rock pore space are still filled with water. After 30 000 time-steps, the water
component has been fully pushed out of the rock. This example only covers binary (oil/water) mixtures of Newtonian fluids, since this is a first and necessary step in the understanding of multiphase fluid flow in porous media (Harting et al. 2004). However, we are able to study the flow of binary immiscible fluids with an additional amphiphilic component in porous media and expect results to be presented elsewhere in the near future.

(c) The cubic gyroid mesophase

It was recently shown by González-Segredo & Coveney (2004b) that the dynamical self-assembly of a particular amphiphile mesophase, the gyroid, can be modelled using the lattice Boltzmann method. This mesophase was observed to form from a homogeneous mixture, without any external constraints imposed to bring about the gyroid geometry, which is an emergent effect of the mesoscopic fluid parameters. It is important to note that this method allows examination of the dynamics of mesophase formation, since most treatments to date have focussed on properties or mathematical description (Seddon & Templer 1993; Große & Brauckmann 1997; Schwarz & Gompper 1999; Gandy & Klinowski 2000) of the static equilibrium state. In addition to its biological importance, there have been recent attempts (Chan et al. 1999) to use self-assembling gyroids to construct nanoporous materials. During the gyroid self-assembly process, several small, separated gyroid-phase regions or domains may start to form, and then grow. Since the domains evolve independently, the independent gyroid regions will in general not be identical, and can differ in orientation, position, or unit cell size; grain-boundary defects arise between gyroid domains. Inside a domain, there may be dislocations, or line defects, corresponding to the termination of a plane of unit cells; there may also be localized non-gyroid regions, corresponding to defects due to contamination or inhomogeneities in the initial conditions. Understanding such defects is, therefore, important for our knowledge of the dynamics of surfactant systems, and crucial for an understanding of how best to produce mesophases experimentally and industrially. In small-scale simulations of the gyroid, the mesophase will evolve to fill the simulated region perfectly, without defects. As the lattice size grows, it becomes more probable that multiple gyroid domains will emerge independently, so that grain-boundary defects are more likely to appear, and the time required for localized defects to diffuse across the lattice increases, making it more likely that defects will persist. Therefore, examination of the defect behaviour of surfactant mesophases requires the simulation of very large systems. Figure 4 shows an example of a 128^3 system after 100 000 simulation time-steps. Multiple gyroid domains have formed and the close-up shows the extremely regular, crystalline, gyroid structure within a domain. Figure 5 demonstrates some of the most interesting properties of the gyroid mesophase: two labyrinths mainly consisting of water and oil counterparts are enclosed by the gyroid minimal surface at which the surfactant molecules accumulate. The characteristic triple junctions can be seen clearly.

The TeraGyroid experiment (Blake et al. 2005) addressed a large-scale problem of genuine scientific interest and showed how intercontinental Grids permit the use of novel techniques in collaborative computational science, which can dramatically reduce the time to insight. TeraGyroid used computational steering over a Grid to study the self-assembly and dynamics of gyroid
mesophases using the largest set of lattice Boltzmann simulations ever performed. Around the Supercomputing 2003 conference, we were able to simulate gyroid formation and defect behaviour harnessing the compute power of a large fraction of the UK and US HPC facilities. Altogether we were able to use about 400 000 CPU hours and generate 2 TB of simulation data. In order to make sure our simulations are virtually free of finite size effects, we simulated different system sizes from $64^3$ to $1024^3$, usually for about 100 000 time-steps. In order to study the long-term behaviour of the gyroid mesophase, some simulations have even run for one million time-steps. For 100 000 time-steps we found that $256^3$ or even $128^3$ simulations do not suffer from finite size effects, but after very long simulation times we might even have to move to larger

Figure 4. A volume rendered dataset of a $128^3$ system after 100 000 simulation time-steps. Various gyroid domains have formed and the close-up shows the extremely regular, crystalline, gyroid structure within a domain. (Movie available in online version.)

Figure 5. Structure of the two labyrinths enclosed by a gyroid minimal surface, showing the characteristic triple junctions.
Even with the longest possible simulation times, we were not able to generate a ‘perfect’ crystal. Instead, either differently orientated domains can still be found or individual defects are still moving around. It is of particular interest to study the exact behaviour of the defect movement, which can be done by gathering statistics of the simulation data by counting and tracking individual defects. Gathering useful statistics implies large numbers of measurements and therefore large lattices, which is the reason for the $512^3$ and $1024^3$ simulations performed. The memory requirements exceed the available resources on most supercomputers and limits us to a small number of machines. Also, it requires substantial amounts of CPU time to reach sufficient simulation times. In the case of the $1024^3$ system, 2048 CPUs of a recent Compaq Alpha cluster are only able to simulate about 100 simulation time-steps per hour. Running for 100 000 time-steps would require more than two million CPU hours or 42 days and is therefore unfeasible. Also, handling the data files which are 4 GB each and checkpoint files which are 0.5 TB each is very awkward with the infrastructure available today. In order to be able to gain useful data from the large simulations, we first run a $128^3$ system with periodic boundary conditions, until it forms a gyroid. This

Figure 6. In order to eliminate finite size effects from simulations, very large lattice sizes are needed. If one is interested in the statistical behaviour of defects, then the lattice size has to be increased even more, since otherwise only a limited number of defects can be found in the system. This figure shows a snapshot from what we believe to be the largest ternary lattice Boltzmann simulation ever performed, on a $1024^3$ lattice.

Figure 7. A sheared gyroid mesophase: (a) before the onset of shear, (b) at the onset of shear and (c) after long shear times.
system is then duplicated 512 times to produce a $1024^3$ gyroid system. In order to reduce effects introduced due to the periodic upscaling, we perturb the system and let it evolve. We anticipate that the unphysical effects introduced by the upscaling process will decay after a comparably small number of time-steps, thus resulting in a system that is comparable to one that started from a random mixture of fluids. This has to be justified by comparison with data obtained from test runs performed on smaller systems. Figure 6 shows a snapshot of a volume rendered dataset from the upscaled $1024^3$ system at 1000 time-steps after the upscaling process. The unphysical periodic structures introduced by the individual $128^3$ systems can still clearly be seen.

Currently, work is in progress to study the stability of the gyroid mesophase. We are interested in the influence of perturbation on a gyroid and the strength of the perturbation needed to break up a well-developed mesophase. Similar studies are performed experimentally by applying constant or oscillatory shear. Here, we study the dependence of the gyroid stability on the shear rate, and expect to find evidence of the non-Newtonian properties of the fluid. An example from those studies can be seen in figure 7, which shows three snapshots of the same simulation. The first shows the liquid crystal before the onset of shear, the second only a few hundred time-steps after shear has been turned on and the third image demonstrates how the gyroid melts if the shear stress becomes too strong.

As seen before, simulation data from liquid crystal dynamics can be visualized using isosurfacing or volume rendering techniques. The human eye has a remarkable ability to easily distinguish between regions where the crystal structure is well developed and areas where it is not. However, manual analysis of large amounts of simulation data is not feasible. In the case of the TeraGyroid project, about 2 TB of data would have to be checked and catalogued manually. This task would keep an individual busy for years. Therefore, computational methods for defect detection and tracking are required. Developing algorithms to detect and track defects is a non-trivial task, however, since defects can occur within and between domains of varying shapes and sizes and over a wide variety of length and time-scales. A standard method to analyse simulation data is the calculation of the three-dimensional structure function

$$S(k, t) = (1/V)\phi_k(t)^2,$$

where $V$ is the number of cites of the lattice, $\phi_k(t)$ the Fourier transform of the fluctuations of the order parameter $\phi$ itself, and $k$ is the wave vector (González-Segredo et al. 2003; González-Segredo & Coveney 2004a). $S(k, t)$ can easily be calculated, but only gives general information about the crystal development (Hajduk et al. 1994; Laurer et al. 1997; González-Segredo & Coveney 2004b). It does not allow one to detect where the defects are located or how many there are, nor does it furnish access to information about the number of differently oriented gyroid domains. $S(k, t)$ is given for a $128^3$ system at time-steps $t=10,000, 100,000$ and $700,000$ in figure 8. We simulate for one million time-steps—more than an order of magnitude longer than any other LB3D simulation performed before the TeraGyroid (TeraGyroid 2003) project. The initial condition of the simulation is a random mixture with maximum densities of 0.7 for the immiscible fluids and 0.6 for surfactant. The coupling constant $g_{ss}$ is set to $-0.0045$ and the coupling between surfactant and the other fluids is set to $g_{sc} = -0.006$. In order to compare our data to experimentally obtained SAXS data (Hajduk et al. 1994), we sum the structure factor in the $x$-direction; $X_{\max}$ denotes the value of the largest peak normalized by the number of lattice sites in the direction of summation (128 in this
case; Harting et al. in press). Gyroid assembly is evident due to the eight peaks of the structure factor which become higher with ongoing simulation time. At $t=700\,000$, $X_{\text{max}}$ reaches 197.00 and most of the previously existing domains have merged into a single one. Only a few defects are left of which two can be spotted visually at the right corner of the volume rendered visualization and the centre of the top surface (denoted by the white arrows). The structure factor analysis does not provide any information about the size, position or number of individual defects in the system. Therefore, we developed more advanced algorithms for the detection and tracking of defects. As a first order approach, the data to be analysed can be reduced by cutting the three-dimensional data sets into slabs and projecting them onto a two-dimensional plane. By using a raytracing algorithm for the projection, we obtain regular patterns in areas where the gyroid is perfectly developed and solid planes in defective areas. We developed two algorithms which use the projection data to separate the defective areas from the perfect crystal. The first approach is based on a generic pattern recognition algorithm and should work with all liquid crystals that form a regular pattern, while the second has been developed with our particular problem in mind and is not known to work with systems other than the gyroid mesophase. However, it is about an order of magnitude faster and the general principles underlying it should be applicable to different systems as well. The first approach is based on the regularity or periodicity of patterns and was developed by Chetverikov and Hanbury in 2001 (Chetverikov & Hanbury 2002) who applied it to patterns from the textile industry. It is assumed that defect-free patterns are homogeneous and show some periodicity. The algorithm searches for areas which are significantly less regular (i.e. aperiodic) than the bulk of the dataset by computing regularity features for a set of windows and identifying defects as outliers. The regularity is quantified by computing the periodicity of the normalized autocorrelation function in polar coordinates. In short, for every window a regularity value is computed. If this value differs by more than a defined threshold value from the median of all window regularity values, the area is accordingly classified as a defect. For a more detailed description of the algorithm see (Chetverikov 2000; Chetverikov & Hanbury 2002; Harting et al. in press). The second approach encapsulates knowledge about the patterns produced by regular and defect regions. As a consequence, it is an order of magnitude faster than the pattern recognition code. For each slab image, the algorithm creates a regular mesh in areas where the gyroid structure is well developed, and an irregular mesh in defective areas. The regions of regular mesh are discarded, leaving only mesh that describes the perimeters of defect regions. A flood-fill algorithm is applied to these datasets to locate distinct defect regions. The output data of both detection algorithms for all two-dimensional projections of a three-dimensional dataset can be used to reconstruct three-dimensional volume data that only consists of defect regions. Figure 9 shows reconstructed datasets at $t=340\,000$, 500 000 and 999 000 which have been detected using the pattern recognition approach. However, the results obtained from the mesh generator are similar. Even at $t=340\,000$ a very large region of the system has not yet formed a well-defined gyroid phase. 160 000 time-steps later, the main defects are pillar shaped ones at the centre and at the corners of the visualized systems. Due to the periodic boundary conditions, the corner defects are connected and should be regarded as a single one. As can be seen from the analysis at $t=999\,000$, defects in the gyroid mesophase are very stable in size as well as in their position.
The pattern recognition algorithm is less efficient than mesh generation. However, it is not limited to simulations of gyroid mesophases and more robust with regard to small fluctuations of the dataset. In the gyroid case, it is more efficient to use the results from the mesh generator to select a smaller number of datasets for

Figure 8. Three-dimensional structure factor of the order parameter at time-steps \(t=10000\), \(100000\) and \(700000\), lattice size \(128^3\) and simulation parameters as given in the text. For comparability with SAXS experimental data, we display the total structure factor in the \(x\)-direction \(X = \sum_{k_x} S(k_x, t)\). \(X_{\text{max}}\) denotes the value of the largest peak divided by the number of lattice sites in the direction of summation (128 in this case). The lower half of the figure shows volume rendered visualizations of the corresponding order parameters and the white arrows are a guide for the eye to spot some defective areas at the top surface and the right corner at \(t=700000\).

Figure 9. Volume rendered visualization of the order parameter at \(t=340000\), \(500000\) and \(999000\). Only the defects are shown as they have been isolated from the full datasets using the pattern recognition algorithm. (Movie available in online version.)

The pattern recognition algorithm is less efficient than mesh generation. However, it is not limited to simulations of gyroid mesophases and more robust with regard to small fluctuations of the dataset. In the gyroid case, it is more efficient to use the results from the mesh generator to select a smaller number of datasets for
post-processing using the pattern recognition algorithm since the computational effort involved in the pattern recognition can be substantial. For a more detailed description of the algorithms see (Harting et al. in press). Currently, we are working on more geometrically based algorithms to efficiently detect defects and results will be published elsewhere in the near future.

4. Conclusions

During the last 2 years, we have worked on various scientific projects using our lattice Boltzmann code LB3D. All of these projects reached the limits of the HPC resources available to us today. However, without the benefits obtained from software development within the RealityGrid project, none of these projects would have been possible at all. These improvements include the steering facilities, code optimizations, IO optimizations as well as the platform independent checkpointing and migration routines which have been contributed by various people within the project. Without the lightweight Grid Service Container OGSI::Lite (McKeown 2003) projects like the TeraGyroid experiment would not have been possible since existing middleware toolkits such as Globus are rather heavyweight, requiring substantial effort and local tuning on the part of systems administrators to install and maintain. This effort cannot be expected from the average scientist who is planning to use a computational Grid (Chin & Coveney 2004). The simulation pipeline requires simulation, visualization, and storage facilities to be available simultaneously, at times when their human operators can reasonably expected to be around. This is often dealt with by manual reservation of resources by systems administrators, but the ideal solution would involve automated advance reservation and co-allocation procedures. The most exciting project involving RealityGrid during the last two years was the TeraGyroid experiment. Hundreds of individuals have worked together to build a transcontinental Grid not only as a demonstrator for the Grid techniques available today, but to perform a scientific project. Since we would not have been able to gain as many new results from the simulations performed during that period without the active use of Grid technologies, we have shown that the advent of computational Grids will be of great benefit for computational scientists.

We would like to thank S. Jha, M. Harvey and G. Giupponi (University College London), A.R. Porter and S.M. Pickles (University of Manchester), N. González-Segredo (FOM Institute for Atomic and Molecular Physics), and E.S. Boek and J. Crawshaw (Schlumberger Cambridge Research) for fruitful discussions and E. Breitmoser from the Edinburgh Parallel Computing Centre for her contributions to our lattice Boltzmann code. We are grateful to the UK Engineering and Physical Sciences Research Council (EPSRC) for funding much of this research through RealityGrid grant GR/R67699 and to EPSRC and the National Science Foundation (NSF) for funding the TeraGyroid project. This work was partially supported by the National Science Foundation under NRAC grant MCA04N014 and PACI grant ASC030006P, and utilised computer resources at the Pittsburgh Supercomputer Center, the National Computational Science Alliance and the TeraGrid. We acknowledge the European Synchrotron Radiation Facility for provision of synchrotron radiation facilities and we would like to thank P. Cloetens for assistance in using beamline ID19, as well as J. Elliott and G. Davis of Queen Mary, University of London, for their work in collecting the raw data and reconstructing the X-ray microtomography data sets used in our Bentheimer sandstone images.

Phil. Trans. R. Soc. A (2005)
References


HDF5 2003 HDF5—a general purpose library and file format for storing scientific data. (http://hdf.ncsa.uiuc.edu/HDF5)

Large-scale lattice Boltzmann simulations

McKeown, M. 2003 OGSI::Lite—an OGSI implementation in Perl. See http://www.sve.man.ac.uk/Research/AtoZ/ILCT.

Phil. Trans. R. Soc. A (2005)
Stress response and structural transitions in sheared gyroidal and lamellar amphiphilic mesophases: lattice-Boltzmann simulations

N. González-Segredo, J. Harting, G. Giupponi, and P. V. Coveney
Stress response and structural transitions in sheared gyroidal and lamellar amphiphilic mesophases: Lattice-Boltzmann simulations

Néildo González-Segredo
FOM Institute for Atomic and Molecular Physics (AMOLF), P. O. Box 41883, 1009 DB Amsterdam, The Netherlands

Jens Harting
Institut für Computerphysik, Pfaffenwaldring 27, D-70569 Stuttgart, Germany

Giovanni Giupponi and Peter V. Coveney
Centre for Computational Science, Department of Chemistry, University College London, 20 Gordon Street, London WC1H 0AJ, United Kingdom
(Received 2 September 2005; published 15 March 2006)

We report on the stress response of gyroidal and lamellar amphiphilic mesophases to steady shear simulated using a bottom-up lattice-Boltzmann model for amphiphilic fluids and sliding periodic (Lees-Edwards) boundary conditions. We study the gyroid per se (above the sponge-gyroid transition, of high crystallinity) and the molten gyroid (within such a transition, of shorter-range order). We find that both mesophases exhibit shear thinning, more pronounced and at lower strain rates for the molten gyroid. At late times after the onset of shear, the skeleton of the crystalline gyroid becomes a structure of interconnected irregular tubes and toroidal rings, mostly oriented along the velocity ramp imposed by the shear, in contradistinction with free-energy Langevin-diffusion studies which yield a much simpler structure of disentangled tubes. We also compare the shear stress and deformation of lamellar mesophases with and without amphiphile when subjected to the same shear flow applied normal to the lamellae. We find that the presence of amphiphile allows (a) the shear stress at late times to be higher than in the case without amphiphile, and (b) the formation of rich patterns on the sheared interface, characterized by alternating regions of high and low curvature.

DOI: 10.1103/PhysRevE.73.031503

PACS number(s): 83.50.–v, 83.80.Qr, 61.30.Cz, 61.30.Jf

I. INTRODUCTION

The study of the response to shear in amphiphilic mesophases has been the subject of attention for numerical modelers only in recent years. The interest in the subject is sustained not only by the wide range of applications in materials science and chemical engineering, but also by the need to gain a fundamental understanding of the universal laws governing the self-assembly processes and competing mechanisms present. Hitherto, studies have focused mainly on the structural changes induced by steady and oscillatory shear, near and far from critical points, in polymer systems [1–6]. The morphologies studied include cubic- and wormlike-micellar, lamellar, and hexagonally-packed-tubular mesophases; more complex structures are the so-called bicontinuous mesophases, of which those liquid-crystalline of cubic symmetry have thus far been considered in far less detail.

The amphiphilic gyroid [14,15] is a bicontinuous cubic liquid crystal consisting of multi-or monolayer sheets of self-assembled amphiphile dividing two regions, each containing phases which are mutually immiscible, e.g., aqueous and hydrocarbon species. These sheets or *labyrinths* form a triply periodic minimal surface (TPMS) whose unit cell is of cubic symmetry, has zero mean curvature, with no two points on it connected by a straight segment, and has no reflection symmetries. Their skeletons, i.e., the locus bounded by the TPMS, for each immiscible phase, form double (interweaving), chirally symmetric threefold coordinated lattices. There are lyotropic [14,15] and thermotropic transitions between the gyroid and the microemulsion mesophase, the latter being a bicontinuous mesophase of short-range order. The morphologies in the crossover regions of these transitions show shorter-range order than the gyroid’s and longer-range order than the microemulsion’s, for which reasons they are termed “molten gyroids.”

Bicontinuous cubic mesophases of monoglycerides and the lipid extract from the archebacterium *Sulfolobus solfataricus* have been found at physiological conditions in cell organelles and physiological transient processes such as membrane budding, cell permeation, and the digestion of fats [7]. They can also be synthesized for important applications in membrane protein crystallization, controlled drug release, and biosensors [8,9].

The purpose of this paper is to report on the response to shear of gyroid (G), molten-gyroid (MG), and lamellar (L₉) amphiphilic mesophases simulated using a bottom-up kinetic-theoretic model for fluid flow. The model is based on a lattice-Boltzmann (LB) method, which has proved to be a modeling tool alternative to and more efficient and robust than sophisticated methods based on continuum equations. This LB method adheres to a *bottom-up* complexity paradigm [13] in the sense that it is simple and fully particulate...
and no hypotheses of desirable macroscopic behavior are imposed on the microdynamics—yet we have shown in the past its ability to simulate correct segregation kinetics for immiscible fluids [20] and nonequilibrium self-assembly into amphiphilic mesophases [14,15]. Knowing that such a simple model is capable of simulating these kinetic processes from a purely bottom-up dynamics, in this paper we investigate how hydrodynamic interactions couple with self-assembly and modify the stability and morphology of the mesophases. The model also has the capability to reproduce morphological transitions without having to assume a macroscopic, free-energy model, used in other LB methods [3,10] to compute the diffusive currents substantiating self-assembly.

In addition, since our method models amphiphilic molecules as point dipoles—the simplest possible particulate model for an amphiphile—the rheological features emergent from it are expected to be universal for a broad range of amphiphilic systems. Finally, most of the numerical studies measuring the stress response of complex fluids to shear reported in the literature deal with phase-segregating fluids on one side [16], and the more complicated polymeric [11] and glassy systems on the other [12]. In this respect, the present paper stands somewhere in between these two.

Our paper is structured as follows. In the next section we briefly introduce the model and describe the boundary conditions for the imposition of shear. In Sec. III we report on simulation data and conclude that shear thinning occurs for both G and MG mesophases leading to a transition of tubular and ringlike structures as the strain increases. In Sec. IV we reveal how the presence of an amphiphile in lamellar mesophases induces the formation of rich interfacial patterns surviving shear and allows higher values of stress than in lamellar mesophases without amphiphile. Finally we provide our conclusions in Sec. V.

II. THE MODEL AND THE LEES-EDWARDS BOUNDARY CONDITIONS

We utilized an existing bottom-up lattice-Boltzmann model for amphiphilic fluids [14,15], extended to simulate shear flow by means of Lees-Edwards boundary conditions [16]. The model is in turn based on an extension made to the Shan-Chen bottom-up LB model for immiscible fluids to model amphiphilic-fluid flow, and employs 25 microscopic velocities, of speeds 0, 1, and $\sqrt{2}$, in three dimensions (D3Q25 lattice) [17,18]. The model uses a BGK (Bhatnagar-Gross-Krook, or relaxation-time) approximation to the collision term of the Boltzmann equation for fluid transport, which allows us to simulate, for large enough lattices [19], the Navier-Stokes (NS) momentum-balance equation in the bulk of each immiscible fluid species, namely, “oil” [or “red” (r)] and “water” [or “blue” (b)]. The model allows the simulation of correct phase-segregation kinetics in the absence [20] and presence [15] of a third, amphiphilic surfactantlike (s) dipolar species. The model controls the interparticle forces between r, b, and s species via coupling parameters ($g_{sr}, g_{sb}, g_{ss}$), and transients are controlled via relaxation times for densities ($\varphi^r, \varphi^s, \varphi^b$) with an additional relaxation time for the orientation of the amphiphile dipoles ($\tau^r$). In addition, the model simulates the nonequilibrium self-assembly and relaxation dynamics of sponge ($L_s$) and gyroid mesophases [14,15]. The gyrods that it simulates show rigidity, arising from their crystalline ordering, which decreases as the concentration of amphiphile is reduced; indeed, a lyotropic transition causes the correlation length to decrease toward that of a sponge mesophase through a molten-gyroid state. This idea is central to the work we present here: we shall see that the mesophase’s crystalline ordering enhances its stress response; indeed, we find shear thinning to occur at higher strain rates for gyrods than for sponges.

The Lees-Edwards boundary conditions (LEBCs) were originally proposed by Lees and Edwards in the context of molecular dynamics simulations [16]. They showed that these boundary conditions would give rise to a desired linear, wedged velocity profile while avoiding the troublesome spatial inhomogeneities appearing when solid walls are used to induce the shear flow [21]. A particular realization of the LEBCs on a Cartesian simulation box $[0, N_x] \times [0, N_y] \times [0, N_z]$ is established by letting the periodic images, at $N_x < x < 2N_x$ and $N_y < y < 0$, move parallel to unit vectors $\hat{x}$, respectively, both with speed $U$. The LEBCs, in their original, molecular dynamics form, are expressed as a Galilean transformation on the position $(x, y, z)$ and velocity $(\xi_x', \xi_y', \xi_z')$ coordinates of a molecule, as follows:

$$
x' = x \mod N_x,
\quad y' = y \mod N_y,
\quad z' = \begin{cases} 
(z + \Delta_z) \mod N_z, & x > N_s, \\
N_z \mod N_z, & 0 \leq x \leq N_s, \\
(z - \Delta_z) \mod N_z, & x < 0,
\end{cases}
$$

where $\Delta_z = U \Delta t$ is the image’s shift at time $\Delta t$ after the onset of shear.

An implementation of the LEBCs on our LB dynamics (LB LEBCs) differs from that used in molecular dynamics (MD LEBCs) in that the shift $\Delta_z$ is not in general a multiple of the lattice unit, as Wagner and Pagonabarraga have pointed out [21], and hence an interpolation scheme is needed. This interpolation scheme streams the amphiphile dipoles $d(x)$ and mass densities $n_i(x)$ located at position $x$ on the shearing wall, where $\xi_i$ is the relevant discrete molecular velocity, $k = 1, \ldots, 25$, for each (fluid and amphiphilic) species $\alpha$.

In our LB LEBCs, while the spatial displacement follows Eqs. (1), the velocity shift cannot be enforced by replacing the continuum velocity component $\xi_i$ in Eqs. (2) with the
We sheared two gyroidal mesophases differing in the amount of amphiphile present and the value of the interamphiphile interaction coupling parameter. Each of these structures was allowed to self-assemble from homogeneous mixtures of oil, water, and amphiphile using periodic boundary conditions. They have been appropriately characterized by probing direct and Fourier-space late-time snapshots of the density order parameter $\phi = \rho_{\text{oil}} - \rho_{\text{water}}$, more precisely, they correspond to gyroid [cf. Fig. 5(a)] and molten gyroid mesophases, as previously reported by us [14,15].

A. Stress response and transients

Shear thinning is said to occur when the shear viscosity drops as the strain rate increases. For structured fluids such as those we study in this paper, the dynamic shear viscosity $\eta$ is not expected to be a constant of the strain rate $\dot{\gamma}$; rather, it is a fast transient which, therefore, does not affect the shear response at the late times that we are interested in. In other words, the late-time shear response is insensitive to a small perturbation in the initial condition. This allowed us to take the upscaled, relaxed structure as the initial condition for the molten gyroid.

It is worth noting that we did not require an elongated aspect ratio for the lattices along the direction parallel to the translation of the shearing walls since spatial density fluctuations were much smaller than the lattice size. This is not the case when shearing phase-segregating fluids without an amphiphilic, growth-arresting species, as has been previously reported using LB lattices of up to 128:128:512 sizes and aspect ratio [24].

The common parameters used for both gyroids were oil and water densities flatly distributed in the range $0 < n^{(0)} = n^{(0)} < 0.7$, coupling strengths $g_{\bar{n}}= 0.08, \ g_{\bar{m}} = -0.006$, relaxation times $\tau = \tau = \tau = 1$, and, for the amphiphile’s dipoles, $\beta = 10$ and $d_0 = 1$.

Their differing parameters were surfactant densities, flatly distributed in the initial homogeneous mixture, in the ranges $0 < n^{(0)} < 0.9$ for the gyroid and $0 < n^{(0)} < 0.6$ for the molten gyroid, with coupling strengths $g_{\bar{n}} = -0.0045$ for the gyroid and $g_{\bar{n}} = -0.003$ for the molten gyroid. These values for the gyroid are 50% higher than those for the molten gyroid.

While the gyroid relaxes to a highly crystalline structure [23], the molten gyroid shows both shorter-range order and stronger temporal fluctuations than the former [15]. In order to obtain a sufficiently relaxed molten gyroid as an initial condition for the shear, we took the structure as evolved up to time step 32 500; regarding the gyroid, the time slice chosen was time step 15 000. For practical reasons, instead of letting the molten gyroid self-assemble starting from a homogeneous initial mixture, we upscaled a smaller molten gyroid, previously self-assembled using the same parameters on a 64$^3$ lattice [15], to a 128$^3$ lattice. Upscaling consisted in replicating identical copies of the system: the periodic boundary conditions used to generate the 64$^3$ system (a) guarantee that the density field is smooth across the replica boundaries, yet, for this same reason, (b) produce a molten gyroid with an additional, undesirable long-wavelength fluctuation whose periodicity is half the lattice size. The amplitude of this undesired long-wavelength fluctuation relaxes in time to a vanishingly small value, a fact which provides us with the 128$^3$ mesophase we seek. We observed, however, that this relaxation takes place in fewer than 1000 time steps [23], i.e., it is a fast transient which, therefore, does not affect the shear response at the late times that we are interested in. In other words, the late-time shear response is insensitive to a small perturbation in the initial condition. This allowed us to take the upscaled, relaxed structure as the initial condition for the molten gyroid.

It is worth noting that we did not require an elongated aspect ratio for the lattices along the direction parallel to the translation of the shearing walls since spatial density fluctuations were much smaller than the lattice size. This is not the case when shearing phase-segregating fluids without an amphiphilic, growth-arresting species, as has been previously reported using LB lattices of up to 128:128:512 sizes and aspect ratio [24].

III. SHEARING GYROIDAL MESOPHASES

We sheared two gyroidal mesophases differing in the amount of amphiphile present and the value of the interamphiphile interaction coupling parameter. Each of these structures was allowed to self-assemble from homogeneous mixtures of oil, water, and amphiphile using periodic boundary conditions. They have been appropriately characterized by probing direct and Fourier-space late-time snapshots of the density order parameter $\phi = \rho_{\text{oil}} - \rho_{\text{water}}$, more precisely, they correspond to gyroid [cf. Fig. 5(a)] and molten gyroid mesophases, as previously reported by us [14,15].

The common parameters used for both gyroids were oil and water densities flatly distributed in the range $0 < n^{(0)} = n^{(0)} < 0.7$, coupling strengths $g_{\bar{n}}= 0.08, \ g_{\bar{m}} = -0.006$, relaxation times $\tau = \tau = \tau = 1$, and, for the amphiphile’s dipoles, $\beta = 10$ and $d_0 = 1$.

Their differing parameters were surfactant densities, flatly distributed in the initial homogeneous mixture, in the ranges $0 < n^{(0)} < 0.9$ for the gyroid and $0 < n^{(0)} < 0.6$ for the molten gyroid, with coupling strengths $g_{\bar{n}} = -0.0045$ for the gyroid and $g_{\bar{n}} = -0.003$ for the molten gyroid. These values for the gyroid are 50% higher than those for the molten gyroid.

While the gyroid relaxes to a highly crystalline structure [23], the molten gyroid shows both shorter-range order and stronger temporal fluctuations than the former [15]. In order to obtain a sufficiently relaxed molten gyroid as an initial condition for the shear, we took the structure as evolved up to time step 32 500; regarding the gyroid, the time slice chosen was time step 15 000. For practical reasons, instead of letting the molten gyroid self-assemble starting from a homogeneous initial mixture, we upscaled a smaller molten gyroid, previously self-assembled using the same parameters on a 64$^3$ lattice [15], to a 128$^3$ lattice. Upscaling consisted in replicating identical copies of the system: the periodic boundary conditions used to generate the 64$^3$ system (a) guarantee that the density field is smooth across the replica boundaries, yet, for this same reason, (b) produce a molten gyroid with an additional, undesirable long-wavelength fluctuation whose periodicity is half the lattice size. The amplitude of this undesired long-wavelength fluctuation relaxes in time to a vanishingly small value, a fact which provides us with the 128$^3$ mesophase we seek. We observed, however, that this relaxation takes place in fewer than 1000 time steps [23], i.e., it is a fast transient which, therefore, does not affect the shear response at the late times that we are interested in. In other words, the late-time shear response is insensitive to a small perturbation in the initial condition. This allowed us to take the upscaled, relaxed structure as the initial condition for the molten gyroid.

It is worth noting that we did not require an elongated aspect ratio for the lattices along the direction parallel to the translation of the shearing walls since spatial density fluctuations were much smaller than the lattice size. This is not the case when shearing phase-segregating fluids without an amphiphilic, growth-arresting species, as has been previously reported using LB lattices of up to 128:128:512 sizes and aspect ratio [24].
FIG. 1. Shear stress response of a gyroid mesophase along the direction of the velocity gradient. As initial condition, we have taken a gyroid on an \( N_xN_yN_z=128^3 \) cubic lattice at time step \( t=15\,000 \) of self-assembly [14,15]. The Lees-Edwards walls move with speed \( U=0.10(\text{Ma}=0.17) \). For each \( x \) coordinate, the original field has been averaged on the plane \( [1,N_x] \times [16,N_y-16] \), where the excluded interval on the \( z \) axis accounts for wrapped-round densities. Standard errors of the averages are about \( 6\times10^{-6} \) throughout, and are not shown. Each line represents the response at \( 4 \text{t} \) time steps after the start of steady shear; \( \Delta t=0 \) (dotted line), 100 (dashed-dotted), 800 (dashed), and 9000 (solid), where the last is ca. the time at which the core (i.e., the plane \( x=64 \)) fully responds. From the figure we can see that momentum transfer decreases as it reaches the core from the walls. Also, note that the stress inverts its sign at late times adjacent to the boundaries, \( |x-x_c|\approx2 \) (\( x_0=0,128 \)). All quantities reported are in lattice units.

In the incompressible, low-Mach-number limit, our LB model reproduces the NS equation away from interfaces [17,19], which describes a Newtonian fluid with a viscosity being a well-known function of the relaxation time. The presence of an interface, characterized by an interfacial tension and a bending rigidity, however, introduces anisotropies in the fluid’s stress tensor which can be accounted for by a tensorial effective viscosity. Since the interface may move, at a speed growing with the strain rate, these anisotropies can become unsteady. Our aim is then to measure how this viscosity evolves with the strain and the strain rate.

In order to probe the function \( \eta=\eta(\gamma) \) for both gyroidal mesophases, we measured \( P_{xx} \) for a number of different applied shear rates. The chosen values for \( U \) were such that they remained within the incompressibility limit, i.e., small compared to the speed of sound on the D3Q25 lattice, \( c_s\approx3^{-1/2}=0.58 \). Values chosen were \( U=0.05,0.10,0.15,0.20, \) corresponding to Mach numbers \( \text{Ma}=U/c_s=0.086,0.17,0.26,0.34 \), respectively. All observables we report in this paper are spatial averages, at least on \( x=\text{const.} \) planes where a simple fluid under the same shear would show translational symmetry for the velocity field, i.e., perpendicular to the velocity gradient. Since, for reasons of computational cost, we do not perform averages over the seed used to generate the pseudorandom initial configuration mimicking a homogeneous ternary mixture, we do not provide error bars around averages.

Figure 1 shows the profile of the stress, for the sheared
gyroid, along the applied velocity gradient direction. Several
curves therein depict the transport of momentum toward the
gyroid, along the applied velocity gradient direction. Several
curves therein depict the transport of momentum toward the
gyroid, along the applied velocity gradient direction. Several
curves therein depict the transport of momentum toward the
gyroid, along the applied velocity gradient direction. Several
curves therein depict the transport of momentum toward the
gyroid, along the applied velocity gradient direction. Several
curves therein depict the transport of momentum toward the
gyroid, along the applied velocity gradient direction. Several
curves therein depict the transport of momentum toward the
gyroid, along the applied velocity gradient direction. Several
curves therein depict the transport of momentum toward the
gyroid, along the applied velocity gradient direction. Several
curves therein depict the transport of momentum toward the
gyroid, along the applied velocity gradient direction. Several
curves therein depict the transport of momentum toward the
gyroid, along the applied velocity gradient direction. Several
curves therein depict the transport of momentum toward the
gyroid, along the applied velocity gradient direction. Several
curves therein depict the transport of momentum toward the
gyroid, along the applied velocity gradient direction. Several
curves therein depict the transport of momentum toward the
gyroid, along the applied velocity gradient direction. Several
curves therein depict the transport of momentum toward the
gyroid, along the applied velocity gradient direction. Several
curves therein depict the transport of momentum toward the
gyroid, along the applied velocity gradient direction. Several
curves therein depict the transport of momentum toward the
gyroid, along the applied velocity gradient direction. Several
curves therein depict the transport of momentum toward the
gyroid, along the applied velocity gradient direction. Several
curves therein depict the transport of momentum toward the
gyroid, along the applied velocity gradient direction. Several
curves therein depict the transport of momentum toward the
gyroid, along the applied velocity gradient direction. Several
curves therein depict the transport of momentum toward the
gyroid, along the applied velocity gradient direction. Several
curves therein depict the transport of momentum toward the
gyroid, along the applied velocity gradient direction. Several
curves therein depict the transport of momentum toward the
gyroid, along the applied velocity gradient direction. Several
curves therein depict the transport of momentum toward the
gyroid, along the applied velocity gradient direction. Several
curves therein depict the transport of momentum toward the
gyroid, along the applied velocity gradient direction. Several
curves therein depict the transport of momentum toward the
gyroid, along the applied velocity gradient direction. Several
curves therein depict the transport of momentum toward the
gyroid, along the applied velocity gradient direction. Several
curves therein depict the transport of momentum toward the
gyroid, along the applied velocity gradient direction. Several
curves therein depict the transport of momentum toward the
gyroid, along the applied velocity gradient direction. Several
curves therein depict the transport of momentum toward the
gyroid, along the applied velocity gradient direction. Several
curves therein depict the transport of momentum toward the
gyroid, along the applied velocity gradient direction. Several
curves therein depict the transport of momentum toward the
gyroid, along the applied velocity gradient direction. Several
curves therein depict the transport of momentum toward the
gyroid, along the applied velocity gradient direction. Several
curves therein depict the transport of momentum toward the
gyroid, along the applied velocity gradient direction. Several
curves therein depict the transport of momentum toward the

B. Morphological transitions

Figure 5 shows the configuration of the gyroid in the 40 \(\leq y \leq 52\) slab of the 128:128:128 lattice, before and at late times after applying a shear of \(U=0.20\). The volume-rendering graphical representation employed [25] makes regions where \(\phi > 0.37\) opaque to the lighting rays, assumed to shine normal to the plane of the text and inward; since \(-0.79 < \phi < 0.79\) over the entire system, these regions are the high-density locus of one of the species (say, oil). Before shear, the structure contains highly ordered subvolumes of gyroid symmetry and diagonal length from about 32 to 64 lattice sites [cf. Fig. 5(a)]. This gyroid is hence a collection of subvolumes with a regular tubular structure making up two threefold coordinated, interweaving chiral lattices of which we depict only one. Since the size of the G unit cell is approximately 5–6 lattice units, the depth \((y\) dimension) of the slabs shown in Fig. 5 is of about two gyroid unit cells. As can be seen in Fig. 5(a), the interfaces between these gyroid subvolumes are defective regions where long-range order and symmetry appear to be drastically reduced [14,15]. Two features characterizing them are the spatial variation in coordination number and chirality, seen by the presence of elongated tubules and toroidal rings (cf. Fig. 6).

At \(\Delta t = 21\,000\), which is a late time after the onset of shear and we take as steady state, the structure has lost any resemblance to the initial gyroid, except for the persistence of the toroidal rings [see Fig. 5(c)], which are defects in G. Also, the structure at \(\Delta t = 21\,000\) is essentially the same as that at time step \(\Delta t = 5000\)—it is a nonequilibrium steady state for at least the previous 16 000 time steps, a time longer than that required for the initial configuration to self-assemble from a homogeneous mixture of oil, water, and amphiphile. The structure at \(\Delta t = 21\,000\) consists of an irregular network of mainly the same structural elements characterizing the defective regions before the onset of shear, namely, (a) elongated tubules, with a tendency to align along a direction that is a linear combination of directions \((1, 0, 0)\) and \((0, 0, 1)\), and (b) toroidal, ringlike structures. This description is, by visual inspection, similar for every subvolume of the lattice visualized.

We also looked into the structure of the sheared molten gyroid at late times. In contradistinction to the gyroid’s state at high strain, showing tubules of shape similar to that depicted in Fig. 6 and at an angle with the \(x = \text{const}\) planes, the highly strained molten gyroid displays tubes which are more stretched and aligned along the \(z\) direction. The toroidal rings, also present for the molten gyroid before shear, represent a much smaller volume fraction for the sheared molten gyroid than for the sheared gyroid.

Figure 7 shows the summed structure function \(\Sigma_y S(k,t)\), or “scattering pattern”, of the sheared gyroid mesophase, showing stages of its plastic deformation. Here, \(S(k,t)\) is the structure function, computed according to [15,20]

\[
S(k,t) = \frac{\langle \phi^2(k,t) \rangle}{V}.
\]  

Here, \(k\) is the discrete wave vector, \(V\) is the lattice volume, \(\phi\) is the unit cell volume for the D3Q25 lattice, and \(\phi^2(k,t)\) is the Fourier transform of the fluctuations of \(\phi\). \(S(k,t)\) is the Fourier transform of the autocorrelation function for the order parameter,

\[
C_{\phi\phi}(r,t,t) = \langle \phi(x,t)\phi(x + r,t) \rangle
\]  

where \(r\) is a vector lag and the angular brackets indicate an average over the spatial coordinate \(x\). Figures 7(a), 7(b), and 7(d) are the \(x_2\) “scattering patterns” of the structures in Fig. 5, produced by summing up the structure function along the \(x\) direction. At \(\Delta t = 1000\) (not shown), the maximum intensity is reduced to 29% of its value at \(\Delta t = 0\), while there appear horizontal “smeared out filaments” of very weak intensity, intrinsically related to the shearing process, as we shall conclude from Fig. 8. At \(\Delta t = 5000\) a clear cardioid shape has developed; the fact that it persists for the rest of the simulation confirms our observation that the system reaches a steady state at time step \(\Delta t = 5000\). In addition, there is no trace of gyroidal patterns along the \(x\) direction.

In order to investigate the origin of the cardioid shape, we computed the scattering pattern for a “synthetic gyroid”
STRESS RESPONSE AND STRUCTURAL TRANSITIONS...
good approximation to the Schoen "G" triply periodic minimal surface of Ia3̅d cubic symmetry, referred to as "the ideal gyroid" hereafter [26]. Figure 8 shows the scattering patterns for the unstrained morphology and for dephases $\delta_{\text{min}}=8, 16$.

Comparing structure function maps in Figs. 7 and 8, at the same value of the strain rate, proves useful. For the synthetic gyroid, the strain is controlled by the number of unit cells that the dephase causes the structure to shift at the lattice boundary, following a linear profile as we approach the other boundary going through zero strain at the lattice core. For our simulated amphiphilic gyroid, however, the strain does not follow a linear profile at early times; instead, the strain at our simulated amphiphilic gyroid, however, the strain does not follow a linear profile at early times; instead, the strain at later times, the coordination number can be reduced to 2, describing a "tubule." We also show the skeleton of the "ring" structure ubiquitous in the sheared gyroid at late times, also present in smaller proportion as a defect in the mesophase before the onset of shear. At lower values of density, this ring appears to be toroidal.

IV. A SIMPLER CASE: SHEARING THE LAMELLAR MESOPHASE

In the last section we reported on the gyroid displaying higher shear stress than the molten gyroid. Since the structural transition between these two mesophases can be driven by both the amphiphile density and the interamphiphile coupling parameter, as we have reported in the past [15], our aim in this section is to elucidate the role of the amphiphile density alone on the stress response to shear; we choose the lamellar mesophase as the subject of study, since this is the mesophase with the simplest possible internal interface.

The initial configuration employed was a cubic 128\(^3\) lattice with 16 lamellae, stacked perpendicularly to unit vector $\hat{z}$. The lamellae were of alternating oil-water compositions, separated by a thin monolayer of amphiphile; the thicknesses of the immiscible and amphiphilic lamellae were 7 and 1 lattice sites, respectively. We populated each lattice site with a value of density kept constant over the region corresponding to a given species; each microscopic velocity is assigned the same fraction of this value. We gave amphiphilic regions the densities $n_{10}=0, 0.80, 0.95$, and oil and water regions the densities $n_{01}=n_{H0}=0.7$. Shear was applied perpendicularly to the lamellae with the same LEBCs employed in the last section, with speed $\dot{\gamma}=0.10$.

Before the onset of shear, the case without amphiphile for the lamellar initial condition just described is, a priori, a metastable state in our LB model. In fact, the structure has a stationary morphology since short-range oil-water forces and the absence of fluctuations maintain immiscibility, i.e., a value for the interface steepness $\nabla\phi$; however, a large enough perturbation in $\phi$ may allow a fluctuation in surface tension which drives the entire interface to a radically different shape. Another factor disrupting this lamellar morphology is shear, which may work against the interfacial tension which drives the entire interface to a radically different shape. Another factor disrupting this lamellar morphology is shear, which may work against the interfacial tension by reducing $\nabla\phi$; this can lead to miscibility ($\dot{\gamma}=0$) for high enough strain rates. Despite these arguments, we observed stability for the sheared lamellar mesophase without amphiphile, as we report next.

Figure 9 shows the stress as measured in the same fashion performed on the data plotted in Fig. 3, for several amphiphile densities. The behavior observed is diverse. For zero amphiphile concentration (solid curve), the stress reaches a peak at early times before it proceeds to a second, lower maximum at late times, going through a trough at intermediate times due to the fact that $\nabla\phi$ experiences a transient decrease.

The high-density regions of one of the immiscible species (say, oil) is shown in Fig. 10(a) at late times, $\Delta t=8000$; these are representative of the shape of the oil-water interface. Away from the boundaries ($x=0, 128$), there is a large interfacial area with zero curvature, where we define the curvature as $H = \partial^2 \phi / \partial x^2$ being the curve resulting from projecting the $\phi=0.18$ surface onto the $x_3$ plane. Curiously, we observe three changes of curvature as we follow the curve $x_3(z)$ for $y=$const, namely, $H<0, H>0, H<0, H>0$; instead, we would have expected the steady, late-time configuration for the sheared lamellar mesophase to rather minimize the interfacial area, leaving only one inflection point.
For fluid regimes under conditions of local thermodynamic equilibrium, we can think of H2 as an interfacial free-energy density associated with the curvature [27]; in this case, we would have expected the steady, late-time configuration to also minimize the interfacial free energy.

The stress curve corresponding to $n^{(0)}=0.80$ (cf. Fig. 9) shows the absence of large troughs, as occurs for the $n^{(0)}=0$ case, despite the fact that interfacial tension is drastically reduced by the presence of the amphiphile. In addition, the stress grows at late times to higher values than those achieved by the $n^{(0)}=0$ case. The late-time order-parameter configuration is displayed in Fig. 10(b), showing a rich interfacial pattern. Using the same arguments as those of the last paragraph, this structure could be characterized by a higher curvature energy $\int d^2x H^2$, where $d^2x$ is a measure on the oil-water interface, and $H$ is now defined as the inverse radius of curvature, parametrized on the arclength $s$. Figure 10(b) shows similar regions of high curvature at an equal distance from the shearing walls, where $u_s=\text{const}$, which we shall call nodal planes. Also note that the interface, as approximately depicted by the boundary of the $\phi \approx 0.22$ volume, joins the lattice boundary at an angle close to 90°.

The stress curve for the $n^{(0)}=0.95$ case shows a dramatically different situation for the first 5000 time steps: the presence of a trough, deeper than that present for the $n^{(0)}=0$ density. After that, there appears a shoot-off whereby the stress rapidly grows and equals the late-time value achieved in the $n^{(0)}=0.80$ case, while the order parameter displays a configuration analogous to the $n^{(0)}=0.80$ case [cf. Fig. 10(c)]. By looking at the amphiphile density field $p(x)$ for the case $n^{(0)}=0.95$, we observed that the high-curvature regions arise close to the boundaries first ($\Delta t \approx 1000$), and then rapidly move away from them as the strain progresses.

**V. CONCLUSIONS**

In this paper we have reported on the shear stress response of two gyroidal cubic amphiphilic mesophases previously self-assembled using the same bottom-up LB model we employ here, namely, the gyroid per se, G, which shows high crystallinity at late self-assembly times, and the molten gyroid MG, endowed with shorter-range order and located within the sponge-gyroid lyotropic structural transition [15]. Shear was imposed via sliding periodic (Lees-Edwards) boundary conditions, and we investigated the response to several values of the strain rate. In addition, in order to investigate the dependence of the shear stress on the amphiphilic density, we also sheared a lamellar mesophase, of much simpler morphology than the gyroidal mesophases.
We found that the gyroidal mesophases exhibit shear thinning, more pronounced and at lower strain rates for the MG mesophase than for the G mesophase. In other words, momentum introduced into the system due to shear is transported more easily for the mesophase containing more amphiphile, with longer-range ordering, i.e., the effective viscosity is higher for the G mesophase.

We also found a shear-induced transition from an initial gyroidal morphology (G and MG) to a mesophase characterized by coexisting elongated tubules and toroidal, ringlike structures. The features of this mesophase are in contrast to those of the mesophase reported by Zvelindovsky et al. using free-energy Langevin-diffusion methods by shearing a bicontinuous structure reminiscent of a molten gyroid. The structure they found is of a shorter-range ordering than that of the MG mesophase described here, and the high-strain structure consists of coexisting lamellae and hexagonally packed tubes elongated along the direction of the imposed shear velocity. Our sheared mesophases also show elongated tubes along this direction, but the structure is far more complicated than that found by Zvelindovsky et al. in that it exhibits remnant toroidal rings and “hard shoulders” reminiscent of gyroidal skeletons; hexagonal packing and coexisting lamellae are, on the other hand, absent.

The shear performs a plastic deformation which effectively breaks the links of the gyroidal skeleton; this happens as these links interpose an (oil-water) interface whose normal $\mathbf{n}$ is parallel or antiparallel to the flow streamlines $\mathbf{u}$. In other
words, shear effectively applies a “mixing” force which is in competition with the interparticle forces keeping the mesophase in place, namely, those controlled by coupling parameters $g_{br}, g_{bs}$, and $g_{ss}$. Our hypothesis is that adsorbed dipoles on interfacial regions at an angle $\theta = \angle (\mathbf{u}, \mathbf{n})$ other than $\theta=0, \pi$ require more work from the shear forces to be drawn away from the interface than those regions on which the streamline impinges normally, since the mixing force goes as $\cos \theta$. In particular, since the mixing force vanishes for $\theta=\pi/2$, considerably longer interfaces can survive the flow—shear induces a preferential direction along which the long-range order present before the onset of shear is not reduced. This explains not only the formation of the elongated tubes but also their reconnection (increase in coordination.

FIG. 10. Slabs 0 ≤ $y$ ≤ 8 of the order parameter $\phi$ for sheared lamellar mesophases corresponding to increasing amphiphile density $n^{00}$ as indicated below each relevant panel, at time step $\Delta t=8000$ after the onset of shear and for shear velocity $U=0.10$. The coordinate system is the same as that in Fig. 5. In (a), the regions opaque to incident (volume rendering) light are those for which $\phi \geq 0.18$, where $|\phi| \leq 0.36$ across the system. In (b), the opaque regions are those for which $\phi \geq 0.22$, where $|\phi| \leq 0.45$ across the system. In (c), the opaque regions are those for which $\phi \geq 0.24$, where $|\phi| \leq 0.48$ across the system. It is worth noting that the surfactantless case (a) exhibits a curved interface. The amphiphilic cases (b) and (c) display the formation of irregularities in the interface and nodal planes, as a result of the interamphiphile interaction. All configurations have translational symmetry along the $y$ axis. All quantities reported are in lattice units.
number). In fact, the toroidal, ringlike structures are not only vestigial gyroid defects which have survived the gradient \( Vu \), but are also created anew as a result of reconstructions. It is relevant to point out that Padding and Boek, using a coarse-grained molecular dynamics model for wormlike micelles, reported on the formation of rings when applying steady shear to a wormlike micellar mesophase [6]—this is an “amphiphile-in-water” binary mesophase, in contrast to the “oil-amphiphile-water” ternary mesophases we study in this paper.

By applying shear to a lamellar mesophase we found that the presence of amphiphile on the oil-water interface of the mesophase causes the interface to fold into a wealth of structures with a (discrete) translational symmetry on planes equidistant from the shearing walls and along the direction of the shear velocity. In other words, the interamphiphile force couples the adsorbed amphiphilic dipoles so that the interface locally increases its curvature energy density. It is worth investigating whether this local increase is due to the amphiphile being incapable of sustaining interfacial regions of low curvature under shear, i.e., whether the “breaking” mechanism induced by shear is counteracted by regions of high curvature energy density. Regarding the shear stress, our amphiphile-containing lamellae responded with higher stress at late times than those without amphiphile. This contrasts with the results found for the gyroidal mesophases, and lets us conclude that it is the gyroid’s cubic morphology that allows this structure to be stiffer. Understanding the behavior of the lamellar mesophase under shear requires the study of amphiphile self-assembly under shear, including in- and out-of-plane amphiphilic and associated Marangoni currents, and their coupling to the imposed flow.

**ACKNOWLEDGMENTS**

We thank Dr. Rafael Delgado-Buscalioni, Professor Antonio Coniglio, and Professor Francesco Sciortino for enlightening discussions. We acknowledge Iain Murray, Elena Breitmoser, and Jonathan Chin for their involvement in algorithm implementation and optimization within the Reality-Grid and TeraGyroid projects. This work was supported by the U.K. EPSRC under RealityGrid Grant No. GR/R67699 which also provided access to a 512-processor SGI Origin3800 platform at Computer Services for Academic Research (CSAR), Manchester Computing. U.K. We also thank the Higher Education Funding Council for England (HEFCE) for our on-site 16-node SGI Onyx2 graphical supercomputer.

Emergence of rheological properties in lattice Boltzmann simulations of gyroid mesophases

G. Giupponi, J. Harting, and P. V. Coveney
Emergence of rheological properties in lattice Boltzmann simulations of gyroid mesophases

G. Giupponi\textsuperscript{1}, J. Harting\textsuperscript{2} and P. V. Coveney\textsuperscript{1}

\textsuperscript{1} Centre for Computational Science, Department of Chemistry
University College London - 20 Gordon Street, WC1H 0AJ, London, UK
\textsuperscript{2} Institute for Computational Physics, University of Stuttgart
Pfaffenwaldring 27, 70569 Stuttgart, Germany

received 28 June 2005; accepted in final form 21 December 2005
published online 13 January 2006

PACS. 47.11.+j – Computational methods in fluid dynamics.
PACS. 82.70.Uv – Surfactants, micellar solutions, vesicles, lamellae, amphiphilic systems (hydrophilic and hydrophobic interactions).
PACS. 83.60.-a – Rheology: Material behaviour.

Abstract. – We use a lattice Boltzmann (LB) kinetic scheme for modelling amphiphilic fluids that correctly predicts rheological effects in flow. No macroscopic parameters are included in the model. Instead, three-dimensional hydrodynamic and rheological effects are emergent from the underlying particulate conservation laws and interactions. We report evidence of shear thinning and viscoelastic flow for a self-assembled gyroid mesophase. This purely kinetic approach is of general importance for the modelling and simulation of complex fluid flows in situations when rheological properties cannot be predicted \textit{a priori}.

Introduction. – Lattice Boltzmann (LB) modelling schemes have emerged in the last several years as a powerful approach for simulating the dynamics of a variety of complex systems, from flow with suspended particles to multiphase flow through porous media [1]. Of great interest is the extension of this method to the modelling of flow properties of viscoelastic materials such as polymer melts and amphiphilic fluids, where the dynamics of microscopic structures is coupled to the flow.

Qian and Deng [2] correctly described transverse wave propagation using a lattice BGK model with an \textit{ad hoc} modified equilibrium distribution, while Ispolatov and Grant [3] obtained viscoelastic effects by adding a force to represent memory effects in the LB equation. However, both approaches involve the inclusion of macroscopic parameters such as Young’s modulus [2] or elastic coefficients [3]. These methods cannot be regarded as fully mesoscopic as at least one parameter is imposed on the basis of macroscopic considerations. Giraud \textit{et al.} [4] introduced a single fluid, two-dimensional model to treat a macroscopically viscoelastic fluid and later Lallemand \textit{et al.} [5] extended this model to three dimensions, but to the best of our knowledge no generalization for multiphase flow or flow with suspended particles has yet been implemented. A free-energy Ginzburg-Landau (GL) model can be defined to study rheological properties of
complex fluids [6,7]. A popular LB scheme based on the same GL approach proceeds by defining an equilibrium distribution through the imposition of constraints on macroscopic thermomechanical quantities such as the stress tensor. Using such a scheme, Denninston et al. [8] obtained non-Newtonian flow behaviour (including shear thinning and banding) using an LB algorithm for the hydrodynamics of liquid crystals in the isotropic and nematic phases. However, with such methods the dynamics is not dictated by the mesoscopic processes; numerical instability [9] can make this approach unsuitable for a fully mesoscopic description of the dynamics.

González and Coveney [10], using a fully mesoscopic, kinetic approach which does not require the existence of a thermodynamic potential, obtained a self-assembled gyroid mesophase in the course of simulating an amphiphilic fluid formed by two immiscible fluids and a surfactant species (fig. 1). The term amphiphilic fluid is used to describe a fluid in which at least one species is made of surfactant molecules. Surfactants are molecules comprised of a hydrophilic (water-loving) head group and a hydrophobic (oil-loving) tail. An amphiphilic fluid may contain oil, water, or both fluids in addition to surfactant. Such complex fluids can self-assemble to form ordered structures such as lamellae, micelles, sponge and liquid crystalline (cubic) phases showing pronounced rheological properties [11].

In this letter we use this purely kinetic LB method to model complex flows whose rheological properties are emergent from the mesoscopic kinetic processes without any imposed macroscopic constraints [12]. In particular, we show evidence of the appearance of rheological effects, such as shear thinning and viscoelasticity, for a self-assembled gyroid liquid crystalline cubic mesophase.

The model. – A standard LB system involving multiple species is usually represented by a set of equations [13]:

\[ n_i^\alpha(x + c_i, t + 1) - n_i^\alpha(x, t) = -\frac{1}{\tau_{\alpha}} (n_i^\alpha(x, t) - n_i^{\alpha eq}(x, t)) , \quad i = 0, 1, \ldots, b , \tag{1} \]

where \( n_i^\alpha(x, t) \) is the single-particle distribution function, indicating the density of species \( \alpha \) (for example, oil, water or amphiphile), having velocity \( c_i \), at site \( x \) on a \( D \)-dimensional lattice of coordination number \( b \), at timestep \( t \). The right side of eq. (1) represents the change in the single-particle distribution function due to the collisions. We choose a single relaxation time \( \tau_{\alpha} \), “BGK” form [14] for the collision operator. In the limit of low Mach numbers, the LB equations correspond to a solution of the Navier-Stokes equation for isothermal, quasi-incompressible fluid flow whose implementation can efficiently exploit parallel computers, as the dynamics at a point requires only information about quantities at nearest-neighbour lattice sites. The local equilibrium distribution \( n_i^{\alpha eq} \) plays a fundamental role in the dynamics of the system as shown by eq. (1). In this study, we use a purely kinetic approach, for which the
local equilibrium distribution $n_{i}^{\alpha \text{eq}}(x,t)$ is derived by imposing certain restrictions on the microscopic processes, such as explicit mass and total momentum conservation [15],

$$n_{i}^{\alpha \text{eq}} = \zeta_{\alpha} n_{\alpha} \left[ 1 + \frac{c_{\alpha} \cdot u}{c_{s}^{2}} + \frac{(c_{\alpha} \cdot u)^{2}}{2c_{s}^{4}} - \frac{u^{2}}{2c_{s}^{2}} + \frac{(c_{\alpha} \cdot u)^{3}}{6c_{s}^{6}} - \frac{u^{2}(c_{\alpha} \cdot u)}{2c_{s}^{4}} \right] ,$$  \hspace{1cm} (2)

where $u = u(x,t)$ is the macroscopic bulk velocity of the fluid, defined as $n^{\alpha}(x,t)u^{\alpha} = \sum_{i} n_{i}^{\alpha}(x,t)c_{i}$, $\zeta_{\alpha}$ are the coefficients resulting from the velocity space discretization and $c_{s}$ is the speed of sound, both of which are determined by the choice of the lattice, which is D3Q19 in our implementation. Immiscibility of species $\alpha$ is introduced in the model following Shan and Chen [16,17]. Only nearest-neighbour interactions among the immiscible species are considered. These interactions are modelled as a self-consistently generated mean-field body force:

$$F_{\alpha}(x,t) \equiv -\psi_{\alpha}(x,t) \sum_{\bar{\alpha}} g_{\alpha \bar{\alpha}} \sum_{x'} \psi_{\bar{\alpha}}(x',t)(x' - x) ,$$  \hspace{1cm} (3)

where $\psi_{\alpha}(x,t)$ is the so-called effective mass, which can have a general form for modelling various types of fluids (we use $\psi_{\alpha} = \left[ 1 - e^{-n_{\alpha}} \right]$ [16]), and $g_{\alpha \bar{\alpha}}$ is a force coupling constant whose magnitude controls the strength of the interaction between components $\alpha, \bar{\alpha}$ and is set positive to mimic repulsion. The dynamical effect of the force is realized in the BGK collision operator in eq. (1) by adding to the velocity $u$ in the equilibrium distribution of eq. (2) an increment

$$\delta u^{\alpha} = \frac{\tau_{\alpha} F_{\alpha}}{n^{\alpha}} .$$  \hspace{1cm} (4)

As described above, an amphiphile usually possesses two different fragments, each having an affinity for one of the two immiscible components. The addition of an amphiphile is implemented as in [12]. An average dipole vector $d(x,t)$ is introduced at each site $x$ to represent the orientation of any amphiphile present there. The direction of this dipole vector is allowed to vary continuously and no information is specified for each velocity $c_{i}$, for reasons of computational efficiency and simplicity. Full details of the model can be found in [12] and [18].

Simulation details. – We use LB3D [19], a highly scalable parallel LB code, to implement the model. A single simulation of a $64^{3}$ model, i.e. a point in fig. 2 below, needs around 300 Mbytes of RAM and takes about 50 CPU hours to complete on a single processor machine. The very good scaling of our LB3D code permits us to run all our simulations on multiprocessor machines and computational grids in order to reduce the length of data collection to a few weeks. The simulation parameters are those used in [10] that lead to the formation of gyroid mesophases. We use a $64^{3}$ lattice size with periodic boundary conditions. The simulations were all started from a checkpointed configuration of a 200000 timesteps equilibrated gyroid mesophase [19]. A single unit cell of the gyroid minimal surface is of the order of 5–8 lattice lengths in its linear dimensions. This implies that hundreds of unit cells are present in a $64^{3}$ sample. Therefore, finite-size effects are not expected to affect the qualitative outcome of the simulations. Moreover, some simulations with a $128^{3}$ lattice have been run to confirm the results obtained and the $128^{3}$ simulations reported in [20] are consistent with the results shown here. Nevertheless, such lattice sizes oblige us to study perfect gyroids; much larger system sizes (and computational cost) would be necessary in order to investigate the role of domains and their associated defects [21]. We leave this for future work.

In order to inspect the rheological behaviour of gyroid mesophases, we have implemented Lees-Edwards boundary conditions, which reduce finite-size effects if compared to moving solid walls [22]. This computationally convenient method imposes new positions and velocities on...
Fig. 2 – Plot of $\sigma_{xz}$ (dimensionless units) vs. shear rate $\dot{\gamma}_{xz}$ (Weissenberg number) for an LB simulation of a 64$^3$ gyroid mesophase. $\tau_r$ is the stress relaxation time; see text for discussion. The inset shows the percentage drop of the viscosity as the shear rate $\dot{\gamma}_{xz}$ increases (shear-thinning).

Fig. 3 – Plot of $\sigma_{xx}$ (black dots) vs. time for oscillatory shear ($U = 0.05$ and $\omega = 0.0004$, $0.005$ in eq. (6)) applied to 64$^4$ lattice-Boltzmann gyroid mesophase. All quantities are dimensionless. After a brief transient, $\sigma_{xz}$ oscillates with the imposed frequency $\omega \tau_r$. The strain (dashed line) is plotted (scaled) as a guide to the eye. The continuous line is the best fit line for the $\sigma_{xz}$ data. Shear moduli $G'(\omega \tau_r)$, $G''(\omega \tau_r)$ are calculated as in eq. (9) over a time interval shown by the horizontal straight line.

particles leaving the simulation box in the direction perpendicular to the imposed shear strain while leaving the other coordinates unchanged. Choosing $z$ as the direction of shear and $x$ as the direction of the velocity gradient, we have

$$
\begin{align*}
z' & \equiv \begin{cases} 
(z + \Delta z) \mod N_z, & x > N_x, \\
\mod N_z, & 0 \leq x \leq N_x, \\
(z - \Delta z) \mod N_z, & x < 0,
\end{cases} \\
\end{align*}
$$

$$
\begin{align*}
u_z' & \equiv \begin{cases} 
u_z + U, & x > N_x, \\
u_z, & 0 \leq x \leq N_x, \\
u_z - U, & x < 0,
\end{cases} \\
\end{align*}
$$

where $\Delta z \equiv U \Delta t$, $U$ is the shear velocity, $u_z$ is the $z$-component of $\mathbf{u}$ and $N_z(z)$ is the system length in the $z(z)$-direction. We also use an interpolation scheme suggested by Wagner and Pagonabarraga [23] as $\Delta z$ is not generally a multiple of the lattice site. Cates et al. [24], found pronounced artefacts (lock-ins) in simulations of 2D sheared binary mixtures with multiple Lees-Edwards planes. In our own work, we have not seen any of these artefacts, even in the longest simulations performed, and a linear velocity profile is obtained at steady state.

Consistent with the hypothesis of the LB model, we set the maximum shear velocity to $U = 0.1$ lattice units. This results in a maximum shear rate $\dot{\gamma}_{xz} = \frac{2 \times 0.1}{64} = 3.2 \times 10^{-3}$ in lattice units. Simulations are run for $T = 10000$ timesteps in the case of steady shear. Steady state is reached in the first 1000 timesteps, and the relevant component of the stress tensor is averaged over the last 3000 timesteps. Some measurements were repeated by doubling the simulation time to a total of $T = 20000$ timesteps but no significant differences were found. For oscillatory shear, we set

$$
U(t) = U \cos(\omega t),
$$

where $\omega/2\pi$ is the frequency of oscillation and $U = 0.05$ lattice units. These simulations were run for at least three complete oscillations, $t = 5000$–100000 timesteps.

Results. – In order to investigate rheological properties of the system, we perform simulations of flow under shear, mimicking a rheometer. We set $U = n \times 0.005$, $n = 2$–21 and $\omega = 0$ to impose a stationary Couette flow. Once steady state flow is reached, we collect the
relevant component of the pressure tensor $\sigma_{xz}$. According to Newton’s law for viscous flow of a liquid we have

$$\sigma_{xz} = 2\eta \dot{\gamma}_{xz},$$  \hspace{1cm} (7)

where $\eta$ is the viscosity of the liquid and $\dot{\gamma}_{xz} = \frac{\partial_x u_z + \partial_z u_x}{2}$. In fig. 2 we plot $\sigma_{xz}$ against the imposed non-dimensionalized (see text below) shear rate $\dot{\gamma}_{xz} \tau_r$ (Weissenberg number). We note that the slope of the curve changes as these shear rates increases. This indicates that the viscosity $\eta$ depends on the shear rate $\dot{\gamma}_{xz}$, $\eta = \eta(\dot{\gamma}_{xz})$. We therefore conclude that our model exhibits non-Newtonian flow behaviour. Complex fluids such as amphiphilic mixtures are well known to exhibit such behaviour experimentally [11]. In the inset we calculate the percentage drop of $\eta$ by averaging the slope of the curve over subsets of four data points. We note that the viscosity decreases as the shear rate increases. This behaviour is referred to as “shear thinning” [11]. In general, fluids exhibit macroscopic non-Newtonian properties because of underlying complex mesoscopic interactions and leading to changes in their microstructural properties. In our case, the highly ordered structure of the gyroid mesophase is responsible for this rheological effect, and our LB model correctly captures it. Therefore, considering the underlying physics of mesophases, this simulation result is not only justified but also expected. We note that the predictions of the model could be directly verified, but no experimental evidence for gyroid rheology is currently available in the literature.

For oscillatory shear, in eq. (6) we set $U = 0.05$ lattice units and we span a range of two decades of frequencies by varying $\omega$ between $\omega = 0.0001$ and $\omega = 0.01$ lattice units. For a viscoelastic medium

$$\sigma_{xz} = \gamma_0 [G'(\omega) \sin(\omega t) + G''(\omega) \cos(\omega t)],$$  \hspace{1cm} (8)

where $\gamma_0 = \frac{2U}{dx}$ is the imposed shear strain and $G'(\omega)$, $G''(\omega)$ are, respectively, the storage and loss moduli which respectively measure the elastic and viscous response at any given frequency [11]. In fig. 3 we show the time dependence of $\sigma_{xz}$ for two simulations with different values of $\omega$. After a brief transient, $\sigma_{xz}$ begins to oscillate as predicted by eq. (8) with the imposed frequency $\omega$ and an amplitude $\sigma_0$ and shift $\phi$ that depend on the storage and loss moduli

$$G' = \frac{\sigma_0 \cos(\phi)}{\gamma}, \hspace{1cm} G'' = \frac{\sigma_0 \sin(\phi)}{\gamma},$$  \hspace{1cm} (9)

where $\sigma_0$ and $\phi$ are the fitted values for the amplitude and phase shift. We derive $G'(\omega)$ and $G''(\omega)$ for the two different values of $\omega = 0.005, 0.0004$ by fitting $\sigma_0$ and $\phi$ with the standard Levenberg-Marquardt algorithm over at least three decades (see caption for details). In fig. 4, we plot $\sigma_{xz}$ against the shear strain at different times (Lissajous plots) for the two frequencies of fig. 3. We note that for the highest frequency (left panel), $\sigma_{xz}$ is in phase with the strain, indicating that the gyroid mesophase exhibits a solid-like response at short timescales. For the smaller frequency, the phase shift is almost $\pi/2$, typical of a liquid-like response as dictated by eq. (7). In fig. 5, we plot $G'(\omega \tau_r)$ and $G''(\omega \tau_r)$ for the different simulation values of $\omega$. In order to capture the relevant scales in the model, all quantities are dimensionless. In particular, the frequency $\omega$ has been multiplied by the characteristic stress relaxation time, $\tau_r \sim 220$ simulation timesteps, calculated by fitting the exponential decay of $\sigma_{xz}$ when shear is removed. We note that a crossover between $G'(\omega \tau_r)$ and $G''(\omega \tau_r)$ occurs at $\omega \tau_r \sim 0.05$. This signals a transition between solid- and liquid-like behaviour and indicates that our LB model is capable of predicting viscoelastic flow; see also fig. 1 and caption. We note that a dimensionless crossover point of $\omega \tau_r \sim 1$ consistently links rheological properties of the mixture with the relaxation time of the gyroid mesophase. This is encouraging for the use of this simulation model to inspect the correlation between macroscopic dynamics and mesoscopic
Fig. 4 – Lissajous plots of $\sigma_{xz}$ (in lattice units) vs. dimensionless strain $\gamma_{xz}$. In the left (right) panel, points tend to form a straight line (ellipse, respectively). This indicates that the pressure $\sigma_{xz}$ is in (out of) phase with respect to the strain $\gamma_{xz}$, as typical for solid (liquid)-like phases. The overall behaviour is typical of viscoelastic fluids.

Fig. 5 – Plots of shear moduli $G'(\omega \tau_r)$ (regular triangles), $G''(\omega \tau_r)$ (inverted triangles) calculated for all frequencies used in our simulations. We note that a crossover between liquid- and solid-like behaviour occurs around $\omega = 0.5$. Theoretically, $G' \sim \omega^2$, $G'' \sim \omega$ as $\omega$ goes to 0. Our data agree with this prediction as shown by the dashed lines in the plot. All quantities are dimensionless ($G_0$ is the plateau modulus for $G'$).

structure. Indeed in the case of linear viscoelasticity, $G' \sim \omega^2$, $G'' \sim \omega$ as $\omega$ goes to 0 for any fluids. In fig. 5 we plot lines $\sim \omega$ and $\sim \omega^2$ as a guide for the eye; our data show excellent quantitative agreement with theory. We interpret the elastic response component as due to the stress response to mechanical perturbations of the long-range ordered equilibrium structure of the liquid gyroid crystalline mesophase. We also tried to fit the data in fig. 5 to a simple single relaxation time (Maxwell) model of viscoelasticity but the fit is very poor. This suggests that there are several relaxation times present, as is to be expected here since the viscoelasticity arises from complex mesoscale structure. We note that the existence of a spectrum of relaxation times is consistent with the stretched-exponential behaviour of domain self-assembly in amphiphilic fluid systems found in [10].

Conclusions. – In this letter we use a purely kinetic LB model that leads to the emergence of non-trivial rheological properties. Non-Newtonian behaviour, in this case a decrease of viscosity with shear rate, has been shown for an initially self-assembled gyroid liquid crystalline mesophase. In addition, linear viscoelastic effects in the system are manifest in the simulations. It is notable that our model correctly predicts the theoretical limits for the moduli $G'(\omega)$, $G''(\omega)$ as $\omega$ goes to 0 as well as a crossover in $G'(\omega)$, $G''(\omega)$ at higher $\omega$. We note that, unlike previous approaches, this model does not require any assumptions at the macroscopic level, that is it provides a purely kinetic-theoretical approach to the description of complex fluids. This model can therefore help in understanding complex flows, such as flow of viscoelastic liquids in porous media, colloidal fluids and polymer melts. Work is in progress to assess the generality of this approach within our lattice-Boltzmann model of amphiphilic fluids. In particular, such simulations should enable us to investigate the link between mesoscopic structure and macroscopic dynamics, for example by correlating rheological relaxation times to mesoscale relaxation processes within the amphiphilic fluid.
We would like to thank D. M. A. Buzza for a critical reading of this manuscript, as well as J. Chin and N. González-Segredo for helpful discussions. This research is funded by EPSRC under the RealityGrid grant GR/R67699 which also provided access to UK national supercomputing facilities at CSAR and HCPx. Access to the US TeraGrid and Lemieux at Pittsburgh Supercomputing centre was provided through the National Science Foundation (USA) NRAC and PACS grants MCA04N014 and ASC030006P, respectively. Intercontinental data transfer was effected via UKLight within EPSRC grant GR/T04465. GG would like to acknowledge the support of the European Commission’s Research Infrastructure Activity, contract number 506079 (HPC Europa).

REFERENCES

Structural transitions and arrest of domain growth in sheared binary immiscible fluids and microemulsions

J. Harting, G. Giupponi, and P. V. Coveney
Structural transitions and arrest of domain growth in sheared binary immiscible fluids and microemulsions

J. Harting,1 G. Giupponi,2 and P. V. Coveney2

1Institute for Computational Physics, University of Stuttgart, Pfaffenwaldring 27, 70569 Stuttgart, Germany
2Centre for Computational Science, Department of Chemistry, University College London, 20 Gordon Street, WC1H 0AJ London, United Kingdom

(Received 15 January 2007; published 26 April 2007; publisher error corrected 27 April 2007)

We investigate spinodal decomposition and structuring effects in binary immiscible and ternary amphiphilic fluid mixtures under shear by means of three-dimensional lattice Boltzmann simulations. We show that the growth of individual fluid domains can be arrested by adding surfactant to the system, thus forming a bicontinuous microemulsion. We demonstrate that the maximum domain size and the time of arrest depend linearly on the concentration of amphiphile molecules. In addition, we find that for a well-defined threshold value of amphiphile concentration, the maximum domain size and time of complete arrest do not change. For systems under constant and oscillatory shear we analyze domain growth rates in directions parallel and perpendicular to the applied shear. We find a structural transition from a sponge to a lamellar phase by applying a constant shear and the occurrence of tubular structures under oscillatory shear. The size of the resulting lamellae and tubes depends strongly on the amphiphile concentration, shear rate, and shear frequency.

DOI: 10.1103/PhysRevE.75.041504 PACS number(s): 64.75.+g, 47.11.–j, 82.70.Uv, 83.60.–a

I. INTRODUCTION

Complex fluid mixtures consisting of immiscible fluid species and surfactants are ubiquitous in many industrial applications. These fluid mixtures involve both hydrodynamic flow effects and complex interactions between fluid particles. Typical examples can be found in the food, cosmetic, and chemical industries where surfactants are utilized to stabilize otherwise immiscible fluids. A good example is a barbecue sauce containing large fractions of water and oil or fat. Without any additives the constituents would phase separate, entering a not very appealing demixed state in which the fat accumulates in a thick layer on top of the remainder. Adding an emulsifier or surfactant helps to stabilize the sauce. These molecules are often called amphiphiles and, in their simplest form, are comprised of a hydrophilic (water-loving) head group and a hydrophobic (oil-loving) tail. The surfactant molecules self-assemble on the surface of oil droplets and reduce the surface tension. Thus, the droplets stabilize and remain suspended within the bulk water. A typical emulsifier used by the food industry is egg yolk lecithin. Proteins and emulsifiers with low molecular weight are also common emulsifiers.

Amphiphilic fluids containing at least one species made of surfactant molecules are in general complex fluids that can self-assemble to form ordered structures such as lamellae, micelles, not ordered sponge phases and liquid crystalline (cubic) phases. In general, adding amphiphiles to a binary mixture of otherwise immiscible fluids (for example oil and water) that is undergoing phase separation can cause the demixing process to slow down. If the amphiphile concentration is sufficiently high, the demixing process might eventually arrest completely. It has been shown by Langevin, molecular dynamics, lattice gas, and lattice Boltzmann simulations that the temporal growth law for the size of oil and water domains in a system without amphiphiles follows a power law \( r^n \) [1,2] and crosses over to a logarithmic growth law \( \ln(t)^\theta \), where \( \alpha \), \( \theta \) are fitting parameters and \( t \) is the time [3–5]. A further increase of the surfactant concentration can lead to growth which is well described by a stretched exponential form \( A - B \exp(-Ct^\gamma) \), where capital letters denote fitting parameters [4,5]. By adjusting temperature, fluid composition, or pressure, amphiphiles can self-assemble and force the fluid mixture into a number of equilibrium structures or mesophases. These include lamellae and hexagonally packed cylinders, micellar, primitive, diamond, or gyroid cubic mesophases as well as sponge phases. In this paper we focus on the sponge mesophase, which in the context of our simulations is called a bicontinuous microemulsion since it is formed by the amphiphilic stabilization of a phase-separating binary mixture, where the immiscible fluid constituents occur in equal proportions. Here, the oil and water phases interpenetrate and percolate and are separated by a monolayer of surfactant at the interface.

Such complex fluids are often subject to shear forces and show pronounced rheological properties [6,7]. Often, shear-induced transitions from isotropic to lamellar phases can be observed. These have been studied experimentally in binary [8] and ternary amphiphilic fluids [9,10]. If oscillatory shear is applied, a further transition to a tubular phase or a transition between differently oriented lamellar phases can occur [11–14].

Computationally, such complex fluids are too large and expensive to tackle with atomistic methods such as molecular dynamics, yet they require too much molecular detail for continuum Navier-Stokes approaches. Algorithms that work at an intermediate or “mesoscale” level of description have been developed during the last twenty years, including dissipative particle dynamics [15–17], lattice gas cellular automata [18], the stochastic rotation dynamics [19–21], and the lattice Boltzmann equation [22–24]. In particular, the lattice Boltzmann method has been found highly useful for simulating complex fluid flows in a wide variety of systems. This algorithm, described in more detail in the next section,
is extremely well suited to implementation on parallel computers, which permits very large systems to be simulated, reaching hitherto inaccessible physical regimes.

In this paper we investigate spinodal decomposition and structuring effects in binary immiscible and ternary amphiphilic fluid mixtures under shear by means of large scale three dimensional lattice Boltzmann (LB) simulations. The purely kinetic LB method we use is able to model complex flows whose rheological properties are emergent from the mesoscopic kinetic processes without any imposed macroscopic constraints [25].

Varieties of the lattice Boltzmann method have been used successfully to study the behavior of multiphase flows in the past. A number of authors have investigated spinodal decomposition [26–36] and the same phenomenon has also attracted some interest in the presence of shear, where structural transitions from isotropic to lamellar or tubular phases may occur [14,37–40].

There have been only limited investigations of the influence of amphiphiles on the domain growth within the lattice Boltzmann method, despite the fact that ternary amphiphilic fluids have been studied by a number of authors [24,41–43]. For example, it has been shown that the lattice Boltzmann method can be used to describe the self-assembly and the rheological properties of mesophases including the primitive $P$ phase [44] and the gyroid phase [45]. The gyroid mesophase in particular has been of major interest during the last few years, where the phase formation and structural properties [5,46], the influence of defects [47–50], as well as its properties under shear [45,51] have been investigated.

In this paper, our aim is to focus on the effect of surfactant concentration on the length and time scales of arrested growth and on the changes in structural properties induced by steady or oscillatory shear.

The remainder of the paper is organized as follows. After an introduction to the simulation method and the fluid parameters used, we present results from the simulation of ternary amphiphilic systems with varying surfactant density. Here, we draw on the results of González-Segredo et al. [5] for the simulation parameters used within the present work. Thereafter, we extend our study to systems under constant and oscillatory shear, where we report on the influence of the domain growth rates and structural transitions induced by the shear. Finally, a summary and conclusions are given.

II. SIMULATION METHOD

A set of equations can be used to represent a standard LB system involving multiple species [52]

\[ n_i^\alpha(x + \mathbf{c}_i, t + 1) - n_i^\alpha(x, t) = \Omega_i^\alpha \]

with $i=0,1,\ldots,b$. The single-particle distribution function $n_i^\alpha(x, t)$ indicates the density of species $\alpha$, having velocity $\mathbf{c}_i$, at site $x$ on a $D$-dimensional lattice of coordination number $b$, at time step $t$. The collision operator

\[ \Omega_i^\alpha = -\frac{1}{2} [n_i^\alpha(x, t) - n_i^{\alpha\text{eq}}(x, t)] \quad (2) \]

represents the change in the single-particle distribution function due to the collisions. A popular form is the single relaxation time $\tau_{\alpha}$ linear Bhatnagar-Gross-Krook (BGK) form [53] for the collision operator. It can be shown for low Mach numbers that the LB equations correspond to a solution of the Navier-Stokes equation for isothermal, quasi-incompressible fluid flow. The lattice Boltzmann method is an excellent candidate to exploit the possibilities of parallel computers, as the dynamics at a lattice site requires only information about quantities at nearest neighbor lattice sites [24,49]. The local equilibrium distribution $n_i^{\alpha\text{eq}}$ plays a fundamental role in the dynamics of the system as shown by Eq. (1). In this study, we use a purely kinetic approach, for which the local equilibrium distribution $n_i^{\alpha\text{eq}}$ is derived by imposing certain restrictions on the microscopic processes, such as explicit mass and total momentum conservation [54]

\[ n_i^{\alpha\text{eq}} = \rho^\alpha \left[ 1 + \left( \frac{\mathbf{c}_i u + (\mathbf{c}_i u)^2}{2c_i^4} - \frac{u^2}{2c_i^4} + \left( \frac{\mathbf{c}_i u}{6c_i^6} - \frac{u^2(c_i u)}{2c_i^4} \right) \right), \right] \quad (3) \]

where $\rho^\alpha(x, t) = \sum_i n_i^\alpha(x, t)$ is the fluid density and $\mathbf{u} = \mathbf{u}(x, t)$ is the macroscopic bulk velocity of the fluid, given by $\rho^\alpha(x, t) \mathbf{u} = \sum_i n_i^\alpha(x, t) \mathbf{c}_i$. $c_i$ are the coefficients resulting from the velocity space discretization and $c_i$ is the speed of sound, both of which are determined by the choice of the lattice. We use a D3Q19 implementation, i.e., a three-dimensional lattice with 19 discrete velocities. Immiscibility of species $\alpha$ is introduced in the model following Shan and Chen [55,56], where only nearest neighbor interactions among the species are considered. These interactions are described by a self-consistently generated mean field body force

\[ \mathbf{F}^\alpha(x, t) = -\psi^\alpha(x, t) \sum_{\alpha'} g_{\alpha \alpha'} \sum_{x'} \psi^\alpha(x', t)(x' - x), \quad (4) \]

where $\psi^\alpha(x, t)$ is the so-called effective mass, which can have a general form for modeling various types of fluids [we use $\psi^\alpha = (1 - \epsilon^\alpha)^p$ [55]], and $g_{\alpha \alpha'}$ is a force coupling constant whose magnitude controls the strength of the interaction between components $\alpha$ and $\alpha'$ and is set positive to mimic repulsion. The dynamical effect of the force is realized in the BGK collision operator by adding to the velocity $\mathbf{u}$ in the equilibrium distribution of Eq. (3) the increment

\[ \partial_t \mathbf{u}^\alpha = \frac{\mathbf{F}^\alpha}{\rho^\alpha}, \quad (5) \]

Amphiphiles are introduced within the model as described in [25] and [43]. An amphiphile usually possesses two different fragments, each having an affinity for one of the two immiscible components. The orientation of any amphiphile present at a lattice site $x$ is represented by an average dipole vector $\mathbf{d}(x, t)$. Its direction is allowed to vary continuously and no information is specified for each velocity $\mathbf{c}_i$, for reasons of computational efficiency and simplicity. The am-

\[ \Omega_i^\alpha = -\frac{1}{2} [n_i^\alpha(x, t) - n_i^{\alpha\text{eq}}(x, t)] \quad (2) \]
phiphile density at a given site is given by an additional fluid species with density \( \rho' \). The model has been used successfully to study spinodal decomposition [2,27], the formation of mesophases [5,44–46,48,51,57], and flow in porous media [39].

We use LBM [49], a highly scalable parallel LB code, to implement the model. The very good scaling of our code permits us to run all our simulations on multiprocessor machines and computational grids in order to reduce the length of data collection to a few weeks. Also, we are able to use simulation boxes typically eight times bigger than previous studies so as to minimize finite size effects.

In order to study the rheological behavior of complex fluid mixtures, we have implemented Lees-Edwards boundary conditions, which were originally developed for molecular dynamics simulations [58]. They reduce finite size effects as compared to moving solid walls [58] and have been used in lattice Boltzmann simulations by various authors [37,38,40,59]. This computationally convenient method imposes new positions and velocities on particles leaving the simulation box in the direction perpendicular to the imposed shear strain while leaving the other coordinates unchanged. Choosing \( z \) as the direction of shear and \( x \) as the direction of the velocity gradient, we have

\[
\begin{align*}
 z' &= \begin{cases} 
 (z + \Delta z) \text{ mod } N_z, & x > N_x, \\
 z \text{ mod } N_z, & 0 \leq x \leq N_x, \\
 (z - \Delta z) \text{ mod } N_z, & x < 0,
\end{cases} \\
 u'_x &= \begin{cases} 
 u_x + U, & x > N_x, \\
 u_x, & 0 \leq x \leq N_x, \\
 u_x - U, & x < 0,
\end{cases}
\end{align*}
\]

where \( \Delta z = U \Delta t \), \( U \) is the shear velocity, \( u_x \) is the component of \( u \), and \( N_z(z) \) is the system length in the \( z \) direction. We also use an interpolation scheme suggested by Wagner and Pagonabarraga [38] as \( \Delta z \) is not generally a multiple of the nearest neighbor lattice distance. For oscillatory shear, we set

\[
U(t) = U \cos(\omega t),
\]

where \( \omega/2 \pi \) is the frequency of oscillation.

In nonsheared studies of spinodal decomposition it has been shown that large lattices are needed to overcome finite size effects. There, 128\(^3\) was the minimum acceptable number of lattice sites [2]. More quantitatively, Kendon et al. [33] set \( L/4 \) as the maximum length scale, which is not affected by finite size effects in their spinodal decomposition simulation, where \( L \) is the length of the simulation box. We therefore choose 256\(^3\) for all nonsheared simulations to limit the influence of finite size effects even further. For high shear rates, systems also have to be highly extended in the direction of the applied shear because, if the system is too small, the domains interconnect across the \( z=0 \) and \( z=N_z \) boundaries to form continuous lamellae in the direction of shear [39,49]. Such artifacts need to be eliminated from our simulations. In this case, a good compromise to limit finite size effects and to keep the computational expense as low as possible is a lattice size of 128 × 128 × 512 and this is used here. Mass and relaxation times are always set to unity, i.e., \( \tau^m = 1.0 \), \( \tau^n = 1.0 \). We call the two immiscible fluids “red” and “blue” and set their initial densities to identical values, \( \rho' = \rho^r \). The initial average surfactant density \( \rho^s \) is varied between 0.0 and 0.7. The lattice is then randomly populated with constant initial total fluid densities \( \rho'^{0} = \rho^r + \rho^s + \rho^b \). This is in contrast to previous studies where only \( \rho^r + \rho^b \) was kept constant [5]. The coupling constant in Eq. (4) between red and blue species is set to \( g_{rs} = 0.08 \), the coupling between an immiscible fluid and surfactant to \( g_{rs} = -0.006 \), and the constant describing the strength of the surfactant-surfactant interaction is kept at \( g_{ss} = -0.003 \). All units in this paper are given in lattice units if not stated otherwise. While our method has been used to simulate other mesophases like lamellar phases, the primitive P phase [44], and the gyroid phase [5,46], the parameters used in all simulations presented here are known to produce a sponge phase in the absence of bulk flow. More detailed investigations of the particular choice of coupling constants and how they modify the system’s properties are given in [2,3,45,46,51].

To analyze the behavior of the various simulations, we define the time-dependent lateral domain size \( L(t) \) along direction \( i=x,y,z \) as

\[
L_i(t) = \frac{2\pi}{\sqrt{\langle k_i^2(t) \rangle}},
\]

where

\[
\langle k_i^2(t) \rangle = \frac{\sum_k k_i^2 S(k,t)}{\sum_k S(k,t)}
\]

is the second-order moment of the three-dimensional structure function

\[
S(k,t) = \frac{1}{V} |\psi_k(t)|^2
\]

with respect to the Cartesian component \( i \), \( \langle \cdot \rangle \) denotes the average in Fourier space, weighted by \( S(k,t) \), and \( V \) is the number of nodes of the lattice, \( \psi_k(t) \) is the Fourier transform of the fluctuations of the order parameter \( \phi' = \phi - \langle \phi \rangle \), and \( k_i \) is the \( i \)th component of the wave vector. A projection of the structure function allows us to compare simulation data to scattering patterns obtained in experiments. We obtain those projections by summing up \( S(k,t) \) in one of the Cartesian directions. For example, for the projection in the \( z \) direction this leads to \( S_{\perp}(k_x,k_y,t) = \sum_{k_z} S(k_x,k_y,t) \).

III. RESULTS

A. Ternary amphiphilic systems without shear

Spinodal decomposition of a binary immiscible fluid mixture has been studied extensively within our model by González-Segredo et al. [2]. The authors report domain size scaling as expected for a crossover from diffusive behavior to hydrodynamic viscous growth, i.e., the domain size grows
as $L \times t$, with $\gamma$ being between $1/3$ and 1. Moreover, they find very good agreement with the dynamical scaling hypothesis, recovering the expected universal behavior of the structure function.

If one adds surfactant to a binary immiscible fluid mixture, the surfactant molecules self-assemble at the interface between the two species and slow down the phase-separation process. For sufficiently high surfactant concentrations, domain growth is arrested completely, leading to a stable microemulsion. In [5], González-Segredo et al. extend their work to ternary amphiphilic fluids and study how the phase separation of a binary immiscible fluid mixture is altered by the presence of surfactant. As already described in the Introduction, by gradually increasing the surfactant concentration they find a slowing down of the domain growth, initially from algebraic to logarithmic temporal dependence, and, at higher surfactant concentrations, from logarithmic to stretched-exponential behavior. They also observe a structural transition from sponge to gyroid phases by increasing the amphiphile concentration or varying the amphiphile-amphiphile or amphiphile-binary fluid coupling constants. For growth-arrested mesophases, they observe temporal oscillations of the domain size due to Marangoni flows.

In the present work we use simulation parameters that differ from those of previous studies and are known to produce a sponge phase. In order to avoid effects due to variations of the fluid densities, we also keep the total density in the system constant at 1.6 (in lattice units), while varying the surfactant densities $\rho^s$ between 0.00 and 0.70. Furthermore, our simulation lattices are up to eight times larger in order to keep the influence of finite size effects to a minimum, and we simulate up to 30,000 time steps in order to gain a better understanding of the long time behavior of the system. We study the influence of the amphiphile concentration on the phase-separation process in this section and reproduce previous results with the present parameters. In addition, we study the dependence of the maximum achievable domain size in a stable microemulsion on surfactant concentration as well as the time needed to achieve this state.

To depict the influence of the surfactant density on the phase-separation process, Fig. 1 shows three volume rendered 256$^3$ systems at $t=30,000$ for surfactant densities $\rho^s=0.00$, 0.15, 0.30 (from left to right). For better visibility only one of the immiscible fluid species is shown. Different colors denote the interface and areas of high density of the visualized fluid. The surfactant particles (not shown) are aligned at the interfaces and the second immiscible fluid component fills the void space. After 30,000 time steps the phases have separated to a large extent if no surfactant is present (left). Adding a small amount of surfactant ($\rho^s=0.15$, center) causes the domains to grow more slowly, as depicted by the smaller structures in the volume rendered image. For sufficiently high amphiphile concentrations ($\rho^s=0.30$, right) the growth process arrests with the formation of a stable bicontinuous microemulsion.

![Volume rendered fluid densities of 256$^3$ systems at $t=30,000$ for surfactant densities $\rho^s=0.00$, 0.15, 0.30 (from left to right). For better visibility only one of the immiscible fluid species is shown. Different colors denote the interface and areas of high density of the visualized fluid. The surfactant particles (not shown) are aligned at the interfaces and the second immiscible fluid component fills the void space. After 30,000 time steps the phases have separated to a large extent if no surfactant is present (left). Adding a small amount of surfactant ($\rho^s=0.15$, center) causes the domains to grow more slowly, as depicted by the smaller structures in the volume rendered image. For sufficiently high amphiphile concentrations ($\rho^s=0.30$, right) the growth process arrests with the formation of a stable bicontinuous microemulsion.](image)

For sufficiently high amphiphile concentrations, the phase-separation process, Fig. 1 shows three volume rendered 256$^3$ systems at $t=30,000$ for surfactant densities $\rho^s=0.00$, 0.15, 0.30 (from left to right). For better visibility only one of the immiscible fluid species is shown. Different colors denote the interface and areas of high density of the visualized fluid. The surfactant particles (not shown) are aligned at the interfaces and the second immiscible fluid component fills the void space. After 30,000 time steps the phases have separated to a large extent if no surfactant is present (left). Adding a small amount of surfactant ($\rho^s=0.15$, center) causes the domains to grow more slowly, as depicted by the smaller structures in the volume rendered image. For sufficiently high amphiphile concentrations ($\rho^s=0.30$, right) the growth process arrests with the formation of a stable bicontinuous microemulsion.

In the present work we use simulation parameters that differ from those of previous studies and are known to produce a sponge phase. In order to avoid effects due to variations of the fluid densities, we also keep the total density in the system constant at 1.6 (in lattice units), while varying the surfactant densities $\rho^s$ between 0.00 and 0.70. Furthermore, our simulation lattices are up to eight times larger in order to keep the influence of finite size effects to a minimum, and we simulate for up to 30,000 time steps in order to gain a better understanding of the long time behavior of the system. We study the influence of the amphiphile concentration on the phase-separation process in this section and reproduce previous results with the present parameters. In addition, we study the dependence of the maximum achievable domain size in a stable microemulsion on surfactant concentration as well as the time needed to achieve this state.

To depict the influence of the surfactant density on the phase-separation process, Fig. 1 shows three volume rendered 256$^3$ systems at surfactant densities 0.0 (left), 0.15 (center), and 0.3 (right). As in all figures throughout the paper, for better visibility only one of the immiscible fluid species is shown. Different colors denote the interface and areas of high density of the rendered fluid. The surfactant particles are aligned at the interfaces and the second immiscible constituent fills the void space. After 30,000 time steps the phases have separated to a large extent when no surfactant is present (left). Running the simulation for even longer would result in two perfectly separated phases, each of them contained in a single domain only. If one adds some surfactant ($\rho^s=0.15$, center), the domains grow more slowly, visualized by the smaller structures in the volume rendered image. For sufficiently high amphiphile concentrations ($\rho^s=0.30$, right) the growth process arrests leading to a stable bicontinuous microemulsion with small individual domains formed by the two immiscible fluids.

The projected structure function $S(k_x,k_y,t)$ (“scattering pattern”) is given in Fig. 2 for two surfactant densities $\rho^s=0.00$ (a) and 0.30 (b) at time step $t=10,000$. As can be clearly seen in Fig. 2(a), a strong peak occurs for small values of $k_x$, $k_y$, depicting the occurrence of length scales that are of the order of the system size. For $\rho^s=0.30$, however, the peaks are by a factor of 100 smaller and shifted to larger values of $k_x$, $k_y$. We find a volcano-like scattering pattern, indicating the dominance of small length scales. Since our system is cubic and no shear is applied, the projections of the structure function in $x$ and $y$ directions are equivalent.

To investigate the influence of surfactant more quantitatively, in Fig. 3 the time-dependent lateral domain size $L(t)$ as given in Eq. (9) is shown for a number of surfactant densities $\rho^s$ between 0.00 and 0.50. Since the lattice is cubic here, $L(t)$ behaves identically in all three directions. Figures 3(a) and 3(b) show identical data, but different scalings of the axes. In Fig. 3(a), we plot the data linearly in order to give a better impression of the time dependence of the growth dynamics. However, in order to check which data are best fitted by the various growth laws, we provide a log-log scale plot of the same data in Fig. 3(b). For the first few hundred time steps, the randomly distributed fluid densities

---

**FIG. 1.** (Color online) Volume rendered fluid densities of 256$^3$ systems at $t=30,000$ for surfactant densities $\rho^s=0.00$, 0.15, 0.30 (from left to right). For better visibility only one of the immiscible fluid species is shown. Different colors denote the interface and areas of high density of the visualized fluid. The surfactant particles (not shown) are aligned at the interfaces and the second immiscible fluid component fills the void space. After 30,000 time steps the phases have separated to a large extent if no surfactant is present (left). Adding a small amount of surfactant ($\rho^s=0.15$, center) causes the domains to grow more slowly, as depicted by the smaller structures in the volume rendered image. For sufficiently high amphiphile concentrations ($\rho^s=0.30$, right) the growth process arrests with the formation of a stable bicontinuous microemulsion.
FIG. 2. (Color online) Projected structure function $S_i(k_x,k_y,t)$ for $\rho' = (a) 0.00$ and (b) 0.30 at time step $t=10,000$. For the case without surfactant, a strong peak occurs for small values of $k_x,k_y$, reflecting the dominance of length scales that are of the order of the system size. For $\rho'=0.30$, only much smaller peaks occur for larger values of $k_x,k_y$, indicating that only small length scales are present. All quantities are expressed in lattice units.

FIG. 3. (Color online) Average domain size $L(t)$ for various surfactant densities $\rho'=0.00, 0.10, 0.15, 0.20, 0.25, 0.30, 0.35, 0.40, 0.50$. Axes are linear in (a) and logarithmic in (b). After the initial spontaneous formation of domains, domain growth can be described by a power law. With increasing $\rho'$ domain growth slows down and eventually comes to an end at a maximum domain size $L_{\text{max}}(\rho')$. All quantities are expressed in lattice units.

FIG. 4. (Color online) (a) Maximum domain size $L_{\text{max}}(\rho')$ and (b) time of arrest $t_{\text{arrest}}(\rho')$ for various surfactant densities $\rho'=0.25, 0.30, 0.35, 0.40, 0.50, 0.7$. $L_{\text{max}}(\rho')$ as well as $t_{\text{arrest}}(\rho')$ decrease linearly with increasing surfactant density $\rho'$ and saturate for $\rho'>0.5$. All quantities are expressed in lattice units.

can slow down the growth process, and for high surfactant densities $\rho'>0.25$, the domain growth stops after a few thousand simulation time steps. By adding even more surfactant, the final average domain size becomes very small and does not grow beyond 7.7 lattice sites. We fit our numerical data with the corresponding growth laws and find that for $\rho'$ smaller than 0.15 $L(t)$ is best fitted by a function proportional to $t^a$. For $\rho'$ being 0.15 or 0.20, a logarithmic behavior proportional to $(\ln t)^b$ is observed. Increasing $\rho'$ further results in $L(t)$ being best described by a stretched exponential. These results correspond well with the findings in [5].

We study the dependence of the final domain size $L_{\text{max}}(\rho')$ on the amount of surfactant as depicted in Fig. 4(a). It can be observed that the maximum domain size decreases linearly from 20.9 for $\rho'=0.25$ with increasing $\rho'$ until a threshold value is reached at $\rho'=0.5$, where $L_{\text{max}}(t)=7.7$. Then, $L_{\text{max}}(\rho')$ decreases much more slowly and stays almost constant. The slope of the linear regime corresponds to $-52.8$. The behavior of $L_{\text{max}}(\rho')$ and $t_{\text{arrest}}(\rho')$ is consistent with previous lattice gas [4,60] and lattice Boltzmann studies [5], where the authors determined the dependence of the surface tension at a planar interface between two immiscible fluid species on the surfactant concentration. In [5], the authors found a linear dependence between surface tension and surfactant density, but they did not study such high concentrations as in the current study. In [60], the surface tension approaches zero for high concentrations, i.e., a saturation occurs, causing the size of the individual fluid domains to saturate as well. These effects can be explained as follows. Adding surfactant to a binary fluid mixture causes the amphiphiles to minimize the free energy in the system by assembling at the interface between the two immiscible fluid species. An increase of surfactant concentration causes the interfacial area to be maximized in order to accommodate as much surfactant as possible. The increasing interfacial area causes the individual domains to become smaller and
\( L_{\text{max}}(\rho') \) decreases. If the surfactant concentration becomes very high (\( \rho' > 0.5 \) in our case), \( L_{\text{max}}(\rho') \) saturates due to the maximum possible interfacial area being reached and all available area being covered with surfactant molecules. More amphiphiles accumulating at the interface would lead to very steep and energetically unfavorable gradients of surfactant density in the system. Therefore, further amphiphiles have to reside within the bulk fluid phases, forming micellar structures. Within our model the minimum final domain size corresponds to 7.7 lattice sites. However, this value can be varied by tuning the coupling constants for the amphiphile-amphiphile or amphiphile-fluid interactions. A more thorough investigation of the influence of the particular choice of the coupling constants on the final domain size is of particular interest and will be investigated within a future project.

In Fig. 4(b), the number of simulation time steps \( t_{\text{arrest}}(\rho') \) needed to reach the final domain size is plotted. Since the time it takes for the system to relax to its equilibrium state directly depends on the final domain size, it is consistent with the data presented in Fig. 4(a) that a linear dependence of \( t_{\text{arrest}}(\rho') \) on the surfactant concentration can be observed. While for \( \rho'=0.25 \) 7000 time steps are needed to reach the maximum possible domain size, for \( \rho'=0.5 \) 500 time steps are sufficient. For \( \rho'>0.5 \), \( t_{\text{arrest}}(\rho') \) decreases much more slowly than for \( \rho'<0.5 \). The slope of \( t_{\text{arrest}}(\rho') \) in the linear regime is given by \(-26\,000\).

### B. Steadily sheared systems

If a binary immiscible fluid mixture is driven mechanically by external shear forces, it is known that the evolution of domains and phase-separation processes is changed profoundly [8,45,51]. The most noticeable effect is the formation of a lamellar phase, i.e., elongated domains or lamellae form and align along the flow direction. Due to the anisotropy of the system, the time-dependent domain size \( L(t) \) behaves differently for the different coordinate axes in this case. Furthermore, modified growth exponents are expected due to the anisotropic effects.

As already seen in the previous section, adding amphiphiles to a binary immiscible fluid under shear can change its properties dramatically. The amphiphiles stabilize the interface between the immiscible fluid species and the domain growth is hindered as described in the previous section. Moreover, the amphiphiles might form complex structures which can have an impact on the properties of the sheared fluid leading to non-Newtonian flow [8,45,51].

We study ternary 128×128×512 sized systems under constant shear. The shear rate is set to \( \gamma = 1.56 \times 10^{-3} \) and \( 3.12 \times 10^{-3} \), while the surfactant density is varied between \( \rho'=0.0 \) and 0.4. Figure 5 shows the time-dependent lateral domain size for all three coordinate axes at \( \gamma = 1.56 \times 10^{-3} \).

In the \( x \) direction, which is the axis between the shear planes, the power law regime of \( L_x(t) \) starts at \( t=500 \) for the \( \rho'=0.0 \) case, while for higher \( \rho' \) the initial growth regime is overcome by the power law regime before the first measurement at \( t=100 \). As long as \( \rho'<0.3 \), the growth rate is not hindered by the amphiphiles and domains grow until the end of the simulation. For \( \rho'=0.3 \) the power law regime starts to break down at \( t=900 \) and \( L_x(t) \) saturates at \( t=5000 \). Adding even more surfactant results in an even earlier saturation at \( t=1500 \). The \( y \) direction is the direction parallel to the shear planes and perpendicular to the direction of shear. Since this direction is less affected by the shear forces, \( L_y(t) \) grows faster than \( L_x(t) \) for low surfactant concentrations \( \rho'<0.2 \), causing the domains forming to be extended in the \( y \) direction. For \( \rho'=0.2 \), \( L_y(t) \) and \( L_z(t) \) behave almost identically, while for \( \rho'>0.2 \) a crossover occurs and the maximum attainable value for \( L_x(t) \) is below the result for \( L_z(t) \). In the direction of shear (\( z \) direction), \( L_z(t) \) saturates even for the no surfactant case at \( L_z(t)=25 \) and comes to arrest at even smaller values with increasing \( \rho' \). The complex behavior of \( L_z(t) \) can be better understood by reminding ourselves that the domain size is measured in the direction of the Cartesian coordinate axes. However, individual fluid domains occurring in the system are being elongated due to the shear and try to align with the shear gradient. Thus, they are not parallel to any coordinate axis. Therefore, with a measurement of \( L_z(t) \) one is not able to detect the actual length of individual lamellae, but only their thickness in the \( z \) direction. Similar arguments are valid for \( L_y(t) \), shear causing the measured domain size in the \( x \) direction to be larger than the lamellae’s thickness. For increasing \( \rho' \), the average domain size reduces due to the influence of the amphiphiles, thus causing the individual domains to become smaller. If \( \rho'>0.2 \), the alignment of the domains with the shear causes \( L_z(t) \) to appear larger than \( L_y(t) \). For high surfactant concentrations (\( \rho'=0.4 \)) all three directions behave very similarly: domain growth comes to an end after fewer than 2000 time steps and the final domain size is between 10 and 15 lattice units.
units in all directions, signaling the appearance of a stable microemulsion where the shape of the domains is almost unaffected by the shear. Regular peaks occur in $L_x(t)$ at every 2500 time steps with less pronounced peaks in between them. These peaks can be explained as follows. For the stretching of domains, a certain amount of work against surface tension is needed. On macroscopic scales, the stress tensor does not vanish due to the viscoelastic response of the system [61,62]. On the microscale, however, a breakup and recombination of domains can be observed [63]. These domains grow by diffusion and eventually join each other to form larger structures. If the internal stress becomes too large due to the shear-induced deformation, they break up and start to form again. Assuming a large system with many independent domains growing and breaking incoherently, the only observable effect might be a slowing down of the domain growth. In contrast, if the growth and breakup occur coherently as they do in our simulations, a periodicity in the measured time-dependent domain size can be observed [64]. As can be observed in Fig. 5(e), the frequency of domain breakup is independent of the surfactant concentration, while the height of the peaks decreases with increasing $\rho'$.

Figure 6 shows volume rendered examples of a simulated system with surfactant density $\rho'=0.2$ and a constant shear rate of $\dot{\gamma}=1.56 \times 10^{-3}$, and variable number of time steps $t=1000$ (upper left), $4000$ (upper right), $6000$ (lower left), and $10,000$ (lower right). While the shape of individual domains does not show distinct features at early times of the simulation, elongated structures appear at $t=4000$ and start to become aligned with the shear at $t=6000$. At late stages of the simulation run ($t=10,000$), the lamellar phase characterized by thin and long lamellae filling the whole system can be observed.

and a surfactant density of $\rho'=0.3$ [Figs. 7(d)–7(f)] at $t = 10,000$. The $x$, $y$, and $z$ directions are shown from top to bottom. The shown projections are for a cubic 128$^3$ cutout of the elongated systems. In contrast to the nonsheared case, all three directions show distinct properties. For $\rho'=0.00$, a high peak of $S(k_x, k_y, t)$ can be observed around $k_x=k_y=0$, while in the $x$ and $y$ directions two lower peaks at positions above zero show up. These data can be interpreted as follows. At $t=10,000$, the domain size in the direction of the flow corresponds to the size of the cubic cutout, i.e., 128 lattice sites. In the $x$ and $y$ directions, however, the size of the occurring structures is smaller, indicating the occurrence of very long lamellar structures in the system. By adding surfactant to the system [Figs. 7(d)–7(f)], the occurring length scales depicted decrease by the splitting of the single peak in the $z$ projection and the occurrence of two small peaks at $k_x=0$ and $k_y = \pm 10$. While the peaks in the $y$ direction denote similar length scales as in the $z$ direction, the projections in the direction between the shear planes ($x$) show a different behavior. Here, two parallel structures at $k_x=\pm 10$ and $k_y$ between $-20$ and $20$ indicate a much wider variation of the thickness of the individual domains. This is in contrast to the non-sheared case in Fig. 2, where a volacano-like shape of the structure factor was observed.

Doubling the shear rate to $\dot{\gamma}=3.12 \times 10^{-3}$ results in very similar behavior, as shown in Fig. 8. In the $z$ direction, peaks can now be observed even for $\rho'=0.4$, but $L_x(t)$ is much noisier for lower surfactant concentrations. However, it can still be seen that there is a number of equidistant peaks for $\rho'<0.4$ which occur every 2500 time steps with some additional peaks in between in the case of $\rho'=0.0$ and 0.1. The equidistant peaks occurring with the same frequency as in the $\dot{\gamma}=1.56 \times 10^{-3}$ case shows that the breakup phenomena observed are independent of the shear rate.

A number of experiments have reported that the shear stabilizes the system and causes the phase separation to come to an end with $L_x(t)$ being very large and $L_y(t)$ being much smaller [30,40,65,66]. We are not able to reproduce these results due to the limited size of our simulations: substantially larger systems and higher shear rates need to be studied in order to quantify the arrest of domain growth due to shear. However, the computational resources needed would be at the limit of what can be done on current supercomputers.

We have shown in this section that the dynamical scaling hypothesis does not hold for sheared ternary systems in three dimensions since we indeed find three individual length scales pointing out the transition from the sponge to the lamellar phase: while in the flow direction ($z$), $L(t)$ is determined by the resultant length of the occurring lamellae, in the direction between the shear planes ($x$), the domains grow steadily and exhibit power law behavior up to a maximum that depends on the surfactant concentration. In the $y$ direction, domain growth is not hindered by shear. In fact, $L_y(t)$ grows slightly faster than in the nonsheared case. Increasing the surfactant concentration has a strong impact on domain growth: starting at $\rho'=0.3$, $L_x(t)$ and $L_y(t)$ recover the behav-
ior of the case without shear, i.e., the length scales saturate around 15. In the $x$ direction, however, growth continues up to $L_x(t)/H = 26$. This can be explained as follows. With increasing surfactant concentration, the final domain sizes become smaller, reducing the influence of the shear forces in the $y$ and $z$ directions. In the direction between the shear planes, however, an increase of $L_x(t)/H$ can still be observed because the domains are still being elongated due to shear and try to align with the velocity profile. Thus, they are tilted and their size appears to be smaller than it actually is in the $z$ direction and larger in the $x$ direction.

Our findings are in agreement with Ginzburg-Landau and Langevin calculations as well as two-dimensional lattice Boltzmann simulations of binary immiscible fluid mixtures as presented in [31,36,40]. However, to the best of our knowledge, there are no detailed theoretical studies of the dependence of domain growth properties on the surfactant concentration. The only known work utilizes a Ginzburg-Landau free-energy approach to study sheared microemulsions, but does not vary the amount of surfactant. In addition, the authors only cover two-dimensional systems and are thus unable to describe the behavior of $L_x(t)$ [70].

C. Complex fluids under oscillatory shear

In the case of oscillatory shear, the morphology and the domain growth are altered significantly, although the average deformation is zero after each period of shear. For example, it has been found experimentally for binary fluid mixtures that for very low oscillation frequencies domain growth can be interrupted, or domains can grow on much longer time scales than given by the oscillation frequency. Simulations so far either do not include hydrodynamic effects, or are limited to two dimensions. It has been observed in the two-dimensional lattice Boltzmann studies by Xu et al. that hydrodynamic effects must not be neglected in the case of oscillatory shear since there exists a finite time inversely proportional to the viscosity which is required to
set a linear velocity profile in the system. For high oscillation frequencies, this time is longer than the oscillation period of shear. For low surfactant concentrations \( \rho' < 0.2 \), \( L_y(t) \) even grows slightly faster than in the case without oscillatory movement. For \( \rho' \geq 0.2 \) the maximum domain size obtained is similar to the steady shear case. Due to the non-steady movement of the shear planes, \( L_y(t) \) and \( L_z(t) \) show a richer behavior: both functions show distinct kinks around the reversal points of the shear and for \( \rho' \geq 0.2 \) it is found that the growth rates are smaller than in the case of steady shear. Thus, we can observe the formation of tubular structures which are elongated in the y direction and show similar length scales in the x and z directions. For very high surfactant concentrations \( \rho' = 0.4 \) it is not possible to distinguish between tubular and spherical structures due to the small size of the individual domains.

In the z direction we can still observe peaks related to the formation and breakup, as well as the rotational movement, of domains. However, due to the overlaid effect of the oscillation, these peaks are no longer equidistant as in the steady shear case.

In Fig. 10 we present a system with a ten times larger oscillation frequency, i.e., \( \omega = 0.01 \). Here, the frequency of the oscillations is so high that the fluid is no longer able to follow the movement of the walls. Thus, the influence of the shear on the growth behavior becomes less pronounced, with the domains constantly growing as long as the amount of surfactant present allows it. The growth rates are comparable to those in the non-sheared case here and show identical growth laws as in the non-sheared case. The only difference is

10. As in the case of continuous shear, the domain growth in y direction is almost un influenced by the applied shear forces. In fact, for low surfactant concentrations \( \rho' < 0.2 \), \( L_y(t) \) even grows slightly faster than in the case without oscillatory movement. For \( \rho' \geq 0.2 \) the maximum domain size obtained is similar to the steady shear case. Due to the non-steady movement of the shear planes, \( L_y(t) \) and \( L_z(t) \) show a richer behavior: both functions show distinct kinks around the reversal points of the shear and for \( \rho' \geq 0.2 \) it is found that the growth rates are smaller than in the case of steady shear. Thus, we can observe the formation of tubular structures which are elongated in the y direction and show similar length scales in the x and z directions. For very high surfactant concentrations \( \rho' = 0.4 \) it is not possible to distinguish between tubular and spherical structures due to the small size of the individual domains.

In the z direction we can still observe peaks related to the formation and breakup, as well as the rotational movement, of domains. However, due to the overlaid effect of the oscillation, these peaks are no longer equidistant as in the steady shear case.
that the exponents are found to be smaller while $L_y(t)$ grows slightly faster than $L_x(t)$ and $L_z(t)$, depicting the occurrence of tubular structures in the system. The $z$ direction is the only component of the time-dependent lateral domain size that differs from the unsheared case, because strong oscillations start to appear due to the distortions caused by the moving boundaries.

We have increased the shear rate to $\dot{\gamma}=3.12 \times 10^{-3}$, but the lateral domain sizes obtained are almost identical to those in the $\dot{\gamma}=1.56 \times 10^{-3}$ case. Therefore, we do not present an additional figure, but our findings can be used to argue that for high oscillation frequencies the system behaves in a manner equivalent to the nonsheared case.

In the case of oscillatory shear we have shown the occurrence of tubular structures due to shear-imposed anisotropic domain growth, the slowing down of the domain growth rate depending on the oscillation frequency, as well as the result that a microemulsion with high surfactant concentration stays unaffected by external shear forces. Our results are in agreement with the simulations of Qiu et al. [11] and the two-dimensional lattice Boltzmann simulations of Xu et al. [14].

IV. CONCLUSIONS

In this paper we have presented detailed three-dimensional lattice Boltzmann studies of binary immiscible and ternary amphiphilic fluid mixtures under constant and oscillatory shear.

We have reproduced the well-known power law growth of domains in the case of binary immiscible fluids (spinodal decomposition) which crosses over to a logarithmic law and to a stretched exponential if one increases the surfactant concentration even further. For sufficiently high surfactant concentrations, domain growth can come to an end and the system corresponds to a stable bicontinuous microemulsion. For amphiphile concentrations of up to 30% we find linear dependencies of the time of arrest as well as the maximum domain size on the amphiphile concentration. For concentrations above 30%, neither arrest length nor time of arrest changes any more since the surface tension at the interfaces between the two immiscible fluids is at its minimum. The whole interface is filled with amphiphiles and further amphiphile molecules have to reside within the bulk fluid.

In sheared systems, we have studied the influence of moving boundaries on the effect of domain growth and report domain breakup phenomena depending on the shear rate as well as the amphiphile concentration. Depending on the surfactant concentration and the shear rate, we find a transition from a sponge phase to a lamellar phase.

Under oscillatory shear and with the oscillation frequencies chosen, no linear velocity gradient can build up within a single period of shear. Thus, the domains are constantly rearranging and align with the flow in the vicinity of the shear planes. In the bulk, however, no preferred alignment can be observed. But since the growth in the $y$ direction is not hindered by the shear, tubular structures occur and are best observable for low surfactant concentrations. For very fast oscillations ($\omega=0.01$), the system is not able to follow the external shear at all. Thus, it behaves similarly to a nonsheared one. The only differences are that fluctuations in $L_z(t)$ can be observed due to the oscillatory forces, while the growth in the other two directions is slowed down. For surfactant concentrations $\rho' \leq 0.2$ anisotropic growth in the $x$ and $y$ directions is observed depicting the presence of tubular domains in the system. In future work it would be of interest to study the formation of tubes and their dependency on the
shear rate, oscillation frequency, and surfactant concentration in greater detail. Additionally, the study of asymmetric mixtures, where the concentrations of the two immiscible fluid species differs, remains unexplored.

ACKNOWLEDGMENTS

We are grateful for the support of the HPC-Europa program, funded under the European Commission’s Research Infrastructure activities, Contract No. RI3-CT-2003-506079. We also acknowledge RealityGrid and ESLEA EPSRC Grant Nos. GR/R67699 and GR/T04465, the Höchstleistungszentrum Stuttgart for providing access to their NEC SX8, and the National Science Foundation (NSF) for U.S. TeraGrid usage within NRAC Grant No. MCA04N014. We would like to thank H. Berger, R. Keller, and P. Lammers for technical support and K. Stratford for fruitful discussions.

[47] S. M. Pickles et al., in Proceedings of the Workshop on Case
Studies on Grid Applications at GGF 10, 2004.

Simulations of microfluidic applications: probing the boundary condition
Lattice Boltzmann simulations of apparent slip in hydrophobic microchannels

J. Harting, C. Kunert, and H. J. Herrmann
Lattice Boltzmann simulations of apparent slip in hydrophobic microchannels

J. Harting, C. Kunert and H. J. Herrmann

Institute for Computational Physics, University of Stuttgart
Pfaffenwaldring 27, D-70569 Stuttgart, Germany

received 18 January 2006; accepted in final form 29 May 2006
published online 23 June 2006

PACS. 83.50.Rp – Wall slip and apparent slip.
PACS. 68.08.-p – Liquid-solid interfaces.
PACS. 47.11.-j – Computational methods in fluid dynamics.

Abstract. – Various experiments have found a boundary slip in hydrophobic microchannel flows, but a consistent understanding of the results is still lacking. While Molecular Dynamics (MD) simulations cannot reach the low shear rates and large system sizes of the experiments, it is often impossible to resolve the needed details with macroscopic approaches. We model the interaction between hydrophobic channel walls and a fluid by means of a multi-phase lattice Boltzmann model. Our mesoscopic approach overcomes the limitations of MD simulations and can reach the small flow velocities of known experiments. We reproduce results from experiments at small Knudsen numbers and other simulations, namely an increase of slip with increasing liquid-solid interactions, the slip being independent of the flow velocity, and a decreasing slip with increasing bulk pressure. Within our model we develop a semi-analytic approximation of the dependence of the slip on the pressure.

During the last century it was widely assumed that the velocity of a Newtonian liquid at a surface is always identical to the velocity of the surface. However, in recent years well-controlled experiments have shown a violation of the no-slip boundary condition in sub-micron-sized geometries. Since then, experimental [1] and theoretical works [2], as well as computer simulations [3–9] have tried to improve our understanding of boundary slip. The complex behavior of a fluid close to a solid interface involves the interplay of many physical and chemical properties. These include the wettability of the solid, shear rate, pressure, surface charge, surface roughness, as well as impurities and dissolved gas. Since all those quantities have to be determined very precisely, it is not surprising that our understanding of the phenomenon is still unsatisfactory. Due to the large number of different parameters, a significant dispersion of the results can be observed for ostensibly similar systems [1], e.g., observed slip lengths vary between nanometres [10] and micrometers [11] and while some authors find a dependence of the slip on the flow velocity [12], others do not [11,13]. Most computer simulations apply Molecular Dynamics (MD) and report increasing slip with decreasing liquid density [6,7] or liquid-solid interactions [8,14], while slip decreases with increasing pressure [4]. These simulations are usually limited to some tens of thousands of particles, lengths scales of nanometres and timescales of nanoseconds. Also, shear rates are orders of magnitude higher than in any
experiment [1]. We overcome these limitations using the lattice Boltzmann (LB) algorithm — a powerful method for simulating fluid dynamics [15]. Rather than tracking individual atoms and molecules, the dynamics of the single-particle distribution function \( \eta \) of mesoscopic fluid packets is described. In contrast to MD simulations, this method is less computationally demanding and allows to simulate experimentally accessible length and time scales. Our ansatz differs from other LB approaches where slip is introduced by generalizing no-slip bounce back boundary conditions to allow specular reflections with a given probability [3] or where the viscosity is modified due to local density variations [16]. In both cases, parameters determining the properties at the boundaries are not easily mappable to experimentally available values. Our approach is based on Shan and Chen’s multi-phase LB model [17]. Here, interactions between different species are modelled by mesoscopic forces between the phases. This naturally opens the way to introduce similar interactions between each fluid species and the channel walls, where the strength of the interaction is determined by the fluid densities, free coupling constants, and a wall interaction parameter which is treated in a similar manner as a local fluid density. The model allows the simulation of multi-phase flows along hydrophobic boundaries and is introduced in the following. However, in order to study the influence of hydrophobicity on the boundary slip and to demonstrate the basic properties of the model, we focus on single phase flow in this paper. Results of multi-phase simulations will be presented in a future work. A multi-phase LB system can be represented by a set of equations [18]

\[
\eta_i^\alpha(x + c_i, t + 1) - \eta_i^\alpha(x, t) = \Omega_i^\alpha, \quad i = 0, 1, \ldots, b,
\]

where \( \eta_i^\alpha(x, t) \) is the single-particle distribution function, indicating the amount of species \( \alpha \) with velocity \( c_i \), at site \( x \) on a D-dimensional lattice of coordination number \( b \) (D3Q19 in our implementation), at time-step \( t \). For the collision operator \( \Omega_i^\alpha \) we choose the Bhatnagar-Gross-Krook (BGK) form

\[
\Omega_i^\alpha = -\frac{1}{\tau_\alpha} (\eta_i^\alpha(x, t) - \eta_i^{\alpha\text{eq}}(u^\alpha(x, t), \eta^\alpha(x, t))),
\]

where \( \tau_\alpha \) is the mean collision time for component \( \alpha \) and determines the fluid viscosity. The system relaxes to an equilibrium distribution \( \eta_i^{\alpha\text{eq}} \) which can be derived imposing restriction on the microscopic processes, such as explicit mass and momentum conservation for each species [19]. \( \eta^\alpha(x, t) \equiv \sum_i \eta_i^\alpha(x, t) \) is the fluid density and \( u^\alpha(x, t) \) is the macroscopic velocity of the fluid, defined as \( \eta^\alpha(x, t) u^\alpha(x, t) \equiv \sum_i \eta_i^\alpha(x, t) c_i \). Interactions between different fluid species are introduced as a mean-field body force between nearest neighbors [17]:

\[
F^{\alpha\bar{\alpha}}(x, t) \equiv -\psi^\alpha(x, t) \sum_\alpha g_{\alpha\bar{\alpha}} \sum_{x'} \psi^{\bar{\alpha}}(x', t)(x' - x),
\]

where \( \psi^\alpha(x, t) = (1 - e^{-\eta^\alpha(x, t)/\eta_0}) \) is the so-called effective mass with \( \eta_0 \) being a reference density that is set to 1 in our case [17]. \( g_{\alpha\bar{\alpha}} \) is a force coupling constant, whose magnitude controls the strength of the interaction between component \( \alpha \) and \( \bar{\alpha} \). The dynamical effect of the force is realized in the BGK collision operator in eq. (2) by adding to the velocity \( u \) in the equilibrium distribution an increment \( \delta u^\alpha = \tau_\alpha F^{\alpha\bar{\alpha}}/\eta^\alpha \). For the interaction of the fluid components with the channel walls we apply mid-grid bounce back boundary conditions [15] and assign interaction properties to the wall which are similar to those of an additional fluid species. I.e., we specify constant values for the force coupling constant \( g_{\alpha\bar{\alpha}} = g_{\text{wall}, \alpha} \) and the density \( \eta^{\bar{\alpha}} = \eta^{\text{wall}} \) at wall boundary nodes of the lattice. This results in a purely local force as given in eq. (3) between the flow and the boundaries. Even though one could argue that a single parameter to tune the fluid-wall interaction would be sufficient, we keep our approach as close as possible to
the original idea of Shan and Chen, in order to benefit from the experience obtained from other works using the original model. Furthermore, the additional parameter allows more flexibility when simulating not only a single fluid, but a multi-phase system. The fluid-wall interaction can be linked to a contact angle between fluid droplets and solid walls as it is often used to quantitatively describe hydrophobic interactions [20]. Recently, Benzi et al. have shown how to compute the contact angle within the Shan-Chen model [21]. The same authors also developed an approach to model apparent slip which is related to ours, but instead of using only local fluid-solid interactions, they add an exponential decay of the interaction with distance from the wall [22]. We simulate pressure-driven flow between two infinite planes (Poiseuille flow), where pressure-driven boundary conditions are implemented in a similar way as in most experiments: a fixed pressure is set at the channel inlet and an open boundary at the outlet. The outlet is realized by interpolating the particle distribution function at the end of the channel as given by \( \eta_i(x,t) = 2\eta_i(x-1,t) - \eta_i(x-2,t) \) leading to a linear pressure gradient. Already in 1823, Navier proposed a boundary condition where the fluid velocity at a surface is proportional to the shear rate at the surface, i.e. \( v_z(x_0) = \beta \partial v_z(x)/\partial x \) at \( x = x_0 \) [23]. Following his hypothesis the velocity in flow direction \( (v_z) \) at position \( x \) between the planes is given by

\[
v_z(x) = \frac{1}{2\mu} \frac{\partial P}{\partial z} \left[ h^2 - x^2 - 2h\beta \right], \tag{4}
\]

where \( 2h \) is the distance between the planes, and \( \mu \) the viscosity. In contrast to a no-slip formulation, the last term in eq. (4) linearly depends on the slip length \( \beta \). Since \( \beta \) is typically of the order of nanometers or micrometers, it can be neglected in macroscopic experiments. In order to obtain \( \beta \) from our data, we measure the pressure gradient \( \partial P/\partial z \) at the center of the channel and the velocity profile between the two planes at a fixed position \( z \). \( \beta \) is then obtained by a least square fit with eq. (4).

Our simulation parameters are as follows: the lattice size is kept constant with the channel length \( (z\text{-direction}) \) being 256 sites, the distance between the plates \( 2h \) being 60 sites \( (x\text{-direction}) \). We approximate infinite planes by using a 16-sites-wide channel with periodic boundaries in the \( y\text{-direction} \). In order to assure a fully equilibrated system, we simulate for at least 40000 time steps before measuring and assured our results being independent of the discretization level by comparing to simulations of 28- and 124-sites-wide channels. Each data point in the figures below corresponds to about six hours simulation time on eight IBM Power 4 1.7 GHz CPUs. All units in this paper are in lattice units with the lattice constant \( c \) and timestep \( \Delta t \) set to 1 if not stated otherwise.

The dependence of the slip length \( \beta \) on the interaction parameter \( g_{wall,a} \) is studied for \( \eta^{wall} = 1.0 \) and 5.0. The bulk pressure \( P = \rho c^2 / \Delta \), where \( \rho \) is the fluid density and \( c = 1/\sqrt{3} \) the speed of sound, is kept at \( P = 0.11 \), while the flow velocity is set to \( V = 0.033 \). As shown in fig. 1a, we vary \( g_{wall,a} \) from 0.06 to 0.22 and find a steady increase of \( \beta \) for increasing \( g_{wall,a} \). As expected, the curve for \( \eta^{wall} = 5.0 \) is growing substantially faster than for \( \eta^{wall} = 1.0 \). The maximum available \( \beta \) are at about 5.2 for \( g_{wall,a} = 0.26 \) and \( \eta^{wall} = 1.0 \). At these strong fluid-wall interactions, the force as given in eq. (3) becomes very large and results in a large area of low fluid density close to the wall. Increasing the interaction even further results in numerical instabilities due to too steep density gradients. In order to study the dependence of the slip on other parameters, the coupling constant \( g_{wall,a} \) is kept constant at 0.08 from now on. Figure 1b depicts the dependence of \( \beta \) on \( \eta^{wall} \) for different bulk pressures \( P = 0.033, 0.1, \) and 0.3 and fixed flow velocity \( V = 3.5 \cdot 10^{-3} \) in the system. While all the three graphs grow constantly with increasing \( \eta^{wall} \), the one for \( P = 0.033 \) grows the fastest demonstrating that absolute values for \( \beta \) are higher for lower pressure.
We have measured the magnitude of the boundary slip over a very wide range of flow velocities $V$ from $1 \cdot 10^{-4}$ to $3 \cdot 10^{-2}$ for wall interactions $\eta_{\text{wall}} = 0.0, 0.5, 1.0$, and $2.0$. $V$ is measured at the center of the channel and given on a logarithmic scale in fig. 2. For $\eta_{\text{wall}} = 0.0$ we do not find any boundary slip confirming that our method properly reproduces the no-slip behavior in the interaction-free case. With increasing wall interactions, we achieve an increase of the magnitude of $\beta$ to up to $\simeq 1.1$ for $\eta_{\text{wall}} = 2.0$. We are not able to find any velocity dependence of $\beta$, but find constant slip for fixed fluid-wall interactions, which is consistent with many experiments [13, 24]. The fluctuations of the data for very low flow velocities are due to numerical uncertainties of the fit at very low curvature of the parabolic velocity profile. For $V > 0.01$ we find a slight deviation of $\beta$ from the constant measurements. This is due
J. Harting et al.: LB simulations of apparent slip

Fig. 3 – The fluid density close to the channel walls $\rho_1$ over $g_{wall,\alpha}$ is given for $\eta_{wall} = 1.0, 5.0$ (symbols). The lines correspond to a fit by our semi-analytic approximation.

to a small variation of the bulk pressure from $P = 0.097$ for $V = 1 \cdot 10^{-4}$ to $P = 0.106$ for $V = 0.03$ that cannot easily be avoided for technical reasons. We have checked for a few data points that $\beta$ stays constant if $P$ can be kept at exactly fixed values, too. The slip length being independent of the flow velocity is consistent with many experiments and computer simulations, like the MD simulations of Cottin-Bizonne et al. [25] and the experiments of Cheng et al. [13] and Baudry et al. [26]. We speculate that an increase of $\beta$ with increasing flow velocity as measured by some experiments [12] is due to surface roughness of the channel boundaries or other nonlinear effects. Since our model is not able to treat roughness on an atomic scale, we do not expect to conform with those results. MD simulations which find a non-constant value for $\beta$ operate at very high shear rates which are orders of magnitude higher than what can be obtained by our approach [9].

Computing the exact slip in dependence of the interaction parameters from first principle analytically is a very hard or even impossible task, since our interaction as given in eq. (3) modifies the equilibrium distribution in the BGK operator. Therefore, we present a semi-analytic approximation which utilizes the common two-layer model. Here, it is assumed that a thin fluid layer with thickness $\delta$ and different viscosity as the bulk fluid exists near the channel walls. As calculated by various authors [2], within this model the slip length can be computed as $\beta = (\mu_{bulk}/\mu_1 - 1)\delta$, where $\mu_{bulk}$ is the viscosity of the bulk fluid, and $\mu_1$ the viscosity close to the wall. Since the dynamic viscosity is given by the kinematic viscosity times the fluid density, $\mu = \rho \nu = \rho (2\tau^\alpha - 1)/6$ [15], we write $\beta = (\rho_{bulk}/\rho_1 - 1)\delta$. $\rho_{bulk}$ can be measured in the channel’s center and $\rho_1$ at the first lattice site next to the wall. Figure 3 shows the dependence of $\rho_1$ on $g_{wall,\alpha}$ for $\eta_{wall} = 1.0, 5.0$, $P = 0.11$, and $V = 0.033$. Since $\rho_1$ cannot easily be computed analytically, we postulate an interaction term that depends on the bulk density and the fluid-wall interaction as well as a free fit parameter $k$,

$$I = k F_{wall,\alpha}(x,t)/\rho_{bulk}(x,t)$$

and fit $\rho_1$ with an exponential function $\rho_1 = \rho_{bulk}(x,t) \exp[-I]$. With only a single value for $k$ we are able to utilize this equation to fit $\rho_1$ for all our simulation parameters. $k$ is found to be 8.35 for our data. The lines in fig. 3 illustrate the good quality of our approximation. A similar approach is applied to model the thickness of the layer at the wall which strongly
Fig. 4 – Slip length $\beta$ vs. bulk pressure $P$ for $\eta^{wall} = 0.5, 1.0, 2.0$ (symbols). $\beta$ increases with increasing fluid-wall interactions, but decreases with increasing $P$. This dependence can be described by a semi-analytic equation (lines) which agrees well for small fluid-wall interactions and qualitatively reproduces our data for strong fluid-wall interactions.

depends on the fluid-wall interaction and bulk density. Here, we set $\delta = \exp(I)$. As a result, $\beta$ can be estimated by $\beta = (\exp[I] - 1) \exp[I]$. The semi-analytic approximation is used to fit the dependence of the slip length $\beta$ on the bulk pressure $P$. Figure 4 shows the simulation data (symbols) and the approximation (lines) for wall interactions $\eta^{wall} = 0.5, 1.0, 2.0$. The bulk pressure is varied from 0.03 to 0.33. We find a decrease of $\beta$ with increasing pressure $P$. An increase of $\eta^{wall}$ leads to an increasing slope of the curves and to higher absolute values for $\beta$. Furthermore, we find a decrease of the slip with increasing bulk pressure. These results qualitatively agree with MD simulations [4,8]. Even with a single value for the fit parameter $k$, the semi-analytic description of $\beta$ agrees very well for low fluid-wall interactions. For strong interactions ($\eta^{wall} = 2.0$), the fit qualitatively reproduces the behavior of the slip length. Higher-order terms in the exponential ansatz for $\delta$ are needed for a better agreement.

To demonstrate that our approach is able to achieve experimentally available length and time scales, we scale our simulations to the experimental setup of Tretheway and Meinhart [11]. They use a 30 $\mu$m high and 300 $\mu$m wide microchannel with typical flow velocities of $V = 10^{-2}$ mm/s. For water, they measure a slip length of 0.92 $\mu$m. The Reynolds number $Re = 2hV/\nu$ in their experiment is $\simeq 0.3$. To reproduce the observed slip, we set $g_{wall,\alpha} = 0.16$ and $\eta^{wall} = 1.0$ (see fig. 1a). We are able to cover a wide range of flow velocities, i.e. for the setup given above, velocities can range from as low as $1 \times 10^{-4}$ to as high as 0.05 corresponding to $Re$ between 0.038 and 19. The Knudsen number is given by $Kn = \nu/(c_s h)$ which corresponds to $4.8 \times 10^{-3}$ for the simulations presented here. At these low $Kn$, the hydrodynamic approach is well valid. However, it has been shown that the LB method can be applied for $Kn$ much larger than 1, if one uses modified boundary conditions [27]. Our mesoscopic force is expected to be able to properly describe fluid-wall interactions in such systems as well.

In conclusion, we presented a new approach to investigate boundary slip in hydrophobic microchannels by means of a multi-phase LB model. In contrast to MD simulations, our model is able to reach the length and time scales of typical experiments and is applicable for a wide range of realistic flow velocities. We qualitatively reproduced the dependence of slip on the hydrophobicity of the channel walls and found constant slip for varying flow velocities. The decrease of the slip with increasing pressure can be approximated by a semi-analytic approach. Our results are consistent with MD simulations [4,8,25] and experiments [11].
We would like to thank G. Giupponi, M. Hecht, N. González-Segredo, and V. S. J. Craig for fruitful discussions and acknowledge the Neumann Institute for Computing for providing access to their IBM p690 system.

REFERENCES

On the effect of surfactant adsorption and viscosity change on apparent slip in hydrophobic microchannels

C. Kunert and J. Harting
On the effect of surfactant adsorption and viscosity change on apparent slip in hydrophobic microchannels

Christian Kunert and Jens Harting*

Institute for Computational Physics, University of Stuttgart, Pfaffenwaldring 27, D-70569 Stuttgart, Germany
E-mail: jens@icp.uni-stuttgart.de
*Corresponding author

Abstract: Substantial effort has been invested to understand the effect of boundary slippage in microfluidics. However, a satisfactory understanding is still lacking due to extremely precise experiments required. Here, we apply the lattice Boltzmann method to model hydrophobic microchannel flows. We find a possible explanation for experiments observing slip depending on the flow velocity – in contradiction to previous studies. We show that these observations can be explained by not fully developed flow profiles in the channel. Further, we show a decreasing slip length with increasing viscosity and demonstrate the effect of adding surfactant to fluid flow in hydrophobic microchannels.

Keywords: lattice Boltzmann; microflows; apparent slip.

Reference to this paper should be made as follows: Kunert, C. and Harting, J. (2008) ‘On the effect of surfactant adsorption and viscosity change on apparent slip in hydrophobic microchannels’, Progress in Computational Fluid Dynamics, Vol. 8, Nos. 1–4, pp.197–205.

Biographical notes: Christian Kunert studied Physics in Stuttgart, Germany and submitted his Diploma Thesis on Apparent Slip in Microfluidic Applications in 2006. Since then he works on a PhD project simulating problems related to boundary conditions using the lattice Boltzmann method.

Jens Harting studied Physics in Oldenburg, Germany. After a Diploma Thesis on Bose-Einstein Condensation he worked on path integral Monte Carlo simulations of few electron systems and received his PhD in December 2001. Being interested in simulations of fluids and high performance computing, he joined the group of Peter V. Coveney in London, UK as a Research Fellow from 2002 to 2004. Since then, he has a position at the University of Stuttgart, Germany and works on the simulation of fluid mixtures and particle-laden flows using lattice Boltzmann and other mesoscale techniques.

1 Introduction

Microflow devices are used for chemical, biological, or medical analysis techniques. Putting the ‘lab on a chip’ allows to minimise the time needed for the analysis with only small amounts of fluid. Also, such microdevices are more mobile and allow a parallel treatment of multiple fluids. Other microflow systems are used as sensors and actors for devices like chemical reactors, cars, airplanes and inkjet printers.

In these miniature apparatuses, a number of effects appear which cannot easily be explained with our conventional physical understanding. A common example is the violation of the no-slip boundary condition. The no-slip boundary condition is one of the fundamental assumptions common in classical fluid mechanics, stating that the velocity of a fluid at a wall is equal to the velocity of the wall. For macroscopic applications no-slip is undoubted but during recent years a number of experiments found a violation of the no-slip boundary condition on a length scale of nanometers up to micrometers (Lauga et al., 2005; Neto et al., 2005). Numerous experiments (Lauga et al., 2005; Neto et al., 2005; Vinogradova, 1995; Vinogradova and Yakubov, 2003; Vinogradova, 1998; Henry et al., 2004; Craig et al., 2001; Neto et al., 2003; Bonaccurso et al., 2003) utilise a modified Atomic Force Microscope (AFM) with an oscillating colloidal sphere at the tip of its cantilever to measure the force needed to displace the fluid between the colloidal sphere and a wall. From the detected force, the amount of wall-slippage can be estimated as described in Vinogradova (1995). Other authors like Tretheway and Meinhart apply Particle Image Velocimetry (PIV) to observe the flow near the fluid-wall boundary directly to quantify wall slippage (Tretheway and Meinhart, 2002, 2004). However, it is still an open question if the detected
slip is a fundamental property or appears due to surface variations, uncertainties in the experimental setups, or the complex interactions between fluid and wall.

Instead of the no-slip boundary condition, Navier introduced in 1823 a slip boundary condition where the transversal velocity near the wall \( v_z(x = 0) \) is proportional to the shear rate \( \frac{\partial v_x}{\partial x} \) and the so-called slip length \( \beta \) (Navier, 1823),

\[
v_z(x = 0) = \beta \frac{\partial v_x}{\partial x}|_{x=0}.
\]

(1)

Here, the boundary is at \( x = 0 \); \( z \) is the flow direction and \( v_z \) is the fluid velocity in flow direction, parallel to the wall. The slip length \( \beta \) can be interpreted as the distance between the wall and the virtual point inside the wall at which the extrapolated flow velocity would be zero.

Due to the large number of tunable experimental parameters like temperature, viscosity, flow velocity, pressure, or surface properties, as well as their individual dependencies on each other, it is not possible to cover all occurring phenomena in a single experiment. In fact, a change in the surface properties usually implies a different experimental setup and a change of viscosity without varying the temperature is only possible by a replacement of the fluid. However, such strong interventions might also have an influence on other parameters of the system. In computer simulations it is possible to vary a single parameter of the fluid, e.g., the viscosity or the density, without changing other parameters. This is important to improve our understanding of the effects occurring in microfluidic systems and to further promote the design of such devices.

In addition, computer simulations are able to study the properties of multiphase flows in microchannels with the individual fluid parameters and fluid-fluid interactions being well defined. In particular, the influence of surfactant is of interest here. Surfactant molecules are often called amphiphiles and are comprised of a hydrophilic (water-loving) head group and a hydrophobic (oil-loving) tail. In a non-wetting microchannel filled with water, the surfactant molecules arrange at the interface between water and surface, thus shielding the hydrophobic repulsion of the wall. On the other hand, in a wetting channel an arrangement of surfactant molecules at the boundary causes the otherwise wetting wall to become hydrophobic. As a result an apparent slip occurs.

### 2 Simulation method

The simulation method used to study microfluidic devices has to be chosen carefully. While Navier-Stokes solvers are able to cover most problems in fluid dynamics, they lack the possibility to include the influence of molecular interactions as needed to model boundary slip. Molecular Dynamics (MD) simulations are the best choice to simulate the fluid-wall interaction, but the computer power today is not sufficient to simulate length and time scales necessary to achieve orders of magnitude which are relevant for experiments. However, boundary slip with a slip length \( \beta \) of the order of many molecular diameters \( \sigma \) has been studied with molecular dynamics simulations by Thompson and Troian (1997), Thompson and Robbins (1990), Koplik and Banavar (1998), Cieplak et al. (2001), Koplik et al. (1989), Cottin-Bizonne et al. (2004) and Baudry and Charlaix (2001). They find increasing slip with decreasing liquid density and liquid-solid interactions as well as a decrease of slip with increasing pressure. However, the maximum number of particles that can be simulated on today’s most powerful supercomputers is about 20 billion (Kadau et al., 2004). This corresponds to a volume of less than 1 \( \mu m^3 \), but the typical length scale of a microfluidic device is about 100 \( \mu m \).

A mesoscopic model is able to govern a volume large enough to describe the flow properties and still holds information about the molecular behaviour. The term ‘mesoscopic’ means that the trajectories of single molecules are not simulated in detail but a whole ensemble of ‘quasi particles’ behaves as the corresponding real microscopic system. Due to this coarse-graining, the numerical effort is much smaller than for molecular dynamics simulations because the collision and propagation rules of the used ‘quasi particles’ are much simpler than the ones of real particles. Therefore, much larger particle counts can be simulated for substantially longer times. An example for a mesoscopic simulation method is Stochastic Rotation Dynamics (SRD), which is sometimes called Multi Particle Collision Dynamics (MPCD). In a propagation step, every representative fluid particle is moved according to its velocity to its new position. In the collision step, the simulation volume is split into boxes. In each box the velocity vectors of every single particle are rotated around the mean velocity in a random manner, so that energy and momentum are conserved in every box (Malevanes and Kapral, 1999, 2000). The method is efficient and is used when Brownian motion is required. Its disadvantage is that thermal fluctuations cannot be switched off. Dissipative Particle Dynamics (DPD) also utilises quasi particles which represent a set of molecules. The propagation of such a collective quasi particle is implemented as in molecular dynamics but collisions are dissipative. This method is easy to implement in an existing MD simulation code but the computational costs are still very high.

In this paper we use the lattice Boltzmann method, where one discretises the Boltzmann kinetic equation

\[
\frac{\partial \eta}{\partial t} + v \nabla \eta + \frac{1}{m} \mathbf{F} \nabla \eta = \Omega
\]

(2)

on a lattice. \( \eta \) indicates the probability to find a single particle with mass \( m \), velocity \( v \) at the time \( t \) at position \( x \). The derivatives represent simple propagation of a single particle in real and velocity space whereas the collision operator \( \Omega \) takes into account molecular collisions in which a particle changes its momentum due to a collision with another particle. External forces \( \mathbf{F} \) can be employed to implement the effect of gravity or external fields. To represent the correct physics, the collision operator should conserve mass, momentum, and energy, and should
be Gallilei invariant. By performing a Chapman Enskog procedure, it can be shown that such a collision operator $\Omega$ reproduces the Navier-Stokes equation (Succi, 2001). In the lattice Boltzmann method the time $t$, the position $x$, and the velocity $v$ are discretised.

During the last years a number of attempts to simulate slip within the lattice Boltzmann method have been developed. The most simple idea is to use a partial bounce back boundary condition (Succi, 2001). While full bounce back leads to no slip, full reflection leads to full slip. Partial slip implies that a particle is reflected by the wall with the probability $q$, while it bounces back with probability $(1-q)$. As a result, a finite boundary slip can be observed. Nie et al. (2002) use a Knudsen-number dependent relaxation time in the vicinity of the wall to generate slippage in an ideal gas lattice Boltzmann model.

Our attempt to generate slip involves a repulsive potential at the wall (Harting et al., 2006). This leads to a depletion zone near the wall with a reduced density resulting in an apparent slip at hydrophobic (non wetting) walls. Benzi et al. (2006a) introduced a similar approach but the repulsion there decays exponentially while the potential we are using only takes into account next neighbour lattice sites as described below. Our method is based on Shan and Chen’s multiphase lattice Boltzmann method, i.e., the interaction between the surface and the fluid is simulated similar to the interactions between two fluid phases. This allows us to recycle our well tested parallel 3D multiphase lattice Boltzmann code, as it is presented in Harting et al. (2005) with only minor modifications. It is very advantageous of our model that its parameters can be linked to experimentally available properties, namely the contact angle (Benzi et al., 2006b).

The simulation method and our implementation of boundary conditions are described as follows. A multiphase lattice Boltzmann system can be represented by a set of equations (Benzi et al., 1992)

$$\eta^s_i(x + c_i t + 1) - \eta^s_i(x,t) = \Omega^s_i, \quad i = 0, 1, \ldots , b ,$$  

(3)

where $\eta^s_i(x,t)$ is the single-particle distribution function, indicating the amount of species $i$ with velocity $c_i$, at site $x$ on a $D$-dimensional lattice of coordination number $b$ (D3Q19 in our implementation), at time-step $t$. This is a discretised version of Equation (2) without external forces $F$ for a number of species $\alpha$. For the collision operator $\Omega^s_i$ we choose the Bhatnagar-Gross-Krook (BGK) form (Bhatnagar et al., 1954)

$$\Omega^s_i = -\frac{1}{\tau^s} (\eta^s_i(x,t) - \eta^{eq}_i(u^s(x,t), \eta^s(x,t))),$$  

(4)

where $\tau^s$ is the mean collision time for component $\alpha$ and determines the kinematic viscosity

$$\nu^s = \frac{2\tau^s - 1}{6}$$  

(5)

of the fluid. The system relaxes to an equilibrium distribution $\eta^{eq}_i$ which can be derived imposing restrictions on the microscopic processes, such as explicit mass and momentum conservation for each species (Chen et al., 1991, 1992; Qian et al., 1992). In our implementation we choose for the equilibrium distribution

$$\eta^{eq}_i = \zeta^s \eta^s_i \left[ 1 + \frac{c_i \cdot u}{c_i^2} + \frac{(c_i \cdot u)^2}{2c_i^4} - \frac{u_i^2}{2c_i^2} + \frac{(c_i \cdot u)^3}{6c_i^6} - \frac{u_i^2(c_i \cdot u)}{2c_i^2} \right].$$  

(6)

which is a polynomial expansion of the Maxwell distribution. $c_i$ are the velocity vectors pointing to neighbouring lattice sites. $c_i = 1/\sqrt{3}$ is the speed of sound for the D3Q19 lattice. The macroscopic values can be derived from the single-particle distribution function $\eta^s_i(x,t)$, i.e., the density $\eta^s_i(x)$ of the species $\alpha$ at lattice site $x$ is the sum over the distribution functions $\eta^s_i(x)$ for all lattice velocities $c_i$.

$$\eta^s_i(x,t) = \sum \eta^s_i(x,t).$$  

(7)

$u^s(x,t)$ is the macroscopic velocity of the fluid, defined as

$$\eta^s_i(x,t)u^s_i(x,t) \equiv \sum i \eta^s_i(x,t)c_i.$$  

(8)

Interactions between different fluid species are introduced following Shan and Chen as a mean field body force between nearest neighbours (Shan, 1993; Shan and Chen, 1994).

$$F^s(x,t) = -\psi^s(x,t) \sum_{\alpha} g_{\alpha \alpha} \sum_{x'} \psi^s(x',t)(x'-x).$$  

(9)

where $\psi^s(x,t) = (1 - e^{-\nu^s(x,t)/\tau^s})$ is the so-called effective mass with $\eta^s_i$ being a reference density that is set to 1 in our case (Shan, 1993). $g_{\alpha \alpha}$ is a force coupling constant, whose magnitude controls the strength of the interaction between component $\alpha$ and $\alpha$. The dynamical effect of the force is realised in the BGK collision operator (4) by adding an increment $\delta u^s = \tau^s F^s/\eta^s$ to the velocity $u$ in the equilibrium distribution function (6). For the potential of the wall we attach the imaginary fluid ‘density’ $\eta^{wall}$ to the first lattice site inside the wall. The only difference between $\eta^{wall}$ and any other fluid packages on the lattice $\eta^s$ is that the fluid corresponding to $\eta^{wall}$ is only taken into account for in the collision step, but not in the propagation step. Therefore, we can adopt $\eta^{wall}$ and the coupling constant $g_{\alpha \alpha \alpha}$ in order to tune the fluid-wall interaction. $g_{\alpha \alpha \alpha}$ is kept at 0.08 throughout this paper if not mentioned otherwise and all values are reported in lattice units. Additionally, we apply second order correct mid-grid bounce back boundary conditions between the fluid and the surface (Succi, 2001). Extending our model to a multi-relaxation time scheme would result in a more correct treatment of the boundaries, but the difference in the observed slip lengths is expected to be neglectable since interaction induced by the repulsive force between fluid and wall causes a substantially larger effect.

From molecular dynamics simulations it is known that the fluid-wall interactions causing a slip phenomenon usually take place within a few molecular layers of
the liquid along the boundary surface (Thompson and Troian, 1997; Thompson and Robbins, 1990; Koplik and Banavar, 1998; Cieplak et al., 2001; Koplik et al., 1989; Cottin-Bizonne et al., 2004; Baudry and Charlaix, 2001). Our coarse-grained fluid wall interaction acts on the length scale of one lattice constant and does not take the molecular details into account. Therefore, our implementation is only able to reproduce an averaged effect of the interaction and we cannot fully resolve the correct flow profile very close to the wall and below the resolution of a single lattice spacing. However, in order to understand the influence of the hydrophobicity on experimentally observed apparent slip, it is fully sufficient to investigate the flow behaviour on more macroscopic scales as they are accessible for experimental investigation. Our method could be improved by a direct mapping of data obtained from MD simulations to our coupling constant $g_{\text{wall}}$ allowing a direct comparison of the influence of liquid-wall interactions on the detected slip. This is currently ongoing project and our results will be published elsewhere.

Amphiphiles are introduced within the model as described in Chen et al. (2000) and Nekovee et al. (2000). An amphiphile usually possesses two different fragments, one being hydrophobic and one being hydrophilic. The orientation of any amphiphile present at a lattice site $x$ is represented by an average dipole vector $d(x, t)$. Its direction is allowed to vary continuously and to keep the model as simple as possible no information is specified for velocities $c_i$. The surfactant density at a given site is given by an additional fluid species with density $\eta^{\text{surf}}$, that behaves as every other species $\alpha$. The direction $d(x, t)$ propagates with the fluid field according to

$$\eta^{\text{surf}}(x,t + 1)d(x,t + 1) = \sum_i \eta^{\text{surf}}(x - c_i,t)d'(x - c_i,t)$$

and during the collision step the direction $d$ evolves to the equilibrium direction $d^{eq}$ similar to the BGK operator

$$d'(x,t) = d(x,t) - d(x,t) - d^{eq}(x,t)$$

($d'$ indicates the direction after the collision step). The equilibrium distribution $d^{eq} \approx \frac{2h}{\tau_d}$ is proportional to the so called colour field or order parameter $h$ which represents the distribution of the other species. It is defined as the weighted sum of the densities of all species

$$h(x, t) = \sum_\alpha \epsilon^\alpha \eta^\alpha(x, t).$$

In our case ($\alpha = 2$) we set the weights to $\epsilon^\alpha = \pm 1$, i.e., $h$ corresponds to the density difference between the two species.

The model has been used successfully to study spinodal decomposition (Chin et al., 2002; González-Segredo et al., 2003), binary and ternary amphiphilic fluids under shear (Harting et al., 2004a), the formation of mesophases (Nekovee and Coveney, 2001a, 2001b; González-Segredo and Coveney, 2004; Harting et al., 2004b; Giupponi et al., 2006; González-Segredo et al., 2006), and flow in porous media (Harting et al., 2004c).

Of particular relevance for the present paper is our first article on simulations of apparent slip in hydrophobic microchannels (Harting et al., 2006).

3 Simulation setup

The simulations in this work use a setup of two infinite planes separated by the distance $2d$. We call the direction between the two planes $x$ and if not stated otherwise $2d$ is set to 64 lattice sites. In $y$ direction we apply periodic boundary conditions. Here, eight lattice sites are sufficient to avoid finite size effects, since there is no propagation in this direction. $z$ is the direction of the flow with our channels being 512 lattice sites long. At the beginning of the simulation ($t = 0$) the fluid is at rest. We then apply a pressure gradient $\nabla P$ in the $z$ direction to generate a planar Poiseuille flow. Assuming Navier’s boundary condition (1), the slip length $\beta$ is measured by fitting the theoretical velocity profile,

$$v_z(x) = \frac{1}{2\mu} \frac{\partial P}{\partial z} [d^2 - x^2 - 2d\beta],$$

in flow direction ($v_z$) at position $x$, to the simulated data via the slip length $\beta$. We validate this approach by comparing the measured mass flow rate $\int v_z(x)dx$ to the theoretical mass flow without boundary slip and find a very good agreement. The pressure gradient $\frac{\partial P}{\partial z}$ is realised by a fixed inflow pressure ($P(z = 0) = \beta \eta(z = 0) = 0.3$ if not stated otherwise). At the outflow ($z = z_{\text{max}}$) we linearly extrapolate the density gradient by setting $\eta(z_{\text{max}}) = 2\eta(z_{\text{max}} - 1) - \eta(z_{\text{max}} - 2)$ in order to simulate infinite plates. Therefore, the body force regulates the velocity. The dynamic viscosity $\mu$ as well as the pressure gradient $\frac{\partial P}{\partial z}$ needed to fit Equation (13) are obtained from our simulation data.

In a previous work (Harting et al., 2006), we have shown that this model creates a larger slip $\beta$ with stronger interaction, namely larger $g_{\text{wall}}$ and larger $g_{\text{wall}}'$. The relaxation time $\tau_\alpha$ was kept constant at 1.0 in this study and the maximum available slip length measured was 5.0 in lattice units. For stronger repulsive potentials, the density gradient at the fluid-wall interface becomes too large, causing the simulation to become unstable. At lower interactions the method is very stable and the slip length $\beta$ is independent of the distance $d$ between the two plates and therefore independent of the resolution. We have also shown that the slip decreases with increasing pressure since the relative strength of the repulsive potential compared to the bulk pressure is weaker at high pressure. Therefore, the pressure reduction near the wall is less in the high pressure case than in the low pressure one. Furthermore, we have demonstrated that $\beta$ can be fitted with a semi analytical model based on a two viscosity model.
On the effect of surfactant adsorption and viscosity

4 Results

We have studied the dependence of the slip length $\beta$ on the flow velocity for a wide range of velocities of more than three decades as it can be seen in Figure 1(a) and in Harting et al. (2006). In the figure, we show data for different fluid-wall interactions $0 < \eta_{\text{wall}} < 2.0$ and flow velocities from $10^{-3} < v < 10^{-1}$. Within this region we confirm the findings of many steady state experiments (Cheng and Giordano, 2002; Cheikh and Koper, 2003), i.e., that the slip length is independent of the flow velocity and only depends on the wettability of the channel walls. Experimentalists often present measurements for different shear rates $\dot{S}$, which for Poiseuille flow are given by

$$\dot{S} = \frac{\partial u}{\partial x} \big|_{z=d} = -\frac{\nabla p x}{\mu} \big|_{z=d} = -\frac{\nabla p d}{\mu}. \tag{14}$$

Some dynamic experiments, however, find a shear rate dependent slip (Neto et al., 2003; Zhu and Granick, 2001, 2002). These experiments often utilise a modified atomic force microscope as described in the introduction to detect boundary slippage. Since the slip length is found to be constant in our simulations after sufficiently long simulation times, we investigate the behaviour of the slip during the transient, i.e., for simulation times $t \ll t_c$ with $t_c = L_z/v$ being the self convection time. The flow that is initially at rest has not converged to its final steady state. The time development of the slip length could explain an apparent shear dependence as shown in Figure 1(b), where $\beta$ is plotted over the flow velocity for different fluid-wall interactions at $t = 15,000$. Here, the detected $\beta$ depends very strongly on the flow velocity. The figure shows a qualitative similarity to the data presented by Zhu and Granick (2001), namely there seems to be a critical shear rate at which the slip starts to increase very fast. However, this only holds during the transient as shown in Figure 1 – in the steady state the slip is independent of the velocity.

Figure 2 depicts the time dependence of the measured slip length at constant $\eta_{\text{wall}} = 1.0$ and for final flow velocities $v = 0.7 \times 10^{-3}$, $1.3 \times 10^{-3}$, and $4.0 \times 10^{-3}$. Since for $t < 10,000$ the expected parabolic velocity profile is not developed, we only plot our data for $10,000 < t < 50,000$. It can be observed that the slip length develops to its final value for all three bulk velocities. However, the number of timesteps needed to achieve the steady state of $\beta$ is dependent on $v$. The slip has reached its steady state after the convection time $t_c = L_z/v$, which is the time it takes for an individual fluid element to be transported through the whole system. The slip converges with different rates depending on the flow velocity, but after $50,000$ timesteps the difference between the actual slip length and the converged one is neglectible already. This explains the fluctuations for very low velocities in Figure 1(a). The determination of the correct slip length therefore can only be expected after sufficiently long simulation times. As can be seen from Figure 3, it is not sufficient to just check if the velocity profile seems to have reached its final shape. Here, velocity profiles after $15,000$ and $50,000$ timesteps are shown for a representative simulation run and $\eta_{\text{wall}} = 2.0$. Even though the parabolic velocity profile is already fully developed after $15,000$ timesteps, the measured slip length is $\beta = 0.55 \pm 7 \cdot 10^{-3}$ only, while after $50,000$ timesteps $\beta = 1.088 \pm 7 \cdot 10^{-3}$ is obtained. The solid lines in Figure 3 correspond to a fit of the data with Equation (13).

![Figure 1](image1.png)

![Figure 2](image2.png)

The kinematic viscosity $\nu$ is another important parameter in fluid dynamics. However, in experiments it can only be varied by changing the fluid itself and therefore it is inevitable to change other parameters too. Within the lattice Boltzmann method with BGK collision operator (4), the kinematic viscosity of the fluid is given by Equation (5) and depends on the relaxation time $\tau^\alpha$. Within the Shan-Chen model, a change of $\tau^\alpha$ also has an influence on the effect of the body force that enters the BGK operator to model the fluid-fluid interactions. One might argue that this is not realistic since a change of viscosity does not necessarily modify the fluid-fluid interactions between different species. Additionally, it is known that mid grid
bounce back boundary conditions are second order correct while using the BGK collision operator, as it is used in this paper (Succi, 2001; He et al., 1997). For relaxation times $\tau^\alpha \approx 1$ the error introduced due to the boundary condition is negligible. However, we are interested in studying the dependence of boundary slippage on the fluid’s viscosity. Therefore, we performed simulations with $\eta_{\text{wall}} = 0$, i.e., without any fluid-wall repulsion, to estimate the effect of the error induced by the boundaries. For $\eta_{\text{wall}} = 0$, $\beta$ should be zero as well, but we find the error of the slip length being proportional to $(\tau^\alpha)^2$. This behaviour is expected by the theory of He et al. (1997) and can only be avoided by using a multi relaxation time approach.

For $1 < \tau^\alpha < 3$ the numerical error is less than 5% of the slip length while for larger relaxation times the error increases strongly so that the slip seems to increase. In order to reduce the influence of the error introduced by the single relaxation time method and the particular boundary conditions used, we subtract the slip length determined for $\eta_{\text{wall}} = 0$ from the measured $\beta$ at $\eta_{\text{wall}} \neq 0$. The results are plotted in Figure 4, where we demonstrate a decreasing slip length with increasing viscosity for $\eta_{\text{wall}} = 0.5, 1.0$, and $2.0$. The data shown in Figure 4 can be fitted exponentially as depicted by the solid lines and all three curves converge to zero for high viscosities.

Figure 2 Measured slip length $\beta$ vs. time $t$ for different bulk velocities at constant $\eta_{\text{wall}} = 1.0$. The difference between the converged slip length and the slip length during the transient is greater for slower velocities. After the convection time $t_c = L_x/v$ the slip is converged, but already for $t > 50,000$ only small deviations from the final value can be observed.

Since surfactant molecules consist of a hydrophobic and a hydrophilic part, they like to assemble at the interface between a fluid and wetting or non-wetting walls. As found by experimentalists, in a wetting microchannel, this can cause no slip to switch to partial slip (Cheikh and Koper, 2003; Zhu and Granick, 2002). In a non-wetting environment, the surfactant molecules can shield the hydrophobic repulsion of the surface (Henry et al., 2004). We apply the amphiphilic lattice Boltzmann model as described earlier in this paper to model a fluid within a hydrophobic microchannel that contains a surfactant concentration of up to 33%. The interaction parameters are choosen according to earlier works (Harting et al., 2004a; Nekovee and Coveney, 2001a, 2001b; González-Segredo and Coveney, 2004; Harting et al., 2004b; Giupponi et al., 2006; González-Segredo et al., 2006), in such a way that they are not too strong to cause structuring effects in the flow, but strong enough to have an effect at the fluid-solid boundary. The total density inside our system $\rho^\alpha + \rho^{\text{surf}}$ is kept fixed at 0.3. As initial condition the system is filled with a binary mixture of surfactant and fluid. The orientation of the dipoles is chosen randomly. In Figure 5, we plot the measured slip length for fluid-wall interactions determined by $\eta_{\text{wall}} = 0.5, 1.0$ and $2.0$ vs. the concentration of surfactant. The symbols in Figure 5 are given by the simulation data while the lines correspond to a fit with an exponential function. We find a strong decrease of the slip length with a higher surfactant concentration. For all three values of $\eta_{\text{wall}}$, the measured slip lengths converge to the same value at high surfactant concentrations showing that at high concentrations the amount of surfactant that can assemble at the interface is saturating.

Figure 3 The velocity profile $v(x)$ for $\eta_{\text{wall}} = 2.0$ after $t = 15,000$ and $t = 50,000$ time steps. The lines are the parabolic fit with Equation (13) with a slip length of $\beta = 0.55 \pm 7 \times 10^{-3}$ at $t = 15,000$. After 50,000 time steps the slip length is significantly larger at $\beta = 1.088 \pm 7 \times 10^{-4}$.

In Figure 6 we present a representative density profile of the surfactant for $\eta_{\text{wall}} = 1.0$. The initial amphiphile
concentration is set to 33% here. It can be seen that the concentration at the first lattice site next to the surface increases to 44%, while the bulk concentration stays constant at 32% – a value slightly lower than the initial 33%. This high concentration regime close to the boundary causes the hydrophobic potential of the wall to be shielded and results in a decreasing slip. Our findings are consistent with experimental results (Cheikh and Koper, 2003; Henry et al., 2004; Zhu and Granick, 2002).

Figure 5 Slip length $\beta$ vs. the concentration of surfactant in % for $\eta_{wall} = 1.0$. $\beta$ is steadily decreasing with increasing the surfactant concentration from 0.64 down to 0.19. The dashed line is given by a fit of the data with an exponential function.

Figure 6 A typical profile of the surfactant concentration in $x$ direction, i.e., between the channel walls. Near the surface, the surfactant concentration is substantially higher (44%) than in the bulk (32%) since it is energetically more favourable for the surfactant molecules to arrange at the fluid-surface interface, thus shielding the repulsive potential of the hydrophobic channel wall.

Large amphiphilic molecules or polymer brushes show a shear dependent slip (Fetzer et al., 2005) since they have to align with the shear forces acting on them. The higher the shear force, the more they are rotated causing the effect of shielding the hydrophobicity to be reduced. Since in our model the amphiphiles are point-like, we cannot expect to observe any shear rate dependence of $\beta$.

5 Conclusion

In conclusion we have presented three-dimensional multiphase lattice Boltzmann simulations which govern a wide range of slip phenomena. After demonstrating the validity of our model, we presented studies of the dependence of the boundary slip on the flow velocity. While the slip is independent of the velocity if the system is in the steady state, we find an apparent velocity dependence during early times of the simulation. For small numbers of timesteps, the parabolic velocity profile is already well developed, but due to the system being in a transient state, the detected slip is not correct. This is an important finding for experimental setups since to the best of our knowledge only dynamic experiments find a velocity dependence, while static experiments confirm the slip lengths being independent of the flow velocity. Our findings are in good agreement with most non dynamic experiments (Lauga et al., 2005; Neto et al., 2005) and MD simulations (Cottin-Bizone et al., 2004; Baudry and Charlaix, 2001).

For experimentalists it is a major effort to change the viscosity of the fluid without changing any other parameters of their setup. In computer simulations, however, this can be done easily. In our simulations we found a decrease of the detected slip with increasing viscosity.

With a simple dipole model we were able to simulate the shielding of the repulsive potential between hydrophobic walls and a fluid if surfactant is present in the solution, i.e., the slip length decreases with increasing surfactant concentration. However, we were not able to show a shear dependence as it is seen in experiments with polymer chains. In a future work, we plan to extend our simulations to govern larger molecules which can be affected by a shear flow. Then, we hope to be able to study the shear rate dependence of boundary slippage.

Acknowledgements

We would like to thank S. Succi, L. Biferale, R. Delgado-Buscalioni, and L.S. Luo for fruitful discussions. We acknowledge the Neumann Institute for Computing for providing access to their IBM p690 system and the High Performance Computing Center Stuttgart for the possibility to use their NEC SX8. We are grateful for financial support provided by the Landesstiftung Baden-Württemberg and the DFG within priority program 1164.

References


Roughness induced boundary slip in microchannel flows

C. Kunert and J. Harting
Roughness Induced Boundary Slip in Microchannel Flows

Christian Kunert and Jens Harting

Institute for Computational Physics, University of Stuttgart, Pfaffenwaldring 27, D-70569 Stuttgart, Germany

(Received 24 April 2007; published 23 October 2007)

Surface roughness becomes relevant if typical length scales of the system are comparable to the variations as it is the case in microfluidic setups. Here, an apparent slip is often detected which can have its origin in the misleading assumption of perfectly smooth boundaries. We investigate the problem by means of lattice Boltzmann simulations and introduce an “effective no-slip plane” at an intermediate position between peaks and valleys of the surface. Our simulations agree with analytical results for sinusoidal boundaries, but can be extended to arbitrary geometries and experimentally obtained data. We find that the apparent slip is independent of the detailed boundary shape, but only given by the distribution of surface heights. Further, we show that slip diverges as the amplitude of the roughness increases which highlights the importance of a proper treatment of surface variations in very confined geometries.

DOI: 10.1103/PhysRevLett.99.176001 PACS numbers: 83.50.Rp, 47.61.+m

In microfluidic systems the surface to volume ratio is large causing boundary effects to be significantly more important than in macroscopic devices. Since even on atomic or molecular scales a perfectly smooth surface is an idealized model, the shape of the boundary is an important property. Additionally, it is of technological interest to design surfaces with well defined structures and properties [1,2]. A commonly investigated surface property is the apparent slip originating, for example, from the surface wettability, electrostatic interactions, impurities, or surface structuring [3]. Navier characterized hydrodynamic slip by postulating that the fluid velocity $v(x)$ at the boundary ($x = 0$) is proportional to the shear rate $\frac{\partial u}{\partial x}$ and the slip length $\beta$ [4]. For macroscopic systems the simple no-slip condition ($\beta = 0$) is a valid assumption. However, if the height of surface variations is not small compared to typical length scales of the system, the position of the boundary is not clearly defined and experiments might detect slip due inaccurately determined wall positions. The influence of roughness on the slip length $\beta$ has been investigated by numerous authors. Roughness leads to higher drag forces and thus to no-slip on macroscopic scales, as shown by Richardson [5] and Jansons [6]. This was experimentally demonstrated by McHale and Newton [7]. Jabbarzadeh et al. performed molecular dynamics (MD) simulations of Couette flow between sinusoidal walls and found that slip appears for roughness amplitudes smaller than the molecular length scale [8]. Also, it can cause pockets to be filled with vapor or gas nano bubbles leading to apparent slip [2,9]. Recently, Sbragaglia et al. applied the LB method to simulate fluids in the vicinity of microstructured hydrophobic surfaces [10] and Varnik et al. [11] have shown that even in small geometries rough channel surfaces can cause flow to become turbulent. A common setup to measure slip is to utilize a modified atomic force microscope (AFM) to oscillate a colloidal sphere in the vicinity of a boundary [12–14]. Vinogradova and Yakubov demonstrated that assuming a wrong position of the surface during measurements can lead to substantial errors in the determined slip lengths [14]. They showed that measurements can be interpreted by assuming a modified sphere radius instead of Navier’s slip condition, so that the position of a no-slip wall would be between peaks and valleys of the rough surface. In this Letter we follow this idea. We answer the question at which position the “effective boundary” has to be placed and study the influence of a wrongly determined wall position numerically.

Panzer et al. gave an analytical equation for $\beta$ for small cosine-shaped surface variations [15]. It is applicable to infinite planes separated by a distance $2d$ being much larger than the highest peaks $h_{\text{max}}$. Surface variations are determined by peaks of height $h_{\text{max}}$, valleys at $h_{\text{max}}$ and given by $h(z) = h_{\text{max}}/2 + h_{\text{max}}/2 \cos(qz)$. Here, $q$ is the wave number and the corresponding slip length is found to be

$$\beta = \frac{h_{\text{max}}}{2} \left( 1 + k \frac{1 - \frac{1}{4} k^2 + \frac{15}{64} k^4 + O(k^6)}{1 + k^2(1 - \frac{1}{2} k^2) + O(k^6)} \right).$$

Higher-order terms cannot easily be calculated analytically and are neglected. Thus, Eq. (1) is valid only for $k = qh_{\text{max}}/2 \ll 1$. However, for realistic surfaces, $k$ can become substantially larger than 1 causing the theoretical approach to fail. Here, only numerical simulations can be applied to describe arbitrary boundaries.

We use a 3D LB model as presented in [16–18] to simulate pressure driven flow between infinite rough walls. Previously, we applied the method to study flows of simple fluids and complex mixtures containing surfactant in hydrophobic microchannels [18,19]. Here, we only shortly describe our method and refer to the literature for details.

The lattice Boltzmann equation, $\eta_i(x + \mathbf{e}_i, t + 1) - \eta_i(x, t) = \Omega_i$, with $i = 0, 1, \ldots, b$, describes the time evolution of the single-particle distribution $\eta_i(x, t)$, indicating the amount of quasi particles with velocity $\mathbf{e}_i$ at site $x$ on a 3D lattice of coordination number $b = 19$, at time-step $t$. 

0031-9007/07/99(17)/176001(4)

© 2007 The American Physical Society
We choose the Bhatnagar-Gross-Krook collision operator 
\[ \Omega_i = -\tau^{-1}[\eta_i(x,t) - \eta_i^eq(u(x,t), \eta_i(x,t))] \]
with mean collision time \( \tau = 1 \) and equilibrium distribution \( \eta_i^eq \). Simulation lattices are 256 lattice units long in flow direction and the planes are separated by 62 sites. Periodic boundary conditions are imposed in the remaining direction allowing us to keep the resolution as low as 16 lattice units. A pressure gradient is obtained as described in [18] and midgrid bounce-back boundaries are applied at the surfaces. An effective boundary position can be found by fitting the parabolic flow profile 
\[ v_z(x) = \frac{1}{2\mu} \frac{\partial P}{\partial z} [d^2 - x^2 - 2d\beta] \] (2)
via the distance \( 2d = 2d_{eff} \). \( \beta \) is set to 0 here and viscosity \( \mu \) as well as pressure gradient \( \frac{\partial P}{\partial z} \) are given by the simulation. To obtain an average value for \( d_{eff} \), a sufficient number of individual profiles at different positions \( z \) are taken into account. Alternatively, the mass flow \( \int v(x) dx \) can be computed to obtain \( 2d_{eff} \). Both methods are equivalent and produce identical results. \( d_{eff} \) gives the position of the effective boundary and the effective height \( h_{eff} \) of the rough surface is then defined by \( d = d_{eff} \) (see Fig. 1).

As rough model surfaces we choose a randomly generated roughness and 3 periodic ones for which the average height or average roughness \( R_a \) is given by \( h_{max}/2 \). While random surfaces are truly uncorrelated, periodic structures correspond to transverse stripes. Cosine-shaped boundaries are given by \( h(x) = h_{max}/2 + h_{max}/2 \cos(qx) \), squares have a height of \( h_{max} \) and are separated by \( h_{max} \) lattice sites. Triangular structures are \( 2h_{max} \) wide and \( h_{max} \) high (see Fig. 2). Randomly generated surface structures are created by choosing for every lattice position of the boundary \( h(x) \) as a random integer between 0 and \( h_{max} \). For determining \( h_{eff} \) we average over 5 surfaces generated with different sequences of uniformly distributed random numbers. All walls are geometrically similar; i.e., the effective height \( h_{eff} \) scales linearly with \( h_{max} \).

In Fig. 3 the effective height \( h_{eff} \) obtained from our simulations is plotted versus \( R_a \) for cosine-shaped surfaces with \( qh_{max}/2 = k = 1, \frac{1}{2} \), \frac{1}{2}, \frac{1}{3} \) (symbols). Error bars would be smaller than the symbols and are therefore not shown. Lines are given by the analytical solution of Eq. (1). For \( k < 1 \) the simulated data agrees within 2.5% with Panzer’s prediction. However, for \( k = 1 \) a substantial deviation between numerical and analytical solutions can be observed because Eq. (1) is valid for small \( k \) only. The inset of Fig. 3 depicts the ratio of \( \beta/h_{max} \) according to the theory of Panzer. In the case of large \( k > 1 \), the theory is not able to correctly reproduce the increase of \( \beta \) with increasing \( h_{max} \) anymore. Instead, \( \beta/h_{max} \) becomes smaller again due to missing higher order contributions in Eq. (1). Our simulations do not suffer from such limitations allowing us to study arbitrarily complex surfaces.

In Fig. 4(a) \( h_{eff} \) is plotted versus \( R_a \) for different types of roughness. By performing a linear fit to the data as given by the lines we find for the uniformly distributed roughness that the position of the effective wall is at \( c = 1.84 \) times the average height of the roughness \( R_a = h_{max}/2 \) or at 92% of the maximum height \( h_{max} \). For squares and triangular structures we find constants of proportionality of \( c = 1.90 \) and \( c = 1.69 \) indicating that the shape of the surface variations indeed affects the position of the effective...
boundary. However, the effect of the shape is small compared to the effect of the height of the variations. All surface structures are geometrically similar causing the linear dependence between $h_{\text{eff}}$ and $R_a = h_{\text{max}}/2$ and $c$ being independent of the lattice resolution. When converting our 3D random roughness into a purely 2D structure, the difference in the measured constant of proportionality $c$ is in the range of the error of the fit algorithm. This is a surprising result since in 3D the flow can pass sidewise around a roughness element. The measured $h_{\text{eff}}$ is found to be independent of the flow velocity over more than 3 decades and does not depend on pressure either, i.e., $h_{\text{eff}}$ is independent of the Reynolds number.

In reality high spikes on a smooth surface may occur, so that the average roughness $R_a$ is much smaller than the maximum height $h_{\text{max}}$. To observe such cases we simulate a triangle geometry with additional void space $a$ between the roughness elements. As maximum height $h_{\text{max}}$ we choose 5 and 10 lattice sites. In similarity to Fig. 4(a) we plot the effective surface height $h_{\text{eff}}$ over the average roughness $R_a$ in Fig. 4(b). In this case the average roughness is smaller than the half of $h_{\text{max}}$, i.e., $R_a = \frac{h_{\text{max}}}{2} \leq h_{\text{max}}/2$. The values of $h_{\text{max}} = 5$ are scaled by a factor of 2 to fit them with the values of $h_{\text{max}} = 10$. Because of the geometrical similarity of the surface structure this scaling is possible. For comparison with Fig. 4(a) the linear fit with slope $c = 1.69$ is plotted. In Fig. 4(b) we see that the maximum height has the strongest influence on the effective height $h_{\text{eff}}$ and not the distance $a$. For small $R_a$ created by a large additional distance $a$, $h_{\text{eff}}$ converges to zero corresponding to a flat surface. For small $a$ the data converge to the triangle geometry as given in Fig. 4(a). For a medium $a = 2h_{\text{max}}$ the effective wall $h_{\text{eff}}$ cannot be approximated by the maximum height $h_{\text{max}}$ nor by the average roughness $R_a$.

We obtained AFM data of a gold coated glass surface with a maximum peak to valley distance of 64 nm. The sample size is 1 $\mu$m$^2$ represented by 512 $\times$ 512 data points. A lattice constant of the LB simulation can be scaled to 1.9 nm by setting the relaxation time $\tau$ to 1.15 and by mapping the speed of sound and the viscosity to the values for water ($c_s = 1.5 \times 10^3$ m/s, $\mu = 1.02 \times 10^{-6}$ m$^2$/s). $h_{\text{eff}}$ can then be measured as in previous paragraphs of this Letter by loading the AFM data onto our simulation lattice. For the simulations presented in this paragraph, the channel width is set to 128 lattice units. The simulated effective height of the gold surface is depicted by the square at $R_a = 21$ nm in Fig. 5. Data points at $R_a = 4$ and 8 are obtained by downscaling the original data set. We find that the distribution of surface heights follows a Gaussian distribution and use this distribution to generate an artificial random surface with identical height distribution. In contrast to the AFM data, our data points are fully uncorrelated, while the gold surface shows distinct structural properties as can be observed in the background images of Fig. 5. For artificial surfaces, the average roughness $R_a$ can be scaled by scaling the width of the distribution of random numbers allowing us to determine $h_{\text{eff}}$ for $R_a$ up to 40 nm. As shown by the dotted line, the measured $h_{\text{eff}}$ linearly depends on $R_a$ with a constant of proportionality $c = 1.43$. The data obtained from the gold coated surface follows the same linear dependence demonstrating that the actual surface shape does not influence its effective position, but only the distribution of heights needs to be known.

The most important question to be answered by our simulations is the effect of a wrongly assumed position of a surface on experimental measurements. As mentioned in the introduction many groups use an approaching method to measure the slip length $\beta$. Here, a colloidal

$\beta \approx 2h_{\text{max}}$, the effective wall $h_{\text{eff}}$ cannot be approximated by the maximum height $h_{\text{max}}$ nor by the average roughness $R_a$.

FIG. 4 (color online). (Color online) (a) Effective height $h_{\text{eff}}$ versus $R_a$ for triangles, blocks (see Fig. 2), and an equally distributed random roughness. (b) $h_{\text{eff}}$ versus $R_a$ for triangles with $h_{\text{max}} = 5$ and 10. The distance between triangles $a$ is varied to obtain the given $R_a$. Values of $h_{\text{max}} = 5$ are scaled by a factor of 2.

FIG. 5 (color online). (Color online) Simulated $h_{\text{eff}}$ versus $R_a$ for gold coated glass and a randomly generated surface with Gaussian distributed heights. The background image shows the gold surface (left) and the artificially generated structure (right).
sphere at the tip of a cantilever immersed in a fluid is oscillated in the vicinity of a surface, or the cylinders of a surface force apparatus (SFA) are brought close to each other. The distance between the surfaces can become very small—even down to contact. To study the influence of the roughness on an apparent slip effect, we assume the surface to be placed at $h_{\text{max}}$ as it is commonly done in experiments [13]. Then, we measure the slip length $\beta$ by fitting Eq. (2). The wrong position of the surface causes a substantial error in the detected slip as can be inferred from Fig. 6. Here, $\beta$ is given versus $R_d$ for randomly generated boundaries with the heights of the surface obstacles following the Gaussian distribution given by the AFM data of the gold surface. For small $R_d$ (and thus large separation of the plates) $\beta$ is in the range of $h_{\text{max}} - h_{\text{eff}}$ and can be neglected in most practical cases. However, the detected slip diverges if $R_d$ becomes large and grows to 80 nm for $R_d \approx 55$ nm. Here, a large $R_d$ is equivalent to the channel width becoming very small—an effect also common in typical surface approaching experiments or microchannel flows. For curved surfaces, as they are utilized in SFA or AFM based slip measurements, the detected $\beta$ can be even larger due to higher order components of the flow field. This might explain experiments reporting large slip lengths of $\beta \approx 100$ nm [3,20].

To conclude, we demonstrated that not properly taking into account surface structures of confining geometries can lead to substantial misinterpretations of measured flow properties. Especially since in typical setups not only roughness, but also, e.g., the wettability, electrostatic interactions or dissolved gasses influence the flow, it is essential for a correct understanding to separate the individual effects. We demonstrated that rough surfaces alone can lead to large apparent slip if the distance between boundaries is small—as it is typical in dynamic microfluidic experiments. These findings were obtained from LB simulations of flow between rough plates showing the existence of an imaginary effective plane where the no-slip boundary condition is valid. We compared our results to analytical calculations of Panzer et al. and found good agreement for small variations ($k < 1$). Large and more realistic perturbations ($k > 1$) can only be covered by simulations as presented here. By simulating flow of water along a gold coated surface and a randomly generated one with identical height distribution, we showed that the position of the effective plane is independent of the boundary structure and that only the height distribution is relevant. Further, these simulations depict the applicability of our method to real surface data. Simulations of flow along surfaces generated from AFM data allow to determine how detected slip might have to be corrected to take surface structures into account.

We thank H. Gong for the AFM data and O.I. Vinogradova, M. Rauscher, and M. Hecht for fruitful discussions. This work was financed within the DFG priority program “nano- and microfluidics” and by the “Landesstiftung Baden-Württemberg”. Computations were performed at the Neumann Institute for Computing, Jülich.

Simulation of fluid flow in hydrophobic rough microchannels

C. Kunert and J. Harting
Simulation of fluid flow in hydrophobic rough microchannels

Christian Kunert and Jens Harting*

Institut für Computerphysik, Pfaffenwaldring 27, 70569, Stuttgart, Germany

(Received 29 April 2008; final version received 29 May 2008)

Surface roughness becomes important in microfluidic setups because the surface to volume ratio becomes large. In such setups the surface roughness is not any longer small compared to the length scale of the system and the wetting properties of the wall have an important influence on the flow. However, the knowledge about the interplay of surface roughness and hydrophobic fluid–surface interaction is still very limited because these properties cannot be decoupled easily in experiments.

We investigate the problem by means of lattice Boltzmann (LB) simulations of rough microchannels with a tunable fluid–wall interaction. We introduce an 'effective no-slip plane' at an intermediate position between peaks and valleys of the surface and observe how the position of the wall may change due to surface roughness and hydrophobic interactions.

We find that the position of the effective wall, in the case of a Gaussian distributed roughness, depends linearly on the width of the distribution. Further, we are able to show that roughness creates a nonlinear effect on the slip length for hydrophobic boundaries.

Keywords: lattice Boltzmann; microflows; apparent slip; roughness; hydrophobic fluid–surface interaction

1. Introduction

In microfluidic systems, boundary effects are significantly more important because of the large surface to volume ratio. The influence of the surface topologies and wetting behaviour on the fluid flow is an important issue in the context of microfluidics and cannot be neglected, since a flat non-interacting surface is always an idealisation. In this paper, we utilise lattice Boltzmann (LB) simulations to investigate the combined influence of roughness and wettability on the fluid flow. In the past the influence of roughness was mainly investigated in the context of (turbulent) boundary layer theory. Already in the 1930s, Nikuradse performed experiments with sand glued inside a pipe in order to obtain the so-called skin friction coefficient, which is still commonly used in engineering applications (Schlichting 1979). Until today, every known theory describing general rough surfaces requires at least one empirical parameter to describe the shape of the surface (Landau and Lifschitz 1986).

This leads to the question of which boundary condition has to be applied at a surface. For more than a 100 years the no-slip boundary condition was successfully applied in engineering applications. Nevertheless, Navier (1823) introduced a slip boundary condition

\[ v(x = 0) = \beta \frac{\partial v}{\partial x} \]

saying that the fluid velocity \( v \) at the boundary \( x = 0 \) is proportional to the velocity gradient \( \partial v/\partial x \). The constant of proportionality is given by the slip length \( \beta \). \( \beta \) depends on many parameters like the wettability, the surface roughness or fluid properties, like the viscosity or molecular interactions. Therefore, it has to be seen as an empirical length that contains many to some extent unknown interactions. However, for simple liquids the measured slip lengths are commonly of the order of up to some tens of nanometers.

The influence of roughness on the slip length \( \beta \) has been investigated by numerous authors. On the one hand, roughness leads to higher drag forces and thus to no-slip on macroscopic scales. Richardson showed that even if on a rough surface a full-slip boundary condition is applied, one can determine a flow speed reduction near the boundary resulting in a macroscopic no-slip assumption (Richardson 1973). This was experimentally demonstrated by McHale and Newton (2004). Jansons (1987) has shown analytically that even few perturbations on flat surfaces lead to mesoscopic slip and therefore macroscopically to no-slip. On the other hand, roughness can cause pockets to be filled with vapour or gas nano bubbles, leading to apparent slip (Du et al. 2004, Joseph et al. 2006). Jabbarzadeh et al. performed molecular dynamics (MD) simulations of Couette flow between sinusoidal walls and found that slip appears for roughness amplitudes smaller than the molecular length scale (Jabbarzadeh et al. 2000).
For the creation of superhydrophobic surfaces which are reducing the drag enormously, it is essential that there are variations in the surface height (Herminghaus 2000).

Varnik et al. (2006) applied the (LB) method to show that even in small geometries rough channel surfaces can cause flow to become turbulent.

Recently, Sbragaglia et al. (2006a) applied the LB method to simulate fluids in the vicinity of microstructured hydrophobic surfaces. In an approach similar to the one proposed in this paper, they modelled a liquid–vapour transition at the surface utilising the Shan–Chen multi-phase LB model (Shan and Chen 1993). The authors were able to reproduce the behaviour of the capillary pressure as simulated by Cottin-Bizonne et al. (2004) using MD simulations quantitatively. They further showed that hydrophobic rough surfaces increase the mass flow which corresponds to an apparent slip effect and that there exists a ‘critical roughness’ at which super-hydrophobic effects appear.

A common setup to measure slip is to utilise a modified atomic force microscope (AFM) to oscillate a colloidal sphere in the vicinity of a boundary (Vinogradova 1995, Bonaccorso et al. 2003, Vinogradova and Yakubov 2006). In these measurements, the drag force is measured by the AFM and compared with the theoretical values. By applying the theory of Vinogradova (1995) to quantify the force correction needed, one can find the corresponding slip length $\beta$. Here, the distance between sphere and wall becomes very small. Therefore a correct knowledge of the boundaries’ properties is very important. Vinogradova and Yakubov (2006) demonstrated that assuming a wrong position of the surface during measurements can lead to substantial errors in the determined slip lengths. They showed that measurements can be interpreted by assuming a modified boundary position instead of Navier’s slip condition, so that the position of a no-slip wall would be between peaks and valleys of the rough surface. In this paper, we follow this idea. We answer the question at which distinct position the ‘effective boundary’ has to be placed and study the influence of a wrongly determined wall position numerically.

In previous papers, we presented a model to simulate hydrophobic surfaces with a Shan–Chen based fluid–surface interaction and investigated the behaviour of the slip length $\beta$ (Harting et al. 2006, Kunert and Harting 2008). We showed that the slip length $\beta$ is independent of the shear rate, but depends on the pressure and on the concentration of surfactant added. Recently, we presented the idea of an effective wall for rough channel surfaces (Kunert and Harting 2007b). We investigated the influence of different types of roughness on the position of the effective boundary. In this paper, we show how the effective boundary depends on the distribution of the roughness elements and how roughness and hydrophobicity interact with each other.

This paper is organised as follows: after the introduction we briefly present the simulation technique and the simulated system. We give an overview on the analytical results of Panzer et al. (1992), which can be compared to our simulations. Then we present new results on flow among Gaussian distributed roughness and the correlation between non-wetting and roughness. We close with a conclusion and summary of our results.

2. The model

We use a 3D LB model as presented in Harting et al. (2005, 2006) and Kunert and Harting (2007b) to simulate pressure driven flow between two infinite rough walls that might be wetting or non-wetting. Previously, we applied the method to study flows of simple fluids and complex mixtures containing surfactant in hydrophobic micro-channels (Harting et al. 2006, Kunert and Harting 2008). Since the method is well described in the literature, we only briefly describe it here.

The LB equation,

$$\eta_i(x + c_i t + 1) - \eta_i(x, t) = \Omega_i, \quad i = 0, 1, \ldots, b,$$

with the components $i = 0, 1, \ldots, b$ describes the time evolution of the single-particle distribution $\eta_i(x,t)$, indicating the amount of quasi particles with velocity $c_i$ at site $x$ on a 3D lattice of coordination number $b = 19$, at time-step $t$. We choose the Bhatnagar–Gross–Krook (BGK) collision operator

$$\Omega_i = -\tau^{-1}(\eta_i(x, t) - \eta^\infty_i(u(x, t), \eta(x, t))),$$

with mean collision time $\tau$ and equilibrium distribution $\eta^\infty_i$ (Succi 2001, Harting et al. 2006). As equilibrium function we choose the expansion

$$\eta^\infty_i = -\xi_i \left[ \left( \frac{c_i \cdot u}{c_i^2} \right) + \frac{(c_i \cdot u)^2}{2c_i^4} - \frac{u^2}{2c_i^4} \right].$$

We use the mid-grid bounce back boundary condition and choose $\tau = 1$ in order to recover the no-slip boundary conditions correctly (He et al. 1997). Interactions between the boundary and the fluid are introduced as described in Harting et al. (2006), namely as mean field body force between nearest neighbours as it is used by Shan and Chen for the interaction between two fluid species (Shan and...
International Journal of Computational Fluid Dynamics

IV-4. Roughness induced boundary slip in microchannel flows


\[
F_{\text{fluid}}(x, t) = -q_{\text{fluid}}(x, t) \delta_{\text{fluid, wall}} \times \sum_{x'} \psi_{\text{wall}}(x', t)(x' - x).
\]  

(4)

The interaction constant \( \delta_{\text{fluid, wall}} \) is set to 0.08 if not stated otherwise (Harting et al. 2006) for a detailed investigation of the influence of different values for the interaction constant. The wall properties are given by the so-called wall density \( \eta_{\text{wall}} \). This enters directly into the effective mass \( \phi^+ = 1 - e^{-\eta_{\text{wall}}/\eta^0} \), with the normalised mass \( \eta^0 = 1 \). With such a model we can simulate slip flow over hydrophobic boundaries with a slip length \( \beta \) of up to five in lattice units (Harting et al. 2006). Higher slip lengths are not easy to realise since the pressure difference between the bulk and at the vicinity of the boundary becomes too large, leading to instabilities. It was shown that this slip length is independent of the shear rate, but depends on the interaction parameters and on the pressure.

In this paper, we model Poiseuille flow between two infinite rough boundaries. Simulation lattices are 512 lattice units long in flow direction and the planes are separated by 128 sites between the lowest points of the roughness elements \( h_{\text{max}} \). Periodic boundary conditions are imposed in the remaining direction allowing us to keep the resolution as low as 16 lattice units. A pressure gradient is imposed in the remaining direction allowing us to keep the maximum height \( d_{\text{max}} \) is kept at a constant of 128 lattice sites.

An effective boundary position can be found by fitting the parabolic flow profile

\[
v(x) = \frac{1}{2\mu} \frac{\partial P}{\partial z} [d^2 - x^2 - 2d\beta],
\]  

(5)

via the distance \( 2d = 2d_{\text{eff}} \). With \( \beta \) set to 0 we obtain the no-slip case. The viscosity \( \mu \) and the pressure gradient \( \partial P/\partial z \) are given by the simulation. To obtain an average value for \( d_{\text{eff}} \), a sufficient number of individual profiles at different positions \( z \) are taken into account. Alternatively, the mass flow \( \int \rho v(x) dx \) can be computed to obtain \( 2d_{\text{eff}} \). Both methods are equivalent and produce identical results. \( d_{\text{eff}} \) gives the position of the effective boundary. The effective height \( h_{\text{eff}} \) of the rough surface measured from the minimum height \( h_{\text{min}} \) is then defined by \( d_{\text{max}} - d_{\text{eff}}/2 \) (Figure 1).

3. Analytic theory

Panzer et al. (1992) calculated the slip length \( \beta \) analytically for Poiseuille flow with rough walls by performing a Fourier expansion of the streaming function \( \Psi \) containing the solution of the Navier–Stokes equations in the laminar case. \( \Psi \) is obtained by a Fourier expansion of the boundary surface and of the pressure field and its solution contains information of an effective boundary. The problem of such an approach is that it works only for small wave numbers.

One would have to take into account an infinite number of terms to achieve a result for arbitrary geometries. Panzer et al. (1992) gave an analytical equation for \( \beta \) in the case of small cosine-shaped surface variations. It is applicable to two infinite planes separated by a distance \( 2d \) being much larger than the highest peaks \( h_{\text{max}} \). Surface variations are determined by peaks of height \( h_{\text{max}} \), valleys at \( h_{\text{max}} \) and given by \( h(z) = h_{\text{max}}/2 + h_{\text{max}}/2 \cos(qz) \). Here, \( q \) is the wave number. Since the surfaces are separated by a large distance, the calculated slip length is equal to the negative effective boundary \( h_{\text{eff}} \) that is found to be

\[
h_{\text{eff}} = -\beta
= \frac{h_{\text{max}}}{2} \left( 1 + k \left[ \frac{1}{1 - \frac{1}{2}k^2} + \frac{19}{12}k^4 + O(k^5) \right] \right).
\]  

(6)

The first and \( k \) independent term shows the linear behaviour of the effective height \( h_{\text{eff}} \) on the average roughness \( R_a = h_{\text{max}}/2 \). Higher order terms cannot easily be calculated analytically and are neglected. Thus, Equation (6) is valid only for \( k = qh_{\text{max}}/2 \ll 1 \). However, for realistic surfaces, \( k \) can become substantially larger than 1, causing the theoretical approach to fail. Here, only numerical simulations can be applied to describe arbitrary boundaries.

Figure 1. The effective boundary height \( h_{\text{eff}} \) is found between the deepest valley at \( h_{\text{min}} \) and the highest peak at \( h_{\text{max}} \) and corresponds to an effective channel width \( d_{\text{eff}} \). The baseline is at \( h_{\text{min}} = 0 \). For the geometries in this paper the average roughness is equal to half the maximum height \( R_a = h_{\text{max}}/2 \). The maximum distance between the plates \( d_{\text{max}} \) is kept at a constant of 128 lattice sites.
To test our method we compare our results with the theoretical model of Panzer and Liu (6). In Figure 2, the normalised effective height $h_{eff}/R_a$ obtained from our simulations is plotted versus $k$ for cosine shaped surfaces with $h_{max}/2 = k = 1, 1.2, 1.4$ (symbols). The line is given by the analytical solution of Equation (6). For $k < 1$ the simulated data agrees within 2.5% with Panzer’s prediction. However, for $k = 1$ a substantial deviation between numerical and analytical solutions can be observed, because Equation (6) is valid for small $k$ only. In the case of large $k > 1$, the theory is not able to correctly reproduce the increase of $\beta$ with increasing $h_{max}$ anymore. Instead, $2\beta/h_{max}$ becomes smaller again due to missing higher order contributions in Equation (6). Our simulations do not suffer from such limitations allowing us to study arbitrarily complex surface geometries (Kunert and Harting 2007b).

4. Distribution of roughness

We showed in a recent paper (Kunert and Harting 2007b) that the position of the effective boundary height is dependent on the shape of the roughness elements, i.e. for strong surface distortions it is between 1.69 and 1.90 times the average height of the roughness $R_a = h_{max}/2$. By adding an additional distance between roughness elements, $h_{eff}$ decreases slowly, so that the maximum height is still the leading parameter. We are also able to simulate flow over surfaces generated from AFM data of gold-coated glass used in microflow experiments by Vinogradova and Yakubov (2006). We find that the height distribution of such a surface is Gaussian and that a randomly arranged surface with a similar distribution gives the same result for the position of the effective boundary although in this case the heights are not correlated. To generate this height distribution we use a Gaussian distributed random height for every lattice point on the surface obtained from a Box–Müller based algorithm (Box and Müller 1958). We can set the width of the distribution $\sigma$ and the average height $R_a$. By scaling $\sigma$ with $R_a$, we obtain geometrically similar geometries. This similarity is important because the effective height $h_{eff}$ scales with the average roughness in the case of geometrical similarity (Kunert and Harting 2007b). As an extension of our previous work, we investigate Gaussian distributed heights with different widths $\sigma$. In Figure 3, the effective height $h_{eff}$ is plotted over the average roughness $R_a$ for $0.054 < \sigma R_a < 0.135$. The height of the effective wall depends linearly on $\sigma$ in the observed range as it can be seen in the inset. We find that the effective height $h_{eff}$ can be fitted by

$$h_{eff} = 1 + 3.1 \sigma.$$  

The range of distributions is limited by the resolution of the lattice. If $\sigma R_a$ becomes too small, the surface is nearly a flat wall, while a too large $\sigma$ leads to a porous medium instead of a channel. In order to perform simulations much larger, simulation volumes are needed in the considered case. The effective height $h_{eff}$ ranges from 1.15$R_a$ to 1.45$R_a$. These values are lower than the effective heights for an equally distributed roughness ($1.84 R_a$). We have found in our previous study that experimentally available surfaces commonly have Gaussian distributed roughness. Therefore, our results shown here can help to estimate $h_{eff}$ in real microchannels.

5. Wettability and roughness

Next, we investigate how roughness and the surface wettability act together. Therefore, we perform simulations with rough channels to which we assign a fluid–wall interaction as given in the introduction (4). For $\eta_{wall}$, we

![Figure 2](image1.png)

**Figure 2.** Effective height $h_{eff}$ normalised by the average roughness $R_a$ over $k = h_{max}/2a$ for a cosine geometry. Symbols denote numerical data and the line is given by Equation (6). For $k > 1$, the slope of the theoretical curve becomes negative, demonstrating that the theory fails for more complex surface structures while simulations are still valid in this regime.

![Figure 3](image2.png)

**Figure 3.** Effective height $h_{eff}$ over average roughness $R_a$ for Gaussian distributed height elements with different width of the distribution $\sigma$. Symbols are the simulation results, lines are a linear fit to the data. In the inset the slope of the fitted lines is plotted over $\sigma R_a$. It shows the linear dependence of the effective height on $\sigma$. 
choose 0.5, 1 and 5. For these values and perfectly smooth surfaces we determine the slip length \( \beta \) to be 0.65, 1.13 and 1.3. As roughness we choose an equally distributed roughness so that \( R_a = h_{\text{max}}/2 \). In order to obtain better results we average the values obtained from four different seeds of the random number generator. The seed determines the sequence of random numbers and therefore the shape of the boundary by leaving the averaged parameters constant.

In Figure 4, we plot the effective height of rough hydrophobic walls versus the average roughness \( R_a \). For \( R_a > 4 \) we find a linear dependence between the average roughness \( R_a \) and the effective height \( h_{\text{eff}} \). The interesting point is that the slope for different \( \eta_{\text{wall}} \) is different. That means that the fluid–surface interaction does not cause a simple offset on the effective height \( h_{\text{eff}} \), but that a nonlinear effect is observed.

To decouple the effect of roughness and wettability we determine the slip length by setting the effective distance \( d_{\text{eff}} \) in Equation (5) to the effective distance for a rough no-slip wall. We then fit the corresponding velocity profile via \( \beta \). In Figure 5, we can see that the slip length \( \beta \) for the strong fluid–wall interaction (\( \eta_{\text{wall}} = 5 \)) first decreases with the average roughness and then raises. For a lower interaction, the slip length is constantly growing and leads to an increase of the slip length for weak fluid–wall interaction (\( \eta_{\text{wall}} = 0.5 \)) by a factor of more than three.

There are two contradicting effects in this system and their interplay can explain the observed behaviour. The decrease of the slip length \( \beta \) is due to an increased friction near the boundary at moderate roughness. The increase has its reason in the reduced pressure near the hydrophobic rough surface, so that the fluid ‘feels’ a smoothed effective surface. Unlike the implementation of Sbragaglia et al. (2006b), our model is not able to model the liquid–gas transition near the surface. We only find a density difference between the bulk fluid and the lattice site just next to the surface. Therefore, we cannot easily calculate the contact angle in our simulations (Benzi et al. 2006). However, the general effect of a pressure drop near a hydrophobic surface is the same as in the model used in Sbragaglia et al. (2006b). For a more detailed study on superhydrophobic surfaces, the strong surface variation as well as the liquid–gas transitions have to be taken into account. This is planned as a future project and beyond the scope of the current paper. Nevertheless, it is important to demonstrate the effect of the pure fluid–surface interaction as it is discussed here.

6. Conclusion
In conclusion, we performed LB simulations of pressure driven flow between two rough plates with and without hydrophobic fluid–wall interaction. We could show that the effective height scales linearly with the width of the distribution \( \sigma \), so that the effective height is \( h_{\text{eff}} = 1 + 3.1\sigma \).

For rough surfaces with a hydrophobic fluid–wall interaction, we could show that there exists a strong nonlinear effect that leads to an increase of the slip length \( \beta \) by a factor of three for small interactions. The behaviour of the slip length can only be explained as a coupled effect since the pure roughness and the pure hydrophobic interaction were investigated in previous studies. For further investigations of super hydrophobicity a phase transition model would be needed.
Acknowledgements
We thank H. Gong for the AFM data and O.I. Vinogradova and M. Rauscher for fruitful discussions. This work was financed within the DFG Priority Program ‘nano- and microfluidics’ and by the ‘Landesstiftung Baden-Württemberg’. Computations were performed at the Neumann Institute for Computing, Jülich and at the Scientific Supercomputing Center, Karlsruhe.

References
Slip flow over structured surfaces with entrapped microbubbles

J. Hyväluoma and J. Harting
Slip Flow Over Structured Surfaces with Entrapped Microbubbles

Jari Hyväruloma and Jens Harting

1Department of Physics, University of Jyväskylä, FI-40014 Jyväskylä, Finland
2Institute for Computational Physics, Pfaffenwaldring 27, D-70569 Stuttgart, Germany

(Received 9 January 2008; published 16 June 2008)

On hydrophobic surfaces, roughness may lead to a transition to a superhydrophobic state, where gas bubbles at the surface can have a strong impact on a detected slip. We present two-phase lattice Boltzmann simulations of a Couette flow over structured surfaces with attached gas bubbles. Even though the bubbles add slippery surfaces to the channel, they can cause negative slip to appear due to the increased roughness. The simulation method used allows the bubbles to deform due to viscous stresses. We find a decrease of the detected slip with increasing shear rate which is in contrast to some recent experimental results implicating that bubble deformation cannot account for these experiments. Possible applications of bubble surfaces in microfluidic devices are discussed.

DOI: 10.1103/PhysRevLett.100.246001

The no-slip boundary condition states that the fluid velocity at a fluid-solid interface equals that of the solid surface. Although this boundary condition has been proven valid at macroscopic scales, it has no microscopic justification and is not fulfilled generally [1]. Its validity was debated already in the early days of fluid mechanics and, due to recent developments in microfluidics, the interest in violation of the no-slip boundary condition has revived [2]. In microfluidics, several experiments have found fluid slip at the boundaries of the flow channels [3,4]. As the slip length (defined below) has typically a magnitude measured in nanometers or micrometers, the appearance of slip does not have noticeable ramifications in macroscopic flows. However, in microfluidic devices with large surface-to-volume ratio, surface properties may dramatically affect the flow resistance. The possibility of engineering slip properties in a controlled way is therefore crucial for microfluidic applications.

The slip at fluid-solid boundaries can be quantified by Navier’s boundary condition, which states that the slip velocity is proportional to the velocity gradient, i.e., \( v_s = b(\partial u / \partial z) \) at the surface \( z = z_0 \). Here, \( b \) is the slip length which is the distance below the surface where the velocity linearly extrapolates to zero. Higher slip length means larger slip and lower friction at the boundary.

Mechanisms behind boundary slip include surface roughness and structural characteristics of roughness [5,6], roughness-induced dewetting on hydrophobic surfaces [7,8], dissolved gas and bubbles on the surface [9–11], as well as wetting properties [12]. Usually, roughness decreases the slip length due to increased dissipation and the roughness-induced slip is just an artifact [5]. Theoretically, it was shown that even slippery surfaces, if rough enough, can provide no-slip boundaries [13]. However, if surfaces are hydrophobic, roughness may increase the slip due to a transition to a superhydrophobic (Fakir or Cassie) state [14]. Here, liquid cannot enter through roughness elements but stays at the top of them. Thus, gas bubbles or layers are formed thereby lubricating the flow due to a reduced liquid-solid contact area. By using a surface patterned with a square array of cylindrical holes, Steinberger et al. found that gas bubbles may also cause an opposite effect; i.e., slip is reduced if microbubbles are present in the holes [10]. Numerically, they found even negative slip lengths for flow over such a bubble mattress. Negative slip means that the effective no-slip plane is inside the channel; i.e., the bubbles increase the flow resistance. These mechanisms are related to so-called effective slip and should be distinguished from the (smaller) intrinsic slip on smooth surfaces. Another peculiarity in the boundary slip is the shear-rate dependence observed in some experiments but not in others [2]. The question of whether the shear-rate dependence is a true property of slip is still to be answered.

As the different mechanisms behind the slip phenomenon are strongly intertwined, the experimental study of a single mechanism is a complicated task. Therefore, numerical simulations are attractive as they provide a controllable way to change a single property of fluid or surface while keeping the others unchanged. Most computer simulations thus far have been performed using molecular dynamics [7,15–17]. For computational reasons molecular dynamics is limited to length scales of tens of nanometers and time scales of nanoseconds, which do not comply with the experimentally relevant scales. Therefore, mesoscopic lattice Boltzmann (LB) simulations have recently been applied to study flow in microchannels or along hydrophobic surfaces [5,8,12,18,19]. This method allows one to reach experimentally relevant scales and preserves those interactions needed to describe the underlying physics.

Our simulations utilize the multiphase LB model by Shan and Chen [20]. Dynamics of the method is governed by a discretized Boltzmann equation

\[
 f_i(\mathbf{r} + \mathbf{c}_i, t + 1) - f_i(\mathbf{r}, t) = -\frac{1}{\tau}[f_i(\mathbf{r}, t) - f_{i,eq}(\mathbf{r}, t)].
\]

where \( f_i \) is a distribution function describing the probability of finding a particle at position \( \mathbf{r} \) at time step \( t \), moving
in lattice direction $c_i$. We use a three-dimensional lattice with 19 discrete velocities. The right-hand side of Eq. (1) models the relaxation of the $f_i$ towards a local equilibrium due to collisions among the particles on a time scale given by the relaxation time $\tau$. The mean-field interactions between particles are given by a force

$$F = G_b \psi(\mathbf{r}) \sum_{c_i} t_i \psi(\mathbf{r} + c_i) c_i,$$

where $\psi = 1 - \exp(-\rho / \rho_0)$ is an effective mass ($\rho$ is the fluid density, $\rho_0$ a reference density), $G_b$ tunes the strength of the interaction, and $t_i$'s are weight factors for different lattice directions. This force term leads to a nonideal equation of state with pressure

$$P = c^2 \rho + \frac{1}{2} c^2 G_b \psi^2,$$

where $c$ is the speed of sound, and it enables simulations of liquid-vapor systems with surface tension. To model the wetting behavior at fluid-solid surfaces, a similar interaction is added between the fluid and solid phases, and the contact angle is tuned by setting a density value $\rho_w$ at the boundaries.

In this Letter we investigate liquid slip in Couette flow, where the flow is confined between two parallel walls. One of the walls is patterned with holes and vapor bubbles are trapped to these holes. Steinberger et al. [10] presented finite-element simulations of flow over rigid "bubbles" by applying slip boundaries at static bubble surfaces. The LB method allows the bubbles to deform if the viscous forces are high enough compared to the surface tension. We are also interested in how surface patterning affects the slip properties of these surfaces, and how bubbles could be utilized to develop surfaces with special properties for microfluidic applications.

In our simulations, the lower wall is static and has the topographical patterning whereas the upper one is smooth and moved with velocity $u_0$. The distance between walls is $d = 1 \text{ \mu m}$ (40 nodes) in all simulations, and the area fraction of holes 0.43 unless stated otherwise. The system boundaries are periodic and a unit cell of the regular array is included in a simulation. To trap bubbles to holes, some heterogeneity is needed at the edges of holes in order to pin the contact line. To this end, we use different wettabilities for boundaries in contact with the main channel and with the hole. The protrusion angle $\varphi$ (see Fig. 1 for definition) is varied by changing the bulk pressure of the liquid. A similar technique to form bubbles on structured surfaces was used experimentally by Bremond et al. [22]. The effective slip length can be calculated from the shear stress $\sigma = \mu d u_0 / d z$ acting on the upper wall, which is obtained from the no-slip boundary condition imposed at the fluid-solid boundaries. Thus the effective slip length reads as

$$b = \mu u_0 / \sigma - d,$$

where $\mu$ is the dynamic viscosity of the liquid. $b$ is measured from the top of the structured surface and $G_b$ is chosen such that the density ratio between liquid and gas is 2.2. This ratio is too small for a realistic description of gas bubbles in a liquid. Also, the interface between both phases is of finite width causing the resistance in the vapor phase. These limitations of multiphase LB models do not influence the qualitative insight obtained from our simulations.

In order to study the effect of a modified protrusion angle and different surface patterning, we use three different arrays of bubbles: i.e., a square array, a rectangular array where the distance of bubbles is larger in one direction than in the other, and a rhombic array. These surfaces have cylindrical holes with radius $a = 500 \text{ nm}$ and the area fraction of the holes is equal in all cases. The shear rate is such that the Capillary number $Ca = 0.16$. The Capillary number is the ratio of viscous and surface forces, i.e., $Ca = \frac{\mu u_0 G_b}{\gamma}$, where $G_b$ and $\gamma$ are the shear rate and surface tension, respectively. Snapshots of simulations are shown in Fig. 2 and the slip lengths obtained are shown in Fig. 1. The observed behavior is similar to that reported in [10], where a square array of holes was studied. In particular, we observe that when the protrusion angle is large enough the slip length becomes negative. We also see that the maximum of the slip length is obtained when the protrusion angle equals zero. Since the area fraction of the bubbles is the same in all three cases, the results clearly indicate that slip properties of the surface can be tailored not only by changing the protrusion angle but also by the array geometry.

Next, the shear-rate dependence of the slip length is investigated. As the shear rate and thus the viscous stresses grow, the bubbles are deformed (see Fig. 2) and the flow field is modified. Results are shown in Fig. 3(a), where the slip length is given as a function of the Capillary number for three different protrusion angles. We also calculate the
Taylor deformation $D = (\ell - a)/(\ell + a)$ of the bubbles by fitting an ellipse to the bubble interface. Here, $a$ and $\ell$ are the minor and major axes of the ellipse, respectively. A slightly superlinear relation between $D$ and $Ca$ is observed [Fig. 3(b)] in accordance with a two-dimensional case in Ref. [23]. We find that increasing shear rates cause the slip length to decrease. This behavior is contradictory to those found in some experiments using surface force apparatuses (see, e.g., Ref. [24]), where a strong increase in the slip is observed after some critical shear rate. This shear-rate dependence has been explained, e.g., with formation and growth of bubbles [9,11]. However, one should notice that these experiments are dynamic in nature while we simulate a steady case. In our simulations there is no formation or growth of the bubbles, but we determine the slip for given bubbles of given size. Our results indicate that the deformation of the bubbles and the changes in the flow field thereby occurring cannot be an explanation for the observed shear-rate dependence. On the other hand, our results are consistent with [5], where it is shown that smaller roughness leads to smaller values of a detected slip. In the present case, the shear reduces the average height of the bubbles and thus the average scale of the roughness decreases as well.

As seen above, the slip properties of a bubble mattress can be tailored by changing the surface patterning. Next, we consider a surface which has slots with a width of 1 $\mu$m, and cylindrical bubbles protruding to the channel with an angle $\varphi = 72^\circ$ [Fig. 2(c)]. We apply shear parallel and perpendicular to the slots, while the area fraction of the bubbles ($\phi$) is varied by changing the distance between the grooves ($L$). According to our results (Fig. 4), $b$ strongly depends on the flow direction and even the qualitative behavior changes. When the flow is parallel to the slot a positive slip is observed, but for perpendicular flow the slip becomes negative. This observation can be explained by means of theoretical predictions of Richardson, who showed that slippery surfaces lead to (macroscopic) no-slip boundaries if rough enough [13]. Our results support these predictions. In the case with flow parallel to the bubbles the streamlines are straight and the flow does not “see” any roughness. However, in the perpendicular direction roughness is caused by the bubbles and negative slip is observed. The inset of Fig. 4 depicts the scaling behavior of the slip length as a function of the solid-area fraction for flow perpendicular to the slots. We see that for small solid-area fraction the slip length obeys the scaling law $b \sim -L \log (1 - \phi)$ as was recently predicted in Ref. [6]. Anisotropic behavior has been observed in the case of flat surfaces composed of stripes with no-slip and perfect-slip boundary conditions [25,26]. Our simulations differ from these studies as our bubbles protrude to the channel thus leading to a larger effect than for a flat surface. In order to understand less idealized surface patterns, we study flow over rectangular holes. The aspect ratio of the holes is chosen such that the longer side is 3 times the smaller one. We observe similar qualitative behavior (Fig. 4), but the difference in slip length between the two flow directions is less pronounced. Obviously, by changing the aspect ratio of the hole, the anisotropic behavior of the slip can be tailored. Because of the striking difference between the slip properties in the two perpen-

![FIG. 2 (color online). Snapshots of simulations of bubbles on structured surfaces. Shown are a square array of bubbles with Capillary number (a) $Ca = 0.02$ and (b) $Ca = 0.4$, and (c) a slot with a cylindrical bubble. In each case a unit cell is shown.](image)

![FIG. 3 (color online). (a) Slip length as a function of capillary number for a square array of bubbles with three different protrusion angles, $\varphi = 63^\circ$, $68^\circ$, and $71^\circ$ (from uppermost to lowermost). In the inset cross sections of liquid-gas interfaces for four capillary numbers are shown. (b) Taylor deformation as a function of Capillary number for the bubble with $\varphi = 71^\circ$. These $Ca$ values correspond to shear rates of the order of $10^{-6}$ to $10^{-7}$ s$^{-1}$.]
dicular directions, we believe that this kind of surfaces may find applications in microfluidic devices. Anisotropic surfaces where the slip is positive in one direction and increases with increasing shear rate [24]. We proposed a hole array and that such a surface with bubbles may yield properties of the surface can be tailored by changing the solid-area fraction, where the dashed line is the theoretical prediction for small solid-area fractions.

The inset shows the normalized slip length as a function of the solid-area fraction, where the dashed line is the theoretical prediction for small solid-area fractions.

To conclude, we simulated Couette flow in a microchannel where one of the walls is patterned and micro-bubbles are attached to the pattern. We found that the slip properties of the surface can be tailored by changing the hole array and that such a surface with bubbles may yield negative slip, i.e., increased resistance to flow, if bubbles are strongly protruding to the channel. Our results can be qualitatively compared to previous results [10], but overcome their limitation of a static liquid-vapor interface. This allowed us to study the influence of the shear rate on the deformation of the interface and its effect on the measured slip. We found that the slip decreases with increasing shear rate demonstrating that shear induced bubble deformation cannot explain recent experimental findings where slip increases with increasing shear rate [24]. We proposed a special surface pattern which can be used to produce surfaces where the slip is positive in one direction and negative in the perpendicular one. Such a surface might be useful to construct microfluidic devices with tunable flow throughput which could be controlled by adding bubble ridges parallel or perpendicular to the flow. In addition, we have shown that the throughput could be tailored by tuning the bulk pressure, i.e., the protrusion angle, or the shear rate.

We are grateful to Jyrki Hokkanen (CSC—Scientific Computing Ltd., Espoo, Finland) for the bubble visualizations. This work was supported by the Academy of Finland (Project No. 7117283), the collaborative research center 716, and the DFG program “nano- and microfluidics.” J. Hyvälouma thanks the staff of ICP, Stuttgart for kind hospitality during his stay.

Simulation of particle-laden flows
Transport phenomena and structuring in shear flow of suspensions near solid walls

A. Komnik, J. Harting, and H. J. Herrmann
Transport phenomena and structuring in shear flow of suspensions near solid walls

A Komnik, J Harting and H J Herrmann

Institut für Computerphysik, Universität Stuttgart, Pfaffenwaldring 27, 70569 Stuttgart, Germany
E-mail: komnik@ical.uni-stuttgart.de, jens@ical.uni-stuttgart.de and hans@ical.uni-stuttgart.de

Received 11 October 2004
Accepted 27 November 2004
Published 7 December 2004
Online at stacks.iop.org/JSTAT/2004/P12003
doi:10.1088/1742-5468/2004/12/P12003

Abstract. In this paper we apply the lattice Boltzmann method and an extension to particle suspensions as introduced by Ladd et al to study transport phenomena and structuring effects of particles suspended in a fluid near sheared solid walls. We find that a particle free region arises near walls, which has a width depending on the shear rate and the particle concentration. The wall causes the formation of parallel particle layers at low concentrations, where the number of particles per layer decreases with increasing distance from the wall.

Keywords: rheology, colloids, rheology and rheophysics

ArXiv ePrint: cond-mat/0408029
1. Introduction

Many manufacturing processes involve the transport of solid particles suspended in a fluid in the form of slurries, colloids, polymers, or ceramics. Examples include the transport of solid material such as grain and drug ingredients in water or other solvents through pipelines. Naturally, these systems occur in mud avalanches and the transport of soil in water streams. It is important for industrial applications to obtain a detailed knowledge of those systems in order to optimize production processes and to prevent accidents.

For industrial applications, systems with rigid boundaries, e.g. a pipe wall, are of particular interest since structuring effects might occur in the solid fraction of the suspension. Such effects are known from dry granular media resting on a plane surface or gliding down an inclined chute [1, 2]. In addition, the wall causes a demixing of the solid and fluid components which might have an unwanted influence on the properties of the suspension. Near the wall one finds a thin lubrication layer which contains almost no particles and causes a so-called ‘pseudo-wall slip’. Due to this slip the suspension can be transported substantially faster and less energy is dissipated.

The dynamics of single-particle motion, interaction with other particles, and effects on the bulk properties are well understood if the particle’s inertia can be neglected. If massive particles are of concern, the behaviour of the system is substantially harder to describe. A number of people have studied particle suspensions near solid walls. These include Sukumaran and Seifert who describe the influence of shear flow on fluid vesicles near a wall [3], Raiskinmäki et al who investigated non-spherical particles in shear flow [4], Jässberg et al who researched hydrodynamical forces on particles near a solid wall [5], Qi and Luo who model the rotational and orientational behaviour of spheroidal particles in Couette flows [6], and Ladd who did work on the sedimentation of homogeneous suspensions of non-Brownian spheres [7].
The last four groups of authors use a simulation technique based on the lattice Boltzmann equation (LBE), that we are also going to use in our simulations. Many other authors have studied similar systems theoretically and experimentally. These include Chaoui and Feuillebois who performed theoretical and numerical investigations on a single sphere in a shear flow close to a wall [8]–[12], and Datta and Shukla who published an asymptotic analysis of the effect of roughness on the motion of a sphere moving away from a wall [9]. Berlyand and Panchenko studied the effective shear viscosity of concentrated suspensions by a discrete network approximation technique [13,14] and Becker and McKinley analysed the stability of creeping plane Couette and Poiseuille flows [15]. There has been a theoretical and experimental study on rotational and translational motion of two close spheres in a fluid [16], and a general approach for the simulation of suspensions has been presented by Bossis and Brady [17]–[19] and applied by many authors. Melrose and Ball have performed detailed studies of shear thickening colloids using Stokesian dynamics simulations [20,21]. Suspensions of asymmetric particles such as fibres, polymers, and large molecules have been of interest to many experimentalists and theoreticians, too. These include Schiek and Shaqfeh and also Babcock et al [22,23].

We expect structuring close to a rigid wall at much smaller concentrations than in granular media because of long range hydrodynamic interactions. In this paper, we study these effects by means of particle volume concentrations versus distance to the wall investigations. Autocorrelation functions of these profiles as well as autocorrelation functions of particle distances from a wall give detailed information about the system’s state and time dependent behaviour. We study the dependence of correlation times on shear rates and achieve insight into the connection of the above-mentioned lubrication layer with the shear rate and particle concentration.

The remaining of this paper is organized as follows: after a description of the lattice Boltzmann method and its extension to particle suspensions in the following section we give an overview of the simulation details in section 3; our results are presented in section 4 and we conclude in section 5.

2. The simulation method

The lattice Boltzmann method is a simple scheme for simulating the dynamics of fluids. By incorporating solid particles into the model fluid and imposing the correct boundary condition at the solid/fluid interface, colloidal suspensions can be studied. Pioneering work on the development of this method has been done by Ladd et al [24]–[26] and we use their approach to model sheared suspensions near solid walls.

2.1. Simulation of the fluid

We use the lattice Boltzmann (hereafter LB) simulation technique which is based on the well-established connection between the dynamics of a dilute gas and the Navier–Stokes equations [27]. We consider the time evolution of the one-particle velocity distribution function \( n(r, v, t) \), which defines the density of particles with velocity \( v \) around the space–time point \( (r, t) \). By introducing the assumption of molecular chaos, i.e. that successive
Transport phenomena and structuring in shear flow of suspensions near solid walls

Binary collisions in a dilute gas are uncorrelated, Boltzmann was able to derive the integro-differential equation for \( n \) named after him [27]:

\[
\frac{\partial n}{\partial t} + \mathbf{v} \cdot \nabla n = \left( \frac{\partial n}{\partial t} \right)_{\text{coll}},
\]

where the left-hand side describes the change in \( n \) due to collisions.

The LB technique arose from the realization that only a small set of discrete velocities is necessary for simulating the Navier–Stokes equations [28]. Much of the kinetic theory of dilute gases can be rewritten in a discretized version. The time evolution of the distribution functions \( n \) is described by a discrete analogue of the Boltzmann equation [26]:

\[
n_i(\mathbf{r} + \mathbf{c}_i \Delta t, t + \Delta t) = n_i(\mathbf{r}, t) + \Delta_i(\mathbf{r}, t),
\]

where \( \Delta_i \) is a multiparticle collision term. Here, \( n_i(\mathbf{r}, t) \) gives the density of particles with velocity \( \mathbf{c}_i \) at \( (\mathbf{r}, t) \). In our simulations, we use 19 different discrete velocities \( \mathbf{c}_i \). The hydrodynamic fields, mass density \( \rho \), momentum density \( \mathbf{j} = \rho \mathbf{u} \), and momentum flux \( \Pi \) are moments of this velocity distribution:

\[
\rho = \sum_i n_i, \quad \mathbf{j} = \rho \mathbf{u} = \sum_i n_i \mathbf{c}_i, \quad \Pi = \sum_i n_i \mathbf{c}_i \mathbf{c}_i.
\]

We use a linear collision operator,

\[
\Delta_i(r, t) = M_{ij}(n_j - n^\text{eq}_j),
\]

where \( M_{ij} \equiv \partial \Delta_i(n^\text{eq})/\partial n_j \) is the collision matrix and \( n^\text{eq}_i \) the equilibrium distribution [29], which determines the rate of scattering between directions \( i \) and \( j \). For mass and momentum conservation, \( M_{ij} \) satisfies the constraints

\[
\sum_{i=1}^M M_{ij} = 0, \quad \sum_{i=1}^M \mathbf{e}_i M_{ij} = 0.
\]

We further assume that the local particle distribution relaxes to an equilibrium state at a single rate \( \tau \) and obtain the lattice BGK collision term [30]

\[
\Delta_i = -\frac{1}{\tau} (n_i - n^\text{eq}_i).
\]

By employing the Chapman–Enskog expansion [27,31] it can be shown that the equilibrium distribution

\[
n^\text{eq}_i = \rho \omega^0 [1 + 3 \mathbf{c}_i \cdot \mathbf{u} + \frac{\omega^1}{2} (\mathbf{c}_i \cdot \mathbf{u})^2 - \frac{\omega^2}{2} \mathbf{u}^2],
\]

with the coefficients of the three velocities

\[
\omega^0 = \frac{1}{3}, \quad \omega^1 = \frac{1}{15}, \quad \omega^2 = \frac{1}{36},
\]

and the kinematic viscosity [26]

\[
\nu = \frac{\eta}{\rho \tau} = \frac{2 \tau - 1}{9},
\]

properly recovers the Navier–Stokes equations

\[
\frac{\partial \mathbf{u}}{\partial t} + (\mathbf{u} \nabla) \mathbf{u} = -\frac{1}{\rho} \nabla p + \frac{\eta}{\rho} \Delta \mathbf{u}, \quad \nabla \mathbf{u} = 0.
\]
2. Fluid–particle interactions

To simulate the hydrodynamic interactions between solid particles in suspensions, the lattice Boltzmann model has to be modified to incorporate the boundary conditions imposed on the fluid by the solid particles. Stationary solid objects are introduced into the model by replacing the usual collision rules (equation (6)) at a specified set of boundary nodes by the ‘link-bounce-back’ collision rule [32]. When placed on the lattice, the boundary surface cuts some of the links between lattice nodes. The fluid particles moving along these links interact with the solid surface at boundary nodes placed halfway along the links. Thus, a discrete representation of the surface is obtained, which becomes more and more precise as the surface curvature gets smaller and which is exact for surfaces parallel to lattice planes. Two discretized spherical surfaces near contact are shown as filled symbols in figure 1. Empty symbols denote the fluid, while filled squares and triangles depict the discretized surface. The crosses (C) denote the shared boundary nodes in contrast to the filled circle (E) which is not a shared boundary node since it is placed on individual links for each sphere.

Numerical results of simulations of a stationary Poiseuille flow between two flat surfaces are in good agreement with the theoretical formula [33]
\[ v = \frac{gL^2}{12\eta} \left( 1 - \frac{4(x - L)^2}{L^2} \right). \] (11)

This is demonstrated in figure 2 which shows the velocity profile versus dimensionless distance $x/L$ from the left wall of a fluid with viscosity $\eta = \frac{1}{9}$ and density $\rho = 1$ under...
Figure 2. Poiseuille flow of a fluid with viscosity $\eta = 1/9$ and density $\rho = 1$ under gravity $g = 10^{-4}$ exerted on each lattice point; channel width $L \in \{8; 16; 32\}$. Gravity is set to $g = 5 \times 10^{-5}$ for channel width $L = 16$. The solid curve represents the expected profile (equation (11)).

constant force $g$ exerted on each lattice point in a channel with width $L$. $g$ is set to $10^{-4}$ for $L \in \{8; 16; 32\}$ and to $g = 5 \times 10^{-5}$ for $L = 16$. $\eta, g, \rho, L$ are in lattice units. The solid curve corresponds to the profile as expected from equation (11).

Since the velocities in the lattice Boltzmann model are discrete, boundary conditions for moving suspended particles cannot be implemented directly. Instead, we can modify the density of returning particles in such a way that the momentum transferred to the solid is the same as in the continuous velocity case. This is implemented by introducing an additional term $\Delta b_i$ in equation (2) [24]:

$$\Delta b_i = \frac{2\omega^c \rho_i u_i \cdot c_i}{c_s^2},$$

with $c_s$ being the velocity of sound and coefficients $\omega^c$ from equation (8).

To avoid redistributing fluid mass from lattice nodes being covered or uncovered by solids, we allow interior fluid within closed surfaces. Its movement relaxes to the movement of the solid body on much shorter timescales than the characteristic hydrodynamic interaction [24]. Figure 3 shows a cut through a three-dimensional box containing a sphere $S$ with periodic boundaries on the front, back, and left and right sides. On the top and bottom sides as well as on the sphere surface we use link-bounce-back boundary conditions. The particle is falling under the influence of gravity $g$. The system size is $32 \times 32 \times 32$ lattice constants $a$ and the particle radius is $4a$. At the beginning the particle and the fluid are at rest and after 3000 time steps the particle attains a steady state. The cut in figure 3 has been generated after 5155 time steps, i.e. well after the system has reached the steady state. Its velocity $u$ is $19\%$ higher than $u_\infty$, expected from the Stokes equation in an infinite fluid system [33]. The difference is caused by the fluid vortices $V$ seen in figure 3, which is due to the periodic boundary conditions and could not arise in an infinite system.
2.3. Boundary nodes shared between two particles

If two particle surfaces approach each other within one lattice spacing, no fluid nodes are available between the solid surfaces (figure 1). In this case, mass is not conserved any longer since boundary updates at each link produce a mass transfer $\Delta m a^3 (a \equiv \text{cell size})$ across the solid–fluid interface [24]. The total mass transfer for any closed surface is zero, but if some links are cut by two surfaces, no solid–fluid interface is available any longer. Instead, the surface of each particle is not closed at the solid–solid contacts any longer and mass can be transferred in between suspended particles. Since fluid is constantly added or removed from the individual particles, they never reach a steady state. In such cases, the usual boundary node update procedure is not sufficient and a symmetrical procedure which takes account of both particles simultaneously has to be used [25]. Thus, the boundary node velocity is taken to be the average of those computed from the velocities of each particle. Using this velocity, the fluid populations are updated (equation (12)), and the force is computed; this force is then divided equally between the two particles.

2.4. Lubrication interactions

If two particles are in near contact, the fluid flow in the gap cannot be resolved by the LB method. For particle sizes used in our simulations ($R < 5a$), the lubrication breakdown in the calculation of the hydrodynamic interaction occurs at gaps less than $0.1R$ [32]. This effect ‘pushes’ particles into each other.

Figure 3. A cut through a three-dimensional system after 5155 time steps. The link-bounce-back boundary conditions are implemented on the surface of the sphere and the walls. The particle Reynolds number is $Re = 6$ and $g$ is the gravitational force. The movement of the interior fluid has already relaxed to solid body movement.
To avoid this force, which should only occur at intermolecular distances, we use a lubrication correction method described in [32]. For each pair of particles a force

$$ F_{\text{lub}} = -6\pi \eta \frac{R_1 R_2}{(R_1 + R_2)^2} \left( \frac{1}{h} - \frac{1}{h_N} \right) \mathbf{u}_{12} \cdot \frac{\mathbf{r}_{12}}{|\mathbf{r}_{12}|}, \quad h < h_N $$

(13)

is calculated, where \( \mathbf{u}_{12} = \mathbf{u}_1 - \mathbf{u}_2, \) \( h = |\mathbf{r}_{12}| - R_1 - R_2 \) is the gap between the two surfaces, and the cut-off distance \( h_N = \frac{2}{3}a \) [26]. For particle–wall contacts we apply the same formula with \( R_2 \to \infty \) and \( h = |\mathbf{r}_{12}| - R_1 \). The tangential lubrication can also be taken into account, but since it has a weaker logarithmic divergence and its breakdown does not lead to serious problems, we do not include it in our simulations.

This divergent force can temporarily lead to high velocities, which destabilize the LB scheme. Instabilities can be reduced by averaging the forces and torques over two successive time steps [25]. In [34] an implicit update of the particle velocity was proposed. This method was then generalized and adopted for the LB approach where two particles are in near contact [26,32]. The drawback of this algorithm is the requirement of two sweeps over all boundary nodes. As we study creeping motion, we use the following simple method. High forces can only arise if the lubrication correction is switched on. Therefore, the lubrication correction \( F_{\text{lub}} \) is limited to a value which would cause a particle acceleration of 0.1 Mach s\(^{-1}\). Such a limitation may lead to particle overlap, but we found that on average there are only five occurrences of this limitation per particle within \( 10^6 \) time steps.

2.5. Particle motion

The particle position and velocity is calculated using Newton’s equations

$$ \mathbf{a} = \frac{1}{m} \mathbf{F} = \dot{\mathbf{v}}, \quad \mathbf{v} = \dot{\mathbf{r}}. $$

(14)

The force \( \mathbf{F} \) is obtained from the calculation of the particle–fluid coupling and the lubrication corrections. Then, the equations are discretized and integrated using the Euler–Cromer method [35]. The velocity \( \mathbf{v}_{n+1} \) and position \( \mathbf{r}_{n+1} \) for the time step \( n + 1 \) are obtained by utilizing the velocity, position, and force from time step \( n \) as well as the time step \( \Delta t = 1 \) and particle mass \( m \):

$$ \mathbf{v}_{n+1} = \mathbf{v}_n + \frac{\mathbf{f}_n}{m} \Delta t $$

(15a)

$$ \mathbf{r}_{n+1} = \mathbf{r}_n + \mathbf{v}_{n+1} \Delta t. $$

(15b)

The same method is applied to particle rotation, with position replaced by angles, velocity by angular velocity, force by torque, and mass by moment of inertia. We do not use more sophisticated methods since they either require additional memory and extra calculations (Verlet [36] and Runge–Kutta [37] approaches) or require the solution of an implicit equation for the velocity at each particle boundary node (the velocity Verlet approach) [38]. Since the forces and velocities in our simulation are rather small and the particle kinetic energy is not conserved between collisions (it is changed by particle–fluid interaction), we do not need to consider the negligible numerical inaccuracies of this method.
3. Simulations

The purpose of our simulations is the reproduction of rheological experiment on computers. First, we simulate a representative volume element of the experimental set-up. Then we can compare our calculations with experimentally accessible data, i.e. density profiles and time dependences of shear stresses and shear rates. We also get experimentally inaccessible data from our simulations, such as translational and rotational velocity distributions and particle–particle and particle–wall interaction frequencies.

The experimental set-up consists of a rheoscope with two spherical plates, whose distance apart can be varied. The upper plate can be rotated either by exertion of a constant force or with a constant velocity, while the complementary value is measured simultaneously. The material between the rheoscope plates consists of glass spheres suspended in a sugar-water solution. The radius of the spheres varies between 75 and 150 μm. For our simulations we assume an average particle radius of 112.5 μm. The density and viscosity of the sugar solution can also be changed.

Because the glass and sugar solution have different light absorption constants, the particle concentration can be obtained by spectroscopic methods. Alternatively, the experimental material can be frozen and analysed with an NMR spectroscope and a three-dimensional porosity distribution can be extracted from the data. Details of the experiment which is currently under development can be found in [39]–[42].

A low resolution ($R \sim 2a$) simulation of a system with the same volume as the experiment would need about 10 GB RAM which is about five times as much as is typically available in current workstations. Each time step, the program sweeps at least twice over the full data set. Simulating one minute of real time would need about three years of CPU time. Increasing the resolution or implementing a curved boundary would increase the computation time even more. Therefore, we calculate only the behaviour of a representative volume element which has the experimental separation between walls, but a much lower extent in the other two dimensions than the experiment. In these directions we employ periodic boundary conditions for particles and for the fluid.

Shearing is implemented using the ‘link-bounce-back’ rule with an additional term $\Delta b_{i}$ at the wall, in the same way as already described for particles (equation (12) with $u_{i}$ now being the velocity of the wall). If a fluid node between the particle and the wall is missing, we use the approach for shared boundary nodes as discussed in section 2.3.

To compare the numerical and experimental results, we need to find characteristic dimensionless quantities of the experiment which then determine the simulation parameters. For this purpose we use the ratio of the rheoscope height and the particle size $\lambda$, the particle Reynolds number $Re$, and the volume fraction of the particles $\phi$:

$$\lambda = \frac{R}{L}, \quad Re = \frac{\gamma R^2 \rho_{f}}{\eta}, \quad \phi = \frac{N(4/3)\pi R^3}{V_{s}},$$

with $R$ being the particle radius, $L$ the height of the rheoscope, $\gamma$ the shear rate, $\eta$ the fluid viscosity, and $N$ the number of particles. In the experiment the suspended particles have a slightly lower density than the fluid. Reducing the particle density would cause instabilities in the LB approach. Therefore, we need to change the acceleration of gravity to a value which would cause the same sedimentation or buoyancy velocity $u$. The Stokes
law \[33\] gives the connection between \( u \) and gravity \( g \):
\[
F = 6\pi R\eta u \Leftrightarrow u = \frac{mg}{6\pi R\eta},
\]
(17)
with the effective mass \( m = \frac{4}{3}\pi R^3(\rho_s - \rho_f) \) of the solid particle. Converting \( u \) to the dimensionless velocity \( u' \) (lattice constant/time step) and inserting simulation parameters into the last equation,
\[
g' = \frac{6\pi R'\eta'u'}{m'},
\]
(18)
where \( m' \) is the mass of the particle without interior fluid, \( R' \) the particle radius, and \( \eta' \) the fluid viscosity (equation (9)).

To provide the simulation results with units, we calculate the length of the lattice constant \( a = R/R' \) and the duration of one time step \( \Delta t = \gamma'/\gamma \); using \( R = 1.125 \times 10^{-3} \) m, \( L = 3.375 \times 10^{-3} \) m, \( \rho_f = 1446 \) kg m\(^{-3}\), \( \rho_s = 1180 \) kg m\(^{-3}\), \( \eta = 0.450 \) kg m\(^{-1}\) s\(^{-1}\), \( \gamma = 10 \) s\(^{-1}\), \( R' = \frac{59}{36} \), \( L = 59 \), \( \nu = \frac{1}{9} \), and \( \rho'_f = 0.7 \) we obtain
\[
a = 0.572 \times 10^{-4} \text{ m}, \quad \Delta t = 1.262 \times 10^{-4} \text{ s}.
\]
(19)
In the simulations presented in the next section we vary the particle Reynolds number to find the dependences of the time needed to attain a steady state and the strength of the structuring effects on the shear rate.

Next we vary the particle volume fraction to study the correlation of velocity profiles and particle concentration. Different volume fractions lead to different correlation effects of particle positions and density profiles.

To check our rule of conversion between numerical and experimental data, we will try to change the fluid viscosity without changing the Reynolds number. This leads to different shear rates and consequently different time steps in the simulation. Higher viscosities lead to longer time steps and thus to shorter simulation times.

A system with \( Re = 4 \times 10^{-6} \) needs about 900 s to attain the steady state. 900 s are equivalent to \( 7 \times 10^6 \) iterations. For such a high number of iterations the program requires about 20 CPU days on a 2 GHz AMD Opteron.

4. Results

Figure 4 shows a snapshot of a suspension with 50 spheres after 5772 500 time steps which are equivalent to 729 s. The vector \( g \) represents the direction of gravity and \( \mathbf{v}_S \) depicts the velocity of the sheared wall.

The particles feel a gravitational acceleration \( g = 0.8 \) m s\(^{-2}\), have a mass \( m = 7.7 \times 10^{-8} \) kg, a Reynolds number \( Re = 4.066 \times 10^{-4} \), and a radius \( R = 1.125 \times 10^{-4} \) m. The system size is \( 1.83 \times 10^{-3} \text{ m} \times 1.83 \times 10^{-3} \text{ m} \times 3.375 \times 10^{-3} \text{ m} \) which corresponds to a lattice size of \( 32 \times 32 \times 59 \). The density of the fluid is set to \( \rho_f = 1446 \) kg m\(^{-3}\) and its viscosity is \( \eta = 450 \) mPa s. The walls at the top and the bottom are sheared with a relative velocity \( \mathbf{v}_s = 3.375 \times 10^{-2} \) m s\(^{-1}\). The system size, particle size, and mass, as well as the gravitational force and all fluid parameters, are fixed throughout the paper. After 200 time steps a linear fluid velocity profile can be observed and the particles are inserted in a random fashion: after choosing a random position for the particle, we check whether
Figure 4. A snapshot of a suspension with 50 spheres (radius $R = 1.125 \times 10^{-4}$ m, mass $m = 7.7 \times 10^{-8}$ kg) at time $t = 729$ s which corresponds to 5772500 time steps. The volume of the simulated system is $1.83 \times 10^{-3}$ m $\times$ $1.83 \times 10^{-3}$ m $\times$ $3.375 \times 10^{-3}$ m = $11.3025 \times 10^{-9}$ m$^3$, acceleration of gravity $g = 0.80$ m s$^{-2}$, and shear velocity $v_s = 3.375 \times 10^{-2}$ m s$^{-1}$. The fluid has a viscosity $\eta = 450$ mPa s and density $\rho_f = 1446$ kg m$^{-3}$. This visualization is a typical example for a system that has reached a steady state: all particles have fallen to the ground due to the gravitational force exerted and most of the system has no particles.

In order to quantitatively characterize structuring effects, we calculate the particle density profile of the system by dividing the whole system into layers parallel to the walls and calculating a partial volume $V_{ij}$ for each particle $i$ crossing such a layer $j$. The scalar $V_{ij}$ is given by the volume fraction of particle $i$ that is part of layer $j$:

$$V_{ij} = \pi(R^2(R_{ij}^{\text{max}} - R_{ij}^{\text{min}}) - \frac{1}{3}(R_{ij}^{\text{max}} - R_{ij}^{\text{min}})).$$

(20)

If the component $r_{i,z}$ perpendicular to the wall of the radius vector $r_i$ of the centre of sphere $i$ lies between $r_{j,z}^{\text{min}}$ and $r_{j,z}^{\text{max}}$, we have

$$r_{j,z}^{\text{min}} = (j - \frac{1}{2})\Delta L_z - R, \quad r_{j,z}^{\text{max}} = (j + \frac{1}{2})\Delta L_z + R,$$
Figure 5. Density profiles from simulations with two different shear rates \( \gamma = 10 \, \text{s}^{-1} \) (a) and \( \gamma = 1 \, \text{s}^{-1} \) (b). Other parameters are equal to those given in figure 4. (a) Shows five peaks with separations of about one particle diameter, which reveal the forming of particle layers. The number of particles per layer is decreasing with increasing distance from the wall, and the change in particle numbers is caused by gravity which is directed perpendicular to the wall at \( z = 0 \). Although we used the same gravity and particle numbers, there are only three peaks in (b) and their width is greater than that of the peaks in (a), demonstrating that the structuring effects strongly relate to the shear rate.

and

\[
R_{ij}^{\text{max}} = \begin{cases} R & \text{if } r_{i,z} + R < r_{j}^{\text{max}} \\ r_{j}^{\text{max}} - r_{i,z} & \text{else}, \end{cases}
\]

\[
R_{ij}^{\text{min}} = \begin{cases} -R & \text{if } r_{i,z} - R > r_{j}^{\text{min}} \\ r_{j}^{\text{min}} - r_{i,z} & \text{else}. \end{cases}
\]

Finally, the sum of all weights associated with a layer is divided by the volume of the layer:

\[
\phi_j = \frac{1}{L_x L_y \Delta L_z} \sum_{i=1}^{N} v_{ij}, \quad \Delta L_z = \frac{L_z}{M},
\]

with \( L_x, L_y \) being the system dimensions between periodic boundaries, \( L_z \) the distance between walls, \( M \) the number of layers, and \( \Delta L_z \) the width of a single layer.

Density profiles calculated by this means for systems with two different shear rates \( \gamma = 10 \, \text{s}^{-1} \) and \( \gamma = 1 \, \text{s}^{-1} \) are presented in figure 5. All other parameters are equal to the set given in the last paragraph. The peaks in figure 5 demonstrate that at certain distances from the wall the number of particles is substantially higher than at other positions. The first peak in both figures is at slightly below one particle diameter, which can be explained by a lubricating fluid film between the first layer and the wall which is slightly thinner than one particle radius. Due to the small amount of particles, time dependent fluctuations of the width of the lubricating layer cannot be neglected and a calculation of the exact value is not possible. The five peaks in figure 5(a) have similar distances which are equal to one particle diameter. These peaks can be explained by closely packed parallel layers of
Transport phenomena and structuring in shear flow of suspensions near solid walls

![Graphs showing autocorrelation](image)

**Figure 6.** The autocorrelation $r_\tau$ of the density profiles shown in figure 5. In both plots the autocorrelation converges to a fixed value. The dashed lines correspond to fitted constants and the points to the simulation data. In (a) the system has a shear rate $\gamma = 10 \text{s}^{-1}$ and in (b) $\gamma = 1 \text{s}^{-1}$. The higher shear rate leads to higher particle velocities and therefore to a higher collision frequency. Therefore, this system attains the steady state faster and is less correlated. This is confirmed by the smaller limit of $r_\tau$, which is 0.361 instead of 0.480 in (b).

particles. Due to the linear velocity profile in the $z$ direction of the fluid flow, every layer adopts the local velocity of the fluid resulting in a relative velocity difference between two layers of about $2R\gamma$. These layers stay stable in time with only a small number of particles being able to be exchanged between them.

Figure 5(b) only shows three peaks with larger distances than in figure 5(a). However, the average slopes of the profiles are identical for the two shear rates. For smaller shear rates, velocity differences between individual layers are smaller, too. As a result, particles feel less resistance while moving from one layer to another. Every interlayer transition distorts the well-defined peak structure of the density distribution resulting in there being only three clearly visible peaks in figure 5(b).

With changing time, the first peak stays constant for both shear rates. The shape, number, and position of all other peaks are slightly changing in time.

To acquire a quantitative description of this effect we calculate the autocorrelation function of the density profile (figure 6) $r_\tau^l$ for each individual layer $l$,

$$r_\tau^l(i\Delta t) = \frac{1}{(T - i)} \left( \sum_{j=1}^{T-i} \phi^l(j\Delta t) \phi^l((i + j)\Delta t) \right) \frac{1}{(1/T)} \sum_{j=1}^{T} T(\phi^l(j\Delta t))^2,$$

with $\Delta t$ being the time step, $i$ the current iteration, and $T$ the total number of time steps.

Averaging the $r_\tau^l$ over all $M$ layers gives

$$r_\tau(i\Delta t) = \frac{1}{M} \sum_{l=1}^{M} r_\tau^l(i\Delta t),$$

which is presented in figure 6 for two systems with shear rates $\gamma = 10$ and $1 \text{s}^{-1}$. The autocorrelation starts—as given by the definition—at one. Then, it decreases and converges to constant values at about $i\Delta t = 15 \text{s}$ for $\gamma = 10 \text{s}^{-1}$ and at $i\Delta t = 25 \text{s}$
Transport phenomena and structuring in shear flow of suspensions near solid walls

![Figure 7](image.png)

Figure 7. The autocorrelation \( r_\tau \) of the particle distances from a wall for two systems with shear rates \( \gamma = 10 \text{ s}^{-1} \) (a) and \( \gamma = 1 \text{ s}^{-1} \) (b). All simulation parameters except \( \gamma \) correspond to those given in figure 4. The straight line in the plots with logarithmically scaled \( r_\tau \) axes shows the exponential connection of \( r_\tau \) and time \( t \): \( r_\tau \propto e^{-t/\tau_{\text{corr}}} \). The typical correlation time \( \tau \) evidently depends on the shear rate. We find \( \tau_{\text{corr}} = 5.5 \) and 38.64 s for \( \gamma = 10 \) and 1 \text{ s}^{-1} respectively.

for \( \gamma = 1 \text{ s}^{-1} \). We obtain these values by fitting the data to a constant function using a non-linear least-squares Marquardt–Levenberg algorithm. The computed values of the autocorrelation function are different for the given shear rates: \( r_\tau \) is 0.480 for \( \gamma = 1 \text{ s}^{-1} \) and 0.361 for \( \gamma = 10 \text{ s}^{-1} \). It is evident that for a simulation without shear the autocorrelation converges to one, because after sedimentation the density profile should not change. Thus, \( \phi^l(k\Delta t) \) is almost constant for all \( k \), and \( r_\tau \to \infty \). For \( \gamma \to \infty \) the velocity and the collision frequency are increasing and the correlation decreases for high shear rates. Therefore, the expectation that for smaller shear rates the autocorrelation converges to higher values than for larger shear rates is confirmed.

Another possibility for computing typical correlation times of structured layers is to analyse the autocorrelation of particle distances from one of the walls. For this purpose we replace the volume fraction \( \phi^l(k\Delta t) \) of layer \( l \) by the distance of particle \( l \) from one of the walls \( r^l_z \) in equation (22). Then the acquired data are averaged for all \( N \) particles:

\[
\begin{align*}
    r_\tau(i\Delta t) &= \frac{1}{N} \sum_{l=1}^{N} r^l_z(i\Delta t).
\end{align*}
\]

(24)

The dependence of \( r_\tau \) on time calculated by this means is shown in figure 7, where simulation parameters are as in the previous section. It is possible to fit the data to an exponential function of the form

\[
    r_\tau = \exp\left(-\frac{t}{\tau_{\text{corr}}}\right),
\]

(25)

where \( \tau_{\text{corr}} \) is the characteristic correlation time. We get \( \tau_{\text{corr}} = 5.5 \) s and \( \tau_{\text{corr}} = 38.64 \) s for \( \gamma = 10 \) and 1 \text{ s}^{-1}, respectively. This fully corresponds to the behaviour expected from the density profiles: shorter correlation times are related to higher shear rates. At higher shear rates the mean velocity of the particles is also higher. Thus they collide with other
Transport phenomena and structuring in shear flow of suspensions near solid walls

Figure 8. The dependence of the correlation time $\tau_{\text{corr}}$ on the shear rate $\gamma$. All data points lie on a straight line, which indicates an exponential behaviour of $\tau_{\text{corr}}$: $\tau_{\text{corr}} \propto e^{-\gamma/\gamma_0}$, with $\gamma_0 = 4.78 \, \text{s}^{-1}$.

particles and walls more often. Each collision contributes a random uncorrelated force component to the equation of motion, which reduces the correlation of particle positions.

We also expect a strong dependence on the average particle concentration and different values for the gravitational force. For a larger number of particles in the system, the effective viscosity changes, which influences the collision rate and reduces correlation times.

Also for very high shear rates there should be non-zero correlation times, and we expect a non-linear connection between shear rate and correlation time. Therefore, we did the same calculations for more different shear rates and we plot the correlation times $\tau_{\text{corr}}$ versus shear rates $\gamma$ in figure 8. Rescaling the axis of ordinates logarithmically we obtain a straight line again. For high shear rates the correlation time is decreasing exponentially:

$$\tau_{\text{corr}} = \tau_{\text{corr}}^{\text{max}} \exp\left(-\frac{\gamma}{\gamma_0}\right),$$  \hspace{1cm} (26)

with $\tau_{\text{corr}}^{\text{max}} = 47.24 \, \text{s}$ being the maximum correlation time and $\gamma_0 = 4.78 \, \text{s}^{-1}$ being a characteristic shear rate.

Another interesting property is the distribution of particle distances, which can be acquired by calculating the distances $r(i, j, l)$ of all particle pairs. Because of the periodic boundary conditions we also account for particle pairs if one of them is shifted in one of the nine possible periodic directions. The maximum distance is then limited by the smallest system dimension $L_{\text{min}}$ (here $1.83 \times 10^{-3} \, \text{m}$):

$$p_k = \frac{1}{N^7} \sum_{i=1}^{N} \sum_{j=i+1}^{N} \sum_{l=1}^{9} \left\{ \begin{array}{ll}
\frac{1}{\delta^2} & \text{if } \left[ \frac{r(i, j, l)}{\Delta r} \right] = k \\
0 & \text{else}
\end{array} \right.$$

$$= \frac{1}{N^7} \sum_{i=1}^{N} \sum_{j=i+1}^{N} \sum_{l=1}^{9} \left\{ \begin{array}{ll}
\frac{1}{\delta^2} & \text{if } \left[ \frac{r(i, j, l)}{\Delta r} \right] = k \\
0 & \text{else}
\end{array} \right.$$

$$= \frac{1}{N^7} \sum_{i=1}^{N} \sum_{j=i+1}^{N} \sum_{l=1}^{9} \left\{ \begin{array}{ll}
\frac{1}{\delta^2} & \text{if } \left[ \frac{r(i, j, l)}{\Delta r} \right] = k \\
0 & \text{else}
\end{array} \right.$$
Transp ort phenomena and structuring in shear flow of suspensions near solid walls

Figure 9. Distributions of particle distances (a) and of differences of particle distances from one of the walls (b). All simulation parameters are equal to those given in figure 4. In both figures we show histograms for two different times: $t = 0$ and 865 s. In (a) the dominant peak moves from $\delta \sim 1.2$ to $\delta \sim 1$ showing the compression of the system under the influence of gravity. There is also a shoulder on the right of the peak showing that many particles have distances between one and two particle radii. In (b) we see a linear profile for $t = 0$, caused by the homogeneous particle distribution at the beginning of the simulation. The highest peak is at $\delta = 0$, which is caused by particles belonging to identical layers. The following peak is due to particles from adjacent layers.

with

$$\Delta r = \frac{L_{\text{min}}}{M},$$

$$N' = \sum_{i=1}^{N} \sum_{j=i+1}^{N} \sum_{l=1}^{N} \left\{ \begin{array}{ll} 1 & \text{if } r(i, j, l) < L_{\text{min}} \\ 0 & \text{else,} \end{array} \right.$$

$$r(i, j, l) = |r_i - (r_j + s_l)|,$$

$$s_l = \left( \begin{array}{c} 0 \\ 0 \\ 0 \end{array} \right), \left( \begin{array}{c} \pm L_x \\ 0 \\ 0 \end{array} \right), \left( \begin{array}{c} 0 \\ \pm L_y \\ 0 \end{array} \right), \left( \begin{array}{c} \pm L_x \\ \pm L_y \\ 0 \end{array} \right).$$

Each $p_k$ corresponds to an interparticle distance

$$\delta = (k + \frac{1}{2})\Delta r. \quad (28)$$

In a homogeneous system the number of particles at a distance $\delta$ from a given particle is proportional to the surface of a sphere of radius $\delta$. Thus to avoid overweighting of larger particle distances we divide the number of particles at distance $\delta$ by $\delta^2$ (equation (27)).

In figure 9(a) we present two distributions for a system with 50 particles. The first distribution corresponds to the start of the simulation at $t = 0$ s. It has one peak between $\delta = 1$ and 2, after which it decreases continuously. The measurement in the steady state gives the second distribution at $t = 865$ s. This distribution also has one peak, but it is narrower and much higher than that at $t = 0$ s. The position of this peak corresponds to a distance $\delta$ slightly higher than one particle diameter from each other; i.e. most particles are at a distance of about one particle diameter.
By computing only the distribution of the components of particle distances perpendicular to the walls \( r_z \) for the same system we get the results plotted in figure 9(b). We do not need to account for periodic boundaries here, which results in a smaller number of counted particle pairs and slightly worse statistics:

\[
p_k = \frac{2}{N(N-1)} \sum_{i=1}^{N} \sum_{j=i+1}^{N} \left\{ \begin{array}{ll} 1 & \text{if } \left[ \frac{r_z(i,j)}{\Delta r} \right] = k \\ 0 & \text{else,} \end{array} \right. \quad (29)
\]

with \( N \) being the number of particles in the system, and

\[
r_z(i,j) = |r_{zi} - r_{zj}|, \quad \Delta r = \frac{L_z}{M},
\]

where \( M \) is the resolution of the distribution, \( L_z \) is the distance between the walls, and \( \delta \) is calculated using equation (28). The distributions in figure 9(b) show that for \( t = 0 \) s there are no structured layers. This histogram gives a nearly straight line with a negative slope.

Let us consider a homogeneous system completely filled with spheres, \( x \) being the number of particles in an individual layer. Then, \( x(x-1) \) is the number of particle pairs at distance \( \delta = L_z - 1 \). For \( \delta = L_z - 2 \) there are two pairs of layers at that distance. Thus, we get \( 2x(x-1) \) particle pairs. Reducing the distance by one particle diameter increases the number of possible particle pairs by \( x(x-1) \). The total number of particle pairs is proportional to \( L_z - \delta \). This argumentation is valid for all homogeneously filled systems. This consideration is confirmed by figure 9(b) for \( t = 0 \) s. The slope of the line should be proportional to the volume fraction because \( x \) gets larger for higher particle numbers. After 865 s there is a peak for \( \delta = 0 \) showing that most pairs belong to the same layer. The second peak belongs to \( \delta \sim 1 \), i.e. to particle pairs in adjacent layers.

Both histograms in figure 9 have a clear dependence on time. To visualize that dependence we calculate the mean \( \langle \delta \rangle \) and standard deviation \( \sigma \) of the particle distances:

\[
\langle \delta \rangle = \frac{1}{M} \sum_{i=1}^{M} p_i \delta_i, \quad (31a)
\]

\[
\sigma = \sqrt{\frac{1}{M} \sum_{i=1}^{M} p_i (\delta_i - \langle \delta \rangle)^2}, \quad (31b)
\]

with \( M \) being the resolution of the distribution \( p(\delta) \), and \( \delta_{\text{max}} \) being the maximum particle distance. For the distribution of differences of particle distances from one of the walls, \( \delta_{\text{max}} \) is set to \( 3.375 \times 10^{-3} \) m \( \equiv 15 \) particle diameters. For the distribution of particle distances, \( \delta_{\text{max}} = 1.8 \times 10^{-3} \) m \( \equiv 8.136 \) particle diameters. In figure 10(a) the mean \( \langle \delta \rangle \) (left ordinate) and \( \sigma \) (right ordinate) are plotted over time. For short times, the mean decreases almost linearly to \( t \sim 16 \) s; then the slope approaches 0 and \( \langle \delta \rangle \) fluctuates around \( \delta = 4.65 \) until the end of the simulation is reached (\( t \sim 900 \) s). At the beginning of the simulation it is not possible to recognize the characteristics of the evolution of \( \sigma \).
Transport phenomena and structuring in shear flow of suspensions near solid walls

Figure 10. The mean $\langle \delta \rangle$ and standard deviation $\sigma$ of particle distances (a) and differences of particle distances from one of the walls (b) versus time for a system of 50 particles with radius $R = 1.125 \times 10^{-4}$ m, acceleration of gravity $g = 0.80$ m s$^{-2}$. The Reynolds number is $Re = 4.066 875 \times 10^{-4}$, and the shear rate $\gamma = 10$ s$^{-1}$. In both figures $\langle \delta \rangle$ and $\sigma$ converge to specific values within about 15 s and they fluctuate around fixed values until the end of the simulation is reached ($t \approx 900$ s). In both cases the standard deviation is smaller than the mean. The mean particle distance converges to $\langle \delta \rangle \approx 4.6$ and its standard deviation converges to $\sigma \approx 3.2$. The mean difference of particle distances from the wall converges to $\langle \sigma \rangle \approx 1.2$ and $\sigma \approx 1$.

For times between 6 and 15 s the points of $\sigma$ lie nearly on a straight line. At $t \approx 16$ the slope of $\sigma$ becomes zero and $\sigma$ is fluctuating around a mean $\sigma \approx 3.2$, like $\langle \delta \rangle$.

In Figure 10(b) the mean $\langle \delta \rangle$ and standard deviation $\sigma$ of the differences of particle distances from one of the walls are plotted versus time. To calculate these values we used equations (31). The evolution of $\langle \delta \rangle$ and $\sigma$ is linear between $t = 5$ and 12 s. The slope then vanishes and only some random fluctuations can be seen around $\langle \delta \rangle = 1.4$ and $\sigma = 1.2$ particle diameters for $t \geq 17$ s. Note that the particle distances have already attained a steady state at 15 s while the density profile needs 158 s.

To study the demixing phenomena already demonstrated in figure 5, we analyse the dependence of the particle and fluid velocities on the distance from the wall. Both profiles in Figure 11 are for a system with shear rate $\gamma = 10$ s$^{-1}$ at $t = 865$ s. All other simulation parameters are kept as in the last section. In addition to the velocity profiles we plot a solid line corresponding to the fluid velocity profile of a system without particles. The values of fluid velocities at the walls ($z = 0$ and 15 particle diameters) exactly match the wall velocities: $v(0) = 0$ and $v(15) = 3.375 \times 10^{-2}$ m s$^{-1}$. For $2 < z < 6$ the two profiles agree very well with each other. No particles are present above $z = 6$ and the fluid velocity profile is exactly linear. We do not have any particle velocity data in this case. Below $z = 2$ the profiles separate and for $z < 0.5$ the fluid velocity profile corresponds to the expected profile for a particle free system, while the particle velocities stay constant. This can be seen in Figure 11(b), which shows an enlarged particle velocity profile. The particle velocity converges to $u_s = 1.1 \times 10^{-3}$ m s$^{-1}$ for $z \to 0$. For higher values of $z$ it is linear, but its slope is about 10% smaller than the slope of the solid line. For $z > 6$ the velocity profile is linear, but it rises faster than expected in order to fit the wall velocity
Transport phenomena and structuring in shear flow of suspensions near solid walls

**Figure 11.** The velocity profile of a system with shear rate $\gamma = 10 \, \text{s}^{-1}$ and mean volume fraction $\phi = 0.026$ versus distance from one of the walls $z$. Solid lines correspond to the expected fluid velocity profile in the absence of particles. At the walls ($z = 0$ and $15$) the fluid velocity is identical to the wall velocities and in the particle filled region the fluid and solid velocities are equal, confirming the validity of the no-slip boundary conditions on particle and wall surfaces. The velocity of the solid particles does not disappear at the wall, unlike the fluid velocity, but converges to a fixed value instead.

**Figure 12.** The velocity of the ‘pseudo-wall slip’ versus shear rate for two different volume concentrations. The dependence of the velocity is linear. The slope of the line gives an effective width of the particle free region near the wall. The width is $1.16 \times 10^{-4}$ and $9.23 \times 10^{-5}$ m for $\phi = 0.026$ and $0.053$, respectively. Narrower particle free regions are caused by higher forces due to the weight of particles being above the particle layer near the wall.

at $z = 15$ and to conserve the validity of the no-slip boundary conditions at the walls. Since the particle and fluid velocities are identical for $2 < z < 6$, the no-slip boundary conditions on the particle surface are shown to generate correct results, too.

The dependence of the particle velocity near the wall on the shear rate is studied in figure 11(b) for $\gamma = 1, 0.1$, and $0.25 \, \text{s}^{-1}$ by calculating the particle velocities for $z \rightarrow 0$. Figure 12 depicts these velocities versus shear rate and their linear dependence is clearly
Transport phenomena and structuring in shear flow of suspensions near solid walls

**Figure 13.** Distributions of particle velocities averaged over $5.55 \times 10^6$ time steps of the steady state. (a) Shows the component perpendicular to the wall and (b) that perpendicular to the shear velocity. While the means of both velocity distributions are zero, their widths differ. The movement perpendicular to the wall is restricted by walls and structured layers. Both sets of data cannot be fitted by a Gaussian distribution.

The data from simulations with higher particle concentration ($\phi = 0.053$ instead of $\phi = 0.026$) also give a straight line but with smaller slope. The slopes can be interpreted as an effective width of the particle free layer near the wall, which is $1.16 \times 10^{-4}$ and $9.23 \times 10^{-5}$ m for $\phi = 0.053$ and 0.026, respectively. The value for $\phi = 0.053$ is slightly smaller than the particle radius and in good agreement with observations from the particle concentration profiles in figure 5(a). The smaller width of the particle free layer at higher particle concentrations is caused by the higher pressure on the lowest particle layer. Since the system cross section is the same in the two simulation series, with higher particle number the number of the particle layers increases. Thus, the resulting gravitational force on the lowest layer increases proportionally to the particle number. However, the reciprocal width of the particle free layer is not proportional to the particle number because this layer is caused by the competition of gravity and the resistance to particle motion perpendicular to the wall. This is not constant but rather approximately proportional to the reciprocal value of the distance [43].

We calculate the distributions of velocity components in three directions: perpendicular to the wall (figure 13(a)), parallel to the shear direction (figure 14), perpendicular to the shear direction and parallel to the wall (figure 13(b)). In figure 14 one can clearly recognize three peaks. The first peak is at $1.1 \times 10^{-3}$ m s$^{-1}$, exactly corresponding to the wall slip velocity for the given shear rate. It can be seen that this peak corresponds to the lowest particle layer and all three peaks have a distance of $2.5 \times 10^{-3}$ m s$^{-1}$ which matches the product of the shear rate and particle diameter. We have already seen the formation of particle layers near the wall, with a distance of about one particle diameter (figure 5). Therefore, we assume that each peak in figure 14 is caused by one single particle layer. The height of the peaks decreases with the velocity since the number of particles per layer is being reduced with time (see figure 5). This reduces the probability of finding a particle with the velocity of the layer, which on the other hand is decreasing with the distance from the wall (figure 11). Thus, for higher wall distances we get higher velocities and smaller particle numbers. The width of the peaks...
Figure 14. The distribution of the particle velocity component parallel to the shear direction averaged over $5.55 \times 10^6$ time steps of the steady state. The peaks are separated by $2.5 \times 10^{-3}$ m s$^{-1}$, starting at $1.1 \times 10^{-3}$ m s$^{-1}$. Dividing this velocity difference ($2.5 \times 10^{-3}$ m s$^{-1}$) by the shear rate (i.e. 10 s$^{-1}$) results in the particle diameter, since the average width of the layer corresponds to one particle diameter. These layers move against each other with a relative velocity corresponding to the shear rate.

The means of both distributions are zero as expected. The distribution of particle velocities perpendicular to the wall is narrower because the movement to the wall is restricted by lubrication interactions. The change between the layers is restricted by the differences in layer velocities, but it is not completely impossible. The data for both distributions do not follow a Gauss distribution.

5. Conclusion

We successfully applied the lattice Boltzmann method and its extension to particle suspensions to simulate transport phenomena and structuring effects under shear near solid walls. We adopted the simulation parameters to the experimental set-up of Buggisch et al [39] and are able to obtain not only qualitatively comparable results, but also values that quantitatively correspond to experimentally measured parameters. We hope to be able to report on direct comparisons between our theoretical results and the experimental results of Buggisch et al in the near future.

We have shown that the density profile has several peaks, confirming the formation of particle layers. The density profile is changing in time, but its autocorrelation function converges to a non-zero value. On the other hand, the autocorrelation function of particle distances from a wall converges exponentially to zero, resulting in a fixed correlation time.
This time is exponentially dependent on the shear rate. Furthermore, we have shown that the particle distances attain a steady state at a much earlier state of the simulation than the density profile. We have also shown the occurrence of a ‘pseudo-wall slip’ of particles, exhibited by a particle free fluid layer near the wall. The velocity of this slip has a linear dependence on the shear rate. It is possible to calculate an effective width of the particle free layer, which depends on the particle concentration.

A natural extension of this work would be to increase the size of the simulated system in order to reach the dimensions of the experimental set-up. Even though the number of LB time steps of our simulations is extremely high already, even longer runs would be desirable.

It would also be interesting to study the behaviour of the system for higher particle densities and higher shear rates. However, improvements of the method are mandatory in order to prevent instabilities of the simulation. Without further improvement of the simulation method, the maximum particle volume concentration is limited to 0.3 and the maximum available shear rate is about 10 s$^{-1}$. A possible solution of this well-known problem is the replacement of the velocity update by an implicit scheme [26,32]. The artefacts caused by the interior fluid can be removed by slightly modifying the coupling rules [44]. We have not implemented this because of the high numerical effort, which is caused by the necessity to sweep over all boundary nodes twice, in order to redistribute the mass from nodes being covered by the spheres.

Acknowledgments

We thank Professor Hans W Buggisch and Silke Muckenfuß for fruitful discussions regarding their experimental set-up and for providing the parameters of their experiment. We also thank Thomas Ihle for support during the time of code implementation.

References


Transport phenomena and structuring in shear flow of suspensions near solid walls

[33] Landau L D and Lifschitz E M, 1966 Hydrodynamik (Berlin: Akademieverlag)
[34] Lowe C P, Frenkel D and Masters A J, 1995 J. Chem. Phys. 103 1582
Anomalous distribution functions in sheared suspensions

J. Harting, H. J. Herrmann, and E. Ben-Naim
Anomalous distribution functions in sheared suspensions

J. Harting, H. J. Herrmann and E. Ben-Naim

1 Institute for Computational Physics - Pfaffenwaldring 27, D-70569 Stuttgart, Germany, EU
2 Computational Physics, IHB - Schafmattstr. 6, ETH Zürich, CH-8093 Zürich, Switzerland
3 Theoretical Division and Center for Nonlinear Studies, Los Alamos National Laboratory
Los Alamos, NM 87545, USA

received 16 May 2008; accepted in final form 16 June 2008
published online 27 June 2008

PACS 02.50.Ey – Stochastic processes
PACS 47.55.Kf – Particle-laden flows
PACS 77.84.Nh – Liquids, emulsions, and suspensions; liquid crystals

Abstract – We investigate velocity probability distribution functions (PDF) of sheared hard-sphere suspensions. As observed in our Stokes flow simulations and explained by our single-particle theory, these PDFs can show pronounced deviations from a Maxwell-Boltzmann distribution. The PDFs are symmetric around zero velocity and show a Gaussian core and exponential tails over more than six orders of magnitude of probability. Following the excellent agreement of our theory and simulation data, we demonstrate that the distribution functions scale with the shear rate, the particle volume concentration, as well as the fluid viscosity.

Copyright © EPLA, 2008

Introduction. – To describe the statistics of complex systems, often probability distribution functions (PDF) are utilized. These distributions have been found to be of non-Gaussian shape in numerous fields of physics, including astrophysics [1], flow in porous media [2], turbulence [3], granular media [4–6], or suspensions [7–9]. However, the underlying processes are often not understood. In this letter, we focus on particularly important systems showing non-Gaussian velocity PDFs, namely sedimenting hard-sphere suspensions confined between sheared walls (see fig. 1). They appear in river beds, blood examinations, industrial food production, the application of paint, and many more situations. Detailed experiments have been performed for more than a hundred years, but questions about the microstructure or structural relaxations of the sediment are still not well understood.

Numerous authors have found that the PDF of particle velocities $P(v)$ is not of similar shape as for an ideal gas, i.e. like a Maxwellian. Instead, $P(v)$ can show a pronounced non-equilibrium shape, where the probability of high velocities is substantially larger [7,8]. In this letter we present a single-particle theory and simulations to show that such non-equilibrium distributions can be described as a consequence of an irreversible driving process, where particles on average gain energy by one mechanism, but lose energy by another one. Here, energy is gained from the shear or gravitational forces causing particles to collide. Contrarily, energy is dissipated due to viscous damping.

This causes $P(v)$ to consist of a Gaussian core and exponential tails. Even though we focus on a well-defined system here, the processes described are of general nature and can be applicable to ostensibly different setups.

Experimentally, Rouyer et al. [10] studied quasi-2D hard-sphere suspensions and found a stretched exponential $P(v)$ with concentration-dependent exponents between 1 and 2 corresponding to exponential distributions for high
concentrations and Gaussians for small particle counts. These results contradict theoretical predictions of a transition from exponential to Gaussian with increasing volume concentration [7,8]. However, both experimental and theoretical studies do not present sufficient statistics over more than 2-4 decades. It is important to note that if one does not have enough data points for high quality PDFs, final answers on the nature of the function cannot be given. Indeed, in the process of analyzing our data we found that even stretched exponentials can fit PDFs with purely exponential tails and Gaussian centers if only two to four decades of probability are covered. But as soon as more data is added, the exponential nature of the tails becomes distinct and it is impossible to fit the whole PDF with a single function.

Here, we overcome such limitations by presenting PDFs consisting of up to $10^{10}$ particle displacements each —allowing a statistics superior to any previous work. Our data does not show deviations from purely exponential tails over 6-8 decades of probability. We show that $P(v_i)$ scales linearly with the shear rate, volume concentration and viscosity.

The simulated system is a 3-dimensional setup as shown in fig. 1. Top and bottom walls are at distance $N_y$ and sheared with shear rate $\dot{\gamma} = 2\dot{\gamma}_{\text{shear}}/N_y$. All other boundaries are periodic. A body force $f$ acting on the otherwise neutrally buoyant particles can be added to mimic gravity. If turned on, this force causes strong density gradients in the system as can be observed in fig. 1. We consider 384 to 1728 initially randomly placed suspended particles of equal radius $a$ corresponding to a particle volume concentration $\phi$ between 6.8% and 30.7%. The particle Reynolds number $Re = \gamma 2a/\nu$ is kept between 0.012 and 0.07.

Simulation method. — The simulation method is composed of a lattice Boltzmann solver (LB) for the fluid and a molecular dynamics (MD) algorithm for the motion of particles. This approach and recent improvements were originally introduced by Ladd and coworkers [11–14] and are well established in the literature [11–16]. Thus, we only shortly describe it here.

The LB approach allows to calculate long-range hydrodynamic interactions between particles, by utilizing a discretized version of Boltzmann’s equation [17]. Here, positions $x$ are discretized on a 3D lattice with 19 discrete velocities $c_i$ pointing from a site to its neighbors. Every $c_i$ is related to a single-particle distribution function $f_i(x)$ which is streamed to neighboring sites at every time step. After streaming, a collision takes place where the individual $f_i(x)$ relax towards an equilibrium distribution $f_i^eq$. Local mass and momentum density are given by moments of the $f_i(x)$. The movement of suspended particles is modeled by Newton’s equation of motion and appropriate boundary conditions are imposed at solid/liquid interfaces to exchange momentum. We find a particle radius of $a = 1.25$ lattice sites sufficient, since for larger radii $P(v_i)$ does not change significantly anymore, while the computational effort increases substantially. Also, in low-density simulations long-range hydrodynamic interactions dominate which are correctly reproduced even by small particles. For dense systems, exact lubrication forces between particle pairs and between particles and walls are applied [11–13]. If many particles come close to each other (less than 0.1 lattice spacings), a cluster implicit method is used for updating forces in the MD algorithm [13].

Theoretical aspects. — Our theoretical model is based on the balance between viscous dissipation and shear forcing in steady state. The shear and the resulting particle collisions are modeled by random, diffusive forcing. Due to this forcing, the velocity $v_i$ of the $j$-th particle changes as $\frac{dv_j}{dt} = \xi_j$ where $\xi_j$ is a white noise, $\langle \xi_j \rangle = 0$ and $\langle \xi_j(t)\xi_i(t') \rangle = 2D\delta_{ij}\delta(t-t')$. In parallel, particles slow down because of the viscous fluid. In accordance with the traditional drag law, the velocity decreases according to $dv_j/\eta = \beta v_j$, resulting in the exponential decay $v(t) = v(0)e^{-\beta t}$ in the absence of forcing. We model this viscous damping by reducing each time unit $\Delta t$ the velocity by a factor $\eta = e^{-\beta \Delta t}$, according to $v \rightarrow \eta v$. In a sheared fluid, there is a well-defined time scale for re-encounters with the boundary, setting the time scale for the damping process. This damping process was used by van Zon et al. as a model for forced granular media [18,19].

The velocity distribution obeys the linear but non-local equation

$$\frac{\partial P(v)}{\partial t} = D \frac{\partial^2 P(v)}{\partial v^2} + \frac{1}{\eta} P(\frac{v}{\eta}) - P(v).$$

(1)

The first term on the right-hand side represents changes due to diffusive forcing and the next two terms represent changes due to viscous damping. In our theory, interactions between particles are represented through the random forcing process reflecting that a particle undergoes diffusion as influenced by all other particles. In steady state, the left-hand side of eq. (1) vanishes. We note that the shape of $P(v)$ is independent of the diffusion constant $D$. Indeed, by making the scaling transformation $v \rightarrow v/\eta$
Anomalous distribution functions in sheared suspensions

\[ v \rightarrow v/\sqrt{D} \] we can eliminate \( D \) and assume without loss of generality that \( D = 1 \). The shape of the distribution depends on the dissipation parameter \( \eta \) alone. The moments

\[ M_n = \int dv v^n P(v) \] (2)

satisfy the recursion relation

\[ M_n = n(n - 1)(1 - \eta^n)^{-1} M_{n-2}. \] (3)

Since the distribution is symmetric, \( P(v) = P(-v) \), the odd moments vanish and, starting with \( M_0 = 1 \), the even moments are

\[ M_{2n} = (2n)! \prod_{k=1}^{n} \frac{1}{1 - \eta^{2k}}. \] (4)

Of particular interest is the normalized 4th moment

\[ \kappa = M_4/M_2^2 = 6/(1 + \eta^2). \] (5)

The distribution is close to exponential for strong drag, \( \kappa \rightarrow 6 \) as \( \eta \rightarrow 0 \) and close to Gaussian for very weak drag, \( \kappa \rightarrow 3 \) as \( \eta \rightarrow 1 \). This is confirmed by the limiting behaviors of all moments

\[ M_{2n} \rightarrow \begin{cases} (2n)!, & \eta \rightarrow 0, \\ (2n-1)!!(1 - \eta^{-n})^{-1}, & \eta \rightarrow 1. \end{cases} \] (6)

The leading large-velocity behavior can be derived by using a heuristic argument. For sufficiently large velocities, the term \( \eta^{-1}P(v\eta^{-1}) \) is negligible and hence,

\[ \frac{d^2}{dv^2} P(v) = P(v). \] (7)

Thus, \( P(v) \) has an exponential tail, \( P(v) \sim \exp(-|v|) \). The prefactor can be obtained from the Fourier transform \( F(k) \) that equals an infinite series

\[ F(k) = \int_0^\infty dv e^{ikv} P(v) = \prod_{m=0}^{\infty} \left[1 + k^2 \eta^{2m}\right]^{-1}. \] (8)

This expression follows from the steady-state analog of (1), \( (1 + k^2)F(k) = F(\eta k) \). The simple poles at \( \pm i \) close to the origin imply an exponential decay, i.e.,

\[ P(v) \approx A(\eta) \exp(-|v|), \] (9)

when \( |v| \rightarrow \infty \). Re-summation yields the residue to this pole, and in the prefactor

\[ A(\eta) = \frac{1}{2} \exp\left(\sum_{n=1}^{\infty} \frac{1}{n} \eta^{2n} \right). \] (10)

The exponential behavior is robust in the limit \( v \rightarrow \infty \). However, in the weak-drag limit, \( \eta \rightarrow 1 \), the exponential behavior holds only for extremely large velocities. When \( \eta \rightarrow 1, (\eta = 1 - \varepsilon, \varepsilon \rightarrow 0) \) we expand the denominator in \( A(\eta) \). Keeping only the dominant terms simplifies the sum to

\[ \sum_{n=1}^{\infty} 1/2n^2 = \frac{\pi^2}{12\varepsilon} \] (11)

and, to leading order, the prefactor is \( A \propto \frac{1}{2} \exp[\pi^2/12\varepsilon] \). Therefore, under weak damping, there is a crossover between a Maxwellian behavior as follows from (6) and an exponential one,

\[ P(v) \sim \begin{cases} \exp \left(\frac{\varepsilon v^2}{2}\right), & v \ll \varepsilon^{-1}, \\ \exp \left(\frac{\pi^2}{12\varepsilon^2} - |v|\right), & v \gg \varepsilon^{-1}. \end{cases} \] (12)

The two expressions match \( P(v) \sim \exp(-\varepsilon^{-1}) \) at the crossover velocity \( v \approx \varepsilon^{-1} \). Interestingly, the crossover to a non-Maxwellian does not affect the leading behavior of the moments.

In summary, the theory predicts the non-equilibrium shape of the PDF as an interplay between energy being injected by a diffusive thermostat and dissipation due to the fluid drag. In general, the high-velocity tail is exponential. The theoretical results shown later in this letter are given by a Monte Carlo solution of the steady-state case of eq. (1). In these simulations, \( N \) particles are characterized by a velocity \( v_i \). The velocities change through two independent processes: damping and random forcing. In the damping process, the velocity is reduced by a fixed factor \( v_i \rightarrow \eta v_i \). In the forcing process, the particle velocity changes by a random increment \( v_i \rightarrow v_i + \xi \), where \( \xi \) has zero mean and a unit variance. The steady-state distributions were obtained using over \( 10^{18} \) points from simulations with \( 10^8 \) particles.

**Results.** – First, we consider suspensions with constant \( \phi \) and various shear rates under the influence of a body force \( f \). The dependence of \( P(v_i) \) on \( \gamma^* \) for three representative values is depicted in fig. 2. \( P(v_i) \) is symmetric and \( \langle v_i \rangle = 0 \) for all cases considered in this letter. As shown in the lower inset, the not normalized \( P(v_i) \) widen for higher \( \gamma^* \). However, a very good scaling is observed: all normalized curves collapse onto a single one. In the upper inset we show the influence of \( \gamma^* \), \( v_i \), \( v_i \), and \( v_i^{\text{RMS}} \), as expected from the theory, \( \gamma^* \) only sets a scale for the velocity corresponding to a linear relation between \( \gamma^* \) and \( v_i^{\text{RMS}} \). \( P(v_i) \) (after deducting the shear velocity) and \( P(v_y) \) also show Gaussian cores and exponential tails, although their width and height are different. Thus, the general shape of distributions in different directions is essentially identical and they are therefore not shown. To obtain an insight into the properties of \( P(v_i) \), we compute the cumulant \( \kappa \) and find that for all simulation parameters studied in this letter it varies between 3.8 and 4.6. Knowing \( \kappa \), we can compute \( \eta \approx \sqrt{6/\kappa - 1} \). Due to the large number of data points in our histograms, we calculate \( \kappa \) for periods of 1 million time steps each
and use the arithmetic average of the last 5 million time steps of a simulation run. We find that $\kappa$ varies by up to 10% within a single simulation which is of the same order as the difference of the individual PDFs in fig. 2. Thus, we average the different curves as well in order to obtain a value for the cumulant to be utilized for the Monte Carlo solution of the steady-state case of eq. (1). For the collapse in fig. 2 we get $\eta = 0.73$. As depicted here, the solid line given by the theory and the simulation data excellently agree over the full range of six decades of probability.

Next, we consider neutrally buoyant suspended hard spheres ($f = 0$) under shear. The shear rate is kept fixed and the particle concentration is varied between $\phi = 6.8\%$ and 30.7%. Due to hydrodynamic interactions, the particles tend to move to the center of the system, i.e., to an area where the shear is low creating a depleted region close to the walls. However, this effect does not change the general shape of $P(v)$. The corresponding normalized $P(v_z)$ are presented in fig. 3a. As depicted in the figure, all PDFs except for the lowest particle concentration $\phi = 6.8\%$ (circles) collapse onto a single curve. At very low $\phi$, the tails of $P(v_z)$ are still not fully converged due to the limited number of particle-particle interactions taking place within the simulation time frame. Again, the solid line in fig. 3a is given by the steady-state solution of eq. (1) with $\eta = 0.69$ being obtained from the 4th moment of $P(v_z)$. As before, simulation and theory agree very well. The full circles in fig. 4 depict the dependence of $v_{\text{RMS}}$ on $\phi$. For concentrations of at least $\phi = 13.6\%$, $v_{\text{RMS}}^{\phi}$ can be fitted by a line with slope $1.8 \cdot 10^{-4}$. The disagreement of the linear fit for low $\phi$ is consistent with the not fully converged PDFs, as shown in fig. 3a. By keeping all parameters except the kinematic viscosity $\nu$ constant, the dependence of $v_{\text{RMS}}$ on $\nu$ can be studied. As demonstrated by the squares depicting the dependence of $v_{\text{RMS}}$ on $\nu$ in fig. 4, $v_{\text{RMS}}$ and thus $P(v_z)$ are independent of the viscosity. Thus, the steady-state curve obtained for different volume concentrations is identical to the one for different $\nu$, as shown in fig. 3b. It would be interesting to study the influence of the body force $f$ on the shape of the PDF. However, $f$ and the shear forces are in a subtle interplay since the height of the steady-state sediment depends on both parameters and thus influences the local concentration. To investigate this behavior is beyond the scope of this letter.

**Conclusion.** – To conclude, the non-equilibrium PDFs reported in this letter are a consequence of the irreversible nature of the driving process. On average, particles gain energy by external forces causing particle-particle collisions but lose energy by viscous damping. Unlike in an
ideal gas, these two mechanisms are not interchangeable. In other words, one cannot reverse the arrow of time and observe the same behavior. Our theoretical model captures this irreversibility through the competition between two non-equivalent driving mechanisms: energy dissipation through a multiplicative process and energy injection through an ordinary additive diffusive thermostat.

The theory describes all aspects of the distribution as demonstrated by an excellent agreement with our coupled LB/MD simulations of sheared suspensions: the velocity distribution functions $P(v_z)$ exhibit Gaussian cores and exponential tails over at least 6 orders of magnitude of probability. This finding is consistent with experimental results as given for example in [4]. We also note that the complete shape of the distribution can be characterized by a single parameter, the normalized 4th moment that has a one-to-one correspondence with the theoretical dissipation parameter. Further, we confirmed that $P(v_z)$ scales linearly with the particle volume concentration as well as the shear rate and is independent of the solvent’s viscosity.

While various authors report on transitions between Gaussian and (stretched) exponential tails [7,8,10], our results do not confirm such transitions. A likely reason for the previously published results might be the lack of sufficient statistics. If only two to four orders of magnitude of probability are covered, $P(v_z)$ can be reasonably well fitted by pure Gaussians or distributions with (stretched) exponential tails as well. For a conclusive answer on the shape of the PDF, we found that at least five to six orders of magnitude of probability and a proper normalization are necessary. Experiments in strongly driven granular particles suspended in a fluid provide evidence of exponential velocity distributions [4]. In order to close the question of the general validity of the findings presented in this letter, further high-quality experimental or numerical data would be needed.

***

We thank A. J. C. Ladd for providing his simulation code and for fruitful discussions. This work was supported by the DFG priority program “nano- and microfluidics”, US-DOE grant DE-AC52-06NA25396, and the “Landesstiftung Baden-Württemberg”. HJH thanks the Max Planck prize. We thank P. L. Krapivsky for fruitful discussions and the Institute for Pure and Applied Mathematics at University of California, Los Angeles for hospitality. The computations were performed at the Jülich Supercomputing Centre.

REFERENCES

Simulation of claylike colloids

M. Hecht, J. Harting, T. Ihle, and H. J. Herrmann
We investigate the properties of dense suspensions and sediments of small spherical silt particles by means of a combined molecular dynamics and stochastic rotation dynamics (SRD) simulation. We include van der Waals and effective electrostatic interactions between the colloidal particles, as well as Brownian motion and hydrodynamic interactions which are calculated in the SRD part. We present the simulation technique and first results. We have measured velocity distributions, diffusion coefficients, sedimentation velocity, spatial correlation functions, and we have explored the phase diagram depending on the parameters of the potentials and on the volume fraction.

I. INTRODUCTION

We simulate claylike colloids, for which in many cases the attractive van der Waals forces are relevant. They are often called “peloids” (Greek: claylike). The colloidal particles have diameters in the range of some nm up to some μm. In general, colloid science is a large field, where many books have been published [1–6]. The term peloid originally comes from soil mechanics, but particles of this size are also important in many engineering processes. Our model system of Al₂O₃-particles of diameter 0.5 μm suspended in water is an often used ceramics and plays an important role in technical processes. In soil mechanics [7] and ceramics science [8], questions on the shear viscosity and compressibility as well as on porosity of the microscopic structure which is formed by the particles, arise [9,10]. In both areas, usually high volume fractions (Φ > 20%) are of interest. The mechanical properties of these suspensions are difficult to understand. Apart from the attractive forces, electrostatic repulsion strongly determines the properties of the suspension. Depending on the surface potential, which can be adjusted by the pH value of the solvent, one can either observe the formation of clusters or the particles are stabilized in suspension and do sediment only very slowly. Hydrodynamic effects are also important for sedimentation experiments. Since typical Péclet numbers are of order 1 in our system, Brownian motion cannot be neglected.

In summary, there are many important factors which have to be included in a model which describes peloids in a satisfying way. Such a model is needed to gain a deeper understanding of the dynamics of dense colloidal suspensions. A lot of effort has been invested by applying different simulation methods, which have their inherent strengths but also some disadvantages. Simplified Brownian dynamics (BD), such as in the work of Hütter [11], does not contain long-ranged hydrodynamic interactions among particles at all. The computational cost is low, since hydrodynamics is reduced to a simple Stokes force and thus large particle numbers can be handled. BD with full hydrodynamic interactions utilizes a mobility matrix which is based on the Oseen- or Rotne-Prager-Yamakawa tensor approximations which are exact in the limit of zero Reynolds number and zero particle volume fraction [12,13].

This technique faces the main problem that the computational effort scales with the cube of the particle number due to the inversion of matrices.

The lattice Boltzmann (LB) method, on the other hand, is numerically efficient and intrinsically contains hydrodynamic interactions. Ladd and Verberg give an overview over the LB method and describe how to include stress fluctuations [14]. Adhikari et al. add noise to their model by introducing a noise term for every lattice velocity and node [15]. However, the discussion about the correct inclusion of thermal fluctuations is still ongoing [15,16]. Pair-Drag simulations have been proposed by Silbert et al. [17], which include hydrodynamic interactions in an approximative way. They have focused on suspensions with high densities (up to 50%) of uncharged spherical colloidal particles. Here we use stochastic rotation dynamics (SRD) [18,19], a recently developed method to simulate fluid flow, and combine this with a molecular dynamics (MD) simulation for the suspended particles. SRD is a particle-based method which does not show any numerical instabilities, contains thermal fluctuations intrinsically, and is simple to implement. Many important issues in fluctuation fluid dynamics such as sedimentation [20], vesicles in flow [21], polymers in flow [22], and reacting fluids [23] have been addressed very recently using this method. In this paper, first we discuss the main points of the MD part of our simulation code. Second we present the SRD method in the context of our work, and then we describe two alternative ways of coupling the two parts of the simulation and point out the advantages and disadvantages of these two possibilities. After that, we analyze the time scales which are relevant for the peloids, we want to simulate. Based on the insights of this section we show in the following section how to determine the simulation parameters. Then we describe how we have tested our simulation code and present first results in the following section. Finally in the last section we draw a conclusion and summarize shortly the model we have presented.
II. MOLECULAR DYNAMICS

The colloidal particles in our simulation are represented by three-dimensional spheres. In order to correctly model the colloidal particles is described by Derjaguin-Landau-Verwey-Overbeek (DLVO) theory [10,11,24]. If the colloidal particles are suspended in a solvent, typically water, ions move into solution, whereas their counter-ions remain in the particle due to a difference in electric double layer. It has been shown (see [24]) that the resulting electrostatic interaction between two of these particles can be described by an exponentially screened Coulomb potential

$$V_{\text{Coul}} = \pi \epsilon_0 \varepsilon \left( \frac{4 \kappa T}{z e} \right)^{\frac{1}{2}} \frac{d^2}{r} \exp(-\kappa[r-d]),$$

(1)

where $d$ denotes the particle diameter and $r$ is the distance of the particle centers. $z$ is the charge of the ions, $e$ is the elementary charge, $T$ is the temperature, $\Psi_0$ denotes the effective surface potential, and $\kappa$ is the inverse Debye screening length. In addition the behavior is determined by the attractive van der Waals interaction which can analytically be integrated over the two spheres. This leads to the second part of the DLVO potential:

$$V_{\text{vew}} = -A_B \left[ \frac{d^2}{r - d^2} + \frac{d^2}{r} + 2 \ln \left( \frac{r^2 - d^2}{r^2} \right) \right].$$

(2)

$A_B$ is the Hamaker constant which involves the polarizability of the particles and of the solvent. The DLVO potentials are plotted in Fig. 1 for six typical examples with different depths of the secondary minimum. The primary minimum has to be modeled separately, as discussed below.

To avoid the particles penetrating each other, one needs a repulsive force depending on their overlap. We are using a Hertz force described by the potential

$$V_{\text{Hertz}} = K(d-r)^{5/2} \quad \text{if } r < d,$$

(3)

where $K$ could be expressed by the elastic modulus of $\text{Al}_2\text{O}_3$. This would determine the simulation time step, but to keep computational effort relatively small, we determine the time step using the DLVO potentials as described later on and then choose a value for $K$. Two aspects have to be considered: $K$ has to be big enough, so that the particles do not penetrate each other by more than approximately 10% and it may not be too big, so that numerical errors are kept small, which is the case when the collision time is resolved with about 20 time steps. Otherwise total energy and momentum are not conserved very well in the collision.

Since DLVO theory contains the assumption of linear polarizability, it holds only for large distances—i.e., the singularity when the two spheres touch—does not exist in reality. Nevertheless, there is an energy minimum about 30$k_B$ deep, so that particles which come that close would very rarely become free again. To obtain numerical stability of our simulation, we model this minimum by a parabolic potential, some $k_BT$ deep (e.g., $6k_BT$). The depth of the minimum in our model is much less than in reality, but the probability for particles to be trapped in the minimum has to be kept low enough so that only few of them might escape during simulation time.

The long-range hydrodynamic interaction is taken into account in a separate simulation for the fluid as described below. This can only reproduce interactions correctly down to a certain level. On shorter distances, a lubrication force has to be introduced explicitly in the molecular dynamics simulation as described in [25]. The most dominant mode, the so-called squeezing mode, is an additional force

$$F_{\text{lub}} = -\left( v_{\text{rel}} r \right)^2 \frac{6 \pi \eta r_{\text{rel}}^2}{r - r_1 - r_2},$$

(4)

with

$$r_{\text{rel}} = \frac{r_1 r_2}{r_1 + r_2},$$

(5)

between two spheres with radii $r_1$ and $r_2$ and the relative velocity $v_{\text{rel}}$. $\eta$ is the dynamic viscosity of the fluid. $F_{\text{lub}}$ diverges if particles touch each other. Therefore, we limit the force by introducing a minimum radius, where the force reaches its largest allowed value. The potential is shifted accordingly to smaller particle distances, so that the maximum force is reached for particles touching each other.

The Hertz force also contains a damping term in the normal direction,

$$F_{\text{damp}} = -(v_{\text{rel}} r) \beta \sqrt{r - r_1 - r_2},$$

(6)

with a damping constant $\beta$, and for the transverse direction a viscous friction proportional to the relative velocity of the particle surfaces is applied.
SIMULATION OF CLAYLIKE COLLOIDS

For the integration of the translational motion we utilize a velocity Verlet algorithm [see Chap. 3.2.1] to update the velocity and position of particle $i$ according to the equations

$$x_i(t + \delta t) = x_i(t) + \delta t v_i(t) + \frac{\delta t^2 F_i(t)}{m},$$

(7)

$$v_i(t + \delta t) = v_i(t) + \frac{F_i(t) + F_i(t + \delta t)}{2m}.$$

(8)

For the rotation, a simple Euler algorithm is applied:

$$\omega_i(t + \delta t) = \omega_i(t) + \delta t T_i,$$

(9)

$$\theta_i(t + \delta t) = \theta_i(t) + F(\theta_i, \omega_i, \delta t),$$

(10)

where $\omega_i(t)$ is the angular velocity of particle $i$ at time $t$, $T_i$ is the torque exerted by noncentral forces on the particle $i$, $\theta_i(t)$ is the orientation of particle $i$ at time $t$, expressed by a quaternion, and $F(\theta_i, \omega_i, \delta t)$ gives the evolution of $\theta_i$ of particle $i$ rotating with the angular velocity $\omega_i(t)$ at time $t$.

The concept of quaternions [26] is often used to calculate rotational motions in simulations, because the Euler angles and rotation matrices can easily be derived from quaternions. Using Euler angles to describe the orientation would give rise to singularities for the two orientations with $\theta = \pm 90^\circ$. The numerical problems related to this fact and the relatively high computational effort of a matrix inversion can be avoided using quaternions.

We have switched off dissipative forces and checked if the total energy and each component of the total momentum are conserved. We have verified this for the molecular dynamics simulation for the simulation of the fluid and for the coupled simulation separately.

We also checked that our implementation of the molecular dynamics code is correct by simulating eight large particles with Hertz repulsion and Coulomb friction in a closed box at a volume fraction of $\Phi = 20\%$. We checked that the collisions are realistic—i.e., that the individual angular velocities for two particles interacting in a noncentral collision before and after they have touched are consistent.

III. STOCHASTIC ROTATION DYNAMICS: SIMULATION OF THE FLUID

The stochastic rotation dynamics method introduced by Malevans and Kapral [18, 19] is a promising tool for a coarse-grained description of a fluctuating solvent, in particular for colloidal and polymer suspensions. The method is also known as “real-coded lattice gas” [27] or as “multi particle-collision dynamics” (MPCD) [28]. It can be seen as a “hydrodynamic heat bath,” whose details are not fully resolved but which provides the correct hydrodynamic interaction among embedded particles [29]. SRD is especially well suited for flow problems with Peclet numbers of order 1 and Reynolds numbers on the particle scale between 0.05 and 20 for ensembles of many particles. The numerical effort scales linearly with the number of embedded colloidal particles unlike in Brownian dynamics, and only one random number per node (for the choice of the rotation matrix) is needed in contrast to fluctuating lattice Boltzmann. For Peclet numbers of order 1, about 3–5 SRD particles are required per box (or node) whose positions and velocities can be seen as the degrees of freedom in that node. In three dimensions (3D) this amounts to 18–25 variables per node which is similar to the 15 or 19 speed lattice-Boltzmann method.

While the LB method might be slower than SRD in the regime of large thermal fluctuations it has the advantage that it can be used for almost arbitrarily high Peclet numbers just by reducing the amplitude of the noise. To reduce the noise in SRD, a huge number of fluid particles per node has to be used which makes the method inapplicable at Peclet numbers higher than about 20. Fortunately it has been shown by Padding and Louis [20] that basic properties of sedimentation such as the main settling speed are hardly affected by thermal noise.

The method is based on so-called fluid particles with continuous positions and velocities which follow a simple, artificial dynamics.

The system is coarse grained into cubic cells of a regular lattice with no restriction on the number of particles in a cell. The evolution of the system consists of two steps: streaming and collision. In the streaming step, the coordinate of each particle is incremented by its displacement during the time step. Collisions are modeled by a simultaneous stochastic rotation of the relative velocities of every particle in each cell. The dynamics is explicitly constructed to conserve mass, momentum, and energy, and the collision process is the simplest consistent with these conservation laws. It has been shown that there is an $H$ theorem for the dynamics and that this procedure yields the correct hydrodynamic equations for an ideal gas [18].

Consider a set of $N$ point particles with (continuous) coordinates $\mathbf{r}_i(t)$ and velocities $\mathbf{v}_i(t)$. In the streaming step, all particles are propagated simultaneously by a distance $\mathbf{v}_i \tau$, where $\tau$ is the value of the discretized time step. For the collision step, particles are sorted into cells, and they interact only with members of their own cell. Typically, the simplest cell construction consisting of a hypercubic grid with mesh size $a$ is used. The collision step consists of an independent random rotation of the relative velocities $\mathbf{v}_i - \mathbf{u}_i$, of the particles in each cell, where the macroscopic velocity $\mathbf{u}(\xi, t)$ is the mean velocity of the particles in the cell with coordinate $\xi$. The local temperature $T(\xi, t)$ is defined via the mean-square deviation of the particle velocities from the mean

\footnote{For low Peclet numbers Brownian motion dominates and sedimentation takes place very slowly. The simulations require a huge number of time steps. Then Brownian simulation (BS), including short-range hydrodynamics interactions, might be a more suitable tool, since not the complete velocity field has to be calculated. For very high Peclet numbers, SRD becomes inefficient due to the necessary averaging. For high Reynolds numbers a small time step and high spatial resolution would be necessary, which increases the computational effort extremely.}
locity in a cell. All particles in a cell are subject to the same rotation, but the rotation angles of different cells are statistically independent. There is a great deal of freedom in how the rotation step is implemented [18,30], since, by construction, the local momentum and kinetic energy are invariant. The dynamics is therefore summarized by

\[ \mathbf{r}_i(t + \tau) = \mathbf{r}_i(t) + \tau \mathbf{v}_i(t), \]

\[ \mathbf{v}_i(t + \tau) = \mathbf{u} [\xi_i(t + \tau)] + \omega \xi_i(t + \tau) [\mathbf{v}_i(t) - \mathbf{u} [\xi_i(t + \tau)]], \]

where \( \omega \xi_i \) denotes a stochastic rotation matrix and \( \xi_i \) is the coordinate of the cell occupied by particle \( i \) at the time of the collision. \( \mathbf{u} [\xi] = (1/M) \sum_{k \neq i} \mathbf{v}_k \) is the mean velocity of the particles in cell \( \xi \). \( \omega \) is taken to be a rotation by an angle \( \pm \alpha \), with probability 1/2. We are using rotations about the three coordinate axes with \( \alpha = \pm 90^\circ \), because these are the most simple rotation matrices one can imagine in 3D, since they only contain entries taken out of \{0, \pm 1\}. This has been suggested by Tazel et al. [31]. In every time step for each cell one of these 6 possibilities is chosen with equal probability 1/6. However, any stochastic rotation matrix consistent with detailed balance can be used.

In order to remove low temperature anomalies and to achieve exact Galilean invariance, we use a modification of the original algorithm [30]: all particles are shifted by the same random vector with components in the interval \([-a/2, a/2]\) before the collision step. Particles are then shifted back by the same amount after the collision. The random vectors of consecutive iterations are uncorrelated. Ihle and Kroll have discussed in Refs. [32,33] why this simple procedure works and showed that it leads to transport coefficients independent of an imposed homogeneous flow field. In Refs. [34,35] analytical calculations of the transport coefficient in this method are presented.

Two different methods to couple SRD and MD simulations have been introduced in the literature. We have implemented them both, and we are using them depending on what we plan to measure. The first one [27] is much more accurate in resolving the local velocity field around the colloidal particles. Lubrication effects are reproduced well by this coupling method. The second one [36] resolves the velocity field only down to a length scale of the particle diameter. On the other hand, the method becomes much faster because of the lower resolution. In both coupling methods the long-range hydrodynamic interactions are reproduced.

IV. COUPLING 1: PLACING FLUID PARTICLES OUTSIDE OF COLLOIDAL PARTICLES

In the combined MD and SRD simulation the fluid particles have to interact with the colloidal particles and transfer momentum from one to the other part of the simulation. One possibility to do this is, as suggested by Inoue et al. [27], to check after each streaming step of a fluid particle \( i \) if its new position \( x_i(t + \tau) \) is within a colloidal particle and, if yes, to modify its position and velocity. In this coupling step total momentum has to be conserved, which means, that when modeling the “collision” between the fluid particle and the colloidal particles, one has to make sure that the change of momentum of the fluid particle is transferred to the suspended particle. The calculations described in the following are done in a frame fixed on the colloidal particle.

One can think of several different methods to assign a new position to the fluid particle, which have been shown to work properly.

(i) Place it on the shortest distance to the surface of the colloidal particle and move it with its new velocity half of a time step.

(ii) Calculate the point and the exact time when the fluid particle has entered into the colloidal particle and move it back to there. Then choose a new velocity and move the fluid particle with the new velocity for the remainder of the time step.

Both methods turned out to work, where the second one is more accurate but more computationally intensive as well. Just to place the fluid particle directly on the surface and move it in the next time step turned out to produce an increase of the fluid particle density around the colloidal particle. Anomalies in the fluid temperature could also be found when the fluid particles were placed directly on the colloidal surface.

To increase stability of the simulation the idea is not to conserve energy in every single collision, but to use a thermostat and choose the new velocities according to a given distribution. The new velocities should point from the colloid surface to the outer area. Since the interior of the colloidal particle usually does not contain any fluid particles and the velocity distribution next to a colloidal particle should be independent of neighboring particles, the velocity distribution for the newly chosen fluid particle velocities has to be the same as if the space inside the suspended particle was filled with fluid particles. Assume these imaginary fluid particles having the same density and temperature as in the remainder of the fluid bath. Then, one could evaluate the velocity distribution for the reflected fluid particles by taking the velocity distribution of the imaginary fluid particles passing through the colloidal surface. But it is a nontrivial task to analytically calculate this distribution for a spherical area. However, if the mean free path of the fluid particles is small compared to the diameter of the colloidal particles, we can safely assume the colloidal surface to be an infinitely extended plane separating the space into two regions [27]. Then one finds the following distribution:

\[ p(v_n) \sim v_n \exp(-\beta v_n^2), \]

\[ p(v_t) \sim \exp(-\beta v_t^2), \]

with

\[ \beta = \frac{m_f}{2k_BT}. \]

where \( v_n \) is the normal component and \( v_t \) is the tangential component of the fluid particle velocity in the frame fixed to the surface of the large particle. \( m_f \) is the mass of a fluid particle. In the following sections we describe how \( m_f \) has to
be chosen. $T$ is the temperature to which this thermostat is adjusted, and the whole system will adopt this temperature after a transient time. The tangential component can be obtained by computing $\sqrt{x_1^2 + x_2^2}$ of two independent and Gaussian distributed random variables.

Since the fluid particles of the SRD are artificial particles within the context of this mesoscopic simulation method, their mean free path and their momentum are different from the corresponding values for single solvent molecules. Because of this, there is a depletion force acting on colloidal particles which is much larger than in reality. Depletion forces are only relevant in systems with very big molecules—e.g., polymer solutions with added small particles or binary mixtures of particles with clearly separated diameters. There, each of the small particles carries a considerable momentum—which is also the case in the SRD simulations. Nevertheless, unrealistically high depletion forces can be suppressed by reflecting fluid particles many times: If after the collision step the fluid particle is placed in another colloidal particle, the collision step is repeated for that colloidal particle and so on, until the fluid particle reaches a position outside any colloidal particle or until a maximum number $N_{\text{max}}$ of collisions has been calculated through. We have measured the depletion force and found out that a limit of $N_{\text{max}} = 10$ is a good compromise between computational speed and accuracy. The depletion force does not decay substantially stronger if the limit is increased, but the computational effort still grows with $N_{\text{max}}$ (at most linearly), because some fluid particles are trapped in a small gap between two colloidal particles and jump from one to the other. This in fact would still decrease the depletion force, but in the meantime the calculation for the remaining system is interrupted until finally eventually one single fluid particle is reflected the very last time. It is obvious that this scenario can easily be truncated. The remaining depletion force can be neglected at least in the cases where strong attractive van der Waals forces or strong repulsive electrostatic forces are present.

V. COUPLING II: ROTATING VELOCITIES OF THE COLLOIDAL PARTICLES

A second possibility to couple SRD and MD simulations is to sort the colloidal particles into the SRD boxes and include their velocity in the rotation step. This technique has been used to model protein chains suspended in a liquid [36,37]. The mean velocity in each cell has then to be weighted with the mass of the particle (because the mass of colloidal particles differs at least by one order of magnitude from the one of the fluid particles and their inertia dominates the flow field next to it). The calculation of $\mathbf{u}[\xi(t+\tau)]$ in Eq. (12) is modified to

$$ u(\xi) = \frac{1}{M_{\xi}} \sum_{k \in \xi} v(m_k), \quad (15) $$

where we sum over all colloid and fluid particles in the cell. $m_k$ is the mass of the particle with index $k$ and $M = \sum_{k \in \xi} m_k$ is the total mass contained in the cell.

The coupling acts on the center of mass of the colloidal particles and affects only the fluid particles within the same cell. This means, to affect the same area of the flow field like in reality, one has to choose the cells to be of the same size as the colloidal particles. Obviously, the mesh size is drastically larger than in the first coupling method and the flow field cannot be resolved in detail. The fact that colloidal particles push away the solvent as well as depletion and lubrication forces cannot be reproduced at any level.

VI. TIME-SCALE ANALYSIS

Our system contains many different, let us say $L$, time scales, which differ by several orders of magnitude making brute force numerical simulations very time consuming or even impossible. These time scales can be used to define $L – 1$ dimensionless characteristic numbers, such as the Reynolds or Peclet number, as the ratio of two time scales. If one can manage to adjust the simulation parameters such that all these characteristic numbers are the same as in the experiment, the simulations should be able to exactly reproduce the dynamical behavior of the real system. Of course, therefore one has to change quantities like the temperature or the viscosity of the fluid.

Often, it is sufficient to reproduce only a few of all characteristic numbers exactly—i.e., only those which are believed to be significant for the behavior. For example, in sedimentation processes where the Reynolds number is much smaller than unity, it may be modified to another value, which still fulfills the condition of being much smaller than 1. In both cases the Stokes limit is a valid approximation. As a general rule of thumb, dimensionless numbers of order 1 are important to be reproduced since they represent two competing dynamical effects. The reason to modify the other “insignificant” numbers is to reduce the ratio of the largest to the smallest time scale which determines the numerical effort. In order to decide which are the dimensionless numbers that can be safely modified without changing the physics too much, a detailed analysis of the different time scales is needed.

We start with the largest scales. After some time an isolated spherical particle sedimenting in a liquid reaches the so-called Stokes velocity

$$ uS = \frac{2 R^2 g}{9 \nu} \left( \frac{\rho_w}{\rho_w - \rho} - 1 \right). \quad (16) $$

$\nu$ is the kinematic viscosity, $g$ denotes gravity, $\rho_w$ is the mass density of the particle, and $\rho$, the mass density of the solvent. This velocity is obtained from the force balance between buoyancy and weight of the particle, $F_G = 4\pi(\rho_w - \rho)gR^2/3$, and the drag force in a viscous liquid, $F_D = 6\pi \rho_w \nu R \cdot \mathbf{u}$. The drag force $F_D$ also defines the mobility $\mu = \mathbf{u}/F_D = 1/(6\pi \rho_w \nu R)$ of a spherical particle. The time for a particle to move a distance of its diameter, $2R$, is denoted by
HECT et al.

\[
\tau_s = \frac{2R}{vS} = \frac{9\nu}{Rg(\rho_m - \rho_w) - 1},
\]  

(17)

By means of the Einstein relation \(D = \mu k_B T\) we obtain the diffusion constant \(D\) for the particle,

\[
D = \frac{k_B T}{6\pi \nu \rho_c R}.
\]  

(18)

The mean-square displacement of a diffusing particle in each dimension \(i\) is given by

\[
\langle x_i^2(t) \rangle = 2Dt,
\]  

(19)

hence, the time the particle needs to diffuse a distance of \(2R\) is of the order of

\[
\tau_D = \frac{2R^2}{D} = \frac{12\pi \nu \rho_c R^3}{k_B T},
\]  

(20)

which we call the diffusion time.

The ratio \(\tau_D/\tau_s\) measures the importance of Brownian motion in the system and is called the Peclet number, \(Pe = \tau_D/\tau_s\). It turns out to be close to unity here. Inserting the definitions for \(\tau_D\) and \(\tau_s\), one notices that \(Pe\) depends on the fourth power of the radius \(R\),

\[
Pe = \frac{vSR}{D} = \frac{F_{cG} R}{k_B T} = \frac{4\pi \nu k_B T (\rho_m - \rho_w)}{3k_B T}.
\]  

(21)

Let us consider another time proportional to \(\tau_p\); we assume a regular three-dimensional, cubic array of spheres which are separated by gaps of size \(R/2\). Then, the volume concentration of this suspension is in the intermediate regime, \(\phi = 0.268\). The time one sphere diffuses the distance of a gap is given by \(\tau_D = \tau_p/16\).

Now, let us discuss another important time called the particle relaxation time, which is related to how long it takes the particle to react to an imposed force; i.e., this time measures the inertia effects. Consider Newton’s equation for a particle of mass \(m\) subject to a force \(F\) and a friction coefficient \(\xi\), \(m\partial v/\partial t = -\xi v + F\). Expanding the velocity \(v\) around the stationary state, \(v = v_0 + \delta v\), gives

\[
\frac{\partial \delta v}{\partial t} = -\frac{\xi}{m} \delta v,
\]  

(22)

which leads to an exponential decay on a time scale \(\tau_p = m/\xi\). Identifying the friction \(\xi\) with \(1/\mu\) and inserting the mass leads to

\[
\tau_p = \frac{2R^2}{9 \nu \rho_c}.
\]  

(23)

Now we consider a very short time scale \(\tau_p\); the time fluid momentum diffuses a distance \(2R\)—i.e., 

\[
(2R)^2 = 2v \tau_p
\]

leading to

\[
\tau_p = \frac{2R^2}{\nu},
\]  

(24)

which helps defining the particle Reynolds number as

\[
Re = \frac{\tau_p}{\tau_s} = \frac{Rv_S}{\nu}.
\]  

(25)

Finally, we have to discuss another important short length scale due to a short-range potential among the colloidal particles. This scale usually determines the maximum time step in molecular dynamics. Guided by the analogy to a harmonic oscillator with frequency \(\omega = k/\mu\), we replace the spring constant \(k\) with the second derivative of the interparticle potential \(\partial^2 V(R)/\partial R^2\) and use the period of this oscillation to define the interaction time scale,

\[
\tau_v = \frac{2}{\omega} = \frac{2\pi}{\sqrt{m\omega/\lambda_B}},
\]  

(26)

where we approximate the derivative of the potential by means of the Hamaker constant \(A_B\) as a typical size of the potential and a typical distance \(l\) such as the distance between the surface of the particle and the primary potential minimum due to the combined effect of van der Waals attraction and screened Coulomb repulsion. Comparison of \(\tau_v\) and \(\tau_p\) can answer the question whether the oscillations of two particles around the primary or secondary minimum are visible or whether the creeping or overdamped case is realized where friction is dominating over inertia. Analyzing a harmonic oscillator with damping constant \(\xi\) one finds that creeping being established at

\[
\tau_p = \tau_v \frac{4\pi}{\xi}
\]  

(27)

In using this relation a lubrication force described in Eq. (4) has to be taken into account. This force is proportional to the difference of normal velocities of two approaching particles, and in this sense it can be seen as an additional contribution to the friction coefficient. It becomes huge at short interparticle distances \(d\), and it will turn out later that even without this addition all particles considered here are well inside the creeping regime due to the large friction in water. This is the justification that so far many people used Brownian dynamics for this system instead of molecular dynamics [11,38]. In our situation, including thermal fluctuations and full hydrodynamics consistently is easier to do in molecular dynamics. Moreover, with our parameters the MD is at least competitive or even faster than previous BD calculations.

VII. SIMILARITY CONSIDERATIONS AND DETERMINATION OF SIMULATION PARAMETERS

A. Introduction

The determination of parameters for a mesoscopic model to quantitatively compare with experiment is a nontrivial task. Typical values of the parameters in an experiment are listed in Table I. For these values of the parameters in the experiment all the time scales defined in the previous section are calculated and listed in Table II. This tells us that the Peclet number is \(Pe = \tau_p/\tau_s = 0.74\) and we have a competition between convection due to gravity and Brownian motion. The particle Reynolds number is very small —i.e., \(Re = \tau_p/\tau_s = 4.0 \times 10^{-7}\). The ratio of \(\tau_v\) to \(\tau_p\) is larger than \(4\pi\).
hence, oscillations of particles in their short-range potentials are overdamped, already without considering lubrication forces. We get $\tau_p \ll \tau_D$, since the particles are well relaxed before they hit each other due to Brownian motion. $\tau_e \ll \tau_D$; hence, the transport of momentum through the fluid is much faster than if transported directly by the particle. These are the dynamical characteristics which have to be preserved by any parameter changes; in particular, the Peclet number has to be kept exactly the same. Of course, the static properties such as the ratio of kinetic energy to be kept the same too.

However, using identical parameters as shown in Table I in an MD simulation would require of the order of $10^7$ iterations to see sufficient progress in the sedimentation process. This is an unacceptably high numerical effort, which must be reduced without significantly changing the physics of this process. First, we now show how to choose the parameters for a simulation using the coupling method I. After that, we describe what has to be changed using the coupling II.

B. Determination of the parameters for coupling I

We start by choosing reasonable parameters for the hydrodynamic part of the code—i.e., stochastic rotation dynamics—since this is time consuming and the most storage intensive part of our simulation. For the moment we keep the particle radius constant at $R=0.4 \mu m$. Let $a$ be the lattice constant of the SRD grid. By choosing $a=R/2$ a spherical particle covers about 34 boxes which is a sufficient resolution of the particle. We get $a=0.2 \mu m$. We use an average number of $M=2.5$ fluid particles per box, which leads to $6M=15$ real numbers (3 velocity and 3 position coordinates in 3 dimensions) to be stored for every box. A larger $M$ would reduce Brownian motion and increase CPU-time and storage requirements. Using a smaller number leads to a very long effective mean free path of the fluid particles (sometimes there is only one particle per box and no collision takes place), which results in a large viscosity and a bad resolution of the flow field around the colloidal particles.

Next, we choose the ratio of the mean free path $\lambda = \tau_5 k_B T/m_f$ to the lattice constant. $m_f$ is the mass of the fluid particle and $T$ the effective temperature of the fluid particles which can differ by several orders of magnitude from the real temperature of the experiment as will be explained later. In Ref. [30] it was discovered that a ratio $\lambda/a$ smaller than 0.5 leads to anomalies in the model, which can be corrected by a random shift of the lattice prior to every rotation. Here, we set $\lambda/a=0.12 \mu m$ to have sufficient resolution of the flow and random shifts are not needed.

The rotation angle $\alpha$ is taken to be $90^\circ$ because this gives the most simple rotation matrix.

The exact expression for the shear viscosity for $\alpha=90^\circ$ is given by [34]

$$\nu = \frac{a^2}{18 \delta t} \left( 1 - \frac{e^{-M}}{M} \right) + \frac{k_B T \delta t M + 2}{4 m_f M - 1}. \quad (28)$$

Inserting $M=2.5$ and expressing temperature by means of $\nu$ it follows for our choice of parameters that

$$\nu = 0.3052 \frac{a^2}{\delta t}. \quad (29)$$

In order to reproduce the same diffusion coefficient as seen in experiments, $\delta t$ has to be determined by means of the Einstein relation,

$$D = k_B T \mu = \frac{k_B T}{6 \pi \eta \rho R}. \quad (30)$$

Setting $\rho_w = M m_f/a^3$, using $\nu$ from Eq. (29), and expressing $k_B T/m_f$ by means of $\lambda$ one finds $\delta t = 0.025 a^2/(D R)$. Inserting the diffusion coefficient expected in reality from the Einstein relation, $D=5.49 \times 10^{-11} m^2/s$, we arrive at a time step $\delta t = 0.91 ms$ for the SRD algorithm. This time step is of course too large to resolve the motion of colloidal particles due to interparticle forces and friction. Hence, a two-step method is needed: The trajectory for the colloidal particles is integrated by another, smaller time step $\delta t_c$. This also means that the extensive SRD procedure is only applied every $(\delta t/\delta t_c)$th iteration of the MD algorithm, thus reducing the required computer power substantially.

The way parameters are derived implicitly means that we keep $\tau_5$ and $\tau_D$ as in reality. This corresponds to $\tau_5/0.91 ms=869$ SRD iterations until a colloidal particle has fallen down by one diameter $2R$ which is affordable. The kinematic viscosity in the simulation is much smaller than in nature ($\nu_{model}=1.34 \times 10^{-11} m^2/s$).

Next, one has to check what happens to the particle relaxation time $\tau_p$. The requirement is that it should be much larger than the one given in Table I (in order to increase numerical efficiency) and, on the other hand, it should still be smaller than $\tau_D$ to ensure that particles can relax between consecutive collisions caused by thermal motion. Following Eq. (23), we obtain $\tau_p = 10.36 ms$. This is an acceptable value: it is much larger than the 0.139 $\mu s$ seen in reality and

<table>
<thead>
<tr>
<th>$\tau_5$</th>
<th>$\tau_D$</th>
<th>$\tau_G$</th>
<th>$\tau_Y$</th>
<th>$\tau_P$</th>
<th>$\tau_P$</th>
</tr>
</thead>
<tbody>
<tr>
<td>0.791 s</td>
<td>0.582 s</td>
<td>36.5 ms</td>
<td>7.45 $\mu s$</td>
<td>0.320 $\mu s$</td>
<td>0.139 $\mu s$</td>
</tr>
</tbody>
</table>
still smaller than $\tau_G = 36.5$ ms. Therefore it needs 3.5 SRD steps to relax a particle which means that the process still can be resolved.

Considering momentum transport in the fluid versus direct transport: During time $\tau_D$, momentum in the fluid is transported a distance $x^2 = 2 v r_{\tau_D} = 24.43 a^2$, i.e., $x = 4.94 a = 2.47 R$. Hence, momentum transport in the fluid is only slightly faster than by direct transport, which is still acceptable, even though in the real system it is much faster. This is reflected in a Reynolds number which is larger by a factor of $10^{-6}$. This again reflects the fact that the SRD model is efficient only if the Peclet and Reynolds numbers are in the range between 0.05 and 20.

Now, the gravity constant $g$ of the model has to be determined requiring that the Stokes velocity be the same as given in Table I. Since thermal convection of the fluid is not important for our simulation, we can neglect gravity on the fluid particles. Therefore, there is no buoyancy force in the simulation. We can correct for that by assuming a smaller gravity constant modified by the density ratio of colloid material and fluid. We find

$$g_{\text{model}} = g_{\text{real}} \frac{\rho_{\text{model}}}{\rho_{\text{real}}} \left(1 - \frac{\rho_a}{\rho_{\text{model}}} \right) = 9.78 \times 10^{-5} \text{m/s}^2. \quad (31)$$

As mentioned above, not only the viscosity, but also the temperature in our simulation may be different from the one in nature. To see that we calculate the ratio $\Lambda = \rho_a / k_B T$. In nature we have $\Lambda = 0.942 \times 10^{-2}$ $\text{K}^{-1}$. In the model we get $\Lambda_{\text{model}} = 3.9 \text{Mm}/(a^3 k_B T)$ where we express $k_B T/m_f$ by means of the mean free path and the time step $\lambda^2/(6\epsilon)^2$. One finds that $\Lambda_{\text{model}}$ is scaled by a factor of $7.44 \times 10^4$. The static features have to be reproduced by the model, and therefore we have to keep the ratio of kinetic and potential energy $k_B T / m_f$ constant. This means that the ratio $\rho_a / V_{\text{real}}$ especially $\rho_a / A_f$ has also to be scaled by this factor. We use $A_f = 4.76 \times 10^{-20} \text{J}/(7.44 \times 10^4)$ in the model, corresponding to new $A_f = 6.39 \times 10^{-22}$ J. From Eq. (26) we get a scaled $\tau_V$ of 2.03 ms corresponding to $\tau_V / \tau_f = 0.196$ [which is smaller than 4 ms; see Eq. (27)]. The unscaled value is 53.6. The creeping case is restored by the lubrication force, which we have included in the MD simulation and which grows for smaller gaps between the particles. The lubrication force determines the small iteration time step $\delta t_{\text{MD}}$ for the MD simulation. We chose $\delta t_{\text{MD}} = 2 \mu$s, which is about 200 times larger than it would be if all the original parameters would have been kept and min($\tau_V, \tau_f$), being much smaller, would determine the time step.

Comparing to the SRD time step we see that every 455 small steps one SRD step is performed. We need 869 SRD steps and $4 \times 10^5$ MD steps to see a colloidal particle sinking down by one diameter. The time scales in the simulation are summarized again in Table III.

### C. Determination of the parameters for coupling II

To simulate the same system with coupling method II, we use the same particle radius $R = 0.4 \mu$m. The lattice constant has now to be chosen differently because the colloidal particles are coupled to the SRD simulation as mass points. They have influence on the fluid which is in the same cell, and therefore the size of the cell can be understood as the volume within which the SRD simulation “feels” the colloidal particles and we choose the lattice constant in a way that the volume of the cell is equal to the volume of a colloidal particle: $a = 6.25 \times 10^{-7}$ m. A smaller lattice constant in this context would model smaller colloidal particles in the SRD part of the simulation. The velocity field would be resolved better, but since coupling method II does not allow a resolution smaller than the colloidal particles, one cannot expect to gain any information from the fluid simulation on smaller length scales than the colloidal particle size. Any attempt to increase the resolution of the SRD simulation would only cause a larger computational effort.

Since we do not modify the Peclet number, we have to choose approximately the same number of fluid particles per colloidal particle. Since the box size has increased with respect to the coupling method I, we have to assume more particles per box now. We choose $M = 60$ (which would correspond to two particles per box in the coupling method I, but since the boxes are much larger now; we can slightly reduce the ratio of fluid particles per colloidal particle). We choose $\lambda / a = 0.5$ and use random grid shifts here to avoid the fluid particles interacting too often with the same partners which causes artifacts in their correlation. The rotation angle $\alpha$ is again 90° to achieve very simple matrices. Following the same procedure as for coupling method I [Eq. (28)] we find a time step for the SRD of

$$\delta t = 2.05 \text{ ms}. \quad (32)$$

According to Eq. (29) the viscosity in the simulation results to $\nu_{\text{model}} = 2.29 \times 10^{-11}$ $\text{m}^2/\text{s}$. The gravity constant therefore has to be rescaled by a factor of 58 813, and the temperature and potentials have to be scaled by 43 733.5

The resulting characteristic times are shown in Table IV. $\tau_S$ and $\tau_D$ are again kept as in reality. Now we need $\tau_S / 2.05 \text{ ms} = 385$ SRD iterations until a colloidal particle has fallen down by one diameter, which is much faster than by using coupling method I. $\tau_F$ is still smaller than $\tau_G$. Again $\tau_F$ is still smaller than $\tau_G$. Again $\tau_F$ is still smaller than $\tau_G$. Again $\tau_F$ is still smaller than $\tau_G$. Again $\tau_F$ is still smaller than $\tau_G$. Again $\tau_F$ is still smaller than $\tau_G$.

### TABLE IV. Time scales in the simulation using coupling method II.

<table>
<thead>
<tr>
<th>$\tau_S$</th>
<th>$\tau_D$</th>
<th>$\tau_G$</th>
<th>$\tau_V$</th>
<th>$\tau_F$</th>
<th>$\tau_P$</th>
</tr>
</thead>
<tbody>
<tr>
<td>0.791 s</td>
<td>0.582 s</td>
<td>36.5 ms</td>
<td>2.03 ms</td>
<td>22.9 ms</td>
<td>10.36 ms</td>
</tr>
</tbody>
</table>

5Where we take care of the fact that we do not apply gravity to the fluid particles.
became larger than \( \tau_s \), so that we have to include a lubrication force in the MD part for coupling II as well. Apart from that, the order of the different time scales is reproduced again: \( \tau_p < \tau_s < \tau_0 < \tau_D < \tau_T \).

Momentum is transported 1.34 times faster in the fluid as by the particles themselves. Short-range hydrodynamic interactions which cannot be resolved are in this sense comparable to particle-particle collisions whereas for long-range interactions the slightly faster transport of momentum can reproduce coarse-grained hydrodynamic effects. Again, to model these effects comparable to reality, the Reynolds number has to be much smaller than unity. We find \( \text{Re} = 1.77 \times 10^{-2} \).

If we include lubrication forces in the MD simulation in order to reproduce at least to some extend short-range hydrodynamics, we have to choose the same MD time step as for coupling method I but we need approximately 50% less CPU time for the hydrodynamics. Even though it seems to be a too simplified approach, we can reproduce a volume-fraction-dependent sedimentation velocity as will be described in Sec. X.

VIII. SIMULATION SETUP

A. Boundary conditions

Most simulations have been performed using periodic boundary conditions in all three directions. Then the total momentum may not change in any simulation step if no external forces (like gravity) are applied. If gravity on the colloidal particles is applied in a system with periodic boundary conditions, this would accelerate the whole system, since the total force on the center of mass is not vanishing. In a real system there is friction at the walls and, even more important, there is an equilibrium between hydrostatic pressure acting on the surface of a given volume and gravity acting as a body force. Since we simulate a volume in the center of the suspension, we either have to apply pressure on the walls or, which is easier, make sure that in sum the forces on the center of mass of the whole simulated system vanishes. Therefore, we follow the center of mass; i.e., on particles with higher density their gravity minus buoyancy has to compensate the gravitation on the colloidal particles with a force acting exactly agrees with temperature adjusted by the thermostat. When gravity is applied to the system, particles are accelerated and if in addition periodic boundaries are used, a thermostat is absolutely needed to remove the extra energy introduced by the periodic boundary in \( z \) direction in combination with gravity.

For the following discussion we define that the direction in which eventually gravity is applied is called the \(-z\) direction if a shear force is applied acts in the \( z \) direction. Using closed boundaries wall effects may be introduced; e.g., crystallization starts earlier than in the bulk. This effect could be observed especially when gravity was switched off and only closed-boundary conditions were applied. This is a finite-size effect, which is not that strong if periodic boundaries are applied. But in the case of gravity being applied, the whole system accelerates. To face this problem, three possibilities were tested.

(i) Fix the boundaries only in the \( z \) direction.
(ii) Fix the boundaries in the \( x \) and \( y \) directions and apply a no-slip condition for the fluid.

B. Temperature and thermostat

We have measured the temperature of the colloidal particles for different setups. If damping constants are chosen appropriately, the resulting temperature fits very well the temperature, which we have adjusted for the fluid by the initial conditions. If we additionally switch on a thermostat which we describe in the following, the measured temperature exactly agrees with temperature adjusted by the thermostat. We have measured the temperature of the colloidal particles in case \( \text{Re} = 1.77 \times 10^{-2} \) to calculate the local temperature. This reflects the fact that the mean velocity \( \mathbf{v}_i \) in the cell already contains three degrees of freedom which the particles in the SRD cell have. The choice of \( \gamma \) and the frequency with...
which the thermostat is called to work determine the relaxation rate, with which the system adapts $T^*$. The version described in [26] shows deviations of the achieved temperature for small numbers of particles per cell, whereas our implementation exactly reproduces $T^*$. The thermostat can even be extended to particles of different mass—i.e., colloid and fluid particles where the mass is used as weight factor for all velocities of the simulation.

C. Outlook: Shear

There are several possibilities to shear the system. If one only has MD particles, one can use moving walls either with a spring constant and a friction coefficient or with direct hard reflections, where a moving wall is assumed and the reflection is calculated in the moving frame fixed to the wall.

These approaches of course neglect all effects (like pseudo wall slip), which appear close to a wall in a shear experiment with a suspension. There, shear stress has to be applied to the fluid which then drags the suspended particles. One way to implement this is to add a small velocity offset to all fluid particles which are reflected. Since this approach works well and the colloidal particles are dragged by the fluid, we apply shear in this way to our system.

IX. TESTS OF THE SIMULATION CODE

A. Conservation of energy, velocity distributions

We have checked that the total energy is conserved in the molecular dynamics simulation if all damping constants are switched off. Otherwise or if the total energy even increases in spite of damping constants, the MD time step has been chosen too large. In the SRD simulation energy is conserved as well and if we use coupling method II also for the total system energy is conserved within numerical accuracy. With coupling method I (where a thermostat is already included in the coupling method) or if we switch on an additional thermostat, energy will not exactly be conserved but the system will reach a stable—i.e., equilibrated—state. In that sense, total energy (including thermal energy) will converge to a constant value.

In SRD simulations without any embedded particles, the total energy contains only the kinetic energy of the fluid particles. It is fully determined by the initialization of the particle velocities. We can choose three uniformly distributed random numbers to initialize the three velocity components for the fluid particles. In thermal equilibrium the distribution should be a Gaussian, which in fact can be observed in our simulations after some tens of SRD time steps. If colloidal particles are included into the system, they should reach a thermal equilibrium, at least as long as no external forces are applied. Damping terms would reduce fluctuations, so to check if the colloidal particles reach the same temperature as the fluid particles, damping constants have to be set to zero. Both distributions are shown in Fig. 2. They are both Gaussian with the correct temperature, even though for initialization uniformly distributed random numbers (square well) had been used. The tests are performed with both coupling methods. We have carried out simulations with particle radii of 0.4 $\mu$m and 0.25 $\mu$m, where the Peclet number (for the simulations where gravity is applied) is 0.11 and, of course, it takes much longer to observe sedimentation.

B. Viscosity

The diffusion coefficient of suspended colloidal particles can be used to check if the desired viscosity could really be achieved in the simulation. Using Eq. (18) we can, once we have measured $D$, calculate the kinematic viscosity $\nu$ and compare it to the value we have used to determine the simulation parameters like the SRD time step. We achieve a deviation of less than 20% in a diluted system compared to the theoretical value for an infinitely diluted system. Note that $D$ is a fixed number only in the limit of an infinitely diluted system and only if the interaction potentials between the colloidal particles are exclusively repulsive.

We are using two different methods, either the Green-Kubo method or direct evaluation of the mean-square displacement. The first is very accurate, if only few particles are used, but consumes much computer time and
memory because all particle velocities have to be stored for all time steps used in the calculation. That means, for higher volume fractions, it is more efficient just to sum up all the mean-square displacements within a given period of time. To calculate $D$ using the Green-Kubo method one uses the following relation:

$$g_x(j) = \lim_{T \to \infty} \frac{1}{IM_{\text{Tot}}} \sum_{i=1}^{I} \sum_{j=1}^{M_{\text{Tot}}} v_{x,i}(i) \delta(t) v_{x,i}(i+\delta t),$$

$$D_x = \delta t \left( \frac{1}{2} g_x(0) + \sum_{j=1}^{\infty} g_x(j) \right),$$

where $M_{\text{Tot}}$ is the total number of particles in the system and $I$ is the number of time steps used to calculate the contribution $g_x(j)$. $v_{x,i}(i) \delta(t)$ denotes the $x$ component of the velocity of particle $i$ in the $i$th time step. The sum in the expression for $D_x$ is in principle an infinite one, but since the contributions $g_x(j)$ decay with $j^{-3/2}$, one can truncate this sum after some tens of terms. $D_x$ and $D_y$ can be calculated accordingly. In Fig. 3(a) we show the diffusion coefficient in each direction. In numerical calculations it is impossible to evaluate an infinite sum. In Eq. (34), $I$ is limited at least by the total number of time steps within the simulation and in Eq. (35) the sum therefore is not infinite either. Since the contributions $g_x(j)$ become more and more inaccurate for larger $j$ we truncate the sum after $n$ terms and find that in our simulations for $n \approx 50$ the diffusion coefficient does not change anymore in a systematic way if $n$ is increased further. In Fig. 3(b) the last term of the sum is shown. For larger values, they fluctuate due to the finite sum in Eq. (34) which leads to the inaccuracy in the right part of Fig. 3(a). These fluctuations become smaller for longer simulation runs, but do not change the value of the diffusion coefficient taken as an average from the center part of Fig. 3(a).  

For the mean-square displacement in one direction during a time interval $\Delta t$ we calculate

$$D_x = \frac{1}{2\Delta t M_{\text{Tot}}} \sum_{i=1}^{M_{\text{Tot}}} \left[ x_i(t + \Delta t) - x_i(t) \right]^2$$

and $D_y$ and $D_z$ accordingly. For medium densities we have compared both methods and achieved the same results within error bars. Depending on the number of particles, we use one of both methods.

According to Richardson and Zaki [40], the mean sedimentation velocity of particles suspended in a liquid depends on the volume fraction $\phi$ as

$$v_x(\phi) = v_x(1 - \phi)^l,$$

with a typical exponent $l$ between $\approx 2.5$ and 4 depending on the boundary conditions. For periodic boundary conditions, Peclet number of Pe=1, and Reynolds number Re$\ll 1$ we find an exponent of 3.5 (Fig. 4) even when we use coupling method II, where only the long-range hydrodynamic interaction can be calculated correctly. A similar value is found for

$^3$This can be seen as a smooth cutoff of the sum in Eq. (35).
tractive forces they are less pronounced than if mainly repulsive interaction is present; see Figs. 5 and 6.

In Fig. 5 the particles cluster due to their attractive potentials and form stable configurations. The diameter of the particles is $5 \times 10^{-7}$ m. There is a sharp peak in the spatial correlation function of the particle centers at exactly that distance $2R$, where two particles touch each other in the very left part of the plot (A). Then, for larger particle separation, the correlation function starts to grow and drops suddenly after a peak at 1 $\mu$m (D), which is twice the diameter ($4R$). This is the contribution of two particles touching the same third particle. The distance between them depends on the angle which they form with the particle in the middle, but it is at last twice the diameter, when they are in a straight line, which explains the sudden drop of the correlation function. If several particles stick together, the straight line is stabilized. This explains the peak at the end of this section of the correlation function.

Two more peaks can clearly be assigned to configurations: One of them is from two particles touching two other particles, which themselves touch each other (C). There again the case of all particles being in the same plane can be stabilized by other particles surrounding them. The particles under consideration are then separated by a distance of $2R\sqrt{3}$. But of course, bending this configuration is still a degree of freedom which brings the two particles slightly closer to each other. Thus their contribution to the correlation function is shifted downward. The fourth peak at $2R\sqrt{6}$ reflects two particles, both touching three particles, which themselves are touching each other and define a plane (B). There is no freedom anymore for the two particles touching all the three of them at the same time. One can place one of them at one side of the plane and the other one at the other side.

When the potentials are mainly repulsive and the minimum caused by the van der Waals attraction is only a fraction of $k_B T$, the spatial correlation function looks completely different, as depicted in Fig. 6: The peaks described in the previous paragraphs have disappeared here. The primary peak has moved to a slightly larger distance, since the repulsive potential hinders the particles from touching each other.

In Fig. 7 we compare the correlation function of Fig. 6 with the potential used for that simulation. The maximum of the correlation function coincides with the minimum of the potential, but as the minimum is not very sharp, the particles are not restricted to fixed geometries and are in a steady process of rearrangement which results in broader peaks. This process could also be studied by evaluating the velocity correlation function for the colloidal particles which is related to the viscosity of the sample. The correlation of particles which are several diameters apart is still remarkable, as it is transmitted by the particles in between. The oscillations of the correlation function can be understood as a formation of layers where the probability of finding a particle in a certain layer is higher than in between.

B. Shear

We have carried out simulations with shear and gravity. For the particles the boundaries in the $z$ direction were closed.
and gravity was applied in the negative $z$ direction only to the colloidal particles. For the fluid particles the boundary in the $z$ direction was closed as well and additionally a velocity offset was added to apply a shear in the $x$ direction. Boundaries for fluids and for particles were periodic in the $x$ and $y$ directions. Velocity distribution functions have been evaluated. For the cases we investigated, after a transient they are all Gaussian (Fig. 8).

C. Phase diagram

We have explored the phase diagram for Al$_2$O$_3$ with respect to screening length and effective surface potential. We could identify the regions of suspended single particles and of flocculation (Fig. 9). The transition between these two regions depends on both parameters: the Debye screening length and effective surface potential. It is known that the $pH$ value determines the effective surface potential $\Psi_0$ and that the salt concentration and $pH$ value determine the Debye screening length $\kappa$ [9]. Exact relations between salt concentration and $pH$ value on the one side and $\kappa$ and $\Psi_0$ on the other side are not known a priori for the parameter ranges of our suspensions. There are approximations for very diluted systems and low salt concentrations. It is known that for Al$_2$O$_3$ the surface potential becomes zero for $pH \approx 8.7$ [41]. However, a phase transition between clustering in the upper left part of Fig. 9 and a suspended regime in the lower right part can be found in the simulations in analogy to the experi-

FIG. 7. Plot of the correlation function of Fig. 6 together with the potential used in this simulation. One can see that the maximum of the correlation function occurs for the distance, at which the very shallow secondary minimum of the potential is located.

FIG. 8. Velocity distribution of colloidal particles for each direction. Semilogarithmic plot where deviations from a Gaussian would be visible by deviations from a parabolic profile.

FIG. 9. (Color online) Snapshots from the phase diagram of Al$_2$O$_3$: For the DLVO potentials with different effective surface charge and different screening length one can either observe cluster formation or single particles in suspension. The simulation was done at room temperature for 1 s of real time and a particle diameter of 0.5 $\mu$m. Gravity has not been applied here. The pictures are corresponding to the values written on the axis. For this figure we have chosen the simulation runs for 14% volume fraction with 462 colloidal particles.

FIG. 10. Correlation function and its dependence on the inverse Debye screening length $\kappa$. $\Psi_0=20$ mV and $\kappa=0.14$ have been kept constant. For shorter Debye screening lengths the attractive force becomes stronger and leads to clustering, which is reflected in the appearance of peaks. The single curves have been shifted with respect to each other.
ment. The spatial correlation function can be evaluated for all the simulated cases and it can be used as a tool to identify the two regions of the phase diagram.

Figures 10–13 show selected examples of correlation functions for different parameter sets. The first and second graphs refer to a volume fraction $\Phi=14\%$ which also has been used for the phase diagram of Fig. 9. In Fig. 10 the correlation function has been plotted for every other image of the left column in the phase diagram in Fig. 9. One can see that for suspended particles only the first peak can be found in the correlation function. The secondary minimum in the potential causes the particles to glue for short times before they continue with their diffusion process. With increasing $\kappa$ the secondary minimum approaches the particle surface, and therefore the main peak is shifted to smaller distances. At the same time it becomes deeper so that clusters are formed and more peaks occur. The peak at a distance of $2R_0 \frac{\gamma}{\kappa} \approx 3R$ disappears again, when the attraction becomes stronger since this is a meta stable configuration of particles forming an octahedron. Figure 11 corresponds to the first row of images of Fig. 9. In this case the depth of the secondary minimum is adjusted by changing the effective surface potential. Again the transition between clustering regime and suspension can be observed. The potentials used here are among the ones plotted above in Fig. 1. In Figs. 12 and 13 the dependence of the correlation function on the volume fraction can be seen. In both cases long-range correlations become more pronounced with increasing volume fraction. This is shown for the suspended regime (Fig. 12) and for the clustering regime (Fig. 13), where the transition between the two cases presented here is achieved by a variation of $\kappa$ by only $10\%$.

D. Diffusion

We measured the diffusion coefficient of colloidal particles with attractive potentials. In Fig. 14 we show the diffusion coefficient for $\text{Al}_2\text{O}_3$ with an effective surface potential of $\Psi_0=50$ mV and an inverse Debye screening length of $\kappa=2 \times 10^8$ m$^{-1}$ for room temperature. One can see that the mobility of the particles decays since a cluster formation process takes place and the particles in the cluster are relatively fixed. The remaining mobility consists of two parts: Particles can still, with a nonvanishing probability, leave the cluster by thermal activation and the cluster itself can take part in a diffusion process; it can vibrate or be deformed—all of these are processes which are taking place on much longer time scales than the single-particle diffusion. By studying the dependence of the diffusion coefficient on the potentials and on the volume fraction, one might be able to find an answer to the question as to which of these processes is important for the dynamics of the system in which part of the phase diagram of Fig. 9.

XI. CONCLUSION

We have shown that by combining a stochastic rotation dynamics and a molecular dynamics simulation it is possible...
to study dense colloidal suspensions. We have explained how to determine effective parameters for the simulation (box size $a$, simulation time step $\Delta t$, number of fluid particles per box $M$, etc.). It is possible to relate the simulation to very distinct experimental conditions since all parameters (density, temperature, potentials, etc.) which enter into the description are scaled in a well-defined manner. We have presented first results which demonstrate the power of the model. We have demonstrated that the Richardson-Zaki law is reproduced already with the simple and fast coupling method II, and we have studied the dependence of the pair correlation function on the shape of the interaction potentials. We have shown how one can distinguish if for given Debye screening length $\kappa$, effective surface potential $\Psi_0$, and Hamaker constant $a_H$ if the system is in the clustering or suspended regime.

We are planning to carry out detailed investigations of the properties described in the two preceding sections (diffusion coefficient, correlation functions, sedimentation velocity) as well as cluster size and shape. Then these quantities can be analyzed under shear, their dependence on the shear rate, and the shear viscosity of the suspension, containing the fluid and particles, which both contribute to a complex shear viscosity.

**ACKNOWLEDGMENTS**

This work has been financed by the German Research Foundation (DFG) within project No. DFG-FOB 371, “Pe-loide.” We thank G. Guidelus, G. Huber, M. Kübler, L. Har-nau, J. Reinhagen, S. Richter, and M. Bier for valuable collaboration. We thank M. Strauss, A. Komnik, E. Tuzel, D. M. Kroll, A. J. Wagner, Y. Inoue, and M. E. Cates for helpful discussions. T. Ihle thanks the SFB 404, Project No. A7, of the DFG and ND EPSCoR through NSF Grant No. EPS-0132289 for financial support.

HECHT et al.

Shear viscosity of claylike colloids in computer simulations and experiments

M. Hecht, J. Harting, M. Bier, J. Reinshagen, and H. J. Herrmann
Shear viscosity of claylike colloids in computer simulations and experiments

Martin Hecht,1 Jens Harting,1 Markus Bier,2 Jörg Reinshagen,3 and Hans J. Herrmann 4,5

1Institute of Ceramics in Mechanical Engineering, University of Karlsruhe (TH), Karlsruhe 76131, Germany
2Max-Planck-Institut für Metallforschung, Heisenbergstraße 3, 70569 Stuttgart, Germany
3Institut für Theoretische und Angewandte Physik, Universität Stuttgart, Pfaffenwaldring 57, 70569 Stuttgart, Germany
4Institut für Baustoffe, Schaffenstrasse 5, ETH Zürich, CH-8093 Zürich, Switzerland
5Departamento de Física, Universidade Federal do Ceará Campus do Pici, 60451-970 Fortaleza CE, Brazil

Received 18 January 2006; revised manuscript received 2 May 2006; published 4 August 2006

Dense suspensions of small strongly interacting particles are complex systems that are rarely understood on the microscopic level. We investigate properties of dense suspensions and sediments of small spherical Al2O3 particles in a shear cell by means of a combined molecular-dynamics and stochastic rotation dynamics simulation. We study structuring effects and the dependence of the suspension’s viscosity on the shear rate and shear thinning for systems of varying salt concentration and pH value. To show the agreement of our results with experimental data, the relation between the bulk pH value and surface charge of spherical colloidal particles is modeled by Debye-Hückel theory in conjunction with a 2 pK charge regulation model.

DOI: 10.1103/PhysRevE.74.021403

I. INTRODUCTION

We simulate colloids of silt particles, for which in many cases the attractive Van der Waals forces are relevant. These colloids are sometimes called “peloids” (Greek: claylike). In contrast to clays consisting of thin platelets [1], our particles are in first approximation spherical particles. For real clays, the particle shape and their orientation is of relevance [2–6]. For silt particles, on the other hand, the description is less complex. However, due to the particle size of micrometers and below, the interplay of diffusion, electrostatic repulsion, van der Waals attraction, and hydrodynamics still renders the suspension a very complex system. Colloid science tries to investigate the properties of such suspensions, and there is a vast amount of literature on this subject [7–11]. Colloids in general have various applications ranging from food industry over paintings and cosmetic products to applications in photographic processes. Particles with well defined properties can be used to investigate general properties of soft condensed matter like gelation or crystallization on a larger length scale than on the atomic level. Especially attractive interactions (depletion forces as well as van der Waals attraction) have attracted attention in the recent years [12–16]. In soil mechanics, real samples, e.g., of sediments, can be less characterized and therefore it is more difficult to gain a microscopic picture from which general properties can be derived. Therefore, we have chosen a synthetic Al2O3 powder suspended in water as a model system for silt. The particle diameter is 0.37 μm.

Al2O3 is not only a cheap testing material for investigations related to soil mechanics, but it is also an important material for ceramics. In process engineering, one of the basic questions is how to obtain components of a predefined shape. Wet processing of suspensions, followed by a sinter process, is a common practice here [17]. Nevertheless, to optimize the production process and to improve the homogeneity and strength of the fabricated workpiece, one has to understand the complex rheological behavior of the suspension and its relation to the microscopic structure. This knowledge in turn can be applied to soil mechanics. Shear thinning as observed in our simulations and experiments is an important mechanism for the dynamics of landslides, making them more dangerous.

In this paper, we present our simulation results of sheared suspensions of Al2O3 particles. The overall behavior is strongly determined by the effective interaction potential between the particles in the suspension. The potentials can be related to experimental conditions within Debye–Hückel theory, and thus we can compare our simulation results to experimental data. In contrast to our approach of a direct comparison to experimental data, in the literature simulation results are often compared to analytical calculations.

Many different simulation methods have been developed and applied to colloidal suspensions: Stokesian dynamics (SD) [18–20], accelerated Stokesian dynamics (ASD) [21,22], pair drag simulations [23], Brownian dynamics (BD) [24,25], Lattice-Boltzmann method (LB) [26–29], and Stochastic rotation dynamics (SRD) [30–32]. Due to the complex nature of the problem, all simulation methods have to simplify in some point. Either Brownian motion is neglected or hydrodynamic interactions are included on a very simplified level. In many cases, simulations are done without a quantitative comparison to experiments. In the present paper, we combine molecular dynamics (MD) to simulate the colloidal particles, SRD for the description of the fluid, and a charge regulation model that provides us with realistic parameters for the Derjaguin-Landau-Vervey-Overbeek (DLVO) potentials [33,34] in the MD simulation. We include long-range hydrodynamic interactions on a coarse-grained level in the SRD part, and we only include DLVO pair potentials in the MD part. No electrostatic many-body interactions or electrodynamic interactions are considered, and modifications of the pair potentials due to locally increased colloid concentrations are neglected, too. However, many numerical investigations are based on much simpler models than ours. In our opinion, our model covers the main properties quite well.
Our paper is organized as follows. First we briefly describe our MD implementation, followed by a short sketch of the SRD simulation method and a description of how we have implemented our shear cell. The simulation method is described in detail in Ref. [30]. Then we describe the so-called 2 pK charge regulation model, which relates our simulation parameters with the pH value and the ionic strength $I$ adjusted in the experiment. A short description of the simulation setup and of the experiments carried out follows. After that, we present our simulation results and compare them to the experimental data. Finally, a summary is given.

II. MOLECULAR DYNAMICS

We study colloidal particles, composing the solid fraction, suspended in a fluid solvent. The colloidal particles are simulated with molecular dynamics (MD), whereas the solvent is modeled with stochastic rotation dynamics (SRD) as described in Sec. III.

In the MD part of our simulation, we include effective electrostatic interactions and van der Waals attraction, a lubrication force, and Hertzian contact forces. The electrostatic and van der Waals potentials are usually referred to as DLVO potentials [8–10,24,35,36], which capture the static properties of colloidal particles in aqueous suspensions. The first component is the screened Coulomb term

$$V_{\text{Coul}} = \pi \varepsilon_0 \rho_0 \left[ \frac{2 + \kappa d}{1 + \kappa d} \right] \frac{\kappa}{4 \pi \varepsilon_0} \left( \frac{\kappa d}{4 \pi \varepsilon_0} \right)^2 \exp(-\kappa (d - r)),$$

(1)

where $d$ denotes the particle diameter and $r$ is the distance between the particle centers. $\varepsilon$ is the elementary charge, $T$ is the temperature, $k_B$ is the Boltzmann constant, and $z$ is the valency of the ions of added salt. Within DLVO theory, one assumes linear screening, mainly by one species of ions with valency $\pm 1$ (e.g., $z = +1$ for $\text{NH}_4^+\text{Cl}^-$). The first fraction in Eq. (1) is a correction to the original DLVO potential, which takes the surface curvature into account and is valid for spherical particles [37].

The effective surface potential $\zeta$ is the electrostatic potential at the border between the diffuse layer and the compact layer. It may therefore be identified with the $\zeta$ potential. It includes the effect of the bare charge of the colloidal particle itself, as well as the charge of the ions in the Stern layer, where the ions are bound permanently to the colloidal particle. In other words, DLVO theory uses a renormalized surface charge, which we determine by the model described in Sec. IV.

$\kappa$ is the inverse Debye length defined by $\kappa^2 = \frac{8 \pi \varepsilon_0 d}{}\beta^2$, with the ionic strength $I$. The Bjerrum length $\lambda_B = \frac{e^2}{k_B T}$ measures the distance at which the electrostatic interaction of two elementary charges amounts to $\beta = k_B T$. $\varepsilon_0$ is the permittivity of the vacuum and $\varepsilon_r$ is the relative dielectric constant of the solvent (we use 81 for water, i.e., $\varepsilon_r = 7$ A for room temperature).

The Coulomb term of the DLVO potential competes with the attractive van der Waals term

$$V_{\text{vdW}} = -\frac{A_H}{12} \left( \frac{d^2}{r^3} - \frac{d^2}{r^4} + 2 \ln \left( \frac{r^2 - d^2}{r^2} \right) \right).$$

(2)

$A_H = 4.76 \times 10^{-20}$ J is the Hamaker constant [25], which involves the polarizability of the particles. The singularity of $V_{\text{vdW}}$ for touching particles is removed and the primary minimum is modeled by a parabola as described in Ref. [30].

Long-range hydrodynamic interactions are taken into account in the simulation for the fluid as described below. This can only reproduce interactions correctly down to a certain length scale. On shorter distances, a lubrication force has to be introduced explicitly in the MD simulation. The most dominant mode, the so-called squeezing mode, is an additional force

$$F_{\text{lub}} = -\left( \nu \frac{\dot{r}}{r^2} \right) \frac{6 \pi \eta}{r} \left( \frac{R}{r - d} \right)^2,$$

(3)

between two particles with radius $R$ and relative velocity $\nu \dot{r}$. $\eta$ is the dynamic viscosity of the fluid. In contrast to the DLVO potentials, the lubrication force is a dissipative force. When two particles approach each other very closely, this force becomes very large. To ensure numerical stability of the simulation, one has to limit $F_{\text{lub}}$. We choose a maximum force at a certain gap width $r_e$ and shift the force so that the maximum force cannot be exceeded: Instead of calculating $F_{\text{lub}}(r)$, we take the value for $F_{\text{lub}}(r + r_e)$. In addition, since the force decays for large particle distances, we can introduce a large cutoff radius $r_c$ for which we assume $F_{\text{lub}}(r) = 0$ if $r - d > r_c$. As the intention of $F_{\text{lub}}$ is to correct the finite resolution of the fluid simulation, $r_e$ and $r_c$ have to be adjusted in a way that the dynamic properties, i.e., the viscosity of a simulated particle suspension with weak DLVO interactions, fits the measurements. It turns out that $r_e = \frac{\lambda_B}{40}$ and $r_c = \frac{\lambda_B}{6}$ work best. Our approach for $F_{\text{lub}}$ is similar to the one often used in lattice Boltzmann simulations [26]. In contrast to Ladd [26], we have chosen to use two cutoff radii to be able to treat small and large gaps separately. There are different approaches, e.g., for Stokesian dynamics [18], where the force field is expanded to a multipole series and the far field part is subtracted afterwards.

Finally, we use a Hertz force described by the potential

$$V_{\text{Hertz}} = K(d - r)^{5/2} \quad \text{if } r < d,$$

(4)

where $K$ is the constant that describes the elasticity of the particles in the simulation. The Hertz force avoids that the particles penetrate each other. It also contains a damping term in normal direction,

$$F_{\text{Damp}} = -(\nu \dot{r}) \beta_D \sqrt{d - r},$$

(5)

with a damping constant $\beta_D$.

Since in this work no stress perpendicular to the shear direction is applied, the tangential forces at the particle surface are not of essential importance. To verify this, we have increased the spatial resolution of the fluid simulation, included tangential forces on the particles, and allowed particle rotations. Even though the computational effort was consid-
erably larger, and one could expect that more effects on the length scale below the particle diameter could be covered, one could observe only a change of some percent in the viscosity and in the velocity profile. Due to the DLVO potential and the lubrication force, the particles very rarely get into contact as long as no confining stress is applied. The only case in which particles really touch each other, would be if the \( \zeta \) potential is close to zero at a certain pH value. This pH value is called “isoelectric point.” It depends on the material of the suspended particles and on the solvent. For our system it is at pH 8.7 [36]. In experiments close to the isoelectric point, a solid fraction immediately flocculates out and sediments. In the simulation, one ends up with only one big cluster in the simulation volume, which corresponds to a part of a floc seen in the experiment.

For this study, we do not apply tangential forces and thus, having only central forces, we could neglect rotation of the particles. This reduces the computational effort substantially.

### III. STOCHASTIC ROTATION DYNAMICS (SRD): SIMULATION OF THE FLUID

The stochastic rotation dynamics method (SRD) was first introduced by Malevanets and Kapral [38,39]. The method is also known as “real-coded lattice gas” [31] or as “multiparticle-collision dynamics” (MPCD) and has been successfully applied to simulate many important systems such as complex fluids containing polymers [40,41], vesicles in flow [42], and dynamics of chemical reactions [43]. The method is a promising tool for a coarse-grained description of a fluctuating solvent, e.g., in Ref. [44] the results of simulations of a flow around a cylinder are presented, or in Ref. [32] sedimentation of a particle suspension is studied.

The method is based on so-called fluid particles with continuous positions and velocities. Each time step is composed of two simple steps: one streaming step and one interaction step. In the streaming step the positions of the fluid particles are updated as in the Euler integration scheme known from molecular-dynamics simulations,

\[
\mathbf{r}_i(t+\tau) = \mathbf{r}_i(t) + \tau \mathbf{v}_i(t),
\]

where \( \mathbf{r}_i(t) \) denotes the position of the particle \( i \) at time \( t \), \( \mathbf{v}_i(t) \) is its velocity at time \( t \), and \( \tau \) is the time step used for the SRD simulation. After updating the positions of all fluid particles, they interact collectively in an interaction step that is constructed to preserve momentum, energy, and particle number. The fluid particles are sorted into cubic cells of a regular lattice and only the particles within the same cell are involved in the interaction step. First, their mean velocity \( \mathbf{u}_j(t') = \frac{1}{N_j(t')} \sum_{i \in j} \mathbf{v}_i(t) \) is calculated, where \( \mathbf{u}_j(t') \) denotes the mean velocity of cell \( j \) containing \( N_j(t') \) fluid particles at time \( t' = t + \tau \). Then, the velocities of each fluid particle in cell \( j \) are updated as

\[
\mathbf{v}_i(t+\tau) = \mathbf{u}_j(t') + \Omega_{ij}(t') \cdot (\mathbf{v}_i(t) - \mathbf{u}_j(t')).
\]

\( \Omega_{ij}(t') \) is a rotation matrix, which is independently chosen randomly for each time step and each cell. We use rotations about one of the coordinate axes by an angle \( \pm \alpha \), with \( \alpha \) fixed [45]. The coordinate axis as well as the sign of the rotation are chosen at random, resulting in six possible rotation matrices. The mean velocity \( \mathbf{u}_j(t) \) in the cell \( j \) can be seen as the streaming velocity of the fluid at the position of the cell \( j \) at the time \( t \), whereas the difference \( \{\mathbf{v}_i(t) - \mathbf{u}_j(t')\} \) entering the interaction step can be interpreted as a contribution to the thermal fluctuations. Thus, to calculate the local temperature in the cell under consideration, one has to sum over the squares of this expression.

The method just described is able to reproduce hydrodynamics and thermal fluctuations. To couple the colloidal particles to the streaming field of the solvent, we use “Coupling II” of Ref. [30]: we modify the rotation step of the original SRD algorithm slightly. The colloidal particles are sorted into the SRD cells as well and their velocity enters into the calculation of the mean velocity \( \mathbf{u}_j(t) \) in cell \( j \). Since the mass of the fluid particles is much smaller (in our case it is 250 times smaller) than the mass of the colloidal particles, we have to use the mass of each particle—colloidal or fluid particle—as a weight factor when calculating the mean velocity,

\[
\mathbf{u}_j(t') = \frac{1}{M_j(t')} \sum_{i \in j} \mathbf{v}_i(t) m_i,
\]

with \( M_j(t') = \sum_{i \in j} m_i \),

where we sum over all colloidal and fluid particles in the cell, so that \( N_j(t') \) is the total number of both particles together. \( m_i \) is the mass of the particle with index \( i \) and therefore \( M_j(t') \) gives the total mass contained in cell \( j \) at the time \( t' = t + \tau \). The update rule for the particle velocities \( \mathbf{v}_i(t) \) and positions \( \mathbf{r}_i(t+\tau) \), which we apply, is summarized in Eqs. (6)–(9). This method to couple some embedded material to the SRD simulation is described for different applications in the literature [46,47].

This coupling method does not enforce no-slip boundaries on the particle surface, as in the method suggested by Inoue et al. [31]. Moreover, as very recently discussed by Paddling and Louis [48], purely radial interactions effectively introduce slip boundary conditions. Considering the drag coefficient, a prefactor changes and this could be corrected by assuming a different hydrodynamic radius. We have checked the influence of hydrodynamic interactions by removing the fluid completely and by varying the resolution of the SRD simulation. Also, two different coupling methods as described in Ref. [30] have been applied. Without fluid, the achieved shear rate as well as the viscosity differed strongly, whereas the difference between the two coupling methods was in the order of some percent only. Therefore, we need hydrodynamics to some extent, but we have chosen the coupling method with less computational effort. Very recently, Yamamoto et al. have shown that for colloidal gelation, hydrodynamic interactions are of minor importance [49] for 3D systems, but in contrast to our work, they focus on the static properties of a colloidal system quenched to zero temperature.
In Ref. [30], we have described a simple method to introduce shear at the fluid boundary by adding a velocity offset to all fluid particles reflected at the shear plane. From a constant velocity offset \( \Delta v \) one can calculate the mean shear force

\[
F_s = \left( \sum_{j=1}^{L} \frac{m_j \Delta v_j}{\tau} \right),
\]

where \( L \) denotes the average number of fluid particles crossing through the shear plane in one time step and \( \langle \cdot \rangle \) stands for a time average. \( L \) can be expressed by the mean free path and the number density of fluid particles. This would be a force driven shear, where one has only indirect control on the shear rate \( \dot{\gamma} \) or the shear velocity \( v_s \), respectively. Therefore, we modify the mean velocity \( \bar{v}(t') \) in the cells close to the shear plane by changing the velocity of each fluid particle as well as the velocity of the colloidal particles contained in that specific cell by the difference \( v_s - \bar{v}(t') \). By construction, the mean velocity in these cells is equal to the shear velocity \( v_s \) after that step. At the wall itself we implement full slip boundary conditions for the fluid and for the colloidal particles. The boundary in the direction of the shear profile (direction of the velocity gradient) is chosen to be non-periodic. By doing so, we can also observe phenomena such as wall-slip, nonlinear velocity profiles, or density profiles in our shear cell (see Sec. VII C). In the case of a nonlinear velocity profile, the viscosity is not well defined. We extract the central region of the profile where it is in first approximation linear and estimate there an averaged viscosity. This is the ratio of the velocity gradient and the shear force, which can be calculated in analogy to Eq. (10) by carrying out the sum over all velocity changes made. The region where we estimate the velocity gradient is half the system size.

We have tested a number of boundary conditions and different ways to impose shear, but the method just described turned out to work best. No-slip boundaries at a top and bottom plane seemed to work for high volume fractions and unless the potentials get attractive. As soon as (only slight) cluster formation sets in, the particles concentrate in the center of the system and lose contact with the sheared walls. Shearing only the fluid and not the colloidal particles always works, but the resulting viscosity is much too small. In fact, what one measures is the flow of the fluid streaming around the particles like a flow through a porous medium. The next point is how to determine the shear force and the velocity gradient we need for the calculation of the shear viscosity. The force is always related to any velocity changes made in the system, and its calculation is straightforward in most cases.

The imposed velocity difference divided by the system size perpendicular to the shear plane would give an averaged gradient. For clustered systems, not even the shape of the shear viscosity against shear rate was comparable to the measurements. If the velocity gradient changes within the system (compare Fig. 4), we have to take care that we measure the viscosity in the bulk, i.e., that we take the velocity gradient there. At least if the particles are not too strongly clustered, the slope of the plateau in the center of the system can be taken as a “good” velocity gradient. We use this velocity gradient as achieved shear rate as mentioned above. With this scheme for strongly attractive forces, the obtained viscosity \( \eta(\dot{\gamma}) \) for the simulation stays in the vicinity of the measured curve, whereas for other methods we tried out, the points of the simulation usually ended up far off the measured curve.

Fully periodic boundary conditions for sheared systems, known as Lees-Edwards conditions, would be a good choice for stable suspensions. As soon as clusters are formed, the velocity profile becomes non-linear, as discussed above. But, additionally the location of the cluster, i.e., the position of the plateau, is not fixed anymore to the center of the system, which makes it more difficult to extract the correct velocity gradient. In addition, the shear force would be determined from the velocity changes of the particles passing around the periodic boundaries. If the cluster by chance stays in the center of the system, again only the fluid would be sheared and only indirectly, transmitted by the fluid, the force would be exerted on the particles, as if with closed boundaries only the walls would move and no sheared regions close to the wall were implemented. Together with the periodic boundaries, this would lead to large fluctuations of the shear force, caused by the present position of the cluster. Furthermore, the boundary conditions would have to be consistent for the MD and for the SRD simulation. For the MD part, it is important that the position, where a particle reenters the system after passing around the periodic boundary, is shifted by \( 2\pi r_s / \tau \), with \( \tau \) being the continuously increasing simulation time and \( v_s \) the shear velocity. Additionally, this shift has to be wrapped around the periodic boundaries in shear direction. If we do the same with the fluid particles, the shift could be any value, not necessarily an integer multiple of the fluid box size. What we want to point out is that without any further restrictions, the grid in the SRD rotation step would no longer be regular in this plane, which in addition is the plane where one measures the shear force. To overcome this problem, one can restrict the shear rate to values determined by the SRD grid size and the SRD time step, but the other difficulties mentioned before remain.

IV. THE CHARGE REGULATION MODEL

To determine the effective surface potential that enters the DLVO potential, we use the model described in the following. In reality, the surface charge is achieved by adsorption and desorption of charge determining ions leading to an electrostatic potential difference between surface and bulk, which in turn influences ion adsorption. A full description of this regulation of surface charges requires two parts: the first part describes the relation between surface charge density and surface potential due to the electrolytic environment, whereas the second part quantifies the ion adsorption depending on the surface concentration of charge-determining ions.

Concerning the first part, a relation between the surface charge density \( \sigma \) and the surface potential \( \zeta \) of a charged spherical colloidal particle of radius \( R \) immersed in an electrolytic environment of relative dielectric constant \( \varepsilon_r \), and ionic strength \( I \) is given within Debye-Hückel theory [50,51] by
As mentioned above, we consider the Stern layer as a part of the surface charge. Thus we can identify the effective surface potential in DLVO theory with the ζ potential, and we can thus skip a discussion of bare charge versus effective charge [52–54].

In the second part of our model, the adsorption of charge-determining ions on the surface of the colloidal particle is described by assuming that the only mechanism of adsorption is that of protons (H+) on surface sites (−S). It turned out that this assumption leads to reasonable results for surfaces made of Al2O3. Adsorption is described by the two chemical reactions [55]

\[ -S^+ + H^+ \rightleftharpoons -SH, \]
\[ -SH + H^+ \rightleftharpoons -SH^2+, \]
with the two reaction constants

\[ K_1^c = \frac{[S^+][H^+][\exp(-\beta \epsilon \zeta)]}{[SH]}, \]
\[ K_2^c = \frac{[S^{-}][H^+][\exp(-\beta \epsilon \zeta)]}{[SH^2+]}. \]

In terms of the surface site concentrations, the total number of surface sites per area and the surface charge density are given by \( N_S = [S^-] + [SH] + [SH^2+] \) and \( \sigma = e(-S^-) + e(-SH^-) \), respectively. Defining \( pK_1^c = -\log_{10}(K_1^c) \) and \( pK_2^c = -\log_{10}(K_2^c) \) yields the point of zero charge \( pH_z \), i.e., the pH value of vanishing surface charge, as \( pH_z = \frac{1}{2} (pK_1^c + pK_2^c) \). The surface site density \( N_S \) and the difference \( \Delta pK = pK_1 - pK_2 \) are treated as adjustable parameters.

The above equations lead to the relation

\[ \frac{\sigma}{eN_S} = \frac{\delta \sinh(\phi_0 - \beta \epsilon \zeta)}{1 + \delta \cosh(\phi_0 - \beta \epsilon \zeta)} \]

with the Nernst potential \( \phi_0 = \ln(10) (pH_z - pH) \) and \( \delta = 2 \times 10^{-2}\Delta pK^2 \).

Equations (11) and (16) can be solved self-consistently for \( \zeta \) as a function of \( pH \). For our system of Al2O3 particles, we find \( \Delta pK = 4.2 \) and \( N_z = 0.22/\text{nm}^2 \). With these values the measured \( \zeta \) potential of 52 mV at \( pH = 6 \), \( I = 0.01 \text{ mol/l} \) and up to 120 mV at \( pH = 4 \), \( I = 0.01 \text{ mol/l} \) can be reproduced. For the experimental determination of the \( \zeta \) potential, electrophoretic (Delsa 440SX, Beckman-Coulter GmbH, Germany) and electrokinetic measurements (AcoustoSixer IIs, Coloidal Dynamics Ind., USA) were performed. To calculate the \( \zeta \) potential, Henry’s theory [8] was used. For details, see Ref. [56]. We have to admit that the relation between the directly measured quantities, e.g., electrophoretic mobility, and the \( \zeta \) potential is a subject of current research [57–60].

V. SIMULATION SETUP

In our simulation, we try to model the experimental system as accurately as possible. We start with spherical particles of diameter \( d = 0.37 \mu m \), the mean diameter of the particles used in the experiment. The simulation box is 48d = 17.76 \( \mu m \) long in the \( x \) direction, 24d = 8.88 \( \mu m \) in the \( z \) direction, and 12d = 4.44 \( \mu m \) in the \( y \) direction. To achieve a volume fraction of usually \( \Phi = 35\% \), as in the experiment, we need to simulate 9241 spheres. Our shear direction is the \( x \) direction, and the velocity gradient of the shear flow points in the \( z \) direction; in other words, we shear the upper and lower \( xy \) plane with respect to each other in the \( x \) direction.

We use periodic boundaries in the \( x \) and \( y \) directions and closed boundaries in the \( z \) direction for both fluid and MD particles. The energy supplied by the shear force is dissipated by means of a Monte Carlo thermostat described in Refs. [30,87]. It acts on the fluid particles as well as on the MD particles and conserves the momentum in each SRD cell.

VI. EXPERIMENTAL SETUP

Experiments are carried out with high-purity (99.97%) \( \alpha \)-Al2O3 powder (RCHP DBM, Baikowski Malakoff Industries, Inc., USA). The mean particle diameter is 0.367 \( \mu m \) (Coulter LS Particle Size Analyzer) and the size distribution is narrow (\( d_{50} = 0.176 \mu m \), \( d_{90} = 0.664 \mu m \)). The powder is suspended in bidistilled water (Merck, Germany). The suspension is then dispersed with alumina balls in a ceramic container for 24 h at a low rotational speed to keep the abrasion low. Subsequently, the suspension is degassed at 50 mbar under agitation. Then, in order to reduce the ionic strength to the desired degree, the suspension is purified by the dialysis technique. In this way, the majority of ions are removed and a background electrolyte of a very low salt concentration (5 \times 10^{-4} \text{ mol/l}) is obtained for suspensions of high solids loading. Starting from this master suspension, suspensions with increased ionic strength are obtained by adding different amounts of dry ammonium chloride NH4Cl (Merck, Germany). The \( pH \) of the suspensions is adjusted to \( pH = 6 \) with 0.1 and 1 mol/l hydrochloric acid HCl (Merck, Germany), if necessary. Thereby, the ionic strength and \( pH \) are revised by use of a laboratory \( pH \) and conductivity meter (inoLab pH/Cond Level 2, WTW GmbH, Germany). The electrophoretic mobility of dilute suspensions is measured with a Coulter Delsa 440 SX. Irreversible aggregation due to inhomogeneous salt concentration is not of importance here. If the ionic strength is strongly increased and after that a second dialysis step is performed to remove the ions again, the original viscosity is restored.

The ionic concentrations of selected ions are measured before and after dialysis using inductively coupled plasma–optical emission spectroscopy (ICP-OES, Model JY 70 plus, France). The suspensions are characterized using a Viscolab LC10 rheometer (Viscolab LM rheometer with control unit Viscolab LC10, Physica, Germany) with a cup and bob or a double gap geometry. The measurements are either performed immediately after suspension preparation or they are stored on a roller bank to avoid sedimentation. Sedimentation during the experiment can be excluded, since it takes much longer than the whole experiment, and the shear forces are much larger than gravity. Shear rate controlled experiments are performed at a constant temperature of 20 °C. The
suspensions are sheared at a constant shear rate of $\dot{\gamma} = 300/s$ before starting the actual ramp measurement. The flow of the fluid is plug flow.

The work concentrates on state $A$ ($pH=6, I=3$ mmol/l) in the suspended region, state $B$ ($pH=6, I=7$ mmol/l) close to the border but already in the clustered region, and state $C$ ($pH=6, I=25$ mmol/l) in the clustered region. The borders are not sharp transitions, but notable in a change of the shear viscosity.

Depending on the experimental conditions, one can obtain three different microstructures: a clustered region, a suspended region, and a repulsive structure. The charge regulation model allows us to quantitatively relate the interaction potentials to certain experimental conditions. A schematic picture of the stability diagram is shown in Fig. 1. Close to the isoelectric point ($pH=8.7$), the particles form clusters for all ionic strengths since they are not charged. At lower or higher $pH$ values, one can prepare a stable suspension for low ionic strengths because of the charge, which is carried by the colloidal particles. At even more extreme $pH$ values, one can obtain a repulsive structure due to very strong electrostatic potentials (up to $\zeta=170$ mV for $pH=4$ and $I=1$ mmol/l, according to our model). The repulsive structure is characterized by an increased shear viscosity. In the following, we focus on three states: state $A$ ($pH=6, I=3$ mmol/l) is in the suspended region, state $B$ ($pH=6, I=7$ mmol/l) is a point already in the clustered region but still close to the border, and state $C$ ($pH=6, I=25$ mmol/l) is located well in the clustered region.

When the suspensions are presheared, an occurring discrepancy between the measured viscosity in the increasing ramp and the decreasing one can be minimized. A detailed description of the experiments will be published elsewhere [17,61].

VII. RESULTS

A. Stability diagram

Increasing ramp and the decreasing one can be minimized. A detailed description of the experiments will be published elsewhere [17,61].

To make it easier for the reader to compare our results to other data. When the suspensions are presheared, an occurring discrepancy between the measured viscosity in the increasing ramp and the decreasing one can be minimized. A detailed description of the experiments will be published elsewhere [17,61].

Some typical examples for the different microstructures are shown in Figs. 2(a)–2(d). These examples are meant to be illustrative only and do not correspond exactly to the cases A–C in Fig. 1 denoted by uppercase letters. In the suspended case (a), the particles are mainly coupled by hydrodynamic interactions. One can find a linear velocity profile and a slight shear thinning. If one increases the shear rate $\dot{\gamma}>500/s$ (Pe $>15$), the particles arrange in layers. The same can be observed if the Debye-screening length of the electrostatic potential is increased (b), which means that the solvent contains fewer ions ($I<0.3$ mmol/l) to screen the particles.

FIG. 1. Schematic stability diagram for volume fraction $\Phi = 35\%$ in terms of $pH$-value and ionic strength involving three different microstructures: A clustering regime due to van der Waals attraction, stable suspensions where the charge of the colloidal particles prevents clustering, and a repulsive structure for further increased electrostatic repulsion. This work concentrates on state $A$ ($pH=6, I=3$ mmol/l) in the suspended region, state $B$ ($pH=6, I=7$ mmol/l) close to the border but already in the clustered region, and state $C$ ($pH=6, I=25$ mmol/l) in the clustered region. The borders are not sharp transitions, but notable in a change of the shear viscosity.

FIG. 2. (Color online) Images of four different cases. For better visibility we have chosen smaller systems than we usually use for the calculation of the viscosity. The colors denote velocities: Dark particles are slow, bright ones move fast. The potentials do not correspond exactly to the cases A–C in Fig. 1, but they show qualitatively the differences between the different states: (a) Suspension like in state $A$, at low shear rates. (b) Layer formation, which occurs in the suspension (state $A$) at high shear rates and in the repulsive regime already at moderate shear rates. (c) Strong clustering, like in state $C$, so that the single cluster in the simulation is formed. (d) Weak clustering close to the border like in state $B$, where the cluster can be broken into pieces, which follow the flow of the fluid (plug flow).
particle charges. On the other hand, if one increases the salt concentration, electrostatic repulsion is screened even more, and attractive van der Waals interaction becomes dominant ($I > 4 \text{ mmol/l}$). Then the particles form clusters (c), and viscosity rises. A special case, called "plug flow," can be observed for high shear rates, where it is possible to tear the clusters apart, and smaller parts of them follow with the flow of the solvent (d). This happens in our simulations for $I = 25 \text{ mmol/l}$ (state C) at a shear rate of $\dot{\gamma} > 500/s$ ($Pe > 15$). However, as long as there are only one or two big clusters in the system, it is too small to expect quantitative agreement with experiments. In these cases, we have to focus on state $B (I = 7 \text{ mmol/l})$ close to the border.

In our simulations, we restrict ourselves to the region around pH=6 where we find the border between the suspended region and the clustered regime at about $I = 4 \text{ mmol/l}$ in the simulations as well as in the experiments. Also the shear rate dependence of the viscosity is comparable in simulations and experiments as discussed in Sec. VII C.

In Ref. [62], a qualitative stability diagram similar to Fig. 1 has been shown. The borders there are shifted, since they depend on the threshold value for which one defines that the viscosity has increased. Correspondingly, if one is less sensitive to the viscosity increase, one would still consider the system to be suspended, if only weak cluster formation takes place.

### B. Total energy

In our simulations, we calculate the total energy, because it can be used as a tool to check if the response of the simulation to the variation of any parameter is consistent with the expectations, e.g., a decrease of the surface charge on the colloidal particles should cause the secondary minimum of the DLVO potential to become deeper and thus decrease the total energy, but if the total energy increases, this can be an indication for numerical instabilities.

The total energy comprises the kinetic energy of both fluid and colloidal particles, including thermal motion on the microscopic level, as well as the potential energy due to Coulomb repulsion, van der Waals attraction, and Hertz contact forces. Our simulations are carried out at room temperature ($T=295 \text{ K}$) and constant volume fraction. Supposing a linear velocity profile, the kinetic energy increases quadratically with the shear rate $\dot{\gamma}$. This can be observed if the electrostatic repulsion is, on the one hand, strong enough to prevent cluster formation due to van der Waals attraction and, on the other hand, weak enough, so that the colloidal particles can move relatively freely without undergoing a glass transition or crystalization.

If the interactions are strongly repulsive, i.e., in the case of very low salt concentration, where the Debye-screening length is large, one can see an extra contribution of the electrostatic repulsion to the total energy. If the volume fraction is low, the particles can still find a configuration in which the mean nearest-neighbor distance is larger than the interaction range of the repulsion. But, if the volume fraction is increased, the particles have to be packed closer, which leads to a constant positive offset to the total energy. It only depends on the potentials and on the volume fraction, but not on the shear rate.

In a similar way as for repulsive interactions, one can understand the negative energy contribution in the case of high salt concentrations: The DLVO potentials contain a minimum where attractive Van der Waals interaction is stronger than electrostatic repulsion. Then the particles form clusters and "try" to minimize their energy. In Fig. 3, for small shear rates the values for the energy in the clustered case of state C ($I = 25 \text{ mmol/l}$) are lower than for the suspended case of state A ($I = 3 \text{ mmol/l}$). We have plotted the total energy divided by the total number of all particles (fluid particles plus colloidal particles) and plotted in units of $k_BT$. The solid line is the analytical solution [Eq. (18)] for a linear velocity profile; the dashed lines are a guide to the eye.

\begin{equation}
E_{\text{tot}} = \frac{3}{2} k_BT N_{\text{tot}} + \frac{1}{24} V \bar{\rho} \dot{\gamma} L_z^2,
\end{equation}

where $N_{\text{tot}}$ is the total number of both fluid and colloidal particles, $V$ denotes the volume of the simulated system, $\bar{\rho}$ is the averaged mass density of the suspension, and $L_z$ is the extension in the $z$ direction (perpendicular to the shear plane). State A coincides very well with this curve.

For state C, the behavior is shear-rate-dependent: In contrast to the repulsive case, clusters can be broken up. This happens at a shear rate $\dot{\gamma} = 500/s$ ($Pe = 15$) (see Fig. 3) where one obtains two clusters moving in opposite directions. Since in this case the resistance of the system decreases, the velocity of the two clusters becomes larger. Since both clusters are moved as a whole, their energy becomes even larger than in the suspended case. If one further increases the shear rate, no
(big) clusters can form anymore, and the energies for both salt concentrations are nearly the same and correspond to the kinetic energy of a suspension with a nearly linear velocity profile. For \( \gamma = 1000 \text{ s}^{-1} \) (Pe=29), state C coincides with the analytic curve.

C. Shear profile and shear viscosity

In each of the three regimes a typical velocity profile of the shear flow occurs. For the suspended microstructure, one finds a linear velocity profile [Fig. 4(a)] with nearly Newtonian flow. The particles are distributed homogeneously, thus the density profile is structureless [Fig. 5(a)]. The motion of the particles is only weakly coupled by the hydrodynamic forces. At low shear rates \( \gamma > 500 \text{ s}^{-1} \) (Pe>15), the particles arrange in layers parallel to the shear plane, as can also be seen in the density profile [Fig. 5(b)]. This arrangement minimizes collisions between the particles. As a result, the shear viscosity decreases, as shown in Fig. 6, which we discuss more in detail below. Shear-induced layer formation has been reported in the literature for different experiments [63–67] and Stokesian dynamics simulations [19,68]. For low shear rates, Brownian motion disturbs the layers and prevents their formation. As shown in Ref. [19], hydrodynamic forces can destroy them as well, if the shear rate is high enough. In our simulations, we do not reach these conditions. In the simulations, shear rates up to 2000 \text{ s}^{-1} \) (Pe=59) can be realized before limitations of the simulation method influence the results. The increment of the shear angle in one SRD time step \( \gamma = \gamma_{\text{SRD}} \) amounts to about \( \pi/4 \) then, i.e., the offset in the \( x \) direction between two neighboring layers of SRD cells in the \( z \) direction amounts to one cell per SRD time step.

Furthermore, the volume fraction and the interaction range of the electrostatic repulsion, or the ionic strength, respectively, influence the layer formation: In the repulsive regime, the layers are more clearly already at moderate shear rates. It can be excluded that the effect is purely a finite-size effect, since for unsheared suspensions no layers can be observed, at least some particle diameters from the walls. In the repulsive regime, the particles try to optimize their local structure, but a long-range order as in the case of a sheared system cannot be seen.

In the clustered regime, the clusters move in the fluid as a whole. They are deformed, but since the interparticle forces are stronger than the hydrodynamic forces, the cluster moves more like a solid body than like a fluid. Often there is one big cluster that spans the whole system. The density profile [Fig. 5(c)] increases in the central region and decays at the regions close to the border, since particles from there join the central cluster. When averaging the velocity profile in the shear flow, one finds a very small velocity gradient in the center of the shear cell and fast-moving particles close to the wall, where the shear is imposed [Fig. 4(b)]. The velocity profile is nonlinear on the length scale of the simulations. In the experiment, the physical dimensions are much larger and therefore the velocity profile can become approximately linear again if the system consists of many large clusters. However, due to the computational effort in simulations it is impossible today to measure the shear viscosity for these
strongly inhomogeneous systems. We have scaled our system by a factor of 2 in the $x$ and $z$ directions, keeping the volume fraction $\Phi = 35\%$ constant, but we still observe one big cluster after some hundreds of SRD time steps, i.e., finite-size effects are still present in our simulations.

Closer to the border, clusters can then be broken up into small pieces by the hydrodynamic forces at least for high shear rates. In state C of Fig. 1, this happens for the first time at $\gamma = 500/s$ ($Pe = 15$), so that one can find two clusters in the system moving in opposite directions. The velocity profile of this case is shown in Fig. 4(c). For even higher shear rates or closer to the border (e.g., state B), the clusters are broken into smaller pieces. Then, they move in the shear flow with an approximately linear velocity profile. Due to van der Waals attraction, the system resists with stronger shear forces and the viscosity is higher than in the suspended case (Fig. 6).

In Fig. 6, the simulation results are shown together with the experimental results, both for the two cases of a slightly clustered system in state B ($I = 7$ mmol/l) and a suspension (state A, $I = 3$ mmol/l). For the suspension (state A), the viscosity decreases with the shear rate (“shear thinning”). The experimental data and the simulation are consistent within the accuracy of our model. There are several reasons why our model does not exactly fit the measurements: The most insecure factor that enters into the comparison is the measurement of the $\zeta$ potential. Starting from this point, we set up our charge regulation model to extrapolate to different salt concentrations, assuming two reactions: the only processes that determine the surface charge of the colloidal particles. Furthermore, we have monodisperse spherical particles, which is another simplification in our model. Then, the lubrication force as a correction for the finite resolution of the fluid method can only recover to a certain degree the hydrodynamics on smaller length scales than the cell size of the fluid simulation, e.g., we have not implemented other modes of lubrication than the “squeezing mode” [Eq. (3)].

However, we have done several tests where we have simulated systems with size and charge polydispersity in the order of magnitude corresponding to the experimental conditions. We have tested different sorts of boundary conditions, different ways to implement shear, and different coupling methods between fluid and particles. In most of the tests, the achieved shear viscosity in the simulation did not change notably. In our experience, the way shear is imposed and the particle size have the largest influence on the result.

Finally, one has to keep in mind that the viscosity of the suspension can be varied by more than one order of magnitude, e.g., by changing the ionic strength. In this context, the deviations between simulation and experiment are small. For the slightly clustered case (state B), an increase of the shear viscosity, compared to the suspended case, can be observed in the experiment as well as in the simulations. Shear thinning becomes more pronounced, because clusters are broken up, as mentioned above. However, the shear-rate dependence is stronger in the simulations than in the experiment. This can be the first indication of finite-size effects.

We have studied the dependence of the simulated shear viscosity in dependence of the system size. The effect is most important for low shear rates and thus we carried out several simulations for state A at $\gamma = 20/s$ ($Pe = 0.6$) and for state B at $\gamma = 50/s$ ($Pe = 1.4$). We have chosen these values because clustering is already too strong in state B at $\gamma = 20/s$ ($Pe = 0.6$) to reasonably determine a viscosity, and the system size dependence becomes too small for $\gamma = 50/s$ ($Pe = 1.4$) were state A. In Fig. 7, we plot the squared relative deviation between simulation and measurements against the system size. We do not know if the simulation results would exactly converge to the measured values if the simulated system is large enough. However, the figure shows the trend that the deviation becomes smaller for larger system sizes, but to reach in state B the same accuracy as in state A one would at least have to double the system size in each dimension. It then takes approximately twice as long for the system to relax to a steady state, resulting in a factor of 16 in the computational effort. This indicates that each single point of Fig. 6 would need approximately 3000 CPU hours. For smaller shear rates or even deeper in the clustered regime of the stability diagram, e.g., in state C ($I = 25$ mmol/l), the finite-size effects become more pronounced—ending up in the extreme case of only one big cluster existing in the system. For simulations with good accuracy, the effort again increases at least by the same factor.
Unfortunately, this would be the most interesting case with respect to soil mechanics and to understand landslides, which was our initial motivation. Anyhow, if we compare state $A$ and state $B$, shear thinning becomes stronger with increasing ionic strength. In the experiments, the effect becomes much stronger for larger ionic strengths (up to $I=65 \text{ mmol/l}$), where the viscosity for low shear rates is increased by more than a factor of 10. However, the fact that there is shear thinning and that it depends on the ionic strength is an interesting result, if we recall the fact that the lubrication force in our simulation can be interpreted as a velocity-dependent damping force, which becomes stronger for higher relative velocities. Therefore, one would intuitively expect shear thickening.

Finally, the limitations of the DLVO theory have to be taken into account. DLVO potentials are derived for dilute suspensions and hence large particle distances. This is not fulfilled in our case—and even less inside the clusters. There are theoretical attempts that address the shortcomings of DLVO theory: Explicit simulation of micro-ions [69], density-functional theory [70–72], response theory [73–76], Poisson-Boltzmann cell models [77–79], and full Poisson-Boltzmann theory [80–83], but they have other disadvantages—most of them require a large computational effort. For Poisson-Boltzmann cell models, one assumes homogeneously distributed colloidal particles, so that each of them can be regarded as a representative single particle in a Wigner-Seitz cell. Additionally, depending on the level Poisson-Boltzmann theory is included, a mysterious phase separation could be identified as an artifact of linearization [79]. Full Poisson-Boltzmann theory would require the calculation of the local potential, not only as done in our charge regulation model in the beginning of the simulation, but for the whole simulation box and in each time step. This would in principle provide a better description of the real system, but the computational effort would be much larger, making simulations of several thousands of particles impossible. The same applies to the approach of including the micro-ions explicitly in the simulation. One could obtain three- (many-) body interactions from full Poisson-Boltzmann theory and try to include them as a lookup table in the simulation. However, one would have to decrease the system size to keep the computational effort affordable. For our simulations we need relatively simple pair potentials to keep the computational costs within a limit. Nevertheless, the overall behavior can be reproduced by the simulation on a semiquantitative level. The reason for that might be the fact that in some of the above-mentioned theoretical attempts (density-functional theory and cell models), DLVO-like potentials are obtained with a renormalized charge and screening length. In our charge regulation model, we merely adjust a renormalized charge to the measurements of the $\zeta$ potential. This may be a general explanation why DLVO potentials can often be used, although the assumptions for DLVO theory are not fulfilled [52,54,84–86].

We have carried out simulations in the repulsive region of the stability diagram as well. We find layers parallel to the shear plane in analogy to Fig. 5(b). In contrast to the suspended regime, in the repulsive regime the layer structure is present—at least locally, but orientationally disordered—

\begin{figure}[h]
\centering
\includegraphics[width=0.5\textwidth]{graph.png}
\caption{Viscosity vs volume fraction for the repulsive region ($pH=6$ and $I=0.3 \text{ mmol/l}$). The shear rate was $\dot{\gamma}=100/s$ ($Pe=2.9$). The points are simulation results, the line is a guide to the eye.}
\end{figure}

\bigskip

even if no shear is applied. If shear flow is present, the shear plane marks one orientation that the layer structure adopts. In some cases, for very low ionic strengths one can observe shear bands so that the velocity gradient and thus the viscosity vary strongly in the system. Again, in the experiment, physical dimensions are much larger and on that length scale the velocity profile might be assumed to be linear when enough shear bands are in the system. The shear force and hence the viscosity increase with respect to the suspended regime, due to electrostatic repulsion. One can consider the particles together with the interaction range as soft spheres with an effective radius of the interaction range of the electrostatic repulsion. This effective radius in our case can be about 25% larger than the particle radius. Therefore, a transition to a repulsive structure already occurs in our systems between 35% and 40% volume fraction. Because of the smooth shape of the exponentially screened Coulomb potential, it is not a sharp glass transition as for hard spheres, but smooth and shear-rate-dependent as well. In Fig. 8, we have shown the dependence of the viscosity on the volume fraction for $pH=6$ and $I=0.3 \text{ mmol/l}$. Starting at $\Phi=0.3$, the shear viscosity starts to increase and reaches a value one decade larger beyond $\Phi=0.4$.

\section{VIII. SUMMARY AND OUTLOOK}

We have shown how to relate DLVO potentials to the conditions in a real aqueous suspension of $\text{Al}_2\text{O}_3$ particles. The behavior of shear viscosity has been studied in experiments and in simulations. We have found shear thinning due to a layer formation on the microscopic scale in the case of a suspension. If a clustered system is sheared, clusters are broken up into pieces by the imposed shear, which leads to a stronger shear thinning than in the suspended case. Close to the border, we are able to reproduce the measured shear viscosity in the simulation.

Deep in the clustered regime, we have found that our particles form one big cluster in the system that can be broken up by the hydrodynamic forces of the shear flow. For strongly clustered systems at low shear rate, which would be the most interesting case for soil mechanics, there are strong finite-size effects. One attempt to address this problem is to
increase the size of the simulated system. As we have shown with our work, the computational effort increases to an extent that a parallelized simulation code would be necessary. However, this might not be sufficient, since in both cases, for low shear rates and for strongly attractive interactions, the finite-size effects become stronger. Since in these simulations a considerable amount of the computing time is consumed by the particles inside a cluster, one could think of a more coarse-grained description of the clusters. Nevertheless, input data for such a model could be obtained using our present simulation code. Depending on the model, one might need information about the shear resistance of a single cluster, depending on the cluster size and shape. This would be very difficult to measure, but it could be calculated in a small simulation using our combined MD and SRD algorithm.

ACKNOWLEDGMENTS

This work has been financed by the German Research Foundation (DFG) within the project DFG-FOR 371 “Peloide.” We thank G. Gudehus, G. Huber, M. Külzer, L. Har nau, and S. Richter for valuable collaboration. Parts of this work have resulted from a collaboration with the group of A. Coniglio, Naples, Italy. M. H. thanks him and his group for their hospitality and for the valuable support during his stay there. M. H. thanks the DAAD for the financial support that enabled him to stay. H. J. H. thanks the Alexander von Humbolt Stiftung and the Max Planck Gesellschaft for the Max Planck prize. M. B. is deeply indebted to M. Biesheuvel and S.-H. Behrens for helpful discussions on charge regulation models. The computations were performed on the IBM p690 cluster at the Forschungszentrum Jülich, Germany.


SHEAR VISCOSITY OF CLAYLIKE COLLOIDS IN...
HECHT et al. PHYSICAL REVIEW E 74, 021403 (2006)

[87] Note: The 2 in the denominator of Eq. [33] in Ref. [30] has to be a 3 for the 3D case.
Stability diagram for dense suspensions of model colloidal Al$_2$O$_3$-particles in shear flow

M. Hecht, J. Harting, and H. J. Herrmann
Stability diagram for dense suspensions of model colloidal Al$_2$O$_3$ particles in shear flow

Martin Hecht,1 Jens Harting,1 and Hans J. Herrmann2,3
1Institute for Computational Physics, Pfaffenwaldring 27, D-70569 Stuttgart, Germany
2Computational Physics, IFB, Schaufnatterstrasse 6, ETH Zürich, CH-8093 Zürich, Switzerland
3Departamento de Física, Universidade Federal do Ceará Campus do Pici, 60451-970 Fortaleza CE, Brazil

(Received 22 December 2006; published 21 May 2007)

In Al$_2$O$_3$ suspensions, depending on the experimental conditions, very different microstructures can be found, comprising fluidlike suspensions, a repulsive structure, and a clustered microstructure. For technical processing in ceramics, the knowledge of the microstructure is of importance, since it essentially determines the stability of a workpiece to be produced. To enlighten this topic, we investigate these suspensions under shear by means of simulations. We observe cluster formation on two different length scales: the distance of nearest neighbors and on the length scale of the system size. We find that the clustering behavior does not depend on the length scale of observation. If interparticle interactions are not attractive the particles form layers in the shear flow. The results are summarized in a stability diagram.

DOI: 10.1103/PhysRevE.75.051404 PACS number(s): 82.70.–y, 47.11.–j, 02.70.Ns, 77.84.Nh

I. INTRODUCTION

Colloid science is a very fascinating research field, gaining more and more importance in the last years. It closely connects physics, chemistry, material science, biology, and several branches of engineering technology. According to its key role in modern science a considerable amount of research has been performed to describe colloidal suspensions from a theoretical point of view and by simulations [1–6] as well as to understand the particle-particle interactions [7–12], the phase behavior [13–16], the relevant processes on the microscale, and their influence on macroscopic parameters [17–19]. Colloidal suspensions are in fact complicated systems, since different time and length scales are involved. The particle sizes are on a mesoscopic length scale, i.e., in the range of nanometers up to micrometers. In systems of particles sized on this length scale Brownian motion often cannot be neglected. Depending on the particle sizes, materials, and concentrations, different interactions are of relevance and often several of them are in a subtle interplay: electrostatic repulsion, depletion forces, van der Waals attraction, hydrodynamic interaction, and Brownian motion are the most important influences. The properties of the suspension are strongly depending on the balance of the microscopic forces between the particles. Especially for industrial processes, where one needs to optimize certain material properties a detailed understanding of the relevant influences is needed. The stability of different microstructures and especially the clustering process are key properties which are of interest.

In our work we investigate these properties, focusing on Al$_2$O$_3$ particles of diameter 0.37 µm suspended in water. This is a widely used material in ceramics [20,21]. We have developed a simulation code for a Brownian suspension [22] and have adjusted the simulation parameters so that the simulation corresponds quantitatively to a real suspension such that experimental data can be compared directly. The diffusion coefficient, sedimentation velocity [22], and the viscosity of the suspension can be reproduced [23]. We also have tested the influence of polydispersity and found that its influence on the results is small. It is much more important to choose the correct mean size of the particles [23]. For Al$_2$O$_3$ suspensions attractive van der Waals forces are important for the behavior of this material. Electrostatic repulsion of the charged particles counteracts the attraction and can prevent clustering depending on the particle surface charge. In Ref. [23] we have presented how one can relate parameters of Derjaguin-Landau-Vervey-Overbeek (DLVO) potentials [7,8] with experimental conditions. In the experiment one can control the pH value and the salt concentration. The latter can be expressed by the ionic strength I, which is an effective concentration of all ions present in the solution. Both the pH value and the ionic strength influence the charge of the colloidal particles. We have shown that for not too strongly attractive forces one can obtain reasonable quantitative agreement with experimental results: by adjusting the lubrication force, which represents the short range hydrodynamics, we are able to reproduce rheological data of a real suspension [23].

Three regimes can be identified and plotted in a stability diagram [23], which we want to investigate here in more detail: a clustered regime, in which particles aggregate to clusters, a fluidlike and stable suspension, and a repulsive region, for which the microstructure is similar to the ones known from glassy systems. From our previous work we know that our model works well, even quantitatively, in the suspended regime of the stability diagram and close to the borders between the different microstructures. In this paper we extend our investigations to different pH values, deeper in the clustered regime, and to the repulsive structure. We expect to gain insight to the microscopic structure on a qualitative level.

On these grounds we have explored the stability diagram of Al$_2$O$_3$ suspensions. The particles are uncharged close to the so called “isoelectric point” at pH=8.7. There, for all ionic strengths, the particles form clusters. For lower pH values particles can be stabilized in solution, because they are charged. For low pH values, low salt concentrations, and high volume fractions a repulsive structure can be found.

In the following section we shortly describe our simulation method. After that we discuss the criteria we apply to
characterize the microstructures. We utilize the pair correlation function and the structure factor to characterize the clustering behavior. Both of them in principle contain the same information, but we concentrate on certain peaks in either of them. Each peak in the correlation function and in the structure factor corresponds to a certain length scale and we chose either the correlation function or the structure factor, depending on which of the two quantities is more suitable under numerical criterions to observe on a given length scale. In the section thereafter we describe our simulation setup. In the results section we start with the discussion of the correlation function and the structure factor. Additionally, we evaluate the so-called demixing parameter $\Psi$ [24]. To characterize the repulsive region we evaluate the mean squared displacement (MSD), which shows a plateau, if the particle motion consists of different processes acting on well separated time scales. Finally, the results are summarized in a stability diagram for our $\mathrm{Al}_2\mathrm{O}_3$ suspension. It shows the behavior of the suspension in an intuitive way and helps to design industrial processes using this material.

II. SIMULATION METHOD

Our simulation method consists of two parts: a molecular dynamics (MD) code, which treats the colloidal particles, and a stochastic rotation dynamics (SRD) simulation for the fluid solvent.

In the MD part of our simulation the colloidal particles are represented by monodisperse spheres. We include effective electrostatic interactions and van der Waals attraction, known as DLVO potentials [7,8], a lubrication force, and Hertzian contact forces. DLVO potentials are composed of two terms, the first one being an exponentially screened Coulomb potential due to the surface charge of the suspended particles

$$V_{\text{Coul}} = 4\pi\varepsilon_0 \left[ \frac{2 + \kappa d}{1 + \kappa d} \right] \frac{Z_{\text{eff}}^{\text{aq}} e}{4\kappa d T} \left[ \frac{Z_{\text{eff}}^{\text{aq}} e}{4\kappa d T} \right] \exp(-\kappa(r-d)), \tag{1}$$

where $d$ denotes the particle diameter and $r$ is the distance between the particle centers. $e$ is the elementary charge, $T$ is the temperature, $k_B$ is the Boltzmann constant, and $Z_{\text{eff}}^{\text{aq}}$ is the valency of the ions of added salt. $\varepsilon_0$ is the permittivity of the vacuum, $k_B T$ is the relative dielectric constant of the solvent. $\kappa$ is the inverse Debye length defined by $\kappa^2 = 8\pi\varepsilon_0 \kappa d$, with the ionic strength $I$ and the Bjerrum length $\kappa d \approx 7$ Å. The first fraction in Eq. (1) is a correction to the DLVO potential (in the form used in Ref. [25]), which takes the surface curvature into account and is valid for spherical particles [26].

The effective surface potential $\zeta$ is the electrostatic potential at the border between the diffuse layer and the compact layer. Smoluchowski related it to the electrokinetic mobility of the particle as $\mu = \zeta e / \eta$ [27]. The $\zeta$ potential can be related to the pH value of the solvent with a so-called $2pK$ charge regulation model [23].

The Coulomb term competes with the second part of the DLVO potential which is given by the attractive van der Waals interaction

$$V_{\text{vdw}} = A_{\text{vdw}} \frac{d^2}{r^2} \left[ \frac{d^2}{r^2} + \frac{d^2}{r^2} + 2 \ln \left( \frac{r^2 - d^2}{r^2} \right) \right]. \tag{2}$$

$A_{\text{vdw}} = 4.76 \times 10^{-20}$ is the Hamaker constant [25]. The DLVO potentials exhibit two minima, one primary minimum, where the particles touch each other. Equation (2) diverges where in reality the primary minimum of the potential is located. Therefore we model the primary minimum by cutting off the DLVO potentials for small separations and substituting them by a parabola which is connected in a continuously differentiable manner to the DLVO potential. This cutoff is made at distances of several nanometers, where the potential has already reached negative values, i.e., where the potential descends to the primary minimum. The primary minimum is separated by a potential barrier from the secondary minimum, which occurs at larger particle distances and which is less deep. For low salt concentrations, i.e., large Debye screening lengths, the electrostatic repulsion overcompensates the van der Waals attraction and the secondary minimum disappears.

Long range hydrodynamic interactions are taken into account in the simulation for the fluid as described below. This can only reproduce interactions correctly down to a certain length scale. On shorter distances, a lubrication force has to be introduced explicitly in the molecular dynamics simulation:

$$F_{\text{ab}} = -\dot{\mathbf{v}}_{\text{rel}} \cdot \left( \frac{6\pi \eta d^2}{r^4} \right), \tag{3}$$

with the relative velocity $\dot{\mathbf{v}}_{\text{rel}}$. Projected on the connecting line of the particle centers, $\eta$ is the dynamic viscosity of the fluid. Our implementation of the interaction contains cutoff radii to connect it to the long-range hydrodynamics and to avoid numerical instability for particles touching each other [22,23].

To avoid that the particles penetrate each other, we are using a Hertz force described by the potential

$$V_{\text{Hertz}} = K(d-r)^{5/2} \text{ if } r < d, \tag{4}$$

where $K$ is an interaction constant and additionally a damping term

$$F_{\text{Damp}} = -\dot{\mathbf{v}}_{\text{rel}} \cdot \beta \dot{\mathbf{r}} \text{ if } d-r, \tag{5}$$

with a damping constant $\beta$. For the integration of the translational motion we utilize a velocity Verlet algorithm [28].

For the simulation of a fluid solvent, many different simulation methods have been proposed: Stokesian dynamics (SD) [3,29,30], accelerated Stokesian dynamics (ASD) [31,32], pair drag simulations [4], Brownian dynamics (BD) [25,33], lattice Boltzmann (LB) [1,2,34,35], and stochastic rotation dynamics (SRD) [22,36,37]. These mesoscopic fluid simulation methods have in common that they make certain approximations to reduce the computational effort. Some of them include thermal noise intrinsically, or it can be included consistently. They scale differently with the number of em-
bedded particles and the complexity of the algorithm differs largely. We apply the stochastic rotation dynamics method (SRD) introduced by Malevanets and Kapral \cite{38,39}. It intrinsically contains fluctuations, is easy to implement, and has been shown to be well suitable for simulations of colloidal and polymer suspensions \cite{22,23,36,37,40,41,42} and very recently for star polymers in shear flow \cite{43}. The method is also known as “real-coded lattice gas” \cite{36} or as “multiparticle-collision dynamics” (MPCD) \cite{44}. It is based on so-called fluid particles with continuous positions and velocities. A streaming step and an interaction step are performed alternately. In the streaming step, each particle is moved according to

\[ \mathbf{r}_i(t + \tau) = \mathbf{r}_i(t) + \tau \mathbf{v}_i(t), \]

where \( \mathbf{r}_i(t) \) denotes the position of the particle \( i \) at time \( t \) and \( \tau \) is the time step. In the interaction step the fluid particles are sorted into cubic cells of a regular lattice and only the particles within the same cell interact among each other according to an artificial interaction. The interaction step is designed to exchange momentum among the particles, but at the same time to conserve total energy and total momentum within each cell, and to be very simple, i.e., computationally cheap: each cell \( j \) is treated independently. First, the mean velocity \( \mathbf{u}_j(t') = \frac{1}{N_j(t')} \sum_{i=1}^{N_j(t')} \mathbf{v}_i(t) \) is calculated. \( N_j(t') \) is the number of fluid particles contained in cell \( j \) at time \( t' = t + \tau \). Then, the velocities of each fluid particle in cell \( j \) are rotated according to

\[ \mathbf{v}_i(t + \tau) = \mathbf{u}_j(t') + \Omega_j(t') \cdot [\mathbf{v}_i(t) - \mathbf{u}_j(t')]. \]

\( \Omega_j(t') \) is a rotation matrix, which is independently chosen at random for each time step and each cell. We use rotations about one of the coordinate axes by an angle \( \pi \alpha \), with \( \alpha \) fixed. The coordinate axis as well as the sign of the rotation are chosen at random, resulting in six possible rotation matrices. To remove anomalies introduced by the regular grid, one can either choose a mean free path of the order of the cell size or shift the whole grid by a random vector once per SRD time step as proposed by Ihle and Kroll \cite{45,46}.

Three different methods to couple the SRD and the MD simulation have been introduced in the literature. Inoue et al. \cite{22} proposed a way to implement no slip boundary conditions on the particle surface \cite{36}. Paddling and Louis very recently came up with full slip boundaries, where the fluid particles interact via Lennard-Jones potentials with the colloidal particles \cite{47}. Falck et al. \cite{48} have developed a “more coarse grained” method which we use for the simulations of the present paper and which we describe shortly in the following.

To couple the colloidal particles to the fluid, the colloidal particles are sorted into the SRD cells and their velocities are included in the rotation step. One has to use the mass of each particle—colloidal or fluid particle—as a weight factor when calculating the mean velocity

\[ \mathbf{u}_j(t') = \frac{1}{M_j(t')} \sum_{i=1}^{N_j(t')} \mathbf{v}_i(t) m_i, \]

with

\[ M_j(t') = \sum_{i=1}^{N_j(t')} m_i, \]

where we sum over all colloidal and fluid particles in the cell, so that \( N_j(t') \) is the total number of both particles, fluid plus colloidal ones. \( m_i \) is the mass of the particle with index \( i \) and \( M_j(t') \) gives the total mass contained in cell \( j \) at time \( t' = t + \tau \). In summary, the algorithm for the fluid simulation is described by Eqs. (6)–(9). To some of our simulations we apply shear. This is realized by explicitly setting the mean velocity \( \mathbf{u}_j \) to the shear velocity in the cells close to the border of the system. Both colloidal and fluid particles are included in this additional step. A thermostat is applied to remove the energy introduced to the system by the shear force.

In this paper we examine the microstructures obtained in our simulations for different conditions. We vary the pH value and the ionic strength \( \lambda \). The shear rate \( \dot{\gamma} \) as an external influence is varied as well. We classify the microstructures in three categories: suspended, clustered, and repulsive. In the suspended case, the particles can move freely in the fluid and do not form stable clusters. In the clustered regime the particles form clusters due to attractive van der Waals forces. These clusters can be torn apart if shear is applied. In some of our simulations the clusters are very weakly connected and at small shear rates they are not only broken up into smaller pieces, but they dissolve to freely moving individual particles. In this case, we assign the microstructure to the suspended region, although in complete absence of the shear flow clusters are formed. At the borders between the different regimes in fact no sharp transitions can be observed. The DLVO forces rather steadily increase and compete with the hydrodynamic interactions. Accordingly, in experiments one cannot observe a sudden solidification, but a steadily increasing viscosity when leaving the suspended regime \cite{23}.

Similarly as for attractive forces, repulsive interactions can restrict the mobility of the particles. If this happens, the mean squared displacement of the particles shows a pronounced plateau, as it can be found in glassy systems. However, we speak of a “repulsive structure,” because the change of the viscosity is not as strong as in glasses, where it often changes by many orders of magnitude, when the glass transition is approached. In addition, to claim that a system shows a glassy behavior would require one to investigate the temperature dependence of a typical time (e.g., particle diffusion time) and to show its divergence as the glass temperature is approached. This is difficult to do in the framework of our simulation model \cite{22} and therefore we prefer to speak about a “repulsive structure” which might be identified as a colloidal glass in future work.
In this paper we would like to emphasize the analysis of the microstructure for different conditions. Our aim is to reproduce a so-called stability diagram by simulations. The stability diagram depicts the respective microstructure depending on the pH value and the ionic strength \( I \). We apply different numerical tools to analyze the microstructure in our simulations and finally arrive at a stability diagram shown in Fig. 10, which summarizes the results which we present in the following sections.

The first tool we apply for that is the pair correlation function

\[
g(r) = \frac{V}{N} \left( \sum_i \sum_{j \neq i} \delta(r-r_{ij}) \right)
\]

(10)

(see Ref. [28], p. 55), where \( V \) is the volume, \( N \) is the number of particles, and \( r_{ij} \) is the distance of two particles \( i \) and \( j \); this can be used for a first characterization of the system. It shows maxima at certain typical particle distances, e.g., there is a nearest neighbor peak, and more complicated structures at larger distances, which we have assigned to typical particle configurations for small distances [22]. Here, we use the next-neighbor peak to analyze our data, which provides information of the short-range structure of the suspension. To observe the long-range structure we move on to its complementary quantity: the structure factor, defined by

\[
S(k) = \frac{1}{N} \sum_{m=1}^{N} \exp(ik \cdot r_m),
\]

(11)

where \( N \) is the number of particles, and \( r_m \) is the vector from particle \( l \) to particle \( m \). \( i \) denotes the imaginary unit here. The structure factor is defined in \( k \)-space and it is related to the pair correlation function in real space by a three dimensional Fourier transform:

\[
S(k) - 1 = \int \mathrm{d}r \exp(ik \cdot r)g(r),
\]

(12)

with the density \( \rho \). Therefore in principle the structure factor contains the same information as the pair correlation function. However, due to numerical reasons and our implementation of shear boundary conditions it is easier to observe the long-range structure in the structure factor than in the pair correlation function. For our evaluation it is important that in a finite system \( k \) vectors are restricted to discrete vectors \( 2\pi \{ \frac{2\pi}{L_x}, \frac{2\pi}{L_y}, \frac{2\pi}{L_z} \} \) with \( L_x, y, z \) being the system size in the \( x, y, \) and \( z \) directions and \( n_{x,y,z} \in \mathbb{Z} \). We evaluate \( S(k) \) for these \( k \) vectors and use \( |S(k)| \) for further analysis. Since we do not evaluate the anisotropy of the structure factor in this study, we average over all possible orientations of the \( k \) vector. We sort the absolute values of the \( k \) vectors into intervals and average within these intervals over the values of the structure factors. Let us now discuss some typical features of the structure factor, corresponding to typical length scales present in the system.

One typical length for dense systems is about one particle diameter, the distance particle centers have to keep so that they do not overlap. The corresponding peak of the structure factor is the nearest-neighbor peak at \( k = \frac{2\pi}{d} \). In a single crystalline the peak would be sharp, since the particle positions are well defined, whereas in our case of a suspension there is a certain disorder, which broadens the peak.

Similar to the nearest-neighbor peak another peak can be detected at twice the \( k \) vector, which corresponds to a distance of one particle radius. This does not necessarily mean that there are particles whose centers are only one particle radius apart. It only means that for a certain \( k \) vector the addends in Eq. (11) do not cancel out each other anymore.

As already mentioned, later in this paper we focus on the low-\( k \) peak which corresponds to a length scale of the size of the system.

IV. SIMULATION SETUP

In this study the colloidal particles are represented by three dimensional spheres of \( d = 0.37 \, \mu \text{m} \) in diameter. This is the mean diameter of the particles used in the experiments to which we refer in Ref. [23]. We have simulated a small volume, \( 24d = 8.88 \, \mu \text{m} \) long in the \( x \) direction, which is the shear direction, and \( 12d = 4.44 \, \mu \text{m} \) long in the \( y \) and \( z \) directions. We have varied the volume fraction between \( \Phi = 10\% \) (660 particles) and \( \Phi = 40\% \) (2640 particles). Most of the simulations were performed at \( \Phi = 35\% \) (2310 particles). We use periodic boundaries in the \( x \) and \( y \) directions and closed boundaries in the \( z \) direction [23]. Shear is applied in the \( x \) direction by moving small zones of particles and fluid close to the wall with a given shear velocity. The \( xy \) plane is our shear plane. For simulations without shear, to achieve the best comparability, we use the same boundary conditions and just set the shear rate to \( \gamma = 0 \). In addition we have performed simulations with two different shear rates: with \( \gamma = 100/s \) and with \( \gamma = 500/s \). For better comparability to other works the Peclét number

\[
\text{Pe} = 6\pi \eta R^2 /k_B T,
\]

(13)

which expresses the importance of Brownian motion with respect to the shear flow, is useful. The Peclét numbers in our shear simulations are \( \text{Pe} = 3 \) and \( \text{Pe} = 15 \), respectively.

V. RESULTS AND DISCUSSION

First, we focus on simulations without shear, where one can predict intuitively what should happen. Qualitatively the results are similar to our earlier work [22], but the quantitative relation between the \( \text{pH} \) value and the potentials is new. The relation was presented in Ref. [23], but here we apply it to different cases and we focus more on the characterization of the microstructure. However, given the particle-particle interaction potentials, the microstructure in equilibrium can be predicted easily, at least on a qualitative level. But, the matter changes and gets more sophisticated when shear is applied and an interplay between shear flow and particle-
Figure 1. Dependence of the particle correlation function on the pH value, $I=3$ mmol/l, $\gamma=0/s$, $\Phi=35\%$. The plots for four different pH values are shifted against each other for better visibility by a factor of 3. For $pH=4$ the particles are not clustered. Hence, the structure at $r/\sigma=2$ is less sharp than in the other three curves of the plot and the nearest-neighbor peak ($at \tau=1$) is broad. For $pH=6.5$ slight clustering starts, the structures become sharper. For $pH=7.7$ strong cluster formation is reflected in very sharp structures. For $pH=8.5$ electrostatic repulsion nearly disappears so that no barrier between primary and secondary minimum exists anymore. The particles cannot rearrange anymore, and therefore the structures labeled by the arrows become smoothened compared to the case of $pH=7.7$.

Particle interactions become responsible for the resulting microstructure.

A. Correlation function

For constant ionic strength $I=3$ mmol/l the local microstructure can be examined using the correlation function. Depending on the pH value the behavior of the system changes from a repulsive structure around $pH=4$ to a stable suspension around $pH=6$ towards a clustered region if the pH value is further increased, until the isoelectric point is reached at $pH=8.7$. There clustering occurs in any case, independent of the ionic strength. This can be seen in the structure of the correlation function: electrostatic repulsion prevents clustering (at $pH=4$). Particles are suspended, and there is no fixed long range ordering in the system. The correlation function (Fig. 1) shows a maximum at a typical nearest-neighbor distance slightly above $r/\sigma=1$ with $d$ denoting the particle diameter, then in the layer of next neighbors small correlations can be found (at $r/\sigma=2$). For larger distances the correlation function is rather constant.

When the pH value is increased, the surface charge is lower, which at first causes the particles to approach each other more closely. The maximum of the correlation function is shifted to smaller distances (see Fig. 1, note that the curves are shifted vertically in the plot by a factor of 3 for better visibility). Then, van der Waals attraction becomes more important and clustering begins. One can see this in the correlation function where a sharp structure at particle distances between 1.5 and 2 particle diameters occurs. In a solidlike cluster the position of the next neighbor is fixed more sharply than in the suspension, consequently the nearest-neighbor peak becomes sharper, too, and its height is increased. Close to the isoelectric point ($pH=8.7$) the barrier between primary and secondary minimum disappears. The particles, once clustered, cannot rearrange anymore, and therefore the correlations to the next neighbors become less sharp again (compare the cases of $pH=8.7$ and $pH=7.7$ in Fig. 1 at the positions denoted by the arrows).

Instead of varying the pH value, one can also vary the ionic strength to achieve similar effects. Increasing the ionic strength, experimentally speaking “adding salt” decreases the screening length $1/\kappa$ and therefore the attractive forces become more important: the particles start to form clusters. On the contrary, if the ionic strength is decreased—experimentally speaking a dialysis step is performed—the electrostatic repulsion prevents cluster formation and, for sufficiently strong repulsion, long range correlations occur as soon as the range of the repulsion reaches the mean particle distance.

In Fig. 2 one can see clustering in the primary minimum of the potential as well as in the very shallow secondary minimum. The correlation function plotted there is evaluated after 1000 SRD time steps in a simulation for $pH=7$ and $I=3$ mmol/l at a volume fraction of $\Phi=35\%$ sheared with $\gamma=100/s$. Note that the shear flow is not of essential importance here, but it supports the particles to overcome the potential barrier between the primary and the secondary minimum. The dotted lines in Fig. 2 denote the zero line.

A secondary minimum only exists if the screening length of the electrostatic repulsion is short enough, i.e., if the ionic strength is large enough. According to the depth of the potential, clustering in the primary minimum is associated with much stronger forces than in the secondary minimum.

The effects described up to here can be observed with or without shear qualitatively in an analogous manner. If the suspension is sheared clustering occurs at higher pH values and the peaks found in the correlation function are slightly broadened, because the relative particle positions are less fixed. But a new feature appears, if a stable suspension of not too high volume fraction is sheared. Induced by the shear particles arrange themselves in layers. Regular nearest-neighbor distances in the shear plane cause the correlation function to become more structured even for large distances (see Fig. 3). The long range structure of the pair correlation function appears after a transient time the particles need to
arrange themselves in the layered structure. The curves in Fig. 3 correspond to the same simulation, but for the layered structure. The plots are shifted vertically by a factor of 2 for better visibility.

FIG. 3. Correlation function for \( \dot{\gamma} = 500/s, \text{pH} = 6, I = 0.3 \text{ mmol/l}, \Phi = 32\% \): depending on the simulation time the peaks indicated by the arrows from below split into two (indicated by the arrows from above), and long range correlations occur (for \( \dot{\gamma} > 4 \)). This reflects the process of layer formation and appearance of a regular (hexagonal) order in the layers. From bottom to top several states for increasing time are shown. The plots are shifted vertically by a factor of 2 for better visibility.

Hecht, Harting, and Herrmann

PHYSICAL REVIEW E 75, 051404 (2007)

FIG. 4. Nearest-neighbor peak (primary and secondary minimum of the potential) with \( I = 3 \text{ mmol/l}, \Phi = 35\% \): for low pH values clustering is prevented by the electrostatic repulsion. For high pH values the particles form clusters, which is reflected by an increased nearest-neighbor peak. First, shear prevents clustering, then depending on the shear rate, cluster formation takes place. Low shear rates even support cluster formation at high pH values.

B. Structure factor

The pair correlation function can be used to characterize the local order of the microstructure on the length scale of the particle size. However, to do the characterization on the length scale of the system size, we use the structure factor \( S(k) \). In principle we could integrate the correlation function \( g(r) \) over an interval for large \( r \). However, due to our implementation of shear (see Ref. [23]), we have to close the boundary in the \( z \) direction. Therefore we already find restrictions for \( r \) around half the extension of the system in this direction. We find that it is easier to handle these finite size restrictions by moving on to the structure factor. There, we always find a low-k peak, even if no cluster formation takes place in the simulation. In contrast to experiments, where this peak only appears for clustered samples, it reflects the presence of a typical length of the system size in the simulation. Namely, the finite size of the simulation volume is the typical length, which appears in the structure factor by the low-k peak. This only produces a constant offset when integrating over the low-k peak as we are going to do below in this section. Thus it is easy to handle the influence of the finite system size in \( k \) space.

In Fig. 5 we have plotted several typical structure factors of our simulations. For these plots the pH value is fixed to pH=6. Cases (a) and (b) are sheared with \( \dot{\gamma} = 500/s \) at an ionic strength of \( I = 0.3 \text{ mmol/l} \). In case (a) the volume fraction \( \Phi = 20\% \) is relatively low. Therefore the particles can arrange themselves in layers parallel to the shear plane, which move relatively independently in the shear flow. They have a certain distance fixed in space and time. This can be seen in a sharp peak at a dimensionless \( k \) vector of \( k = 5.2 \), which corresponds to a distance of 1.2 particle diameters. In fact, this is exactly the distance between two neighboring layers, as one can easily verify by counting the layers in a snapshot of the system [Fig. 6(a)]. The particles in the layers do not have a fixed distance and therefore no second-order peak can be observed.

For case (b) the volume fraction is increased to \( \Phi = 35\% \). The particle layers are packed more densely and therefore the interactions between one layer and the neighboring one
become relevant. Particles jump from one layer to the other, which disturbs the flow and therefore the distance between the layers is not fixed anymore. The sharp peak on top of the nearest-neighbor peak disappears. Instead of that, in each layer a regular hexagonal order appears and therefore the second-order peak is much flatter, which depicts that the cluster is fractal.

In case (c) the ionic strength is increased to \( I = 25 \text{ mmol/l} \). The interparticle potentials are attractive enough that aggregation takes place. In this simulation we did not apply shear, therefore one finds only one big cluster in the system [compare Fig. 6(c)]. In the cluster the particles are packed more densely and consistently the nearest-neighbor peak in the structure factor is shifted to larger \( k \) vectors. The volume fraction is \( \Phi = 40\% \) in this case.

In case (d) the volume fraction is decreased to \( \Phi = 10\% \). The particles still form clusters, but their mobility is not high enough to create one compact cluster. The system has a fractal structure [see Fig. 6(d)]. This can be seen in the structure factor as well: The slope of the low-\( k \) peak is flatter in this case compared to cases (a)–(c). A flatter slope of the low-\( k \) peak is typical for structure factors of fractal objects. The fractal dimension of the cluster extracted from the slope of the low-\( k \) peak is 2.5. In experiments this relation is often used to determine the fractal dimension of a sample: Lattuada et al. [54] have evaluated the fractal dimension of agglomerates of latex particles from the slope of the structure factor. McCarthy et al. [55] give an introduction to scattering intensities at fractal objects, without mentioning the structure factor, but their arguments refer to the contribution of the structure factor on the scattering intensity. The underlying mechanism which is responsible for these structures is cluster cluster aggregation [56].

In Fig. 7 we show the dependence of the low-\( k \) peak of the structure factor on the \( \rho \)H value. Here we have integrated over dimensionless \( k \) vectors smaller than 3 which means we have captured structures larger than twice a particle diameter. A large integral over the low-\( k \) peak is due to a large inhomogeneity in the system. In one part of the system particles are present and in the other part they are not. In other words, we observe the process of cluster formation on a length scale of the system size. Without shear, particles cluster in the secondary minimum for all \( \rho \)H values. If the system is slightly sheared (\( \gamma = 100/s \)) clustering is suppressed for low \( \rho \)H values. Starting at \( \rho \)H=6 cluster formation starts and is even supported by the shear flow for \( \rho \)H values larger than 7.5. For large shear rates (\( \gamma = 500/s \)) cluster formation is suppressed by the shear flow. By analyzing the low-\( k \) peak of the structure factor one observes on the length scale of the
The same behavior of the system can be seen by analyzing the pair correlation function, as we have already shown in Fig. 4. In that case one analyzes the number of nearest neighbors, that means one observes the length scale of a particle diameter. Nevertheless, both graphs show the same behavior of the system, i.e., we have a consistent picture of the cluster formation process on the length scale of the nearest neighbors and on the length scale of the system size.

Thus we have confirmed that the cluster formation process is not limited to length scales smaller than our system size. This is reflected especially by the transition between pH = 7–8 and its shear rate dependence in the plots in Figs. 7 and 4. There is a strong similarity of the two plots, which are obtained by two evaluation methods referring to two different length scales. This confirms that the plots do not only reflect how clusters are formed on the respective length scale, but that the clustering process is a phenomenon which can be observed on any length scale by applying a suitable method to characterize it.

C. Density inhomogeneity

Another way to observe the cluster formation process is provided by the “demixing parameter” defined in Ref. [24] as follows: the system is divided into $n^3$ cubes and the particle density $\rho_p$ is evaluated in each cube. Then the demixing parameter is the mean squared deviation of the density

$$\Psi_d = \frac{1}{n^3} \sum_{k=0}^{n^3} (\rho_p - \bar{\rho})^2,$$  \hspace{1cm} (14)

where $\bar{\rho}$ is the mean density. For our elongated system we modify this definition and use $2n$ cubes in the $x$ direction, resulting in $2n^3$ cubes in total. The demixing parameter implicitly contains information about the density of the clusters. It is the density fluctuation of the whole system. If the clusters are more compact, voids have to appear and the demixing parameter increases, since the distribution of the local density becomes less homogeneous. For $I=5\text{ mmol/l}$ the demixing is even suppressed completely.

FIG. 8. Demixing parameter for constant pH = 6 and volume fraction $\Phi=35\%$ for three different shear rates: $\gamma=0/\text{s}$ (a), 100/\text{s} (b), and 500/\text{s} (c). Without shear only very weak demixing takes place. For $\gamma=100/\text{s}$ the strongest demixing can be observed for the system with the highest ionic strength $I=20\text{ mmol/l}$. For lower ionic strengths the effect decreases until it disappears completely at $I=1\text{ mmol/l}$, where the repulsive regime is reached. If the shear rate is further increased ($\gamma=500/\text{s}$), the clusters are less stable so that the system becomes less inhomogeneous. For $I=5\text{ mmol/l}$ the demixing is even suppressed completely.

FIG. 7. Low-$k$ peak for different pH values and different shear rates. The ionic strength $I$ is kept constant at $I=3\text{ mmol/l}$ and the volume fraction is always $\Phi=35\%$. For $\gamma=0/\text{s}$ the particles tend to cluster in the secondary minimum of the potential. This clustering can easily be broken up, if shear is applied. If the pH value is increased, shear cannot prevent cluster formation anymore. At low shear rates ($\gamma=100/\text{s}$) clustering is even enhanced, since the particles are brought closer to each other by the shear flow. Note that the offset of the plots reflects the finite size of our system combined with the closed boundary conditions (see the text).
decreased, the clustering effect decreases as well, until it
disappears completely at $I=1\ \text{mmol/l}$, where the...

There, also the intensity of the respective peaks for $\dot{\gamma}=100/s$ is less than for $\dot{\gamma}=100/s$. This shows again that large shear rates can inhibit cluster formation, whereas moderate shear rates can support the clustering process [the plots in Fig. 8(b) for $\dot{\gamma}=100/s$ are steeper than the ones in Fig. 8(a) for $\dot{\gamma}=0/s$]. In addition to the information already contained in other quantities we have presented, Fig. 8 shows the time evolution. One can see three regimes of time evolution: for very short times all plots are nearly constant. This means that the cluster formation has not yet started. Then, the plots for attractive potentials rise, which shows the onset of cluster formation. For large times the slope of the plots decreases, which reflects that the clusters have been formed and no more “demixing” takes place. Without shear [see Fig. 8(a)], it is remarkable that for the largest ionic strength the growth of the demixing parameter first becomes constant after 0.3 s, whereas for the other cases where clustering occurs, it grows further. This reflects that in the case of $I=20\ \text{mmol/l}$ the attraction is that strong that no reordering of the particles is possible anymore.

D. Repulsive regime

To characterize the repulsive regime, we evaluate the mean squared displacement for the particles. In Fig. 9 we plot the mean squared displacement for different ionic strengths. The $p\text{H}$ value is kept constant at $p\text{H}=6$ and the volume fraction is $\Phi=35\%$ for this plot. Three different regimes can be identified. For very short times, the ballistic regime, particles move on short distances without a notable influence by their neighbors. The distances are in the order of some percent of the particle diameter and the times are a few SRD time steps. For larger times the particles interact with their neighbors and therefore their mobility is limited due to collisions with the neighbors. This is reflected in the mean squared displacement by a plateau of reduced slope, which is more pronounced the more the mobility of the particles is restricted. For even larger time scales collective motion starts, i.e., clusters or groups of particles move, or single particles can escape from a cage formed by its neighbors.

Depending on the ionic strength different effects are important and thus the shape of the curve is different. For large ionic strengths the particles form clusters and these clusters may drift or rotate in the system. Then the collective motion is more dominant and the mean squared displacement grows faster than in single particle diffusion. The mean squared displacement does not show a plateau, then. But in the repulsive regime, the neighbors limit the motion of the particles, and the slope of the plateau is flatter, i.e., the plateau is even more pronounced, compared to the suspended case. In the repulsive regime the particles tend to arrange themselves in layers when shear is applied [23] and long range correlations can be found in unsheared systems [22].

E. Stability diagram

The results of the investigations presented in this paper can be summarized in a stability diagram for our Al$_2$O$_3$ suspension (Fig. 10). Three different microstructures can be identified: a repulsive structure, a suspended region, and a clustered region. In contrast to our previous work [22,23], we have explored the parameter space more in the repulsive regime and deeper in the clustered region. We use the mean
squared displacement, the demixing parameter $\Psi$ [24], the correlation function, and the structure factor to decide to which of the three microstructures a certain point in the stability diagram belongs. However, the borders between the regions are not sharp and they depend on the shear rate. We have indicated the crossover regions by the shaded patterns in the stability diagram. If the volume fraction is decreased, the region of the repulsive structure becomes smaller.

To decide if a state is in the suspended region or in the repulsive one of the stability diagram, we have compared the plots of the mean squared displacement for the simulations without shear. If the plateau was pronounced there, we have counted the state as repulsive. As a second criterion one can compare the pair correlation function. If there are long range correlations even though the system is not sheared, then the microstructure is the repulsive one. Finally, the shear force can be used to localize the border to the repulsive regime. For a given shear rate and a fixed volume fraction, the shear force depends on the particle interactions. If the shear force increases compared to a state well in the suspended regime, the motion of the particles is blocked by the electrostatic interaction in the repulsive regime.

As we have mentioned in Sec. III, without shear weak clustering can be seen in the suspended case as well, since there is no barrier for the particles to enter the secondary minimum of the DLVO potential, but the clusters can be broken up again very easily. Thus to decide if a state belongs to the clustered or to the suspended regime, we first study the snapshots of the system. If we see no clusters there, the clustered regime can be excluded. But, if we see clusters, we next check the density of the clusters and the onset time of the increase of the demixing parameter $\Psi$, which is a measure for the time it takes to form clusters. Both the density and the time are indications for the stability of the clusters. If they grow slowly and their density is low, we count the state to the suspended regime. The demixing parameter $\Psi$ shown in Fig. 8 implicitly contains information about the density of the clusters. It is the density fluctuation of the whole system. If the clusters are more compact, voids have to appear and the demixing parameter increases, since the distribution of the local density is broadened thereby. The stability diagram obtained by these criteria is consistent with the results of the simulations with shear flow, shown in Figs. 4 and 7. Especially, the increased cluster formation for $f=3$ mmol/l starting between pH=7 and 8 is reflected in an increased nearest-neighbor peak in Fig. 4, and low-k peak in Fig. 7, respectively, and in a border between suspended and clustered regime in Fig. 10. The repulsive structure for pH=4 and $f=3$ mmol/l cannot be recognized in Figs. 4 and 7, but a pronounced layer formation.

VI. SUMMARY AND OUTLOOK

We have simulated colloidal particles in shear flow and investigated how the clustering process due to attractive DLVO potentials is affected by the hydrodynamic forces. We find a consistent behavior on different length scales. The nearest-neighbor peak of the pair correlation function has been used to observe the direct neighborhood of the particles and the low-k peak of the structure factor to keep track of the length scales up to the system size. In both cases a suppression of the cluster formation by the shear flow can be seen at low pH values. For large pH values low shear rates even support the clustering process. In contrast, for high shear rates it suppresses the cluster formation. We have evaluated the mean squared displacement and the demixing parameter $\Psi$ [24] in order to draw the stability diagram as given in Fig. 10. To our knowledge this stability diagram for Al2O3 suspensions is reproduced quantitatively for the first time from simulations. It helps to predict the behavior of a real suspension. Our findings on the cluster formation process suggest that soft stirring can enhance the cluster formation in industrial processing of this material. Further investigations can be carried out on the fractal dimension and its dependence on the experimental conditions. The low-k peak of the structure factor can be used for that. The cluster size distribution could as well deliver interesting insights useful to design industrial processes. Apart from that one can apply our algorithm to different materials. To do so, one has to change the interaction constants and especially for the electrostatic repulsion the calibrations we have presented in Ref. [23] are necessary. We expect that the stability diagram does not change qualitatively, but the position of the borders will be different.

ACKNOWLEDGMENTS

This work has been financed by the German Research Foundation (DFG) within the project DFG-FOR 371 “Peloide.” We thank G. Gudehus, G. Huber, M. Külzer, L. Harnau, M. Bier, J. Reinhagen, and S. Richter for valuable collaboration. This work resulted in large part from the collaboration with the group of A. Comiglio, Universität di Napoli “Federico II,” Naples, Italy. M.H. thanks him and his group for his hospitality and for the valuable support during his stay there. M.H. thanks the DAAD for financial support which enabled him the stay. The computations were performed on the IBM p690 cluster at the Forschungszentrum Jülich, Germany and at HLRS, Stuttgart, Germany.

STABILITY DIAGRAM FOR DENSE SUSPENSIONS OF...

PHYSICAL REVIEW E 75, 051404 (2007)


Formation and growth of clusters in colloidal suspensions

M. Hecht, J. Harting, and H. J. Herrmann
Formation and growth of clusters in colloidal suspensions

Martin Hecht and Jens Harting
Institute for Computational Physics, Pfaffenwaldring 27, D-70569 Stuttgart, Germany
{hecht,jens}@icp.uni-stuttgart.de

Hans J. Herrmann
Computational Physics, IFB, Schafmattstr. 6, ETH Zürich, CH-8093 Zürich, Switzerland
hans@icp.uni-stuttgart.de

Received January 12, 2007
Revised Day Month Year

Depending on the pH-value and salt concentration of Al₂O₃ suspensions different microstructures can form. Especially the clustered one is of major interest for industrial purposes as found in the production of ceramics. In this paper we investigate the clustered microstructure by means of a coupled Stochastic Rotation Dynamics (SRD) and Molecular Dynamics (MD) simulation. In order to gain statistics within a system containing numerous clusters, large simulation volumes are needed. We present our parallel implementation of the simulation algorithm as well as a newly developed cluster detection and tracking algorithm. We then show first results of measured growth rates and cluster size distributions to validate the applicability of our method.

Keywords: Stochastic Rotation Dynamics; Molecular Dynamics; colloids; clustering

PACS Nos.: 82.70.-y, 47.11.+j, 02.70.Ns

1. Introduction

Colloidal suspensions in general are present everywhere in our daily life. Paintings, cosmetic products, and different kinds of food are some examples. They behave in a complex way, since different time and length scales are involved. The particle sizes are on a mesoscopic length scale, i.e., in the range of nanometers up to micrometers. Depending on the particle sizes, materials, and concentrations, different interactions are of relevance and often several of them are in a subtle interplay: electrostatic repulsion, depletion forces, van der Waals attraction, hydrodynamic interaction, Brownian motion, and gravity are the most important influences. Here, we are interested in colloids, where attractive van der Waals interaction is important for the description, i.e., where under certain circumstances cluster formation plays an important role\textsuperscript{1,2}. To model these systems experimentally, Al₂O₃ suspensions are commonly used\textsuperscript{3,4}. Al₂O₃ is also a common material in the ceramics industry. There, wet processing of suspensions, followed by a sinter process is a common practice. The stability of the resulting workpiece strongly depends on the properties of the
clusters formed before the sintering process. The size distribution, stability and local porosity of the clusters as well as the time dependence of their formation are only a few of the parameters of influence. In our work we investigate cluster formation of a sheared solution of spherical Al$_2$O$_3$ particles of diameter $0.37 \, \mu\text{m}$ in water. Cluster formation can have different reasons: depletion forces$^{5,6,7}$, like-charge attraction mediated by the counterions in the solvent$^{8,9,10}$, or, as in our case, van der Waals attraction$^{11,12,13}$. The shear flow can either support cluster formation at low shear rates, or it can suppress cluster formation at high shear rates as we have shown in$^{14}$. We adjust the simulation parameters so that the simulation corresponds quantitatively to a real suspension with 5% volume concentration under shear. The shear rate is kept fixed at $\dot{\gamma} = 20/\text{s}^{15,16}$. For Al$_2$O$_3$ suspensions attractive van der Waals forces compete with electrostatic repulsion. Depending on the particle surface charge, clustering due to attractive van der Waals forces can dominate or be prevented. We have presented how one can relate parameters of DLVO potentials$^{17,18}$ with experimentally tunable parameters, i.e., the pH-value and the salt concentration expressed by the ionic strength $I$, influence the charge of the colloidal particles$^{16}$. We explored the stability diagram of Al$_2$O$_3$ suspensions and reproduced that the particles are uncharged close to the so called “isoelectric point” at pH = 8.7, where they form clusters regardless of the ionic strength. For lower pH-values particles can be stabilized in solution. For very low pH-values, low salt concentrations, and high volume fractions a repulsive structure can be found. The particle size is on a mesoscopic length scale, where Brownian motion is relevant and long range hydrodynamic interactions are of importance. Therefore, we use “Stochastic Rotation Dynamics” (SRD), which includes both, hydrodynamics and Brownian motion for the description of the fluid solvent$^{19,20}$.

2. Simulation algorithm

Our simulation method is described in detail in$^{15,16}$ and consists of two parts: a Molecular Dynamics (MD) code, which treats the colloidal particles, and a Stochastic Rotation Dynamics (SRD) simulation for the fluid solvent. In the MD part we include effective electrostatic interactions and van der Waals attraction, known as DLVO potentials$^{17,18}$, a lubrication force and Hertzian contact forces. DLVO potentials are composed of two terms, the first one being an exponentially screened Coulomb potential due to the surface charge of the suspended particles

$$V_{\text{Coul}} = \pi \varepsilon_r \varepsilon_0 \left[ \frac{2 + \kappa d}{1 + \kappa d} \cdot \frac{4k_B T}{ze} \tanh \left( \frac{ze \zeta}{4k_B T} \right) \right]^2 \times \frac{d^2}{r} \exp(-\kappa[r-d]), \quad (1)$$

where $d$ denotes the particle diameter, $r$ the distance between the particle centers, $e$ the elementary charge, $T$ the temperature, $k_B$ the Boltzmann constant, and $z$ is the valency of the ions of added salt. $\varepsilon_0$ is the permittivity of the vacuum, $\varepsilon_r = 81$ the relative dielectric constant of the solvent, $\kappa$ the inverse Debye length defined by
\( \kappa^2 = 8\pi \ell_B I \), with ionic strength \( I \) and Bjerrum length \( \ell_B = 7 \text{ Å} \). The first fraction in Eq. (1) is a correction to the DLVO potential (in the form used in \(^{11}\)), which takes the surface curvature into account and is valid for spherical particles. The effective surface potential \( \zeta \) can be related to the pH-value of the solvent with a 2\( pK_H \) charge regulation model\(^{16}\). The Coulomb term competes with the attractive van der Waals interaction \( \left( A_H = 4.76 \cdot 10^{-20} \text{ J} \right) \) is the Hamaker constant\(^{11}\)

\[
V_{\text{vdW}} = -\frac{A_H}{12} \left[ \frac{d^2}{r^2 - d^2} + \frac{d^2}{r^2} + 2 \ln \left( \frac{r^2 - d^2}{r^2} \right) \right].
\]

The attractive contribution \( V_{\text{vdW}} \) is responsible for the cluster formation we observe. However, depending on the pH-value and the ionic strength, it may be overcompensated by the electrostatic repulsion. When particles get in contact, the potential has a minimum. However, Eq. (2) diverges due to the limitations of DLVO theory. We cut off the DLVO potentials and model the minimum by a parabola. The particle contacts are modeled as Hertzian contacts and for non-touching particles, below the resolution of the SRD algorithm short range hydrodynamics is corrected by a lubrication force, which we apply within the MD framework, as we have explained in Ref. \(^{15,16}\). For the integration of translational motion of the colloidal particles we utilize a velocity Verlet algorithm\(^{21}\) and for the fluid we apply the Stochastic Rotation Dynamics method (SRD)\(^{19,20}\). It intrinsically contains fluctuations, is easy to implement, and has been shown to be well suitable for simulations of colloidal and polymer suspensions\(^{22,23,24,25,15,16}\). The method is also known as “Real-coded Lattice Gas”\(^{22}\) or as “multi-particle-collision dynamics” (MPCD)\(^{24}\) and is based on coarse-grained fluid particles with continuous positions and velocities. A streaming step and an interaction step are performed alternately. In the streaming step, each particle \( i \) is moved according to \( \mathbf{r}_i(t + \tau) = \mathbf{r}_i(t) + \tau \mathbf{v}_i(t) \), where \( \mathbf{r}_i(t) \) denotes the position of the particle \( i \) at time \( t \) and \( \tau \) is the time step. In the interaction step fluid particles are sorted into cubic cells of a regular lattice and only the particles within the same cell interact according to an artificial collision rule which conserved energy and momentum. First, for each independent cell \( j \) the mean velocity \( \mathbf{u}_j(t') = \frac{1}{N_j(t')} \sum_{i=1}^{N_j(t')} \mathbf{v}_i(t) \) is calculated. \( N_j(t') \) is the number of fluid particles contained in cell \( j \) at time \( t' = t + \tau \). Then, the velocities of each fluid particle are rotated according to

\[
\mathbf{v}_i(t + \tau) = \mathbf{u}_j(t') + \Omega_j(t') \cdot [\mathbf{v}_i(t) - \mathbf{u}_j(t')].
\]

\( \Omega_j(t') \) is a rotation matrix, which is independently chosen at random for each time step and cell. Rotations are about one coordinate axes by a fixed angle \( \pm \alpha \). To couple colloidal particles and the fluid, the particles are sorted into SRD cells and their velocities are included in the rotation step. The masses of colloidal and fluid particles are used as a weight factor for the mean velocity

\[
\mathbf{u}_j(t') = \frac{1}{M_j(t')} \sum_{i=1}^{N_j(t')} \mathbf{v}_i(t) m_i, \quad \text{with} \quad M_j(t') = \sum_{i=1}^{N_j(t')} m_i.
\]
We sum over all colloidal and fluid particles in the cell to obtain their total number $N_j(t')$. $m_i$ is the mass of particle $i$ and $M_j(t')$ gives the total mass contained in cell $j$ at time $t' = t + \tau$. We apply shear by explicitly setting the mean velocity $\mathbf{u}_j$ to the shear velocity in the cells close to the border of the system. A thermostat removes the energy introduced to the system by the shear force.

A single simulation run as presented in our previous papers took between one and seven days on a 3GHz Pentium CPU. However, for strongly clustering systems we easily end up with only a single cluster inside the simulation volume. In order to be able to gather statistics on cluster growth and formation, as well as to minimize finite size effects, we parallelized our code. While MD codes have been parallelized by many groups, only few parallel implementations of a coupled MD and SRD program exist. This is in contrast to the number of parallel implementations of other mesoscopic simulation methods like for example the lattice Boltzmann method. A possible explanation is that SRD is a more recent and so far not as widely used algorithm causing the parallelization to be a more challenging task. In order to push the development in this field we provide some details of our implementation in this section.

We utilize the Message Passing Interface (MPI) to create a C++ code based on domain decomposition for both involved simulation methods. In the MD code the position of neighbouring particles is needed to compute the interactions. Since the interactions have a limited range, and a linked cell algorithm is already used in the serial code, we apply linked cells here as well. Particle positions at the border of the domain of each processor are communicated to the neighbouring processors for the calculation of the forces. Then, the propagation step is performed and particle positions are updated, whereby the particles crossing a domain boundary are transferred from one processor to the other one.

Since (in principle), fluid particles can travel arbitrary large distances in one time step, one either has to limit the distance they can move, or one needs all-to-all communication between the processors. Even though the mean free path in our systems is small enough to limit communication to nearest neighbours only, the current version of our code tries to be as general as possible and allows fluid particles to move to any possible position in the total simulation volume within a single timestep. First, we determine locally which fluid particles have to be sent to which destination CPU and collect all particles to be sent to the same destination into a single MPI message. If no particles are to be sent, a zero dummy message is transmitted. On the receiving side, MPI_Probe with the MPI_ANY_SOURCE option is utilized to determine the sender’s rank and the number of particles to be accommodated. Now, MPI_Recv can be used to actually receive the message. All processors send and receive in arbitrary order, thus waiting times are kept at a minimum allowing a very efficient communication. The standard MPI all-to-all communication procedure should be less efficient since the size of every message would be given by the size of the largest message. However, we still do find a substantial communication overhead from our benchmark tests of the scalability of the code. Due to this
overhead, we are currently limited to 32-64 CPUs on an IBM p690 cluster. In order to achieve Gallilean invariance, a random shift of the SRD lattice is performed for every rotation step\textsuperscript{26,27}. Since the domains managed by each CPU do not move, this would include the borders between the processors to cross SRD cells, which is undesirable. Therefore, we keep the position of the lattice fixed and shift the fluid particle positions before sorting them into the cells instead. After the rotation step they are shifted back.

3. Results

We study the formation of clusters for systems containing a volume concentration of 5\% of colloidal particles (=1320 MD particles), a shear rate of $\dot{\gamma} = 20$/s, ionic strengths $I = 3\text{mmol/l}$ and $7\text{mmol/l}$, and pH = 6 and 7. To demonstrate the effect of clustering, in Fig.1 snapshots from a typical simulation of a $8.88\mu m^3$ system with periodic boundaries at $I = 7\text{mmol/l}$ and pH = 6 at different times are shown. While at the beginning of the simulation (a), freely moving particles can be observed, small clusters appear after $t = 0.26s$ (b). After $t = 1.06s$, all particles are contained within three individual clusters (c) and after $t = 4.22s$ only a single cluster is left in the system. For an investigation of the formation and movement of clusters, substantially larger systems are needed. Therefore, we scale up the simulation volume to $17.76\mu m^3$ containing 10560 MD particles and $1.3 \times 10^7$ fluid particles. Due to the computational demands of the fluid solver, a single simulation of 5s real time requires about 5000 CPU hours on 32 CPUs of an IBM p690 system.

We developed a cluster detection algorithm which not only examines a certain configuration at a fixed time, but also takes account of the time evolution of clusters. This algorithm works as follows: a cutoff radius is introduced, below which two particles are considered to be connected. If they are separated further, they are considered as being not directly connected. However, they might both be connected to a third particle. Therefore we have to check all particle pairs for possible connections. If there are no further connections, the particles are considered not to belong to any cluster. Otherwise four cases have to be distinguished: If both particles are not part of any cluster, a new cluster is created and both particles are assigned to it (1). If one particle is already part of a cluster, the other one is assigned to the same cluster (2). If both particles belong to different clusters, the clusters are united, i.e., all particles of the smaller cluster are assigned to the larger one (3). If it is found that for a particle pair to be checked later on, both particles already belong to the same cluster, nothing has to be done (4). This pairwise checking is optimized by a linked cell algorithm, so that only particle pairs of the same and of neighbouring cells are checked. Additionally, clusters need to be tracked in time, i.e., the clusters found within a time step have to be identified with the clusters of the previous time step. This is done by assigning an identification number (“cluster ID”) for every cluster. Since every particle has a unique identification number, assigning the ID of the cluster it belongs to solves the problem. According to which cluster ID the
particles were assigned in the previous time step, the ID is assigned to the new cluster. Again four different cases have to be considered: if both particles belonged to the same cluster, and therefore refer to the same cluster ID, this ID is assigned to the new cluster (1). If one of the particles did not belong to any cluster in the previous time step, a new cluster has formed during the last time step and has to be provided with a new ID (2). If only one particle was part of a cluster, the ID it provides is preserved (3). If the particles are assigned to different cluster IDs one of those IDs has to be chosen for the new cluster. We decide for the one referring to the larger cluster of the previous time step or choose randomly if both clusters are of identical size (4). Finally one has to check if the cluster IDs are unique. If several clusters are assigned to the same ID the largest one keeps the ID and the smaller ones are assigned to new ones.

The strength of our algorithm is the possibility to track individual particles and their assignment to different clusters in time. Clusters grow and break into pieces and we can follow the trajectory of each particle in this scenario. This enables us to draw cluster assignment trees like the one in Fig. 2. In contrast to conventional algorithms, where clusters cannot be tracked in time, the clusters are sorted here on

Figure 1. Snapshots of a simulation of an 8.88\,\mu m^3 system, filled with \( \Phi = 5\% \) MD particles of diameter \( d = 0.37\,\mu m \) under shear with \( \dot{\gamma} = 20\,s\).
the $x$ axis and keep their position. The lines are obtained by plotting the assignment of the particles to the clusters and their distance depicts the cluster sizes, i.e., if at time $t$ a cluster contains a fraction $p$ of all particles in the simulation, a fraction $p$ is reserved for this cluster on the $x$-axis and the line is plotted at the center of this region. Consequently, if only a single cluster is left in the system, the corresponding line is drawn at $x = 0.5$. Depending on the inter particle forces, different structures can be identified, meaning different scenarios like breaking up of large clusters or unification of smaller ones. We are planning to study systematically the dependence of the structures seen in such cluster tree plots on the inter particle forces determined by the pH-value and the ionic strength $I$ in a future work.

In Fig. 3 we present the time dependence of the mean cluster size (a) and of the number of clusters in the system (b). We find that both observables can be fitted by a power law of the form $A \cdot (t + B)^C$, where $A, B, C$ are fitting parameters. The lines in the figure correspond to the fit and the symbols to the simulation data. The parameters $A, B, C$ to fit the simulation data shown in Fig. 3 are listed in table 3. It would be of great interest to investigate if a general scaling behavior can be observed depending on the volume concentration, the ionic strength and the pH value. However, for this a detailed investigation of the parameter space would be needed which will be the focus of a future work.

4. Conclusion

In this paper we have demonstrated an efficient way to parallelize a combined SRD and MD code and presented our new cluster detection algorithm that is able to not only detect clusters, but also to track their positions in time. We applied this algorithm to data obtained from large scale simulations of colloidal suspensions in the clustering regime and find that the time dependence of the mean cluster size
Figure 3. The time dependence of the mean cluster size is plotted for different simulation parameters (a). Fig. b) depicts the time dependence of the number of clusters found in the system. Each curve is shifted vertically by a factor of two for better visibility. While the symbols correspond to simulation data, lines are given by a power law fit.

<table>
<thead>
<tr>
<th>conditions</th>
<th>number of clusters</th>
<th>mean cluster size</th>
</tr>
</thead>
<tbody>
<tr>
<td>pH</td>
<td>mmol/l</td>
<td>A</td>
</tr>
<tr>
<td>6</td>
<td>3</td>
<td>1.5 · 10^6</td>
</tr>
<tr>
<td>7</td>
<td>3</td>
<td>5.27 · 10^6</td>
</tr>
<tr>
<td>6</td>
<td>7</td>
<td>1.46 · 10^8</td>
</tr>
<tr>
<td>7</td>
<td>7</td>
<td>1.01 · 10^8</td>
</tr>
</tbody>
</table>

Table 1. Parameters for the fit of the simulation data

and the number of clusters in the system can be well described by power laws.

Acknowledgments

This work has been financed by the German Research Foundation (DFG) within the project DFG-FOR 371 “Peloide”. We thank G. Gudehus, G. Huber, M. Külzer, L. Harnau, M. Bier, J. Reinshagen, S. Richter, and A. Coniglio for valuable collaboration. We also thank the German-Israeli Foundation (GIF) for support. The computations were performed on the IBM p690 cluster at the Forschungszentrum Jülich, Germany and at HLRS, Stuttgart, Germany.

Bibliography


10 M. Hecht, J. Harting, and H.J. Herrmann

Computational steering of cluster formation in Brownian suspensions

M. Hecht and J. Harting
Computational steering of cluster formation in Brownian suspensions

Martin Hecht, Jens Harting *

Institut für Computerphysik, Pfaffenwaldring 27, 70569 Stuttgart, Germany

A R T I C L E   I N F O

Keywords:
Colloid
Suspension
Simulation
Steering

A B S T R A C T

We simulate cluster formation of model colloidal particles interacting via DLVO (Derjaguin, Landau, Vervey, Overbeek) potentials. The interaction potentials can be related to experimental conditions, defined by the pH-value, the salt concentration and the volume fraction of solid particles suspended in water. The system shows different structural properties for different conditions, including cluster formation, a glass-like repulsive structure, or a liquid suspension. Typically, many simulations are required to explore the parameter space. In order to reduce the computational effort and data storage requirements, we developed a steering approach to control a running simulation and to detect interesting transitions from one region in the configuration space to another. The advantages of the steering approach and the restrictions of its applicability due to physical constraints are illustrated by several example cases.

© 2009 Elsevier Ltd. All rights reserved.

1. Introduction

Soft matter physics is a large field which has gained more and more importance during the last years. It comprises for example complex fluids, biological systems like membranes, solutions of large molecules like proteins, or suspensions of small soft or solid particles, which are commonly called “colloids”. Since one can find examples for these materials nearly everywhere in everyday life (in medicine, food industry, paintings, glue, blood, ceramics…), scientific research can be very relevant for practical applications. In this context, it is fair to say that colloid science is presently one of the most active fields of research. Considerable effort has been invested to describe colloidal suspensions from a theoretical point of view and by simulations [1–5], as well as to understand the particle–particle interactions [6–11] and the phase behavior [12–16]. One of the main characteristics of soft matter and especially of colloidal systems, is the key role of a so-called mesoscopic scale, which is a length and time scale much larger than the atomistic scale, but still much smaller than the macroscopic scale. Material properties on the macroscopic scale, e.g., the viscosity of a suspension, are determined by effects on the mesoscopic scale.

This mesoscopic level allows to control the parameters of a certain material by manipulating the processes on the mesoscopic length scale. For colloids there are many ways to control the interactions between the individual colloidal particles. For example, it is possible to initiate clustering in a colloidal suspension by changing the particles’ interactions [17–23].

However, on the mesoscopic length scale, often many different effects are in a subtle interplay making it difficult to provide quantitative predictions. Computer simulations can help to study these systems in detail, e.g., to observe the response of the system on a change in the particles’ interactions. Such a change can be achieved by adding salt to a suspension or by tuning the pH-value [18,19,22,24]. However, in mesoscopic systems where many different effects contribute to the overall behavior, the parameter space is quite large and one has to perform a large number of different simulations, each of which requiring a lot of computing resources, to gain an understanding of all inter-relationships.

* Corresponding author.
E-mail address: jens@icp.uni-stuttgart.de (J. Harting).

© 2009 Elsevier Ltd. All rights reserved.
doi:10.1016/j.camwa.2009.02.018
Therefore, it is useful to be able to steer the simulation on-line. Steering in this context can mean changing the interaction potentials or other parameters like an externally applied shear rate. It can also mean going back in the simulation time and following a different path starting from an earlier configuration, both options induced by interaction with the user [25,26]. For our study we have modeled an aqueous suspension of Al2O3-particles. The particles are monodisperse spheres of 0.37 μ m in diameter. They interact via DLVO (Derjaguin, Landau, Vervey, Overbeek) potentials [6,7] as well as a repulsive force ensuring excluded volume for the particles. The properties of this model system have been investigated by simulations and experiments in previous works [27,24,22,28–30]. In this paper we discuss how simulation steering can help to explore the parameter space and which problems may occur when using steering techniques.

In the following section we describe our simulation method and focus on the implementation of the steering. Then, we discuss some examples of simulations where steering helps to explore the parameter space, but we also give examples in which a steered simulation might lead to different results if compared to a non-steered simulation. Finally, we summarize our results and draw a conclusion.

2. Simulation Method

Our simulation method is described in detail in Refs. [27,24,28] and consists of two parts: a Molecular Dynamics (MD) code, which treats the colloidal particles, and a Stochastic Rotation Dynamics (SRD) simulation for the fluid solvent. In the MD part we include effective electrostatic interactions and van der Waals attraction, known as DLVO potentials [6,7]. The repulsive term results from the surface charge of the suspended particles

\[ V_{\text{cold}} = \pi \varepsilon r_0 \left[ \frac{2 + k \varepsilon}{1 + k \varepsilon} \cdot \frac{4 k_b T}{2 \varepsilon} \tanh \left( \frac{2 \varepsilon}{4 k_b T} \right) \right]^2 \cdot \frac{d^2 e^{-|r-d|}}{r}, \]

where \( d \) denotes the particle diameter, \( r \) the distance between the particle centers, \( e \) the elementary charge, \( T \) the temperature, \( k_b \) the Boltzmann constant, and \( z \) is the valency of the ions of added salt. \( r_0 \) is the permittivity of the vacuum, \( \varepsilon_r = 81 \) the relative dielectric constant of the solvent, \( k \) the inverse Debye length defined by \( k^2 = 8 \pi \varepsilon_r k_b T \), with \( l \) being the ionic strength and the Bjerrum length \( \ell_B = 7 \) Å. We have related the effective surface potential \( \varepsilon \) to the \( \phi \)-value and the ionic strength of the solvent by means of a charge regulation model in our previous work [24]. Thus, the particle interaction potentials can be related to distinct experimental conditions. The second term of the DLVO potentials which does not depend on the \( \phi \)-value or the ionic strength is the attractive van der Waals interaction

\[ V_{\text{vdW}} = -A_h \left[ \frac{d^2}{r^2 - d^2} + \frac{d^2}{r^2} + \frac{2}{2} \ln \left( \frac{r^2 - d^2}{r^2} \right) \right], \]

\( A_h = 4.76 \cdot 10^{-22} \) J is the Hamaker constant [17]. The attractive contribution \( V_{\text{vdW}} \) competes with the repulsive term and is responsible for the cluster formation as one can observe for conditions in which the attraction dominates.

Since DLVO theory is based on the assumption of large particle separations, it does not correctly reproduce the primary minimum in the potential, which should appear at particle contact. Therefore, we cut off the DLVO potentials and model the minimum by a parabola as described in Refs. [27,28]. To ensure excluded volume of the particles we use a repulsive (Hertzian) potential. Below the resolution of the SRD algorithm short-range hydrodynamics is corrected by a lubrication force within the MD framework as explained in Refs. [27,24,28].

For the simulation of a fluid solvent, many different simulation methods have been proposed: direct Navier–Stokes solvers [31–34], Stokesian Dynamics (SD) [35,36,3], Accelerated Stokesian Dynamics (ASD) [37,38], pair drag simulations [4], Brownian Dynamics (BD) [17,39], Lattice Boltzmann (LB) [1,12,40,41], and Stochastic Rotation Dynamics (SRD) [42,43,27]. These mesoscopic fluid simulation methods have in common that they impose certain approximations to reduce the computational effort. Some of them include thermal noise intrinsically, or it can be included consistently. They scale differently with the number of embedded particles, and the complexity of the algorithm differs largely. In particular, there are big differences in the concepts how to couple the suspended particles to the surrounding fluid.

We apply the Stochastic Rotation Dynamics method (SRD) introduced by Malevanets and Kapral [44,45]. It intrinsically contains fluctuations, is easy to implement, and has been shown to be well suitable for simulations of colloidal and polymer suspensions [42,43,46–48,27,24]. The method is also known as “Real-coded Lattice Gas” [42] or as “Multi-Particle-Collision Dynamics” (MPCD) [49,50]. It is based on the so-called fluid particles with continuous positions and velocities. A streaming step and an interaction step are performed alternately. In the streaming step, each particle \( i \) is moved according to

\[ \mathbf{r}_i(t + \tau) = \mathbf{r}_i(t) + \tau \mathbf{v}_i(t), \]

where \( \mathbf{r}_i(t) \) denotes the position of particle \( i \) at time \( t \), and \( \tau \) is the time step. In the interaction step, the fluid particles are sorted into cubic cells of a regular lattice, and only the particles within the same cell interact with each other according to an artificial interaction. The interaction step is designed to exchange momentum among the particles, but at the same time to conserve total energy and total momentum within each cell, and to be very simple, i.e., computationally cheap. Each cell \( j \) is treated independently: first, the mean velocity \( \mathbf{u}_j(t') = \frac{1}{N_j(t')^2} \sum_{i=1}^{N_j(t')} \mathbf{v}_i(t) \) in cell \( j \) is calculated. \( N_j(t') \) is the number of...
fluid particles contained in cell $j$ at time $t' = t + \tau$. Then, the velocities of each fluid particle in this cell are rotated according to

$$v(t + \tau) = u_j(t') + \Omega_j(t') \cdot (v(t) - u_j(t')).$$

(4)

$\Omega_j(t')$ is a rotation matrix, which is independently chosen at random for each time step and each cell. We use rotations about one of the coordinate axes by an angle $\pm \alpha$, with $\alpha$ fixed. The coordinate axis as well as the sign of the rotation are chosen at random, resulting in 6 possible rotation matrices. However, there exists great freedom to choose the rotation matrices. Any set of rotation matrices satisfying the detailed balance for the space of velocity vectors could be used here. To remove anomalies introduced by the regular grid, one can either choose the mean free path to be sufficiently large or shift the whole grid by a random vector once per SRD time step [51,52].

For the coupling of the SRD and the MD simulation, three main methods have been introduced in the literature. Inoue et al. proposed a way to implement no slip boundary conditions on the particle surface [42]. To achieve full slip boundary conditions, Lennard-Jones potentials can be applied for the interaction between the fluid particles and the colloidal particles [45,53]. A more coarse grained method was originally designed to couple the monomers of a polymer chain to the fluid [54,55], but in our previous work [27,24,22] we have demonstrated that it can also be applied to colloids, as long as no detailed spatial resolution of the hydrodynamics is required. We use this coupling method in our simulations and describe it shortly in the following.

To couple the colloidal particles to the fluid, they are sorted into the SRD cells and included in the SRD interaction step. The stochastic rotation is performed in momentum space instead of the velocity space to take into account the difference of inertia between light fluid and heavy colloidal particles. We have described the simulation method in more detail in Refs. [27,24,28].

In the present work we report several studies of pressure filtration, cluster formation in a steered simulation, and sedimentation. All these studies are proofs of principle, and therefore small simulations are performed. Typical parameters for these simulations are as follows: on average there are 60 fluid particles in each cubic box of 0.296 $\mu$m extension in each Cartesian direction. All these volumes of the box is set to be identical to the volume of a colloidal particle with a diameter of 0.37 $\mu$m. The system size is 30 to 60 boxes in each direction. By adjusting the diffusion constant, the density and the viscosity to a certain suspension [24], the scaling scheme we have presented in Ref. [27] yields a scaled temperature of 14 mK and a scaled viscosity of 4.78 - $10^{-8}$ Pa s. To preserve the correct dynamics, characterized by the dimensionless numbers (Re, Pe, Kn,...) one has to rescale the potentials and all driving forces in the MD scheme by the same scaling factor, as well.

Let us now shortly sketch the technical realization of the steering interface in our simulation code. The program is an object oriented code written in C++. Each object contains virtual routines save() and load(), which write the data of the object to a buffer, or load it from there, respectively. The buffer contains a plain text description of all variables contained in the object including their values. This is similar to the C++ source code one would write to initialize an instance of that class. The work flow of a simulation, including the actual MD loop as well as data input/output tasks is described in this manner using specialized “workstep-classes”.

Each workstep-class provides a specialized work() -routine, which performs different actions depending on the actual class type of the respective object. One of the workstep-classes is designed to change a specified object by using its save() and load() routine. First, the current values are stored to a temporary buffer, then one or several variables may be overwritten by new values, and finally the object to be modified is loaded again from the temporary buffer. The description of the changes may be read from standard input or from a file, similar to the simulation setup, which is read during the initialization of the simulation. workstep-class objects can also be included into the work flow at later times, or may even be disconnected from the usual work flow and bound to standard UNIX system signals. By default a workstep writing particle positions to standard output and a workstep to change objects getting its buffer from standard input are bound to the SIGTTOU("terminal output") and SIGTIN("terminal input") signal, respectively. This allows to embed the simulation program into a framework of shell scripts which generate the appropriate input, redirect the output to a visualization tool and send the signals according to the user interaction. This can be realized in a client–server fashion, even on different hosts and platforms using appropriate scripts and TCP/IP connections.

3. Results

We now turn to a discussion of the advantages of such a steering approach for computer simulations. We highlight some pitfalls resulting from the physical background in the context of steering. This section is subdivided into several subsections each of them focusing on a particular example to illustrate the steering approach in practice.

3.1. Pressure filtration

In this example a filtration of a suspension is simulated. The suspended particles cannot pass the filter, whereas the fluid passes through the filter without resistance in an idealized filtration process. The suspended particles agglomerate in front

---

1 The workstep concept was originally introduced by M. Strauß in his SRD code [56].
Fig. 1. Local density expressed in terms of volume fraction of the solid particles constituting the filter cake depending on the pH-value. The ionic strength is kept constant at \( I = 3 \) mmol/l and the volume fraction at \( \Phi = 5\% \). The plots for different pH-values are shifted against each other vertically by 0.2 for better visibility. The shape of the profile differs since the spacing of the particles depends on their interactions.

of the filter and form a so-called filter cake. Since the dynamics of the particles is not only governed by the hydrodynamics of the fluid, but also by their DLVO interaction, the density and the structure of the filter cake depend on the pH-value and the ionic strength.

In our simulation we drive the fluid by an applied force pointing downwards acting in a small region close to the upper boundary of the system. For the fluid we apply fully periodic boundary conditions, whereas the boundaries for the suspended particles are closed in z-direction. In this way, the fluid is forced to stream in vertical direction and to drag the particles to the bottom of the system. We choose a constant ionic strength of \( I = 3 \) mmol/l and a constant volume fraction of \( \Phi = 5\% \), and simulate the filtration process for several pH-values.

Depending on the interactions, different internal structures of the filter cake are formed when the density of the particles increases at the bottom of the system. To gain good statistics for the structure, large simulations are needed. However, in a filtration process, usually low initial densities are used, so that very large simulation volumes are needed to obtain large particle numbers for good statistics. On the other hand, large simulation volumes also imply large data files and long transient times until the filter cake is formed and a considerable part of the particles has reached the bottom of the system.

This is a typical problem, in which one would like to observe the running simulation occasionally to check if the filter cake is already formed and when the structure does not change significantly anymore. One would like to measure pressure profiles, local streaming velocities, or simply the final density profile of the filter cake. Here, steering means to initiate data acquisition by user interaction. This is of advantage if an action, like starting to average certain data, should be initiated when conditions are fulfilled, which are difficult to check automatically. Since the density and structure of the filter cake are a priori unknown, an automatic check is difficult to implement.

Additionally, the upper boundary of the filter cake is diffusive and depends on the conditions (pH-value and ionic strength) as well. In Fig. 1 the dependence of the density profiles of filter cakes obtained in simulations for different pH-values are shown. With increasing pH-value the height of the filter cake increases and the density profile changes slightly. Some voids in the filter cake diminish the local density if compared to a dense packing. At the bottom of the filter cake, layers of particles are present, whereas in the upper regions the structure becomes more irregular. However, the structure of the sediment and its height will also depend on the pressure exerted on the fluid and probably on the initial volume fraction. Larger simulations than this proof of principle are needed to be able to quantify the pH-value dependence of the height of the filter cake. However, simulations without a driving force on the fluid can also help to understand this system. Therefore, we explore the parameter space in the absence of a driving force in the following subsection.

3.2. Observation of cluster formation

Since several parameters, namely the volume fraction, pH-value, ionic strength, and external driving forces influence the system, a first step to understand its behavior is to examine it in the absence of external forces. Still, two parameters—the pH-value and the ionic strength—govern the dynamics of the system. If the potentials are attractive, cluster formation can be observed [57], provided the attraction is strong enough compared to thermal fluctuations. In our previous work [22,28] we have explored this part of the parameter space for some distinct volume fractions by performing numerous individual simulations. However, with the newly implemented steering approach one can detect boundaries between different regions of the stability diagram more quickly and more sensitively.
Fig. 2. Memory effects in a steered simulation: the grayscale denotes the coordination number. Particles closer than 2.4 radii are considered to be neighbors. Dark particles in this sense are highly coordinated, bright ones have low coordination numbers. (a) Right after initialization, homogeneous distribution of the particles, (b) cluster formation after tuning the potentials to be attractive, (c) still inhomogeneous regions after steering the simulation out of the clustered region, (d) the system is homogeneous again when diffusion has counterbalanced the density inhomogeneities. The steering path in the parameter space as well as the points (a)–(d) thereon are illustrated in Fig. 3.

In Fig. 2 we illustrate the effects of changing the properties of the interaction potentials. All snapshots correspond to the same simulation and represent different points of the trajectory in parameter space the system is steered through: (a) The initial situation with repulsive potentials, shortly after initialization of the system, (b) after crossing the boundary to the clustered region, (c) shortly after steering back to the suspended regime, where still some inhomogeneities in the system can be observed, and, finally, (d) after some time when diffusion has restored a homogeneous system. This time depends on the distance between the clusters and on the characteristic diffusion time of the particles. The steering path in the configuration space is shown in Fig. 3. The points at which the snapshots of Fig. 2 are taken are marked by black dots. In the insets of Fig. 3 the total potential is plotted for several cases. During the simulation we change the pH-value and the ionic strength, and we drive the system along a closed trajectory in parameter space. We start at low ionic strength \( I = 1 \text{ mmol/l} \) and \( \text{pH} = 4 \) (potential shown) and first increase the pH-value (potential shown for \( I = 1 \text{ mmol/l} \) and \( \text{pH} = 7 \)), then increase the ionic strength, then reduce the pH-value again while further increasing the ionic strength (potential shown for \( I = 15 \text{ mmol/l} \) and \( \text{pH} = 4 \)), and finally decrease the ionic strength to return to the starting point. We choose the trajectory such that the barrier between the primary and the more shallow secondary minimum of the DLVO potential does not go below 5 \( k_B T \). Since the secondary minimum becomes deeper during the simulation, cluster formation can be observed. However, the barrier between the primary and the secondary minimum of the DLVO potentials prevents irreversible aggregation in the primary minimum. A case in which the barrier vanishes and irreversible aggregation would appear is also shown in the upper right inset (\( I = 15 \text{ mmol/l} \) and \( \text{pH} = 7 \)). In any case, in our simulation all clusters dissolve again after steering back to a lower pH-value and lower ionic strength (see Fig. 2 (d)). This confirms that we have reversible clustering in the secondary minimum of the DLVO potentials.

One of the advantages of a steered simulation is that one can detect the onset of the cluster formation more accurately than by starting individual simulations for different conditions. One might predict the boundary by evaluating the depth of the secondary minimum in the potential, but since also the hydrodynamic damping forces and an eventually applied driving pressure influence this process the prediction of the boundary can become more complicated. We have experienced similar difficulties in the context of shear flow simulations of the same colloidal suspension [24,22].

A quantity to support the visualization of clusters is the coordination number. We consider two particles whose centers are closer than 2.4 particle radii as neighbors and define the coordination number of each particle as the number of its neighbors.
neighbors in the sense just described. The particles in Fig. 2 are drawn in grayscale corresponding to their coordination number, where dark particles are highly coordinated particles and bright ones are those with a low coordination number. At the beginning of the simulation the particles are distributed homogeneously in simulation volume (a). After steering the simulation into the region of attractive potentials, cluster formation can be observed (b). When the potentials are made less attractive again, the clusters dissolve (c). However, the particles preferably stay at their positions and thus, the inhomogeneities in the system do not disappear immediately, which can also be seen in the dark color denoting high coordination numbers. After some time, diffusion restores the homogeneity of the system (d). However, if the pH-value or the ionic strength are increased too much, so that the barrier between the primary and the secondary minimum of the DLVO potentials vanishes, the clustering process becomes irreversible. We avoided this by choosing an appropriate steering path (Fig. 3).

In this example, the simulation “remembers” the interaction of the user. Even if the steering path is selected carefully inside the reversible range, the simulated suspension needs a characteristic time to relax after changing the interactions. The interaction by the user can be seen as a perturbation in the physical sense and the system needs time to adopt to the new situation. In these cases special care is advised when steering a simulation. We illustrate this by another example in the following subsection.

3.3. Sedimentation: Hydrodynamic interaction

Not only the particle positions and velocities, but also hydrodynamic interactions can influence the behavior of the system. Therefore, inhomogeneous particle distributions induced by first steering the simulation into a clustered region of the parameter space might influence the result. In the following paragraph we show that a sedimentation process can be accelerated, if cluster formation by attractive interactions creates density inhomogeneities in the system.

The friction force a single particle feels in a sedimentation process can be calculated analytically, but it differs if other particles are present. In Ref. [27] we have confirmed that the sedimentation velocity in our simulation depends on the volume fraction, as it is well known from sedimentation theory [58]. But even at constant volume fraction the sedimentation velocity can be different in different configurations. If the particles form clusters, it settles down and the fluid streams around the whole cluster. The resistance is much less compared to the case when the particles are distributed homogeneously and the fluid streams around each of them separately (compare Ref. [28] and references therein). In Fig. 4 we have plotted the sedimentation velocity evaluated in several simulations, which only differ by the potentials. The volume fraction is the same for all of them. However, as one can see in the figure, the sedimentation velocity is smaller for low pH-values. Around pH = 4 the sedimentation velocity of an isolated particle of 0.6 diameters/s is reached [28].

This can be explained as follows: since for increasing pH-value the potentials become attractive, the particles form clusters and settle down faster. This effect is purely due to the hydrodynamics of the system. Since the flow field depends on all particle positions and velocities not only at a given time, but also on their history, steering in this context may be very dangerous. When long-range interactions (like hydrodynamics) which also depend on the history of the system, are important, steering should be avoided. In this case, to obtain reasonable results, one has to start again from an initial configuration when changing the interactions.
4. Conclusion

By bearing in mind the underlying physics, one can roughly estimate the limits of steered simulations. Let us summarize the points we have brought forward in this paper. First of all, steering can save computing time when aiming at a rough understanding of the influences on a system, especially when looking for the “interesting points” in parameter space, e.g., when searching for transition lines between different phases. We have shown this by the onset of cluster formation due to modifying the interaction potentials in a model suspension. Steering can be used as well to start data acquisition after a transient at the beginning of a simulation or to adjust the frequency for the data acquisition according to the current state of the simulation. This is especially interesting for large simulations with long transient times, as illustrated by the simulation of filter flow. We also explained why in some cases steering may be seen as a perturbation of the system, especially when the interaction potentials are changed. Special care is advised when dealing with non-ergodic systems, memory effects, or long-range interactions, like the role of hydrodynamics in sedimentation as in our last example.

Acknowledgments

We thank H.J. Herrmann for valuable collaboration and his support. The High Performance Computing Center Stuttgart, the Scientific Supercomputing Center Karlsruhe and the Neumann Institute for Computing in Jülich are highly acknowledged for providing the computing time and the technical support needed for our research. J.H. thanks the DFG for financial support within the priority program “nano- and microfluidics”. M.H. thanks the ICMMES for travel funds and the DFG for financial support within SFB 716.

References

Colloids dragged through a polymer solution: experiment, theory and simulation

C. Gutsche, F. Kremer, M. Krüger, M. Rauscher, R. Weeber, and J. Harting
Colloids dragged through a polymer solution: Experiment, theory, and simulation

Christof Gutsche,1 Friedrich Kremer,1 Matthias Krüger,2,3 Markus Rauscher,2,3 Rudolf Weeber,3 and Jens Harting3
1Institut für Experimentalphysik I, Universität Leipzig, 04103 Leipzig, Germany
2Max-Planck-Institut für Metallforschung, Heisenbergstr. 3, 70569 Stuttgart, Germany
and Institut für Theoretische und Angewandte Physik, Universität Stuttgart, Pfaffenwaldring 57, 70569 Stuttgart, Germany
3Institut für Computeralphysik, Universität Stuttgart, Pfaffenwaldring 27, 70569 Stuttgart, Germany
(Received 9 November 2007; accepted 3 July 2008; published online 22 August 2008)

We present microrheological measurements of the drag force on colloids pulled through a solution of λ-DNA (used here as a monodisperse model polymer) with an optical tweezer. The experiments show a drag force that is larger than expected from the Stokes formula and the independently measured viscosity of the DNA solution. We attribute this to the accumulation of DNA in front of the colloid and the reduced DNA density behind the colloid. This hypothesis is corroborated by a simple drift-diffusion model for the DNA molecules, which reproduces the experimental data surprisingly well, as well as by corresponding Brownian dynamics simulations. © 2008 American Institute of Physics. [DOI: 10.1063/1.2965127]

I. INTRODUCTION

Complex fluids in general and colloid-polymer mixtures in particular are an ideal model system for studying the structure and phase behavior of multiparticle systems. However, they also play a large role in many technological processes such as oil recovery and food science, as well as in most biological systems. For these systems, in addition to the equilibrium properties, the dynamics is important. While for many applications the bulk rheological properties are most relevant, in biological systems, where the size of many constituents is on the microscale, a strong size dependence of the rheological properties has been observed and micron sized probe particles have been used to test the rheology (e.g., by measuring the friction coefficient) on the relevant small length scales, see, e.g., Refs. 1–3.

The friction of probes moving through a solution of macromolecules or of colloids and the diffusion constant of such probe particles are related quantities. The first has been mainly discussed in the context of sedimentation.4–11 However, the major differences between sedimentation and nanorheology with colloidal probe particles is that in a sedimenting system all particles experience a force (usually gravity) and therefore move, which creates a solvent backflow responsible for a large part of the reduction in the sedimentation velocity as compared to that of a single isolated particle.

The diffusion constant of suspended particles has been discussed in literature extensively,12–18 and as expected, it is consistently smaller than that in the pure solvent, while the friction constant (as well as the solution’s macroscopic viscosity) is always larger than that in the pure solvent.13,14 If one assumes that the Brownian motion is a Markov process, the diffusion constant is coupled to the friction coefficient (the proportionality constant between force and velocity in the linear regime) via the fluctuation-dissipation theorem. Also the latter is directly related to the viscosity via the solution of the Stokes equation. For a large, heavy, and isolated particle suspended in a simple solvent, this is a reasonable approximation,19,20 but in a suspension of nonvanishing density, the timescale for the dynamics of the surrounding particles and the timescale for the dynamics of the probe particle cannot be separated in general.18 In addition, structural changes in the solution in the vicinity of the probe particle can enhance21,22 or reduce23,24 the friction coefficient. Therefore, one cannot expect a direct relation between the diffusivity of a particle and the macroscopic viscosity of the suspension as it is measured in a viscosimeter. This violation of the Stokes–Einstein relation has been observed experimentally in light scattering experiments25–27 and by x-ray photon correlation spectroscopy.28

In order to elucidate the dynamics of colloids in polymer solutions in more detail, we perform—under well controlled conditions—experiments in which single isolated colloids are held in a stream of λ-DNA solution using optical tweezers. This technique allows us to move a colloid through a highly monodisperse polymer solution at a given velocity (rather than with a given driving force such as in sedimenting systems) and to measure the drag force on the colloid with picosecond resolution at the same time. For high DNA concentrations we find a significantly higher drag force than that predicted by the Stokes equation for the homogeneous solution (which is in apparent contradiction to the observation of an enhanced diffusivity of colloids in similar polymer solutions); however, the force scales approximately linearly with the velocity up to high Peclet numbers. We compare the experimental results to a simplified dynamic density func-

314 V-8. Colloids dragged through a polymer solution

tional theory for the noninteracting polymers in a flowing solvent and to Brownian dynamics (BD) simulations.

II. EXPERIMENT
A. Materials and methods

Figure 1 illustrates the experimental setup, in which one colloid is held by an optical trap surrounded by a polymer solution of λ-DNA. The DNA stock solution (10 mM tris-HCl, 1 mM ethylenediaminetetraacetic acid (EDTA), 500 µg/ml λ-DNA) was obtained by New England BioLabs (Germany) and diluted to its final concentrations in deionized water. An inverted microscope (Axiovert S 100 TV, Carl Zeiss, Jena, Germany) is used to image the colloid and the optical trap is realized with a diode pumped neodymium doped yttrium aluminum garnet laser (1064 nm, 1 W, LCS-DTL 322; Laser 2000, Wessling, Germany). Its power is stabilized to achieve long-term stability. Additionally, the profile of the laser beam is monitored. After passing an optical isolator, a quarter-wave-plate is used in order to produce circularly polarized light to exclude effects due to reflection differences of the mirrors between the p and s polarizations of the laser light. The beam is expanded and coupled into the back aperture of the microscope objective (Plan-Neofluor 100×1.3 oil, Carl Zeiss, Jena, Germany). Video imaging and the optical position detection are accomplished by a digital camera (KPF 120, Hitachi, Düsseldorf, Germany). The optical stage is positioned in three dimensions with nanometer resolution using piezoactuators (P-5173CD, Physik Instrumente, Karlsruhe, Germany). The sample cell consists of a closed chamber that can be flushed by a syringe pump with varying solutions. The viscosities are measured with an Ostwald viscosimeter at the same temperature (25 °C room temperature) and with the same λ-DNA solutions as used in the experiment.

B. Data analysis and calibration

The position of the bead in the optical trap is determined by image analysis [Fig. 2(a)]. For that a sequence of images is recorded (repetition rate of 30 Hz) and analyzed based on the Levenberg–Marquardt algorithm [Fig. 2(b)]. As fit function,

\[ I = I_0 + A (1 - ad \exp(-d^2)), \]

\[ d = \left( \frac{x - x_0}{r} \right)^2 + \left( \frac{y - y_0}{r} \right)^2, \]

is used, where \((x_0, y_0)\) is the center position, \(r\) is the optical radius, \(A\) is the amplitude of the profile relative to the background image intensity \(I_0\), and \(a\) is a constant, which is needed in order to consider the diffuse ring around the colloid. A variation in \(r\), \(A\), or \(a\) in an image sequence indicates the motion in the \(z\) direction. The calibration of the optical trap is based on Stokes’ law for the pure solvent (here water), as described in detail elsewhere. A typical force constant of the trap is 0.085 pN/nm, corresponding to forces

![FIG. 1. (Color online) Illustration of the experimental setup. The colloid (green) surrounded by coils of DNA (blue) is held in an optical trap (red). Due to short ranged repulsion, the DNA’s center of mass can approach the colloid only up to the green dashed circle.](image)

![FIG. 2. (a) Microscope image of a single colloid (1.12 µm radius) in the optical trap; (b) fit of the intensity distribution for the image shown in (a) using the Levenberg–Marquardt algorithm.](image)

![FIG. 3. Drag force \(F\) on colloids of radii \(R_c=1.5 \mu m\) (boxes), 1.12 \(\mu m\) (circles), and 0.66 \(\mu m\) (diamonds) in \(\rho_{w} = 30 \mu g/ml\) (solid symbols) and 20 \(\mu g/ml\) (half-filled symbols), as well as in pure water (open symbols), as a function of the pulling speed \(a\). Also shown are linear fits to the data from which one can extract a density dependent viscosity via the Stokes formula Eq. (3). However, as shown in Fig. 4, these viscosities are inconsistent as they depend on the colloid radius.](image)
in the range between 0 and 50 pN, which can be determined with an accuracy of ±0.15 pN.

In the experiments, no sign of irreversible adsorption of λ-DNA molecules to the colloid is detected. While switching between different flow velocities, the same forces at the same speed are obtained in the range of our uncertainties and after flushing back to pure water, the Stokes force on the colloid could be reproduced.

C. Experimental results

Figure 3 shows the measured force on colloids of sizes $R_C=1.5$, 1.12, and 0.66 $\mu$m as a function of the dragging velocity $u$ in pure water as well as for DNA concentrations $\rho_{DNA}=20$ $\mu$g/ml and $\rho_{DNA}=30$ $\mu$g/ml. Linear fits to the data analyzed in terms of the Stokes drag formula for a colloid of diameter $R_C$ in a DNA solution of viscosity $\eta$.

$$F = 6\pi\eta R_C u,$$ (3)

yield the DNA concentration dependent viscosities in Table I (shown in Fig. 4). The viscosities obtained in this manner are significantly larger than the viscosities measured in a viscometer for the same DNA solution and they increase with the colloid radius $R_C$. In other words, the drag forces measured are much larger than the Stokes drag forces computed from the independently measured viscosities. This is a strong indication that on top of the Stokes drag, a second mechanism plays a role.

With the viscosities measured in the viscometer we can also extract an effective hydrodynamic radius $R_{eff}$ from a fit of Eq. (3) to the data in Fig. 3. The resulting $R_{eff}$, as a function of $\rho_{DNA}$, are shown in Table II. $R_{eff}/R_C$ increases with $\rho_{DNA}$ but it also varies with $R_C$: the value for $R_C=1.5$ $\mu$m is much larger than the values for the smaller colloids. If the increase in $R_{eff}$ would be due to adsorption of DNA molecules to the colloid, one expects the absolute increase in $R_{eff}$ to a first approximation to be independent of $R_C$.

The molecular weight of a λ-DNA molecule is $M=31.5 \times 10^6$ amu=$5.23 \times 10^{-11}$ $\mu$g and its contour length is about 16 $\mu$m, leading to a radius of gyration of about $R_g=0.5$ $\mu$m. This corresponds roughly to a hydrodynamic radius of $R_H=0.33$ $\mu$m ($R_H=0.662 R_g$). Based on the hydrodynamic radius and the radius of gyration, we get molecular volumes $V_H=1.5 \times 10^{-13}$ ml and $V_g=5.2 \times 10^{-13}$ ml respectively. For the highest DNA concentrations used in the experiment, i.e., $\rho_{DNA}=50$ $\mu$g/ml, this corresponds to volume packing fractions of 14% and 50%, respectively. With this, we can extract the intrinsic viscosity of the DNA solution from the viscometer data and get $[\eta]=1.55$. Assuming the intrinsic viscosity for hard-sphere suspensions $[\eta]=\frac{3}{5}$, as predicted in Refs. 32 and 33, we get an effective hydrodynamic radius of $R_{eff}=0.28$ $\mu$m.

Figure 5 shows the drag force normalized to the pulling velocity $u$ on a colloid of diameter $R_C=1.12$ $\mu$m measured as a function of the DNA density $\rho_{DNA}$. The data point at $\rho_{DNA}=0$ is normalized to the Stokes drag force with the viscosity of water $\eta_{w}=10^{-3}$ N s/m$^2$. Clearly, the additional drag due to the presence of the DNA in the solution is not linear in $\rho_{DNA}$ and larger than the drag expected from the

![FIG. 4. Viscosity as a function of DNA concentration extracted from linear fits to the data shown in Fig. 3 using the Stokes formula Eq. (3) compared to the viscosities measured in a viscometer (see Table I). From the viscometer data, one obtains an intrinsic viscosity of the DNA of $[\eta]=1.55$.](image)

**TABLE II. Effective colloid radius $R_{eff}$ extracted from the linear fits to the data in Fig. 3.**

<table>
<thead>
<tr>
<th>$\rho_{DNA}$</th>
<th>0 $\mu$g/ml</th>
<th>20 $\mu$g/ml</th>
<th>30 $\mu$g/ml</th>
</tr>
</thead>
<tbody>
<tr>
<td>$R_{eff}$=1.50 $\mu$m</td>
<td>1.50 $\mu$m</td>
<td>1.87 $\mu$m</td>
<td>2.72 $\mu$m</td>
</tr>
<tr>
<td>$R_{eff}$=1.12 $\mu$m</td>
<td>1.12 $\mu$m</td>
<td>1.14 $\mu$m</td>
<td>1.27 $\mu$m</td>
</tr>
<tr>
<td>$R_{eff}$=0.66 $\mu$m</td>
<td>0.66 $\mu$m</td>
<td>0.73 $\mu$m</td>
<td>0.79 $\mu$m</td>
</tr>
</tbody>
</table>

![FIG. 5. The measured drag force $F$ on a colloid of radius 1.12 $\mu$m normalized to the pulling velocity $u$ as a function of the DNA concentration (open symbols). For the clarity of the presentation only a subset of the experimental data is shown. The data collapse demonstrates that $F$ is proportional to $u$; however, the dependence on $\rho_{DNA}$ is nonlinear. The drag is also significantly larger than expected from the increased viscosity as measured in a viscometer (+). In pure water we obtain $F/u=0.021$ pN/$\mu$m/s (dashed horizontal line). Also shown are simulation results for polymers with modified mobility (as explained in the text, solid symbols). A fit for concentrations between 0 and 20 $\mu$g/ml (solid line) highlights the nonlinearity of the force and the linearity of the viscosity in the density.](image)
increased viscosity. The error bars for small velocities in Fig. 5 are large due to the uncertainties in determining the velocity. In Fig. 6 the measured drag force on the colloids is compared to the calculated Stokes drag force \( F_F \) in the solution of DNA [see Eq. (3)]. The forces are to a good approximation linear in the velocity. The difference between the drag forces for 20 \( \mu g/ml \) DNA and for pure water is only significant for velocities larger than 400 \( \mu m/s \). For 40 \( \mu g/ml \) DNA we can observe a significant difference already at velocities larger than 100 \( \mu m/s \). The experimental resolution limits the measurement of the DNA induced drag force to relatively large velocities and to large DNA concentrations. However, Fig. 6(b) shows that the measured drag forces cannot be explained simply by the increased viscosity \( \eta \) DNA of the solution alone.

III. DRIFT-DIFFUSION MODEL

The experimental observation that the drag force on the colloid cannot be explained simply by the increased viscosity in the DNA solution indicates that the dynamics of the DNA in the vicinity of the moving colloid has to be taken into account. The DNA dynamics is governed by the interplay of direct intermolecular interactions, hydrodynamics, and the internal degrees of freedom of the DNA coil. It has been shown that an additional drag force due to the rearrangement of solute particles in the vicinity of a dragged colloid can be already obtained in a simple drift-diffusion (DD) model. Here we employ the same model to calculate this additional drag force \( F_D \) and compare it to the experimental values. Within this model, the DNA molecules are idealized as mutually noninteracting particles having a finite hard core interaction radius with the colloid. The origin of this additional force is illustrated in Fig. 7. Due to the repulsion between the DNA coils and the colloid, the center of mass of the DNA can approach the colloid surface only up to a distance of roughly \( R_g \), i.e., the radius of gyration, which creates a forbidden zone for the DNA. The solvent molecules, however, are much smaller and enter this zone, such that the solvent flow field has a component normal to the surface of the forbidden zone pointing inwards in front of the colloid and outwards behind the colloid. DNA coils advected with the solvent will therefore accumulate in front of the colloid and their density will be reduced in its back. This inhomogeneous DNA distribution will lead to an inhomogeneous osmotic pressure and therefore to a force \( F_p \) on the colloid.

We will quantify the rearrangement of DNA molecules by calculating the average concentration \( C(r) \) (in units of molecules per volume) near the colloid, which in the simple DD model described in Refs. 21 and 22 is given by the stationary solution of the Smoluchowski equation

\[
\nabla \cdot D \nabla C = D \Delta C
\]

in a frame of reference comoving with the colloid, with the solvent velocity field \( \mathbf{v} \) and the DNA (zero concentration) diffusion constant \( D \). In contrast to Ref. 21, \( \mathbf{v}(r) \) is not uniform but it is the solution of Stokes’ equation for a sphere of radius \( R_c \) translating with velocity \( \mathbf{u} \) through a resting solvent \( (\mathbf{v} \to \mathbf{u} \text{ for } |r| \to \infty) \) (see, e.g., Ref. 34 for details),

\[
\mathbf{v}(r) + \mathbf{u} = \frac{3R_c}{4r} \left[ 1 + \frac{R_c^2}{3r^2} \right] \mathbf{u} + \left[ 1 - \frac{R_c^2}{3r^2} \right] \mathbf{f}(r \cdot \mathbf{u})
\]

with the unit vector \( \mathbf{f} = r/|r| \). As we model the interaction potential between the colloid and the DNA as a hard-sphere potential, the centers of the DNA coils are excluded from a
sphere of radius $R_C + R_g$ around the center of the colloid (see Fig. 7). Therefore, the boundary condition for Eq. (4) on this sphere is

$$ \langle \mathbf{e} \cdot \mathbf{j} \rangle_{r=R_C+R_g} = 0, \tag{6} $$

i.e., the DNA current $\mathbf{j} = \nabla r = -D \nabla C$ normal to the sphere’s surface has to vanish. Far from the colloid, the DNA density should be constant, i.e., $C(\mathbf{r}) = C_0$ for $|\mathbf{r}| \to \infty$. As the mutual interaction between the DNA molecules is neglected, the local pressure $p(\mathbf{r})$ on the colloid surface can be calculated from the ideal gas law $p(\mathbf{r}) = k_B T C(\mathbf{r})$. Integrating the local pressure over the surface yields the force $F_P$ on the colloid.

The solution of Eq. (4) depends only on the dimensionless velocity $u^* = u/[R_C + R_g]/D$ (the Peclet number) of the colloid and the size ratio of the involved particles $R_g/R_C = 0$, i.e., if the colloid is much larger than the DNA coils ($R^* \to 1$), the surface of the forbidden zone coincides with the colloid surface and the DNA behaves like a solvent molecule. In this limit the solution of Eq. (4) is $C(\mathbf{r}) = C_0$, the DNA molecules do not accumulate in front of the colloid, and therefore the additional drag force $F_P$ vanishes.

With a hydrodynamic radius of $R_H = 0.33 \mu m$ the Stokes–Einstein relation leads to a diffusion constant $D = 6 \times 10^{-13} \ m^2/s$. Therefore, the smallest velocities in the experiments ($u = 50 \ \mu m/s$) correspond to Peclet numbers $u^*$ larger than 100 (depending on the colloid radius). While for small $u^*$ Eq. (4) can be solved analytically in linear order in the Peclet number, we have to use numerical methods for the experimental velocities. To this end, we expand the density field $C(\mathbf{r})$ in spherical harmonics up to order $N$ and obtain a system of $N+1$ ordinary differential equations for the $|\mathbf{r}|$-dependent expansion coefficients, which we solved numerically with AUTO 2000 (Ref. 35) (for details, see Ref. 22).

For large $u^*$ a fine numerical discretization (large $N$) is needed since the thickness of the region in front of the colloid in which the DNA density is enhanced decreases with $u^*$, while the density in this region increases with $u^*$. We therefore use $N = 100$, which allows us to calculate reliable solutions up to $u^* = 100$. For such high velocities the drag force $F_P$ is well approximated by an affine function (see Fig. 8 and also Ref. 21).

$$ F_P(R^*, u^*) = (R_C + R_g)^2 k_B T C_0 \alpha(R^*) + \beta(R^*) u^*, \tag{7} $$

which we use to extrapolate to velocities larger than $u^* = 100$ (see Fig. 8). The coefficients for fits to the numerical data in the range $80 \leq u^* \leq 100$ for the experimentally relevant values $R_C = 1.50, 1.12, 0.66 \ \mu m$ and $R_g = 0.5 \ \mu m$ are given in Table III. In order to calculate $F_P$ also for other values of $R^*$ and to test the sensitivity of our result to variation in $R_g$, we use the linear interpolation of the coefficients $\alpha(R^*)$ and $\beta(R^*)$.

Table III. Fitting coefficients for the extrapolation of the force $F_P$ to large velocities (Peclet numbers) $u^*$ for the three colloid sizes used in the experiments. The radius of gyration is assumed to be $R_g = 0.5 \ \mu m$.

<table>
<thead>
<tr>
<th>$R_C$ (( \mu m ))</th>
<th>$R^* = R_C/R_g + R_g$</th>
<th>$\alpha(R^*)$</th>
<th>$\beta(R^*)$</th>
</tr>
</thead>
<tbody>
<tr>
<td>0.66</td>
<td>0.57</td>
<td>3.31</td>
<td>0.126</td>
</tr>
<tr>
<td>1.12</td>
<td>0.69</td>
<td>2.43</td>
<td>0.063</td>
</tr>
<tr>
<td>1.50</td>
<td>0.75</td>
<td>1.76</td>
<td>0.036</td>
</tr>
</tbody>
</table>

For the ideal gas law, which we use to calculate the local osmotic pressure on the colloid surface, we need the number density $C_0$ rather than the mass density $\rho_{DNA}$, as used in the experiments. With the molecular weight $M = 5.23 \times 10^{-11} \ \mu g$ we get $C_0 (m^-3) = 1.91 \times 10^{10} \rho_{DNA} (\mu g/ml)$.

Figure 6 compares the forces on a colloid of size $R_C = 1.12 \ \mu m$ predicted by the DD model to the experimental data. In order to demonstrate the sensitivity of $F_P$ on $R_g$, we show $F_P$ for the estimated radius of gyration, $R_g = 0.5 \ \mu m$, as well as for a slightly higher value of $R_g = 0.7 \ \mu m$. At a concentration of $\rho_{DNA} = 20 \ \mu g/ml$ [Fig. 6(b)], within the experimental error, the friction of the colloid in the DNA solution can be explained by the increased viscosity in the solution ($F^*_d$). Nevertheless, theory and experiment are in better correspondence (in particular, for larger velocities) if one takes into account the additional force $F_P$ for a DNA radius of $R_g = 0.5 \ \mu m$. At a DNA concentration of $\rho_{DNA} = 40 \ \mu g/ml$ [Fig. 6(b)], the drag force is much larger and it cannot be explained by the increased viscosity only ($F^*_d$). Taking into account $F_P$ calculated with the DD model still yields a too small drag force (even for $R_g = 0.7 \ \mu m$). As presented in this section, we calculate $F_P$ in linear order in $\rho_{DNA}$, i.e., we do not take into account interactions among the DNA molecules. The experimentally measured drag force as a function of the DNA concentration (see Fig. 5) clearly shows a nonlinearity, indicating that these interactions are relevant for concentrations larger than $\rho_{DNA} = 20 \ \mu g/ml$. Therefore we performed BD simulations as presented in Sec. IV.
IV. BROWNIAN DYNAMICS SIMULATIONS

We simulate the experiments with colloids of radius $R_C=1.12 \, \mu m$ using a modified BD method, which, to some extent, includes the hydrodynamics caused by the dragged colloid, as explained in Ref. 22 and summarized below. The polymers and the colloid are modeled as hard spheres with their respective radii. For the polymers we use the radius of gyration $R_g=0.5 \, \mu m$. As the polymers drag along most of the water contained in their volume, the polymers are assigned the mass of water contained in a sphere with radius $R_g$. The number of polymers in the system is chosen such that the number density (rather than the mass density) of the experiments is reproduced. We use a rectangular simulation volume of $400 \times 40 \times 40 \, \mu m^3$ with periodic boundary conditions in all three directions. For a polymer concentration of $40 \, \mu g/ml$ this corresponds to 244 736 polymers. The large system size is necessary due to the long range hydrodynamic interactions caused by the dragged colloid.

The colloid is trapped in a moving parabolic potential $V(r)=\frac{1}{2}a r^2$, mimicking the optical tweezer. The potential has a spring constant of $a=7.5 \times 10^{-3} \, pN/\mu m$, which gives a better signal to noise ratio than the experimental value of $8.5 \times 10^{-2} \, pN/\mu m$. Figure 9 shows a snapshot of our simulation setup.

In conventional BD, the two most important aspects of hydrodynamics felt by the suspended particles are taken into account, namely, the Stokes friction and the Brownian motion. Correspondingly, this is done by adding to a molecular dynamics simulation two additional forces. The Langevin equation describes the motion a Brownian particle with radius $R$ at position $r(t)$ as

$$m \ddot{r}(t) = 6 \pi \eta r(t) + F_{\text{rad}}(t) + F_{\text{ext}}(r,t), \quad (9)$$

where the first term models the Stokes friction in a solvent of viscosity $\eta$, $F_{\text{rad}}(r,t)$ is the sum of all external forces such as gravity, forces exerted by other suspended particles, and, for the colloid, the optical trap. $F_{\text{ext}}(r,t)$ describes the thermal noise, which gives rise to the Brownian motion. The random force on different particles, as well as the force on the same particle at different times, is assumed to be uncorrelated. It is further assumed to be Gaussian with zero mean. The mean square deviation of the Gaussian (i.e., the amplitude of the correlator) is given by the fluctuation-dissipation theorem as

$$\langle |F_{\text{rad}}|^2 \rangle = 12 \pi \eta R \kappa T. \quad (10)$$

This conventional BD scheme is widely used to simulate suspensions (e.g., Refs. 36–38) because it is well understood, not difficult to implement, and needs much less computational resources than a full simulation of the fluid. However, this simulation method does not resolve hydrodynamic interactions between particles. In particular, the long-ranged hydrodynamic interactions between the dragged colloid and the surrounding polymers are not modeled. However, in the system we consider, these interactions are important as the dragged colloid moves quickly and has a strong influence on the flow field around it. Therefore, the BD scheme is modified such that the effect caused by the flow field around the dragged colloid is included. This is achieved by calculating the friction force on the polymers with respect to a resting fluid ($F=6 \pi \eta R \dot{u}$) but with respect to the flow field caused by the moving colloid, as given by Eq. (5). The friction force then is

$$F = 6 \pi \eta R (\dot{u} - v(r)), \quad (11)$$

where $v(r)$ is the flow field around the moving colloid at a position $r$ with respect to the colloid’s center. This correction leads to the inclusion of two hydrodynamics-mediated effects. Due to the large component of the flow field along the direction of motion both in front and behind the colloid, polymers are dragged along. Also, as shown in Fig. 7, the advective flow advects polymers around the moving colloid, i.e., obstacles are moved out of the way to its sides. Both these effects lead to a reduction in drag force on the driven colloid.

In order to reduce computation time, physical quantities are rescaled: the simulation is carried out at a lower temperature. To compensate for this, the viscosity of the fluid as well as all energies are scaled by the same factor. In the present simulations, we scale down the temperature by a factor of 37 500. This scaling leaves the diffusion constant as well as the relative importance of diffusion and motion caused by external forces unchanged. However, it allows for a much larger numerical time step (in this case 60 $\mu s$).

From the simulation data, it is possible to measure the effective polymer concentration around the dragged colloid. To accomplish this, we take about 2000 snapshots of the simulation and move each snapshot such that the position of the colloid coincides in each snapshot. We calculate the probability for each of the $200 \times 200$ bins to be occupied by a polymer by averaging over all snapshots.

The pictures in Figs. 10(a) and 10(b) depict density maps for a concentration of $10 \, \mu g/ml$ and $\nu=40 \, \mu m/s$ (in order to obtain a two-dimensional image we take only polymers in a slice $1 \, \mu m$ above and below the colloid center into account). It can be observed that there is a region of high polymer concentration in front and at the sides of the colloid, whereas the region behind the colloid is not yet completely refilled by the polymers. While the accumulation of polymers is stronger at higher concentrations, the depletion region behind the driven colloid is more pronounced at lower concentrations. Figure 10(c) shows the normalized average density of polymers in the direction of motion for concentrations of 10 and $40 \, \mu g/ml$ and $\nu=50 \, \mu m/s$. In front of the
colloid, a sharp peak can be observed. For high polymer concentrations, the probability to find a polymer in front of the colloid is close to 1. The region right behind the colloid is almost clear of polymers because the polymers get advected away from the colloid before they can diffuse into this region.

Our model reproduces very well the linear relation between drag force and the drag velocity for different polymer concentrations. As higher drag velocities require a larger system (even with periodic boundary conditions) and short numerical time steps, we are limited to about 80 μm/s by the available computational resources and time. However, the linearity of the drag force with respect to the velocity at sufficiently high velocities allows to extrapolate to the higher velocities used in the experiments. Figure 6 includes a linear fit to the simulation data, which agrees well with the experimental data.

As in the experiments, with Eq. (3) the measured drag force can be interpreted in terms of an effective viscosity. Figure 11 shows the dependency of this viscosity on the velocity for pure water and for polymer concentrations of 20, 40, and 50 μg/ml. Even for a polymer concentration of 50 μg/ml this viscosity is independent of the drag velocity for \( u > 60 \ \mu \text{m/s} \), but for lower velocities, we find that the viscosity increases, especially at higher polymer concentrations.

For a fixed drag velocity and varying polymer concentrations, the linear relation between force and concentrations for low polymer concentrations and the stronger nonlinear increase for higher concentrations can also be observed, as shown in Fig. 5. It can be seen that there is good quantitative agreement between simulations and experiments.

V. DISCUSSION

The experiments clearly show that the drag force on colloids pulled through a solution of λ-DNA with an optical tweezers cannot be explained by the Stokes force for the viscosity of the solution. It is much higher and increases nonlinearly with the DNA concentration but approximately linearly with the drag velocity. The resolution of the force measurement is ±0.15 pN and, therefore, the difference between the forces on the colloid in the solution and in pure water can be measured only for large concentrations or, in the case of smaller concentrations, only for large velocities. This limits the overlap between the DD theory and the experiments to the lowest concentration of \( \rho_{\text{DNA}} = 20 \ \mu \text{g/ml} \) used in the experiment.

At this concentration, the volume fraction of the solution taken by the DNA coils (based on \( R_g \approx 0.5 \ \mu \text{m} \)) is 0.1 and DNA-DNA interactions can be neglected. The agreement between theory and experiment is very good. For the highest concentration used in the experiments, \( \rho_{\text{DNA}} = 50 \ \mu \text{g/ml} \), the volume fraction is 0.5 and DNA-DNA interactions have to be taken into account.

In the BD simulations DNA-DNA interactions have been taken into account. As in the experiments and in the DD theory, the drag force increases linearly with the velocity, at least for large velocities. The nonlinear increase in the drag force with the DNA concentration is also observed in the BD simulations. The hydrodynamic flow field around the colloid (see Fig. 7) is explicitly included in the BD simulation scheme as well as in the DD calculations. It has a component normal to the direction of motion of the colloid, which efficiently reduces the number of DNA molecules accumulated in front of the colloid, while it fills the depletion zone at its back. As demonstrated by the excellent agreement with the experimental data, this hydrodynamic interaction between the colloid and the DNA is the dominant hydrodynamic force in the system. A proper treatment of these forces allows to achieve quantitative accuracy of the simulations.

The agreement between theory and experiment for low concentrations but high velocities is particularly remarkable.
considering the simplicity of the models, which neglect many aspects of hydrodynamic interactions or include them in a rather simplistic way (e.g., in terms of an increased viscosity in the DD model). The internal structure of the DNA molecules is neglected in both the DD model and the BD simulations. Apparently, the deformation of the DNA coils in the accumulation region in front of the colloid is not significant. This can be only explained by the proximity of the stagnation point for the solvent flow, which limits the normal component of the solvent velocity field. At this point the shear rate is also smaller than that at the colloid's side, where it goes up to over 700 1/s for \( u = 500 \mu m/s \). With a relaxation time on the order of 0.1 s, this amounts to a Weissenberg number \( Wi \) on the order of 70, i.e., much larger than 1. Therefore, the polymer coils should be significantly distorted and viscoelastic effects are to be expected.

In summary, we present a first direct experimental observation of jamming-induced drag enhancement on colloids in polymer solutions. First theoretical approaches to this problem as well as the BD simulations presented in this paper are in agreement with the experiments and represent a first quantitative modeling of the micro rheology of a complex fluid. However, further refinement is needed. Also the apparent contradiction to observations of an enhanced diffusivity of colloids in polymer solutions as compared to the Stokes–Einstein relation based on the viscosity of the bulk polymer solution needs to be addressed. We conjecture that the structure of the polymer solution in the vicinity of the colloid is significantly different for the fast and steady unidirectional motion (with viscosities in the highly nonlinear regime) discussed in this paper and for the slow random motion in diffusion.

ACKNOWLEDGMENTS

F.K., J.H., and M.R. acknowledge support from the Deutsche Forschungsgemeinschaft within the priority program SPP 1164 “Micro- and Nanofluids.” J.H. and R.W. also acknowledge fruitful discussions with M. Hecht. The computations were performed at Forschungszentrum Jülich and at the Scientific Supercomputing Center in Karlsruhe.

16G. K. Batchelor, J. Fluid Mech. 137, 467 (1983), CORRIGENDUM.
Bibliography


[276] The Level 2 Grid Project, http://www.grid-support.ac.uk/l2g/index.html. 61


[344] B. Ovryn. *Three-dimensional forward scattering particle image velocimetry applied to a microscopic field-of-view.* Exp. in Fluids, 29, S175, 2000. 313


Acknowledgments

It is a difficult or even impossible task to thank everybody who contributed to my work and progress during the last six years. However, I would like to thank my closest scientific collaborators and mentors for all the discussions, patience and scientific contributions of which quite a few have found their way into the papers presented in this thesis.

My sincere gratitude belongs to Peter V. Coveney, Hans J. Herrmann, and Hans R. Trebin for their support during the past years. Peter V. Coveney sparked my interest in mesoscopic simulation methods, grid computing and computational steering. Hans J. Herrmann drew my attention to suspensions and gave me the freedom and advice needed to develop my own directions of research. Last but not least, Hans R. Trebin’s support was of great help after Hans J. Herrmann left the University of Stuttgart.

I benefited from enjoyable and stimulating collaborations with my colleagues at the Centre for Computational Science in London, namely Sandra Barsky, Pascal Boulet, Jonathan Chin, Simon Clifford, Rafael Delgado-Buscalioni, Giovanni Giupponi, Nelido Gonzalez-Segredo, Matt Harvey, Shantenu Jha, Maddalena Venturoli, Shunzhou Wan, and many other colleagues from the RealityGrid project.

When moving to Stuttgart, I found a lively institute full of very friendly and hard working scientists. Also, during the last few years I learnt to know many colleagues from other institutions who contributed to my work during discussions, seminars, conferences, and also social events. Since it is certain that I would forget some people if I would try to count them all, the following names are meant to be representative: Eli Ben-Naim, Markus Bier, Bibhu Biswal, Roel Dullens, Ramon Garcia-Rojo, Christof Gutsche, Rudolf Hilfer, Jari Hyvälomma, Thomas Ihle, Badr Kaoui, Alexander Kornik, Pedro Lind, Sean Mc Namara, Ariel Narvaez, Frank Raischel, Markus Rauscher, Jörg Reinshagen, Aniruddha Sarkar, Sebastian Schmieschek, Hansjoerg Seybold, Sauro Succi, Olga Vinogradova, and Martin Wackenhut.

Special thanks belong to Martin Hecht, Christian Kunert, and Rudolf Weeber. Without their contributions to our collaborations, the current thesis would not have been possible.

I really appreciate the endless amounts of time Frank Huber, Marlies Parsons, and Henriette Patzelt invested to solve my technical and administrative problems. Thanks to Florian Zeller for helping me with the bibliography of this thesis.

The computer simulations presented in this thesis were mainly performed at supercomputing centres in Germany, the UK, and the US. I would like to thank the Höchstleistungsrechenzentrum Stuttgart, the Jülich Supercomputing Centre, the Scientific Supercomputing Center Karlsruhe, the Edinburgh Parallel Computing Centre, the Computer Services for Academic Research in Manchester, as well as the Pittsburgh Supercomputing Center for their excellent technical support. Financial support from the Landesstiftung Baden-Württemberg and the German science foundation was very profitable to me in order to realize my ideas and to be able to meet other scientists at various conferences and meetings.

And I would like to thank Christine for all her patience, support and love.